VoIP Topics

Contents

Articles

References

Article Licenses

Voice over IP

Voice over IP (**VoIP**, or **voice over Internet Protocol**) commonly refers to the communication protocols, technologies, methodologies, and transmission techniques involved in the delivery of voice communications and multimedia sessions over Internet Protocol (IP) networks, such as the Internet. Other terms commonly associated with VoIP are *IP telephony*, *Internet telephony*, *voice over broadband* (VoBB), *broadband telephony*, *IP communications*, and *broadband phone*.

Internet telephony refers to communications services —voice, fax, SMS, and/or voice-messaging applications— that are transported via the Internet, rather than the public switched telephone network (PSTN). The steps involved in originating a VoIP telephone call are signaling and media channel setup, digitization of the analog voice signal, encoding, packetization, and transmission as Internet Protocol (IP) packets over a packet-switched network. On the receiving side, similar steps (usually in the reverse order) such as reception of the IP packets, decoding of the packets and digital-to-analog conversion reproduce the original voice stream.[1] Even though IP Telephony and VoIP are terms that are used interchangeably, they are actually different; IP telephony has to do with digital telephony systems that use IP protocols for voice communication, while VoIP is actually a subset of IP Telephony. VoIP is a technology used by IP telephony as a means of transporting phone calls.[2]

Early providers of voice over IP services offered business models (and technical solutions) that mirrored the architecture of the legacy telephone network. Second generation providers, such as Skype have built closed networks for private user bases, offering the benefit of free calls and convenience, while denying their users the ability to call out to other networks. This has severely limited the ability of users to mix-and-match third-party hardware and software. Third generation providers, such as Google Talk have adopted [3] the concept of Federated VoIP - which is a complete departure from the architecture of the legacy networks. These solutions typically allow arbitrary and dynamic interconnection between any two domains on the Internet whenever a user wishes to place a call.

VoIP systems employ session control protocols to control the set-up and tear-down of calls as well as audio codecs which encode speech allowing transmission over an IP network as digital audio via an audio stream. The choice of codec varies between different implementations of VoIP depending on application requirements and network bandwidth; some implementations rely on narrowband and compressed speech, while others support high fidelity stereo codecs. Some popular codecs include u-law and a-law versions of G.711, G.722 which is a high-fidelity codec marketed as HD Voice by Polycom, a popular open source voice codec known as iLBC, a codec that only uses 8 kbit/s each way called G.729, and many others.

VoIP is available on many smartphones and Internet devices so that users of portable devices that are not phones, may place calls or send SMS text messages over 3G or Wi-Fi.[4]

Protocols

Voice over IP has been implemented in various ways using both proprietary and open protocols and standards. Examples of the network protocols used to implement VoIP include:

- H.323
- Media Gateway Control Protocol (MGCP)
- Session Initiation Protocol (SIP)
- Real-time Transport Protocol (RTP)
- Session Description Protocol (SDP)
- Inter-Asterisk eXchange (IAX)
- Jingle XMPP VoIP extensions

The H.323 protocol was one of the first VoIP protocols that found widespread implementation for long-distance traffic, as well as local area network services. However, since the development of newer, less complex protocols

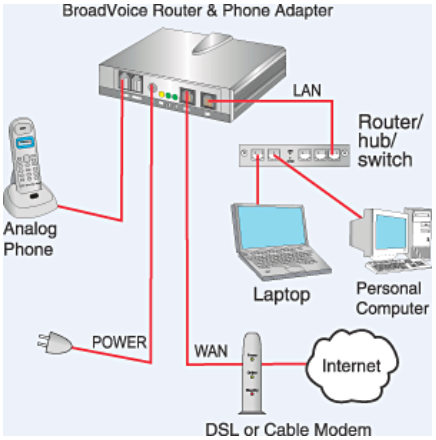
such as MGCP and SIP, H.323 deployments are increasingly limited to carrying existing long-haul network traffic. In particular, the Session Initiation Protocol (SIP) has gained widespread VoIP market penetration.

A notable proprietary implementation is the Skype protocol, which is in part based on the principles of peer-to-peer (P2P) networking.

Adoption

Consumer market

A major development that started in 2004 was the introduction of mass-market VoIP services that utilize existing broadband Internet access, by which subscribers place and receive telephone calls in much the same manner as they would via the public switched telephone network (PSTN). Full-service VoIP phone companies provide inbound and outbound service with Direct Inbound Dialing. Many offer unlimited domestic calling for a flat monthly subscription fee. This sometimes includes international calls to certain countries. Phone calls between subscribers of the same provider are usually free when flat-fee service is not available A VoIP phone is necessary to connect to a VoIP service provider. This can be implemented in several ways:



Example of residential network including VoIP

- Dedicated VoIP phones connect directly to the IP network using technologies such as wired Ethernet or wireless Wi-Fi. They are typically designed in the style of traditional digital business telephones.
- An analog telephone adapter is a device that connects to the network and implements the electronics and firmware to operate a conventional analog telephone attached through a modular phone jack. Some residential Internet gateways and cablemodems have this function built in.
- A softphone is application software installed on a networked computer that is equipped with a microphone and speaker, or headset. The application typically presents a dial pad and display field to the user to operate the application by mouse clicks or keyboard input.

PSTN and mobile network providers

It is becoming increasingly common for telecommunications providers to use VoIP telephony over dedicated and public IP networks to connect switching centres and to interconnect with other telephony network providers; this is often referred to as "IP backhaul".[5][6]

Smartphones and Wi-Fi enabled mobile phones may have SIP clients built into the firmware or available as an application download.

Corporate use

Because of the bandwidth efficiency and low costs that VoIP technology can provide, businesses are migrating from traditional copper-wire telephone systems to VoIP systems to reduce their monthly phone costs. In 2008, 80% of all new PBX lines installed internationally were VoIP.[7]

VoIP solutions aimed at businesses have evolved into unified communications services that treat all communications—phone calls, faxes, voice mail, e-mail, Web conferences and more—as discrete units that can all be delivered via any means and to any handset, including cellphones. Two kinds of competitors are competing in this space: one set is focused on VoIP for medium to large enterprises, while another is targeting the small-to-medium business (SMB) market.[8]

VoIP allows both voice and data communications to be run over a single network, which can significantly reduce infrastructure costs.[9]

The prices of extensions on VoIP are lower than for PBX and key systems. VoIP switches may run on commodity hardware, such as PCs or Linux systems. Rather than closed architectures, these devices rely on standard interfaces.[9]

VoIP devices have simple, intuitive user interfaces, so users can often make simple system configuration changes. Dual-mode phones enable users to continue their conversations as they move between an outside cellular service and an internal Wi-Fi network, so that it is no longer necessary to carry both a desktop phone and a cellphone. Maintenance becomes simpler as there are fewer devices to oversee.[9]

Skype, which originally marketed itself as a service among friends, has begun to cater to businesses, providing free-of-charge connections between any users on the Skype network and connecting to and from ordinary PSTN telephones for a charge.[10]

In the United States the Social Security Administration (SSA) is converting its field offices of 63,000 workers from traditional phone installations to a VoIP infrastructure carried over its existing data network.[11][12]

Advantages

There are several advantages to using voice over IP. The biggest single advantage VoIP has over standard telephone systems is cost. In addition, international calls using VoIP are usually very inexpensive. One other advantage, which will become much more pronounced as VoIP use climbs, calls between VoIP users are usually free. Using services such as TrueVoIP, subscribers can call one another at no cost to either party.[13]

Operational cost

VoIP can be a benefit for reducing communication and infrastructure costs. Examples include:

- Routing phone calls over existing data networks to avoid the need for separate voice and data networks.[14]
- The ability to transmit more than one telephone call over a single broadband connection.
- Secure calls using standardized protocols (such as Secure Real-time Transport Protocol). Most of the difficulties of creating a secure telephone connection over traditional phone lines, such as digitizing and digital transmission, are already in place with VoIP. It is only necessary to encrypt and authenticate the existing data stream.

Challenges

Quality of service

Communication on the IP network is inherently less reliable in contrast to the circuit-switched public telephone network, as it does not provide a network-based mechanism to ensure that data packets are not lost, and are delivered in sequential order. It is a best-effort network without fundamental Quality of Service (QoS) guarantees. Therefore, VoIP implementations may face problems mitigating latency and jitter.[15][16]

By default, network routers handle traffic on a first-come, first-served basis. Network routers on high volume traffic links may introduce latency that exceeds permissible thresholds for VoIP. Fixed delays cannot be controlled, as they are caused by the physical distance the packets travel; however, latency can be minimized by marking voice packets as being delay-sensitive with methods such as DiffServ.[15]

A VoIP packet usually has to wait for the current packet to finish transmission, although it is possible to preempt (abort) a less important packet in mid-transmission, although this is not commonly done, especially on high-speed links where transmission times are short even for maximum-sized packets.[17] An alternative to preemption on slower links, such as dialup and digital subscriber line (DSL), is to reduce the maximum transmission time by reducing the maximum transmission unit. But every packet must contain protocol headers, so this increases relative

header overhead on every link traversed, not just the bottleneck (usually Internet access) link.[17]

DSL modems provide Ethernet (or Ethernet over USB) connections to local equipment, but inside they are actually Asynchronous Transfer Mode (ATM) modems. They use ATM Adaptation Layer 5 (AAL5) to segment each Ethernet packet into a series of 53-byte ATM cells for transmission and reassemble them back into Ethernet packets at the receiver. A virtual circuit identifier (VCI) is part of the 5-byte header on every ATM cell, so the transmitter can multiplex the active virtual circuits (VCs) in any arbitrary order. Cells from the *same* VC are always sent sequentially.

However, the great majority of DSL providers use only one VC for each customer, even those with bundled VoIP service. Every Ethernet packet must be completely transmitted before another can begin. If a second VC were established, given high priority and reserved for VoIP, then a low priority data packet could be suspended in mid-transmission and a VoIP packet sent right away on the high priority VC. Then the link would pick up the low priority VC where it left off. Because ATM links are multiplexed on a cell-by-cell basis, a high priority packet would have to wait at most 53 byte times to begin transmission. There would be no need to reduce the interface MTU and accept the resulting increase in higher layer protocol overhead, and no need to abort a low priority packet and resend it later.

ATM has substantial header overhead: 5/53 = 9.4%, roughly twice the total header overhead of a 1500 byte Ethernet packet. This "ATM tax" is incurred by every DSL user whether or not he takes advantage of multiple virtual circuits - and few can.[15]

ATM's potential for latency reduction is greatest on slow links, because worst-case latency decreases with increasing link speed. A full-size (1500 byte) Ethernet frame takes 94 ms to transmit at 128 kbit/s but only 8 ms at 1.5 Mbit/s. If this is the bottleneck link, this latency is probably small enough to ensure good VoIP performance without MTU reductions or multiple ATM VCs. The latest generations of DSL, VDSL and VDSL2, carry Ethernet without intermediate ATM/AAL5 layers, and they generally support IEEE 802.1p priority tagging so that VoIP can be queued ahead of less time-critical traffic.[15]

Voice, and all other data, travels in packets over IP networks with fixed maximum capacity. This system may be more prone to congestion and DoS attacks[18] than traditional circuit switched systems; a circuit switched system of insufficient capacity will refuse new connections while carrying the remainder without impairment, while the quality of real-time data such as telephone conversations on packet-switched networks degrades dramatically.[15]

Fixed delays cannot be controlled as they are caused by the physical distance the packets travel. They are especially problematic when satellite circuits are involved because of the long distance to a geostationary satellite and back; delays of 400–600 ms are typical.

When the load on a link grows so quickly that its switches experience queue overflows, congestion results and data packets are lost. This signals a transport protocol like TCP to reduce its transmission rate to alleviate the congestion. But VoIP usually uses UDP not TCP because recovering from congestion through retransmission usually entails too much latency.[15] So QoS mechanisms can avoid the undesirable loss of VoIP packets by immediately transmitting them ahead of any queued bulk traffic on the same link, even when that bulk traffic queue is overflowing.

The receiver must resequence IP packets that arrive out of order and recover gracefully when packets arrive too late or not at all. Jitter results from the rapid and random (i.e., unpredictable) changes in queue lengths along a given Internet path due to competition from other users for the same transmission links. VoIP receivers counter jitter by storing incoming packets briefly in a "de-jitter" or "playout" buffer, deliberately increasing latency to improve the chance that each packet will be on hand when it is time for the voice engine to play it. The added delay is thus a compromise between excessive latency and excessive dropout, i.e., momentary audio interruptions.

Although jitter is a random variable, it is the sum of several other random variables that are at least somewhat independent: the individual queuing delays of the routers along the Internet path in question. Thus according to the central limit theorem, we can model jitter as a gaussian random variable. This suggests continually estimating the

mean delay and its standard deviation and setting the playout delay so that only packets delayed more than several standard deviations above the mean will arrive too late to be useful. In practice, however, the variance in latency of many Internet paths is dominated by a small number (often one) of relatively slow and congested "bottleneck" links. Most Internet backbone links are now so fast (e.g. 10 Gbit/s) that their delays are dominated by the transmission medium (e.g., optical fiber) and the routers driving them do not have enough buffering for queuing delays to be significant.

It has been suggested to rely on the packetized nature of media in VoIP communications and transmit the stream of packets from the source phone to the destination phone simultaneously across different routes (multi-path routing).[19] In such a way, temporary failures have less impact on the communication quality. In capillary routing it has been suggested to use at the packet level Fountain codes or particularly raptor codes for transmitting extra redundant packets making the communication more reliable.

A number of protocols have been defined to support the reporting of quality of service (QoS) and quality of experience (QoE) for VoIP calls. These include RTCP Extended Report (RFC 3611), SIP RTCP Summary Reports, H.460.9 Annex B (for H.323), H.248.30 and MGCP extensions. The RFC 3611 VoIP Metrics block is generated by an IP phone or gateway during a live call and contains information on packet loss rate, packet discard rate (because of jitter), packet loss/discard burst metrics (burst length/density, gap length/density), network delay, end system delay, signal / noise / echo level, Mean Opinion Scores (MOS) and R factors and configuration information related to the jitter buffer.

RFC 3611 VoIP metrics reports are exchanged between IP endpoints on an occasional basis during a call, and an end of call message sent via SIP RTCP Summary Report or one of the other signaling protocol extensions. RFC 3611 VoIP metrics reports are intended to support real time feedback related to QoS problems, the exchange of information between the endpoints for improved call quality calculation and a variety of other applications.

Layer-2 quality of service

A number of protocols that deal with the data link layer and physical layer include quality-of-service mechanisms that can be used to ensure that applications like VoIP work well even in congested scenarios. Some examples include:

- IEEE 802.11e is an approved amendment to the IEEE 802.11 standard that defines a set of quality-of-service enhancements for wireless LAN applications through modifications to the Media Access Control (MAC) layer. The standard is considered of critical importance for delay-sensitive applications, such as voice over wireless IP.
- IEEE 802.1p defines 8 different classes of service (including one dedicated to voice) for traffic on layer-2 wired Ethernet.
- The ITU-T G.hn standard, which provides a way to create a high-speed (up to 1 gigabit per second) Local area network using existing home wiring (power lines, phone lines and coaxial cables). G.hn provides QoS by means of "Contention-Free Transmission Opportunities" (CFTXOPs) which are allocated to flows (such as a VoIP call) which require QoS and which have negotiated a "contract" with the network controllers.

Susceptibility to power failure

Telephones for traditional residential analog service are usually connected directly to telephone company phone lines which provide direct current to power most basic analog handsets independently of locally available power.

IP Phones and VoIP telephone adapters connect to routers or cable modems which typically depend on the availability of mains electricity or locally generated power.[20] Some VoIP service providers use customer premises equipment (e.g., cablemodems) with battery-backed power supplies to assure uninterrupted service for up to several hours in case of local power failures. Such battery-backed devices typically are designed for use with analog handsets.

Some VoIP service providers implement services to route calls to other telephone services of the subscriber, such a cellular phone, in the event that the customer's network device is inaccessible to terminate the call.

The susceptibility of phone service to power failures is a common problem even with traditional analog service in areas where many customers purchase modern telephone units that operate with wireless handsets to a base station, or that have other modern phone features, such as built-in voicemail or phone book features.

Emergency calls

The nature of IP makes it difficult to locate network users geographically. Emergency calls, therefore, cannot easily be routed to a nearby call center. Sometimes, VoIP systems may route emergency calls to a non-emergency phone line at the intended department; in the United States, at least one major police department has strongly objected to this practice as potentially endangering the public.[21][22]

A fixed line phone has a direct relationship between a telephone number and a physical location. If an emergency call comes from that number, then the physical location is known.

In the IP world, it is not so simple. A broadband provider may know the location where the wires terminate, but this does not necessarily allow the mapping of an IP address to that location. IP addresses are often dynamically assigned, so the ISP may allocate an address for online access, or at the time a broadband router is engaged. The ISP recognizes individual IP addresses, but does not necessarily know to which physical location it corresponds. The broadband service provider knows the physical location, but is not necessarily tracking the IP addresses in use.[22]

There are more complications since IP allows a great deal of mobility. For example, a broadband connection can be used to dial a virtual private network that is employer-owned. When this is done, the IP address being used will belong to the range of the employer, rather than the address of the ISP, so this could be many kilometres away or even in another country. To provide another example: if mobile data is used, e.g., a 3G mobile handset or USB wireless broadband adapter, then the IP address has no relationship with any physical location, since a mobile user could be anywhere that there is network coverage, even roaming via another cellular company.

In short, there is no relationship between IP address and physical location, so the address itself reveals no useful information for the emergency services.

At the VoIP level, a phone or gateway may identify itself with a SIP registrar by using a username and password. So in this case, the Internet Telephony Service Provider (ITSP) knows that a particular user is online, and can relate a specific telephone number to the user. However, it does not recognize how that IP traffic was engaged. Since the IP address itself does not necessarily provide location information presently, today a "best efforts" approach is to use an available database to find that user and the physical address the user chose to associate with that telephone number—clearly an imperfect solution.[22]

VoIP Enhanced 911 (E911) is a method by which VoIP providers in the United States support emergency services. The VoIP E911 emergency-calling system associates a physical address with the calling party's telephone number as required by the Wireless Communications and Public Safety Act of 1999. All VoIP providers that provide access to the public switched telephone network are required to implement E911,[22] a service for which the subscriber may be charged. Participation in E911 is not required and customers may opt-out of E911 service.[22]

One shortcoming of VoIP E911 is that the emergency system is based on a static table lookup. Unlike in cellular phones, where the location of an E911 call can be traced using Assisted GPS or other methods, the VoIP E911 information is only accurate so long as subscribers are diligent in keeping their emergency address information up-to-date. In the United States, the Wireless Communications and Public Safety Act of 1999 leaves the burden of responsibility upon the subscribers and not the service providers to keep their emergency information up to date.[22]

Lack of redundancy

The historical separation of IP networks and the PSTN provided redundancy when no portion of a call was routed over IP network. An IP network outage would not necessarily mean that a voice communication outage would occur simultaneously, allowing phone calls to be made during IP network outages. When telephone service relies on IP network infrastructure such as the Internet, a network failure can isolate users from all telephony communication, including Enhanced 911 and equivalent services in other locales. However, the network design envisioned by DARPA in the early 1980s included a fault tolerant architecture under adverse conditions.

Number portability

Local number portability (LNP) and Mobile number portability (MNP) also impact VoIP business. In November 2007, the Federal Communications Commission in the United States released an order extending number portability obligations to interconnected VoIP providers and carriers that support VoIP providers.[23] Number portability is a service that allows a subscriber to select a new telephone carrier without requiring a new number to be issued. Typically, it is the responsibility of the former carrier to "map" the old number to the undisclosed number assigned by the new carrier. This is achieved by maintaining a database of numbers. A dialed number is initially received by the original carrier and quickly rerouted to the new carrier. Multiple porting references must be maintained even if the subscriber returns to the original carrier. The FCC mandates carrier compliance with these consumer-protection stipulations.

A voice call originating in the VoIP environment also faces challenges to reach its destination if the number is routed to a mobile phone number on a traditional mobile carrier. VoIP has been identified in the past as a Least Cost Routing (LCR) system, which is based on checking the destination of each telephone call as it is made, and then sending the call via the network that will cost the customer the least.[24] This rating is subject to some debate given the complexity of call routing created by number portability. With GSM number portability now in place, LCR providers can no longer rely on using the network root prefix to determine how to route a call. Instead, they must now determine the actual network of every number before routing the call.

Therefore, VoIP solutions also need to handle MNP when routing a voice call. In countries without a central database, like the UK, it might be necessary to query the GSM network about which home network a mobile phone number belongs to. As the popularity of VoIP increases in the enterprise markets because of least cost routing options, it needs to provide a certain level of reliability when handling calls.

MNP checks are important to assure that this quality of service is met. By handling MNP lookups before routing a call and by assuring that the voice call will actually work, VoIP service providers are able to offer business subscribers the level of reliability they require.

PSTN integration

E.164 is a global FGFnumbering standard for both the PSTN and PLMN. Most VoIP implementations support E.164 to allow calls to be routed to and from VoIP subscribers and the PSTN/PLMN.[25] VoIP implementations can also allow other identification techniques to be used. For example, Skype allows subscribers to choose "Skype names"[26] (usernames) whereas SIP implementations can use URIs[27] similar to email addresses. Often VoIP implementations employ methods of translating non-E.164 identifiers to E.164 numbers and vice-versa, such as the Skype-In service provided by Skype[28] and the ENUM service in IMS and SIP.[29]

Echo can also be an issue for PSTN integration.[30] Common causes of echo include impedance mismatches in analog circuitry and acoustic coupling of the transmit and receive signal at the receiving end.

Security

VoIP telephone systems are susceptible to attacks as are any Internet-connected devices. This means that hackers who know about these vulnerabilities (such as insecure passwords) can institute denial-of-service attacks, harvest customer data, record conversations and break into voice mailboxes.[31][32][33]

Another challenge is routing VoIP traffic through firewalls and network address translators. Private Session Border Controllers are used along with firewalls to enable VoIP calls to and from protected networks. For example, Skype uses a proprietary protocol to route calls through other Skype peers on the network, allowing it to traverse symmetric NATs and firewalls. Other methods to traverse NATs involve using protocols such as STUN or Interactive Connectivity Establishment (ICE).

Many consumer VoIP solutions do not support encryption, although having a secure phone is much easier to implement with VoIP than traditional phone lines. As a result, it is relatively easy to eavesdrop on VoIP calls and even change their content.[34] An attacker with a packet sniffer could intercept your VoIP calls if you are not on a secure VLAN. However, physical security of the switches within an enterprise and the facility security provided by ISPs make packet capture less of a problem than originally foreseen. Further research has shown that tapping into a fiber optic network without detection is difficult if not impossible. This means that once a voice packet is within the Internet backbone it is relatively safe from interception.

There are open source solutions, such as Wireshark, that facilitate sniffing of VoIP conversations. Securing the content of conversations from malicious observers requires encryption and cryptographic authentication which is sometimes difficult to find at a consumer level. The existing security standard Secure Real-time Transport Protocol (SRTP) and the new ZRTP protocol are available on Analog Telephone Adapters (ATAs) as well as various softphones. It is possible to use IPsec to secure P2P VoIP by using opportunistic encryption. In 2005, Skype invited a researcher, Dr Tom Berson, to assess the security of the Skype software, and his conclusions are available in a published report.[35]

Securing VoIP

To prevent the above security concerns government and military organizations are using voice over secure IP (VoSIP), secure voice over IP (SVoIP), and secure voice over secure IP (SVoSIP) to protect confidential and classified VoIP communications.[36] Secure voice over secure IP is accomplished by encrypting VoIP with protocols such as SRTP or ZRTP. Secure voice over IP is accomplished by using Type 1 encryption on a classified network, like SIPRNet.[37][38][39][40][41] Public Secure VoIP is also available with free GNU programs and in many popular commercial VoIP programs via libraries such as ZRTP.[42]

Caller ID

Further information: Caller ID spoofing

Caller ID support among VoIP providers varies, but is provided by the majority of VoIP providers.

Many VoIP carriers allow callers to configure arbitrary caller ID information, thus permitting spoofing attacks.[43] Business-grade VoIP equipment and software often makes it easy to modify caller ID information, providing many businesses great flexibility.

The Truth in Caller ID Act became law in on December 22, 2010. This bill proposes to make it a crime in the United States to "knowingly transmit misleading or inaccurate caller identification information with the intent to defraud, cause harm, or wrongfully obtain anything of value ...".[44] Rules implementing the law were adopted by the Federal Communications Commission on June 20, 2011.[45]

Compatibility with traditional analog telephone sets

Some analog telephone adapters do not decode pulse dialing from older phones. They may only work with push-button telephones using the touch-tone system. The VoIP user may use a pulse-to-tone converter, if needed.[46]

Fax handling

Support for sending faxes over VoIP implementations is still limited. The existing voice codecs are not designed for fax transmission; they are designed to digitize an analog representation of a human voice efficiently. However, the inefficiency of digitizing an analog representation (modem signal) of a digital representation (a document image) of analog data (an original document) more than negates any bandwidth advantage of VoIP. In other words, the fax "sounds" simply do not fit in the VoIP channel. An alternative IP-based solution for delivering fax-over-IP called T.38 is available. Sending faxes using VoIP is sometimes referred to as FoIP, or Fax over IP.[47]

The T.38 protocol is designed to compensate for the differences between traditional packet-less communications over analog lines and packet based transmissions which are the basis for IP communications. The fax machine could be a traditional fax machine connected to the PSTN, or an ATA box (or similar). It could be a fax machine with an RJ-45 connector plugged straight into an IP network, or it could be a computer pretending to be a fax machine.[48] Originally, T.38 was designed to use UDP and TCP transmission methods across an IP network. TCP is better suited for use between two IP devices. However, older fax machines, connected to an analog system, benefit from UDP near real-time characteristics due to the "no recovery rule" when a UDP packet is lost or an error occurs during transmission.[49] UDP transmissions are preferred as they do not require testing for dropped packets and as such since each T.38 packet transmission includes a majority of the data sent in the prior packet, a T.38 termination point has a higher degree of success in re-assembling the fax transmission back into its original form for interpretation by the end device. This in an attempt to overcome the obstacles of simulating real time transmissions using packet based protocol.[50]

There have been updated versions of T.30 to resolve the fax over IP issues, which is the core fax protocol. Some newer high end fax machines have T.38 built-in capabilities which allow the user to plug right into the network and transmit/receive faxes in native T.38 like the Ricoh 4410NF Fax Machine.[51] A unique feature of T.38 is that each packet contains a portion of the main data sent in the previous packet. With T.38, two successive lost packets are needed to actually lose any data. The data you lose will only be a small piece, but with the right settings and error correction mode, there is an increased likelihood that you will receive enough of the transmission to satisfy the requirements of the fax machine for output of the sent document.

Support for other telephony devices

Another challenge for VoIP implementations is the proper handling of outgoing calls from other telephony devices such as digital video recorders, satellite television receivers, alarm systems, conventional modems and other similar devices that depend on access to a PSTN telephone line for some or all of their functionality.

These types of calls sometimes complete without any problems, but in other cases they fail. If VoIP and cellular substitution becomes very popular, some ancillary equipment makers may be forced to redesign equipment, because it would no longer be possible to assume a conventional PSTN telephone line would be available in consumer's homes.

Legal issues

As the popularity of VoIP grows, governments are becoming more interested in regulating VoIP in a manner similar to PSTN services.[52]

Throughout the developing world, countries where regulation is weak or captured by the dominant operator, restrictions on the use of VoIP are imposed, including in Panama where VoIP is taxed, Guyana where VoIP is prohibited and India where its retail commercial sales is allowed but only for long distance service.[53] In Ethiopia, where the government is monopolizing telecommunication service, it is a criminal offense to offer services using VoIP. The country has installed firewalls to prevent international calls being made using VoIP. These measures were taken after the popularity of VoIP reduced the income generated by the state owned telecommunication company.

European Union

In the European Union, the treatment of VoIP service providers is a decision for each Member State's national telecoms regulator, which must use competition law to define relevant national markets and then determine whether any service provider on those national markets has "significant market power" (and so should be subject to certain obligations). A general distinction is usually made between VoIP services that function over managed networks (via broadband connections) and VoIP services that function over unmanaged networks (essentially, the Internet). The relevant EU Directive is not clearly drafted concerning obligations which can exist independently of market power (e.g., the obligation to offer access to emergency calls), and it is impossible to say definitively whether VoIP service providers of either type are bound by them. A review of the EU Directive is under way and should be complete by 2007.

India

In India, it is legal to use VoIP, but it is illegal to have VoIP gateways inside India. This effectively means that people who have PCs can use them to make a VoIP call to any number, but if the remote side is a normal phone, the gateway that converts the VoIP call to a POTS call should not be inside India.

In the interest of the Access Service Providers and International Long Distance Operators the Internet telephony was permitted to the ISP with restrictions. Internet Telephony is considered to be different service in its scope, nature and kind from real time voice as offered by other Access Service Providers and Long Distance Carriers. Hence the following type of Internet Telephony are permitted in India : (a) PC to PC; within or outside India (b) PC / a device / Adapter conforming to standard of any international agencies like- ITU or IETF etc. in India to PSTN/PLMN abroad. (c) Any device / Adapter conforming to standards of International agencies like ITU, IETF etc. connected to ISP node with static IP address to similar device / Adapter; within or outside India. (d) Except whatever is described in condition (ii) above, no other form of Internet Telephony is permitted. (e) In India no Separate Numbering Scheme is provided to the Internet Telephony. Presently the 10 digit Numbering allocation based on E.164 is permitted to the Fixed Telephony, GSM, CDMA wireless service. For Internet Telephony the numbering scheme shall only conform to IP addressing Scheme of Internet Assigned Numbers Authority (IANA). Translation of E.164 number / private number to IP address allotted to any device and vice versa, by ISP to show compliance with IANA numbering scheme is not permitted. (f) The Internet Service Licensee is not permitted to have PSTN/PLMN connectivity. Voice communication to and from a telephone connected to PSTN/PLMN and following E.164 numbering is prohibited in India. [54]

Middle East

In the UAE and Oman it is illegal to use any form of VoIP, to the extent that Web sites of Skype and Gizmo5 are blocked. Providing or using VoIP services is illegal in Oman. Those who violate the law stand to be fined 50,000 Omani Rial (about 130,317 US dollars) or spend two years in jail or both. In 2009, police in Oman have raided 121 Internet cafes throughout the country and arrested 212 people for using/providing VoIP services.

South Korea

In South Korea, only providers registered with the government are authorized to offer VoIP services. Unlike many VoIP providers, most of whom offer flat rates, Korean VoIP services are generally metered and charged at rates similar to terrestrial calling. Foreign VoIP providers encounter high barriers to government registration. This issue came to a head in 2006 when Internet service providers providing personal Internet services by contract to United States Forces Korea members residing on USFK bases threatened to block off access to VoIP services used by USFK members as an economical way to keep in contact with their families in the United States, on the grounds that the service members' VoIP providers were not registered. A compromise was reached between USFK and Korean telecommunications officials in January 2007, wherein USFK service members arriving in Korea before June 1, 2007, and subscribing to the ISP services provided on base may continue to use their US-based VoIP subscription, but later arrivals must use a Korean-based VoIP provider, which by contract will offer pricing similar to the flat rates offered by US VoIP providers.[55]

United States

In the United States, the Federal Communications Commission now requires all interconnected VoIP service providers to comply with requirements comparable to those for traditional telecommunications service providers. VoIP operators in the US are required to support local number portability; make service accessible to people with disabilities; pay regulatory fees, universal service contributions, and other mandated payments; and enable law enforcement authorities to conduct surveillance pursuant to the Communications Assistance for Law Enforcement Act (CALEA). "Interconnected" VoIP operators also must provide Enhanced 911 service, disclose any limitations on their E-911 functionality to their consumers, and obtain affirmative acknowledgements of these disclosures from all consumers.[56] VoIP operators also receive the benefit of certain US telecommunications regulations, including an entitlement to interconnection and exchange of traffic with incumbent local exchange carriers via wholesale carriers. Providers of "nomadic" VoIP service—those who are unable to determine the location of their users—are exempt from state telecommunications regulation.[57]

Another legal issue that the US Congress is debating concerns changes to the Foreign Intelligence Surveillance Act. The issue in question is calls between Americans and foreigners. The National Security Agency (NSA) is not authorized to tap Americans' conversations without a warrant—but the Internet, and specifically VoIP does not draw as clear a line to the location of a caller or a call's recipient as the traditional phone system does. As VoIP's low cost and flexibility convinces more and more organizations to adopt the technology, the surveillance for law enforcement agencies becomes more difficult. VoIP technology has also increased security concerns because VoIP and similar technologies have made it more difficult for the government to determine where a target is physically located when communications are being intercepted, and that creates a whole set of new legal challenges.[58]

Pronunciation

The acronym *VoIP* has been pronounced variably since the inception of the term. Apart from spelling out the acronym letter by letter, *vē'ō'ī'pē* (*vee-oh-eye-pee*), there are three likely possible pronunciations: *vō'ī'pē*[needs IPA] (*vo-eye-pee*) and *vō'ip*[needs IPA] (*vo-ipp*), have been used, but generally, the single syllable *vŏy'p*[needs IPA] (*voyp*, as in *voice*) may be the most common within the industry.[59]

Historical milestones

- 1973: Network Voice Protocol (NVP) developed by Danny Cohen and others to carry real time voice over Arpanet
- 1974: The Institute of Electrical and Electronic Engineers (IEEE) published a paper titled "A Protocol for Packet Network Interconnection".[60]
- 1974: Network Voice Protocol (NVP) first tested over Arpanet in August 1974, carrying 16k CVSD encoded voice - first implementation of Voice over IP
- 1977: Danny Cohen, Vint Cert, Jon Postel agree to separate IP from TCP, and create UDP for carrying real time traffic
- 1981: IPv4 is described in RFC 791.
- 1985: The National Science Foundation commissions the creation of NSFNET.[61]
- 1986: Proposals from various standards organizations for Voice over ATM, in addition to commercial packet voice products from companies such as StrataCom
- 1992: Voice over Frame Relay standards development within Frame Relay Forum
- 1994: First Voice Over IP application (Freeware for Linux) [62]
- 1995: VocalTec releases the first commercial Internet phone software.[63][64]
  - Beginning in 1995, Intel, Microsoft and Radvision initiated standardization activities for VoIP communications system.[65]
- 1996:
  - ITU-T begins development of standards for the transmission and signaling of voice communications over Internet Protocol networks with the H.323 standard.[66]
  - US telecommunication companies petition the US Congress to ban Internet phone technology.[67]
- 1997: Level 3 began development of its first softswitch, a term they coined in 1998.[68]
- 1999:
  - The Session Initiation Protocol (SIP) specification RFC 2543 is released.[69]
  - Mark Spencer of Digium develops the first open source private branch exchange (PBX) software (Asterisk).[70]
- 2004: Commercial VoIP service providers proliferate.

References

[1] "Voice over Internet Protocol. Definition and Overview" (http://www.teliqo.com/voip/). International Engineering Consortium. 2007. . Retrieved 2009-04-27.

[2] "IP Telephony Vs VoIP" (http://www.networkstraining.com). . Retrieved 27 April 2011.

[3] "XMPP Federation" (http://googletalk.blogspot.com/2006/01/xmpp-federation.html). Google Talkabout. 2006. . Retrieved 2012-05-11.

[4] Booth, C (2010). "Chapter 2: IP Phones, Software VoIP, and Integrated and Mobile VoIP". *Library Technology Reports* **46** (5): 11–19.

[5] "Carriers look to IP for backhaul" (http://www.commsdesign.com/showArticle.jhtml;jsessionid=YBQXDJILLQX0IQSNDLPSKH0CJUNN2JVN?articleID=159907138). Telecommunications Online. January 21, 2009. . Retrieved 2009-01-21.

[6] "Mobile's IP challenge" (http://www.totaltele.com/View.aspx?ID=77588&t=4). Total Telecom. December 8, 2005. . Retrieved 2009-01-21.

[7] Michael Dosch and Steve Church. "VoIP In The Broadcast Studio" (http://www.axiaaudio.com/tech/voip/default.htm). Axia Audio. . Retrieved 2011-06-21.

[8] Callahan, Renee (December 9, 2008). "Businesses Move To Voice-Over-IP" (http://www.forbes.com/2008/12/09/skype-vonage-ringcentral_leadership_clayton_in_rc_1209claytonchristensen_inl.html). forbes.com. . Retrieved 2009-03-03.
[9] Korzeniowski, Peter (January 8, 2009). "Three Technologies You Need In 2009" (http://www.forbes.com/2009/01/08/small-business-voip-ent-tech-cx_bm_0108bmightytech09.html/). Forbes.com. . Retrieved 2009-03-02.
[10] "Skype For Business" (http://www.skype.com/business/allfeatures/3skypephone/). skype.com. . Retrieved 2009-03-16.
[11] William Jackson (2009-05-27). "SSA goes big on VOIP" (http://gcn.com/Articles/2009/06/01/SSA-VOIP-implementation.aspx?s=gcndaily_280509&Page=1). Government Computer News. . Retrieved 2009-05-28.
[12] "Social Security to Build "World's Largest VOIP"" (http://www.govtech.com/gt/275677). Government Technology. . Retrieved 2009-05-29.
[13] voip cute (http://voipcute.blogspot.com/2012/01/voice-over-ip-voip-services.html), VOIP Advantages
[14] FCC.gov (http://www.fcc.gov/voip/), What are some advantages of VOIP?
[15] "Quality of Service for Voice over IP" (http://www.cisco.com/en/US/docs/ios/solutions_docs/qos_solutions/QoSVoIP/QoSVoIP.html). . Retrieved May 3, 2011.
[16] Prabhakar, G.; Rastogi, R.,& Thotton, M (2005). "OSS Architecture & Requirements for VoIP Networks". *Bell Labs Technical Journal* **10** (1): 31–45.
[17] "Quality of Service for Voice over IP" (http://www.cisco.com/en/US/docs/ios/solutions_docs/qos_solutions/QoSVoIP/QoSVoIP.html#wp1029054). . Retrieved May 3, 2011.
[18] VOIP – Vulnerability over Internet Protocol (http://www.continuitycentral.com/feature074.htm)
[19] IEEE Multipath routing with adaptive playback scheduling for Voice over IP in Service Overlay Networks (Abstract) (http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=4520089)
[20] ICT Regulation Tool Kit – 4.4 VOIP – Regulatory Issues – Universal Service (http://www.ictregulationtoolkit.org/en/Section.3083.html)
[21] Letter from the City of New York to the Federal Communications Commission (http://gullfoss2.fcc.gov/prod/ecfs/retrieve.cgi?native_or_pdf=pdf&id_document=6517587651)
[22] "FCC Consumer Advisory VoIP and 911 Service" (http://www.fcc.gov/cgb/consumerfacts/voip911.pdf). . Retrieved May 2, 2011.
[23] Keeping your telephone number when you change your service provider – FCC (http://www.fcc.gov/cgb/consumerfacts/numbport.html)
[24] VoIpservice.com (http://www.voipservice.com/sip-trunking)
[25] "RFC 3824 – Using E.164 numbers with the Session Initiation Protocol (SIP)" (http://www.packetizer.com/rfc/rfc3824/). The Internet Society. June 1, 2004. . Retrieved 2009-01-21.
[26] "Create a Skype Name" (http://www.skype.com/help/guides/createskypename_windows/). Skype. . Retrieved 2009-01-21.
[27] "RFC 3969 – The Internet Assigned Number Authority (IANA) Uniform Resource Identifier (URI) Parameter Registry for the Session Initiation Protocol (SIP)" (http://www.packetizer.com/rfc/rfc3969/). The Internet Society. December 1, 2004. . Retrieved 2009-01-21.
[28] "Your personal online number" (http://www.skype.com/allfeatures/onlinenumber/). Skype. . Retrieved 2009-01-21.
[29] "Application-level Network Interoperability and the Evolution of IMS" (http://ipcommunications.tmcnet.com/hot-topics/MCP/articles/1311-application-level-network-interoperability-the-evolution-ims.htm). TMCnet.com. May 24, 2006. . Retrieved 2009-01-21.
[30] Packetcable Implementation P557 – Jeff Riddel – ISBN 1-58705-181-8 Google Books Preview (http://books.google.com/books?id=8CNBbrxytcAC&pg=PA557&lpg=PA557&dq=PSTN+gateway+VOIP+impedance+mismatch)
[31] Taub, Eric (April 2, 2008). "VOIP System Security: Time to Worry, or Maybe Not" (http://bits.blogs.nytimes.com/2008/04/02/voip-system-security-time-to-worry-or-maybe-not/?scp=4&sq=voip&st=cse/). New York Times. . Retrieved 2009-03-02.
[32] Stanton, Ray (Secure VoIP- An Achievable Goal). *Computer Fraud & Security* **4**: 11–14. doi:10.1016/S1361-3723(06)70333-5.
[33] Stanton, R. (2006). "Secure VoIP- an achievable goal". *Computer Fraud & Security* **4**: 11–14. doi:10.1016/S1361-3723(06)70333-5.
[34] "Examining Two Well-Known Attacks on VOIP" (http://www.circleid.com/posts/examining_two_well_known_attacks_on_voip1/). CircleID. . Retrieved 2006-04-05.
[35] Skype.com (http://download.skype.com/share/security/2005-031 security evaluation.pdf), "Skype Security Evaluation", Tom Berson/Anagram Laboratories
[36] Disa.mil (http://iase.disa.mil/stigs/stig/VoIP-STIG-V2R2.pdf), Internet Protocol Telephony & Voice over Internet Protocol Security Technical Implementation Guide
[37] IJCSNS.org (http://paper.ijcsns.org/07_book/200706/20070610.pdf), Secure Voice-over-IP
[38] Sans.org (http://www.sans.org/reading_room/whitepapers/voip/secure_voice_over_ip_322), SANS Institute InfoSec Reading Room
[39] JHU.edu (http://www.clsp.jhu.edu/~cwhite/papers/asilo_04_LossConceal_final.pdf), Packet Loss Concealment in a Secure Voice over IP Environment
[40] GDC4S.com (http://www.gdc4s.com/documents/D-VIPER-14-1007_p11.pdf), State-of-the-art voice over IP encryptor
[41] Networkworld.com (http://www.networkworld.com/news/2009/041609-cellcrypt-secure-voip-heading-to.html), Cellcrypt secure VOIP heading to BlackBerry.
[42] Freesorftwaremagazine.com (http://www.freesoftwaremagazine.com/columns/secure_voip_calling_free_software_right_to_privacy), Secure VOIP calling, free software, and the right to privacy
[43] VOIPSA.org (http://voipsa.org/blog/2006/09/29/hello-mom-im-a-fake/), Blog: "Hello Mom, I'm a Fake!" (Telespoof and Fakecaller).

[44] 111th Congress (2009) (January 7, 2009). "S. 30 (111th)" (http://www.govtrack.us/congress/bills/111/s30). *Legislation*. GovTrack.us. . Retrieved June 25, 2012. "Truth in Caller ID Act of 2009"
[45] Federal Communications Commission (June 22, 2011). "Rules and Regulation Implementing the Truth in Caller ID Act of 2009" (http://www.fcc.gov/document/rules-and-regulation-implementing-truth-caller-id-act-2009). *fcc.gov*. . Retrieved June 25, 2012.
[46] Oldphoneworks.com (http://www.oldphoneworks.com/antique-phone-parts/parts-and-pieces/pulse-to-tone-converters/)
[47] Voip-Info.org (http://www.voip-info.org/wiki/view/FoIP), "FoIP".
[48] Soft-Switch.org (http://soft-switch.org/foip.html), Faxing over IP networks
[49] Umass.edu (http://www-net.cs.umass.edu/kurose/transport/UDP.html), UMass Discussion on UDP transmission Characteristics.
[50] Faqs.org (http://www.faqs.org/rfcs/rfc3362.html), RFC 3362-T.38
[51] FaxSuperstore.com (http://www.faxsuperstore.com/ricoh-4410nf.html), Ricoh 4410NF
[52] "Global VOIP Policy Status Matrix" (http://www.ipall.org/matrix/). Global IP Alliance (http://www.ipall.org). . Retrieved 2006-11-23.
[53] Proenza, Francisco J.. "The Road to Broadband Development in Developing Countries is through Competition Driven by Wireless and VOIP" (http://www.e-forall.org/pdf/Wireless&VOIP_10July2006.pdf) (PDF). . Retrieved 2008-04-07.
[54] (http://www.scribd.com/doc/101919043/TECHNICAL-NOTE-ON-ILLEGAL-INTERNATIONAL-LONG-DISTANCE-TELEPHONE-EXCHANGE-IN-INDIA) by Harish Kumar Gangwar , Technical Note on illegal ILD telephone Exchange in India
[55] Stripes.com (http://www.stripes.com/article.asp?section=104&article=41826&archive=true), Stars and Stripes: USFK deal keeps VoIP access for troops
[56] GPO.gov (http://www.access.gpo.gov/nara/cfr/waisidx_07/47cfr9_07.html), 47 C.F.R. pt. 9 (2007)
[57] FCC.gov (http://www.fcc.gov/voip/)
[58] Greenberg, Andy (May 15, 2008). "The State Of Cybersecurity Wiretapping's Fuzzy Future" (http://www.forbes.com/2008/05/15/wiretapping-voip-lichtblau-tech-security08-cx_ag_0515wiretap.html/). www.forbes.com. . Retrieved 2009-03-02.
[59] Voip | Define Voip at Dictionary.com (http://dictionary.reference.com/browse/voip). Dictionary.com. Retrieved 2011-07-15.
[60] Vinton G. Cerf, Robert E. Kahn, "A Protocol for Packet Network Intercommunication", IEEE Transactions on Communications, Vol. 22, No. 5, May 1974, pp. 637–648.
[61] "The Launch of NSFNET" (http://www.nsf.gov/about/history/nsf0050/internet/launch.htm). The National Science Foundation. . Retrieved 2009-01-21.
[62] "MTALK-Readme" (ftp://sunsite.unc.edu/pub/Linux/apps/sound/talk/mtalk.README) (TXT). Sunsite.edu. . Retrieved 2012-04-29.
[63] Keating, Tom. "Internet Phone Release 4" (http://blog.tmcnet.com/blog/tom-keating/docs/cti-buyers-guide-1996.pdf) (PDF). Computer Telephony Interaction Magazine. . Retrieved 2007-11-07.
[64] "The 10 that Established VOIP (Part 1: VocalTec)" (http://www.ilocus.com/2007/07/the_10_that_established_voip_p_2.html). iLocus. . Retrieved 2009-01-21.
[65] The free Library RADVision and Intel Target Compatibility Between RADVision's H.323/320 Videoconferencing Gateway And Intel's Business Video Conferencing And TeamStation Products. (http://www.thefreelibrary.com/RADVision+and+Intel+Target+Compatibility+Between+RADVision's+...-a019467970) June 2, 1997 VoiP Developer Solutions (http://www.radvision.com/Developer-Solutions/)
[66] "H.323 Visual telephone systems and equipment for local area networks which provide a non-guaranteed quality of service" (http://www.itu.int/rec/T-REC-H.323-199611-S/en). ITU-T. . Retrieved 2009-01-21.
[67] "RFC 2235" (http://www.faqs.org/rfcs/rfc2235.html). R. Zakon. . Retrieved 2009-01-21.
[68] "The 10 that Established VOIP (Part 2: Level 3)" (http://www.ilocus.com/2007/07/the_10_that_established_voip_p_1.html). iLocus. July 13, 2007. . Retrieved 2007-11-07.
[69] "RFC 2543, SIP: Session Initiation Protocol" (http://www.ietf.org/rfc/rfc2543.txt). Handley,Schulzrinne,Schooler,Rosenberg. . Retrieved 2009-01-21.
[70] "What is Asterisk" (http://www.asterisk.org/about). Asterisk.org. . Retrieved 2009-01-21.

External links

- Voice over IP (http://www.dmoz.org/Business/Telecommunications/Services/VoIP/) at the Open Directory Project

Global Dialing Scheme

The **Global Dialing Scheme** (GDS) is numbering plan for H.323 audio-visual communication networks (often used for videoconferencing). Based on the numerology provided by the United Nations International Telecommunications Union, GDS numerology resembles the international telephone system numbering plan, with some exceptions.

The Global Dialing Scheme uses a hierarchy of gatekeepers to route call set-up information nationally and internationally. National gatekeepers have knowledge of all zones within a country, World gatekeepers have knowledge of all National gatekeepers.[1]

Each basic number consists of four parts: <IAC><CC><OP><EN>.[2]

GDS History

In 2000, Victor Reis at HEAnet, Egon Verharen at SURFNet and Steve Williams at The Welsh Video Network began meeting by IP videoconference to discuss the requirements for an international H.323 videoconferencing dialling scheme that would allow E.164 dialling, avoiding clashes through the duplication of number spaces. Tim Poe from the University of North Carolina joined this meeting bringing USA involvement through VIDEnet and the NASM group.

It was envisaged that the GDS would be in place only for a few years until SIP/e-mail address dialling was adopted by vendors - in 2012 the GDS is still in use.

GDS Around the World

GDS is used heavily in many of the European countries. Educonf as part of GÉANT maintains a connectivity table[3] and interactive map[4] which displays which countries current run and maintain active GDS gatekeepers. Many countries also have a series of test number that can be dialed for both technical testing and GDS connectivity verification.

GDS in North America

The North American root gatekeepers serve the United States and its territories, Canada, Bermuda, and many Caribbean nations, including Anguilla, Antigua & Barbuda, Bahamas, Barbados, British Virgin Islands, Cayman Islands, Dominica, Dominican Republic, Grenada, Jamaica, Montserrat, St. Kitts and Nevis, St. Lucia, St. Vincent and the Grenadines, Trinidad and Tobago, and Turks & Caicos. Their purpose is to resolve h.323 numbers at the '001' prefix level under the Global Dialing Scheme (GDS) plan.

Terminology in this document follows the <IAC><CC><OP><EN> format of general GDS documentation.The ‘001’ above refers to the IAC of 00 and the CC of 1 for North America.

This North American node of the Global Dialing Scheme utilizes an enhanced version of the North American Numbering Plan (NANP) to distribute addresses. The address space is divided into two parts: North American E.164 Space and North American Super Space. North American E.164 Space correlates to existing telephone number assignments and is well-suited for IP telephony applications. North American Super Space utilizes unused NANP address space starting with 0 or 1 to create an address space that is separate from existing telephone numbering addresses. This North American Super Space is well suited to video over IP or other all-IP applications that desire to be distinct from telephony applications and NANP regulations.

GDS users in North America may request addresses in either or both spaces, if needed.

North American E.164 Space

Addresses allocated from this range will be based upon the ITU-T e.164 telephone number assigned to the current subscriber of a range of telephone numbers, rather than to the service provider carrying those numbers. For example, if a university held +1.919.226.6100 through +1.919.226.6199, then that university would be eligible for the GDS prefix 00191922661. That university could assign the remaining two digits to endpoints 00-99. Aside from maintaining direct inward dial (DID) capability for endpoints, there is no reason to limit endpoint numbering to two digits. For example, the university might use five digit endpoint numbers for a total address space of 001919226610000 through 001919226619999, yielding 10,000 usable addresses. Organizations that do not have a DID range may use this extension technique to map their entire address space onto a single 10-digit telephone number. Implementors of voice over IP applications may wish to adhere more strictly to the NANP numbering convention.

North America Super Space

The organizational prefixes <OP> of addresses in North American Super Space (NASS) start with a 0 or 1 immediately following the country code (1). These digits are not assigned under the NANP, being used rather for special indications as described in 1947 by AT&T and Bell Laboratories.

NASS addresses are of the form:

001PX9<EN>

Where P is a 0 or 1. X is a variable length string of digits consisting of any digit between 0 and 8. 9 is used as a delimiter. <EN> is a variable length user-defined number consisting of any digits 0-9. Thus, NASS addresses are variable in length.

Some examples of fully qualified GDS NASS addresses; all address below contain the GDS IAC (00) and CC (1) with endpoint numbers indicated as <EN>:

0010 Reserved

00119<EN> (OP = 19)

001109<EN> (OP = 109)

001119<EN> (OP = 119)

001129<EN> (OP = 129)

001139<EN> (OP = 139)

001189<EN> (OP = 189)

00110123456789<EN> (OP = 10123456789)

Actual Dialing With the onset of Cisco Telepresence which uses the Cisco Call Manager (CUCM), some modifications have been made for dialing with North America. Specifically within North America, the need to dial the '00' as part of the GDS string is no longer required. This was done to eliminate end-user confusion and bring GDS closer to look like real world dialing numbers. It was also done because the Cisco Call Manager only routes legit phone-based e.164 numbers even if they are not going over the telephone lines. So in order to allow SIP based devices to take place and use GDS, the removal of the 00 requirement was applied. Dialing with the '00' will still work, however it is not required as the translation for handling the '00' is done on the back end. Currently this only applies to GDS zones within North America.

References

[1] http://www.wvn.ac.uk/en/support/globaldiallingschemeexplained/
[2] GDS : The Global Dialing Scheme Explained (http://commons.internet2.edu/gds.html)
[3] http://educonf.geant2.net/monitoring/gds/: The Global Dialing Scheme Connectivity Matrix
[4] http://educonf.geant2.net/directory/map/: The Global Dialing Scheme Global Map

External links

- http://www.vide.net/help/gdsintro.shtml
- http://www.wvn.ac.uk/en/support/globaldiallingschemeexplained/
- http://www.vide.net/compliant/workgroups/nasm/
- http://commons.internet2.edu/gds.html
- http://educonf.geant2.net
- http://educonf.geant2.net/directory/map/

Session Description Protocol

The **Session Description Protocol** (**SDP**) is a format for describing streaming media initialization parameters. The IETF published the original specification as an IETF Proposed Standard in April 1998,[1] and subsequently published a revised specification as an IETF Proposed Standard as RFC 4566 in July 2006.[2]

SDP is intended for describing multimedia communication sessions for the purposes of session announcement, session invitation, and parameter negotiation. SDP does not deliver media itself but is used for negotiation between end points of media type, format, and all associated properties. The set of properties and parameters are often called a *session profile*. SDP is designed to be extensible to support new media types and formats.

SDP started off as a component of the Session Announcement Protocol (SAP), but found other uses in conjunction with Real-time Transport Protocol (RTP), Real-time Streaming Protocol (RTSP), Session Initiation Protocol (SIP) and even as a standalone format for describing multicast sessions.

Session description

A session is described by a series of fields, one per line.[3] The form of each field is as follows.

```
<character>=<value>
```

Where `<character>` is a single case-significant character and `value` is structured text whose format depends upon attribute type. Values are typically a UTF-8 encoding.[4] Whitespace is not allowed immediately to either side of the =.[5]

Within an SDP message there are three main sections, detailing the *session*, *timing*, and *media* descriptions. Each message may contain multiple *timing* and *media* descriptions. Names are only unique within the associated syntactic construct, i.e. within the *session*, *time*, or *media*.[6]

Optional values are specified with `=*` and each field must appear in the order shown below.

```
Session description
   v=  (protocol version)
   o=  (originator and session identifier)
   s=  (session name)
   i=* (session information)
   u=* (URI of description)
   e=* (email address)
```

```
        p=* (phone number)
        c=* (connection information—not required if included in all media)
        b=* (zero or more bandwidth information lines)
        One or more time descriptions ("t=" and "r=" lines; see below)
        z=* (time zone adjustments)
        k=* (encryption key)
        a=* (zero or more session attribute lines)
        Zero or more media descriptions
```

```
     Time description
        t=  (time the session is active)
        r=* (zero or more repeat times)
```

```
     Media description, if present
        m=  (media name and transport address)
        i=* (media title)
        c=* (connection information—optional if included at
             session level)
        b=* (zero or more bandwidth information lines)
        k=* (encryption key)
        a=* (zero or more media attribute lines)
```

Attributes

SDP uses attributes to extend the core protocol. Attributes can appear within the Session or Media sections and are scoped accordingly as *session-level* or *media-level*. New attributes are added to the standard occasionally through registration with IANA.[7] Attributes take two forms:

- A property form: `a=<flag>` conveys a property of the session.
- A value form: `a=<attribute>:<value>` provides a named parameter.

Time Formats

Absolute times are represented in Network Time Protocol format (the number of seconds since 1900). If the stop time is 0 then the session is "unbounded." If the start time is also zero then the session is considered "permanent." Unbounded and permanent sessions are discouraged but not prohibited. Intervals can be represented with Network Time Protocol times or in typed time: a value and time units (days ('d'), hours ('h'), minutes ('m') and seconds ('s')) sequence.

Thus an hour meeting from 10am on 1 August 2010, with a single repeat time a week later at the same time can be represented as:

```
        t=3487140000 3487143600
        r=604800 3600 0
```

Or using typed time:

```
        t=3487140000 3487143600
        r=7d 3600 0
```

Notes

[1] Handley, Mark; Van Jacobson (1998-04). "SDP: Session Description Protocol (RFC 2327)" (http://tools.ietf.org/html/rfc2327). IETF. . Retrieved 2008-04-19.

[2] Handley, Mark; Van Jacobson, Colin Perkins (2006-07). "SDP: Session Description Protocol (RFC 4566)" (http://tools.ietf.org/html/rfc4566). IETF. . Retrieved 2008-04-19.

[3] Each line is separated from the next by a carriage return/line feed sequence. Implementations are allowed to relax this to omit the carriage return and supply only the line feed.

[4] *session information* and *session name* values are subject to the encoding specified in any *charset* attribute of the section.

[5] RFC 4566 Section 5

[6] An In-Depth Overview of SDP (http://www.konnetic.com/Documents/KonneticSDPTechnicalOverview.pdf)

[7] Internet Assigned Numbers Authority SDP site (http://www.iana.org/assignments/sdp-parameters)

References

External links

- Rosenberg, J.; Schulzrinne, H. (2002-06). "An Offer/Answer Model with the Session Description Protocol(RFC 3264)" (http://tools.ietf.org/html/rfc3264). IETF. Retrieved 2010-07-25.

Session Initiation Protocol

The **Session Initiation Protocol** (**SIP**) is an IETF-defined signaling protocol widely used for controlling communication sessions such as voice and video calls over Internet Protocol (IP). The protocol can be used for creating, modifying and terminating two-party (unicast) or multiparty (multicast) sessions. Sessions may consist of one or several media streams.

Other SIP applications include video conferencing, streaming multimedia distribution, instant messaging, presence information, file transfer and online games.

The SIP protocol is an Application Layer protocol designed to be independent of the underlying Transport Layer; it can run on Transmission Control Protocol (TCP), User Datagram Protocol (UDP), or Stream Control Transmission Protocol (SCTP).[1] It is a text-based protocol, incorporating many elements of the Hypertext Transfer Protocol (HTTP) and the Simple Mail Transfer Protocol (SMTP).[2]

History

SIP was originally designed by Henning Schulzrinne and Mark Handley in 1996. In November 2000, SIP was accepted as a 3GPP signaling protocol and permanent element of the IP Multimedia Subsystem (IMS) architecture for IP-based streaming multimedia services in cellular systems. The latest version of the specification is RFC 3261 from the IETF Network Working Group published in June 2002.[3]

The U.S. National Institute of Standards and Technology (NIST), Advanced Networking Technologies Division provides a public domain implementation of the JAVA Standard for SIP[4] which serves as a reference implementation for the standard. The stack can work in proxy server or user agent scenarios and has been used in numerous commercial and research projects. It supports RFC 3261 in full and a number of extension RFCs including RFC 3265 (Subscribe / Notify) and RFC 3262 (Provisional Reliable Responses) etc.

Protocol design

SIP employs design elements similar to the HTTP request/response transaction model.[5] Each transaction consists of a client request that invokes a particular method or function on the server and at least one response. SIP reuses most of the header fields, encoding rules and status codes of HTTP, providing a readable text-based format.

Each resource of a SIP network, such as a user agent or a voicemail box, is identified by a uniform resource identifier (URI), based on the general standard syntax[6] also used in Web services and e-mail. A typical SIP URI is of the form: `sip:username:password@host:port`. The URI scheme used for SIP is `sip:`.

If secure transmission is required, the scheme `sips:` is used and mandates that each hop over which the request is forwarded up to the target domain must be secured with Transport Layer Security (TLS). The last hop from the proxy of the target domain to the user agent has to be secured according to local policies. TLS protects against attackers which try to listen on the signaling link. It does not provide real end-to-end security, since encryption is only hop-by-hop and every single intermediate proxy has to be trusted.

SIP works in concert with several other protocols and is only involved in the signaling portion of a communication session. SIP clients typically use TCP or UDP on port numbers 5060 and/or 5061 to connect to SIP servers and other SIP endpoints. Port 5060 is commonly used for non-encrypted signaling traffic whereas port 5061 is typically used for traffic encrypted with Transport Layer Security (TLS). SIP is primarily used in setting up and tearing down voice or video calls. It also allows modification of existing calls. The modification can involve changing addresses or ports, inviting more participants, and adding or deleting media streams. SIP has also found applications in messaging applications, such as instant messaging, and event subscription and notification. A suite of SIP-related Internet Engineering Task Force (IETF) rules define behavior for such applications. The voice and video stream communications in SIP applications are carried over another application protocol, the Real-time Transport Protocol (RTP). Parameters (port numbers, protocols, codecs) for these media streams are defined and negotiated using the Session Description Protocol (SDP) which is transported in the SIP packet body.

A motivating goal for SIP was to provide a signaling and call setup protocol for IP-based communications that can support a superset of the call processing functions and features present in the public switched telephone network (PSTN). SIP by itself does not define these features; rather, its focus is call-setup and signaling. The features that permit familiar telephone-like operations: dialing a number, causing a phone to ring, hearing ringback tones or a busy signal - are performed by proxy servers and user agents. Implementation and terminology are different in the SIP world but to the end-user, the behavior is similar.

SIP-enabled telephony networks can also implement many of the more advanced call processing features present in Signaling System 7 (SS7), though the two protocols themselves are very different. SS7 is a centralized protocol, characterized by a complex central network architecture and dumb endpoints (traditional telephone handsets). SIP is a peer-to-peer protocol, thus it requires only a simple (and thus scalable) core network with intelligence distributed to the network edge, embedded in endpoints (terminating devices built in either hardware or software). SIP features are implemented in the communicating endpoints (i.e. at the edge of the network) contrary to traditional SS7 features, which are implemented in the network.

Although several other VoIP signaling protocols exist (such as BICC, H.323, MGCP, MEGACO), SIP is distinguished by its proponents for having roots in the IP community rather than the telecommunications industry. SIP has been standardized and governed primarily by the IETF, while other protocols, such as H.323, have traditionally been associated with the International Telecommunication Union (ITU).

The first proposed standard version (SIP 1.0) was defined by RFC 2543. This version of the protocol was further refined to version 2.0 and clarified in RFC 3261, although some implementations are still relying on the older definitions.

Network elements

SIP also defines server network elements. Although two SIP endpoints can communicate without any intervening SIP infrastructure, which is why the protocol is described as peer-to-peer, this approach is often impractical for a public service. RFC 3261 defines these server elements.

User Agent

A *SIP user agent* (UA) is a logical network end-point used to create or receive SIP messages and thereby manage a SIP session. A SIP UA can perform the role of a *User Agent Client* (UAC), which sends SIP requests, and the *User Agent Server* (UAS), which receives the requests and returns a SIP response. These roles of UAC and UAS only last for the duration of a SIP transaction.[7]

A SIP phone is a SIP user agent that provides the traditional call functions of a telephone, such as dial, answer, reject, hold/unhold, and call transfer.[8][9] SIP phones may be implemented as a hardware device or as a softphone. As vendors increasingly implement SIP as a standard telephony platform, often driven by 4G efforts, the distinction between hardware-based and software-based SIP phones is being blurred and SIP elements are implemented in the basic firmware functions of many IP-capable devices. Examples are devices from Nokia and Research in Motion.[10]

In SIP, as in HTTP, the user agent may identify itself using a message header field 'User-Agent', containing a text description of the software/hardware/product involved. The User-Agent field is sent in request messages, which means that the receiving SIP server can see this information. SIP network elements sometimes store this information,[11] and it can be useful in diagnosing SIP compatibility problems.

Proxy server

An intermediary entity that acts as both a server (UAS) and a client (UAC) for the purpose of making requests on behalf of other clients. A proxy server primarily plays the role of routing, which means its job is to ensure that a request is sent to another entity "closer" to the targeted user. Proxies are also useful for enforcing policy (for example, making sure a user is allowed to make a call). A proxy interprets, and, if necessary, rewrites specific parts of a request message before forwarding it.

Registrar

A server that accepts REGISTER requests and places the information it receives in those requests into the location service for the domain it handles which registers one or more IP addresses to a certain SIP URI, indicated by the `sip:` scheme, although other protocol schemes are possible (such as `tel:`). More than one user agent can register at the same URI, with the result that all registered user agents will receive a call to the SIP URI.

SIP registrars are logical elements, and are commonly co-located with SIP proxies. But it is also possible and often good for network scalability to place this location service with a redirect server.

Redirect server

A user agent server that generates 3xx (Redirection) responses to requests it receives, directing the client to contact an alternate set of URIs. The redirect server allows proxy servers to direct SIP session invitations to external domains.

Session border controller

Session border controllers Serve as *middle boxes* between UA and SIP server for various types of functions, including network topology hiding, and assistance in *NAT traversal*.

Gateway

Gateways can be used to interface a SIP network to other networks, such as the public switched telephone network, which use different protocols or technologies.

SIP messages

SIP is a text-based protocol with syntax similar to that of HTTP. There are two different types of SIP messages: requests and responses. The first line of a request has a *method*, defining the nature of the request, and a Request-URI, indicating where the request should be sent.[12] The first line of a response has a *response code*.

For SIP requests, RFC 3261 defines the following methods:[13]

- REGISTER: Used by a UA to indicate its current IP address and the URLs for which it would like to receive calls.
- INVITE: Used to establish a media session between user agents.
- ACK: Confirms reliable message exchanges.
- CANCEL: Terminates a pending request.
- BYE: Terminates a session between two users in a conference.
- OPTIONS: Requests information about the capabilities of a caller, without setting up a call.

A new method has been introduced in SIP in RFC 3262: [14]

- PRACK (Provisional Response Acknowledgement): PRACK improves network reliability by adding an acknowledgement system to the provisional Responses (1xx). PRACK is sent in response to provisional response (1xx).

The SIP response types defined in RFC 3261 fall in one of the following categories:[15]

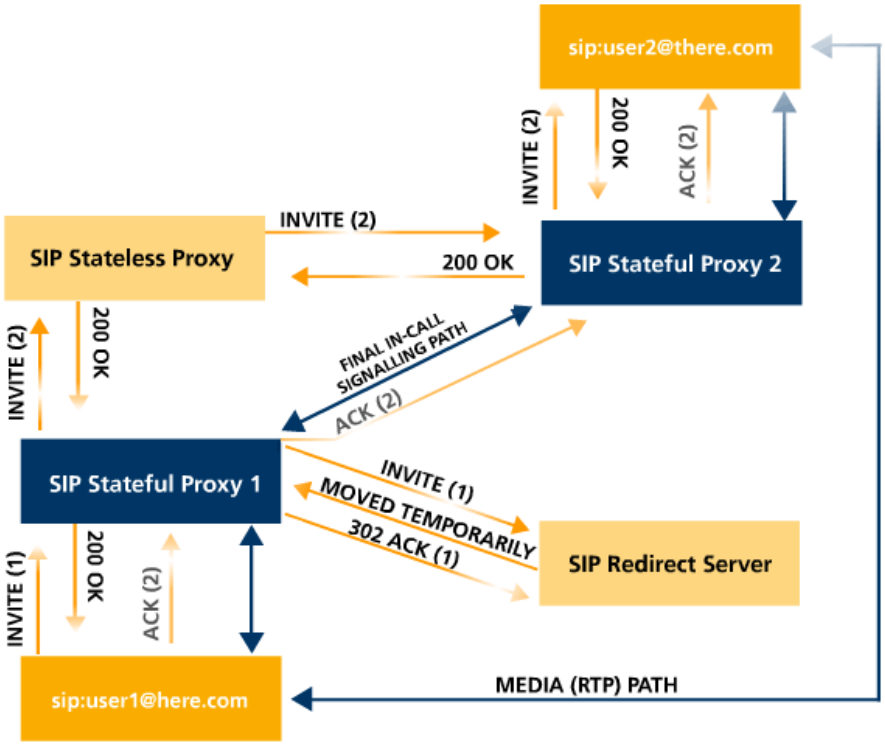
- Provisional (1xx): Request received and being processed.
- Success (2xx): The action was successfully received, understood, and accepted.
- Redirection (3xx): Further action needs to be taken (typically by sender) to complete the request.
- Client Error (4xx): The request contains bad syntax or cannot be fulfilled at the server.
- Server Error (5xx): The server failed to fulfill an apparently valid request.
- Global Failure (6xx): The request cannot be fulfilled at any server.

Transactions

SIP makes use of transactions to control the exchanges between participants and deliver messages reliably. The transactions maintain an internal state and make use of timers. *Client Transactions* send requests and *Server Transactions* respond to those requests with one-or-more responses. The responses may include zero-or-more Provisional (1xx) responses and one-or-more final (2xx-6xx) responses.

Transactions are further categorized as either *Invite* or *Non-Invite*. *Invite* transactions differ in that they can establish a long-running conversation, referred to as a *Dialog* in SIP, and so include an acknowledgment (ACK) of any non-failing final response (e.g. 200 OK).

Because of these transactional mechanisms, SIP can make use of un-reliable transports such as User Datagram Protocol (UDP).



If we take the above example, User1's UAC uses an *Invite Client Transaction* to send the initial INVITE (1) message. If no response is received after a timer controlled wait period the UAC may have chosen to terminate the transaction or retransmit the INVITE. However, once a response was received, User1 was confident the INVITE was delivered reliably. User1's UAC then must acknowledge the response. On delivery of the ACK (2) both sides of the transaction are complete. And in this case, a Dialog may have been established.[16]

Instant messaging and presence

The Session Initiation Protocol for Instant Messaging and Presence Leveraging Extensions (SIMPLE) is the SIP-based suite of standards for instant messaging and presence information. MSRP (Message Session Relay Protocol) allows instant message sessions and file transfer.

Conformance testing

TTCN-3 test specification language is used for the purposes of specifying conformance tests for SIP implementations. SIP test suite is developed by a Specialist Task Force at ETSI (STF 196).[17] The SIP developer community meets regularly at the SIP Forum SIPit [18] events to test interoperability and test implementations of new RFCs.

Applications

The market for consumer SIP devices continues to expand; there are many devices such as SIP Terminal Adapters, SIP Gateways, and SIP Trunking services providing replacements for ISDN telephone lines.

Many VoIP phone companies allow customers to use their own SIP devices, such as SIP-capable telephone sets, or softphones.

SIP-enabled video surveillance cameras can make calls to alert the owner or operator that an event has occurred; for example, to notify that motion has been detected out-of-hours in a protected area.

SIP is used in audio over IP for broadcasting applications where it provides an interoperable means for audio interfaces from different manufacturers to make connections with one another.[19]

SIP-ISUP interworking

SIP-I, or the Session Initiation Protocol with encapsulated ISUP, is a protocol used to create, modify, and terminate communication sessions based on ISUP using SIP and IP networks. Services using SIP-I include voice, video telephony, fax and data. SIP-I and SIP-T[20] are two protocols with similar features, notably to allow ISUP messages to be transported over SIP networks. This preserves all of the detail available in the ISUP header, which is important as there are many country-specific variants of ISUP that have been implemented over the last 30 years, and it is not always possible to express all of the same detail using a native SIP message. SIP-I was defined by the ITU-T, where SIP-T was defined via the IETF RFC route.[21]

References

[1] RFC 4168, *The Stream Control Transmission Protocol (SCTP) as a Transport for the Session Initiation Protocol (SIP)*, IETF, The Internet Society (2005)

[2] Johnston, Alan B. (2004). *SIP: Understanding the Session Initiation Protocol, Second Edition*. Artech House. ISBN 1-58053-168-7.

[3] "SIP core working group charter" (http://www.ietf.org/dyn/wg/charter/sipcore-charter.html). Ietf.org. 2010-12-07. . Retrieved 2011-01-11.

[4] "JAIN SIP project" (http://java.net/projects/jsip). . Retrieved 2011-07-26.

[5] William Stallings, p.209

[6] RFC 3986, *Uniform Resource Identifiers (URI): Generic Syntax*, IETF, The Internet Society (2005)

[7] RFC 3261, *SIP: Session Initiation Protocol*

[8] Azzedine (2006). *Handbook of algorithms for wireless networking and mobile computing* (http://books.google.com/books?id=b8oisvv6fDAC&pg=PT774). CRC Press. p. 774. ISBN 978-1-58488-465-1. .

[9] Porter, Thomas; Andy Zmolek, Jan Kanclirz, Antonio Rosela (2006). *Practical VoIP Security* (http://books.google.com/books?id=BYxdyekyRlwC&pg=PA76). Syngress. pp. 76–77. ISBN 978-1-59749-060-3. .

[10] ""BlackBerry MVS Software"" (http://na.blackberry.com/eng/services/business/blackberry_mvs/). Na.blackberry.com. . Retrieved 2011-01-11.

[11] "User-Agents We Have Known " (http://www.voipuser.org/forum_topic_14998.html)VoIP User.org

[12] Stallings, p.214
[13] Stallings, pp.214-215
[14] http://www.ietf.org/rfc/rfc3262.txt
[15] Stallings, pp.216-217
[16] James Wright. "SIP - An Introduction" (http://www.konnetic.com/Documents/KonneticSIPIntroduction.pdf) (PDF). Konnetic. . Retrieved 2011-01-11.
[17] Experiences of Using TTCN-3 for Testing SIP and also OSP (http://portal.etsi.org/ptcc/downloads/TTCN3SIPOSP.pdf)
[18] http://www.sipit.net/
[19] Jonsson, Lars; Mathias Coinchon (2008). "Streaming audio contributions over IP" (http://tech.ebu.ch/webdav/site/tech/shared/techreview/trev_2008-Q1_coinchon.pdf) (PDF). *EBU Technical Review*. . Retrieved 2010-12-27.
[20] "RFC3372: SIP-T Context and Architectures" (http://www.ietf.org/rfc/rfc3372.txt). 2002-09. . Retrieved 2011-01-11.
[21] White Paper: "Why SIP-I? A Switching Core Protocol Recommendation" (http://www.4gamericas.org/documents/3G_Americas_SIP-I_White_Paper_August_2007-FINAL.pdf)

External links

- Computers/Internet/Protocols/SIP/ (http://www.dmoz.org/Computers/Internet/Protocols/SIP//) at the Open Directory Project
- Henning Schulzrinne's SIP homepage (http://www.cs.columbia.edu/sip/) hosted by Columbia University
- SIP Latest specifications (http://www.sipknowledge.com/eBooks.htm) hosted by SIPKnowledge
- IANA: SIP Parameters (http://www.iana.org/assignments/sip-parameters)
- IANA: SIP Event Types Namespace (http://www.iana.org/assignments/sip-events/sip-events.xhtml)

List of SIP response codes

SIP responses are the codes used by Session Initiation Protocol for communication. They complement the SIP Requests, which are used to initiate action such as a phone conversation. Note that the Reason Phrases of the responses listed below are only the recommended examples, and can be replaced with local equivalents without affecting the protocol.

1xx—Provisional Responses

100 Trying

> Extended search being performed may take a significant time so a forking proxy must send a 100 Trying response

180 Ringing

> Destination user agent received INVITE, and is alerting user of call.[1]

181 Call is Being Forwarded

> Servers can optionally send this response to indicate a call is being forwarded.[1]

182 Queued

> Indicates that the destination was temporarily unavailable, so the server has queued the call until the destination is available. A server may send multiple 182 responses to update progress of the queue.[1]

183 Session in Progress

> This response may be used to send extra information for a call which is still being set up.[1]

199 Early Dialog Terminated

> Can used by User Agent Server to indicate to upstream SIP entities (including the User Agent Client (UAC)) that an early dialog has been terminated.[2]

2xx—Successful Responses

200 OK

Indicates the request was successful.

202 Accepted

Indicates that the request has been accepted for processing, but the processing has not been completed.

204 No Notification

Indicates the request was successful, but the corresponding response will not be received.[3]

3xx—Redirection Responses

- 300 Multiple Choices
- 301 Moved Permanently
- 302 Moved Temporarily
- 305 Use Proxy
- 380 Alternative Service

4xx—Client Failure Responses

400 Bad Request

The request could not be understood due to malformed syntax.[1]

401 Unauthorized

The request requires user authentication. This response is issued by UASs and registrars. [1]

402 Payment Required

Reserved for future use. [1]

403 Forbidden

The server understood the request, but is refusing to fulfill it.

404 Not Found (User not found)

The server has definitive information that the user does not exist at the domain specified in the Request-URI. This status is also returned if the domain in the Request-URI does not match any of the domains handled by the recipient of the request.

405 Method Not Allowed

The method specified in the Request-Line is understood, but not allowed for the address identified by the Request-URI.

406 Not Acceptable

The resource identified by the request is only capable of generating response entities that have content characteristics not acceptable according to the Accept header field sent in the request.

407 Proxy Authentication Required

The request requires user authentication. This response is issued by proxys. [1]

408 Request Timeout

Couldn't find the user in time.[1]

409 Conflict

User already registered (RFC 2543)

410 Gone

The user existed once, but is not available here any more.[1]

412 Conditional Request Failed

Conditional Request Failed (RFC 3903)

413 Request Entity Too Large

Request body too large.[1]

414 Request-URI Too Long

The server is refusing to service the request because the Request-URI is longer than the server is willing to interpret.[1]

415 Unsupported Media Type

Request body in a format not supported.[1]

416 Unsupported URI Scheme

Request-URI is unknown to the server.[1]

417 Unknown Resource-Priority

Unknown Resource-Priority (RFC 4412)

420 Bad Extension

Bad SIP Protocol Extension used, not understood by the server.[1]

421 Extension Required

The server needs a specific extension not listed in the Supported header.[1]

422 Session Interval Too Small

It is generated by a UAS or proxy when a request contains a Session-Expires header field with a duration below the minimum timer for the server (RFC 4028)

423 Interval Too Brief

Expiration time of the resource is too short.[1]

424 Bad Location Information

Bad Location Information (RFC 6442)

428 Use Identity Header

Use Identity Header (RFC 4474)

429 Provide Referrer Identity

Provide Referrer Identity (RFC 3892)

433 Anonymity Disallowed

Anonymity Disallowed (RFC 5079)

436 Bad Identity-Info

Bad Identity-Info (RFC 4474)

437 Unsupported Certificate

Unsupported Certificate (RFC 4474)

438 Invalid Identity Header

Invalid Identity Header (RFC 4474)

480 Temporarily Unavailable

Callee currently unavailable.[1]

481 Call/Transaction Does Not Exist

Server received a request that does not match any dialog or transaction.[1]

482 Loop Detected.

Server has detected a loop.[1]

483 Too Many Hops

Max-Forwards header has reach value '0'.[1]

484 Address Incomplete

Request-URI incomplete.[1]

485 Ambiguous

Request-URI is ambiguous.[1]

486 Busy Here

Callee is busy.[1]

487 Request Terminated

Request has terminated by bye or cancel.[1]

488 Not Acceptable Here

Some aspects of the session description of the Request-URI is not acceptable.[1]

489 Bad Event

Bad Event (RFC 3265)

491 Request Pending

Server has some pending request from the same dialog.[1]

493 Undecipherable

Request contains an encrypted MIME body, which recipient can not decrypt.[1]

494 Security Agreement Required

Security Agreement Required (RFC 3329)

5xx—Server Failure Responses

- 500 Server Internal Error
- 501 Not Implemented: The SIP request method is not implemented here
- 502 Bad Gateway
- 503 Service Unavailable
- 504 Server Time-out
- 505 Version Not Supported: The server does not support this version of the SIP protocol
- 513 Message Too Large
- 580 Precondition Failure [4]

6xx—Global Failure Responses

- 600 Busy Everywhere
- 603 Decline
- 604 Does Not Exist Anywhere
- 606 Not Acceptable

References

[1] Rosenberg, J.; Schulzrinne, H.; Camarillo, G.; Johnston, A.; Peterson, F.; Sparks, R.; Handley, M.; Schooler, E. (June 2002). *SIP: Session Initiation Protocol* (https://tools.ietf.org/html/rfc3261). IETF. RFC 3261. . Retrieved January 20, 2012.

[2] Holmberg, C. (May 2011). *Session Initiation Protocol (SIP) Response Code for Indication of Terminated Dialog* (https://tools.ietf.org/html/rfc6228). IETF. RFC 6228. .

[3] Niemi, A. (May 2010). Willis, D. ed. *An Extension to Session Initiation Protocol (SIP) Events for Conditional Event Notification* (https://tools.ietf.org/html/rfc5839). IETF. RFC 5839. .

[4] Rosenberg, J. (October 2002). Camarillo, G.; Marshall, W. eds. *Integration of Resource Management and Session Initiation Protocol (SIP)* (https://tools.ietf.org/html/rfc3312). IETF. RFC 3312. .

External links

- Mapping SIP Error Messages to DSS1 codes (http://www.en.voipforo.com/SIP/SIP_error_messages.php)
- Session Initiation Protocol (SIP) Parameters (http://www.iana.org/assignments/sip-parameters) Contains a registry of different SIP parameters, including response codes

SIP Trunking

SIP trunking is a Voice over Internet Protocol (VoIP) and streaming media service based on the Session Initiation Protocol (SIP)[1] by which Internet telephony service providers (ITSPs) deliver telephone services and unified communications to customers equipped with SIP-based private branch exchange (IP-PBX) and Unified Communications facilities. Most Unified Communications software applications provide voice, video and other streaming media applications such as desktop sharing, web conferencing, and shared whiteboard.

Domains

The architecture of SIP trunking provides a partitioning of the Unified Communications network into two different domains of expertise:[2]

- Private Domain: a VoIP solution realized at the customer's home that takes advantage of phone and unified communication services;
- Public Domain: full VoIP access solution to the PSTN / PLMN property and responsibility of the ITSP that provides phone service.

The interconnection between the two domains must occur through a SIP trunk.

The interconnection between the two domains, created by transport via the Internet Protocol (IP), involves setting specific rules and regulations as well as the ability to handle some services and protocols that fall into the well-defined name of SIP trunking.

The ITSP is completely responsible to the applicable regulatory authority regarding all the following law obligations of the Public Domain:[3]

- Tracking traffic;
- Identification identity of users;
- Implementation of the lawful interception mechanisms.

The private domain instead, by nature, is not subject to particular constraints of law, and may be either the responsibility of the ITSP, the end user (enterprise) or of a third party who provides the voice services to the company.

Architecture

In each domain there are elements that perform the characteristic features requested to that domain, in particular the result (as part of any front-end network to the customer) is logically divided into two levels:

- The control of access (Class 5 softswitch);
- Network-border elements[4][5] that separate the Public Domain from the Private Domain, implementing all the appropriate ITSP phone security policies.

The private domain consists of three levels:

- Corporate-Border Elements that separate the Public Domain from the Private Domain, implementing all the appropriate company security policies.
- Central Corporate Switching Node;
- IP-PBXs.

References

[1] "SIP Trunking Network" (http://www.siptrunk.org/). Ingate Systems. .

[2] Gaboli, Ivan; Puglia, Virgilio (Jan 2011). "SIP Trunking the route to the new VoIP services" (http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5682153). *Kaleidoscope: Beyond the Internet? – Innovations for future networks and services, 2010 ITU-T, 13-15 Dec 2010*. IEEE. ISBN 978-1-4244-8272-6. .

[3] "Legal issues in different countries" (http://voippla.net/link4.html). .

[4] "Role of Border Element" (http://www.cisco.com/en/US/prod/collateral/routers/ps9343/white_paper_c11-540690.html). Cisco. .

[5] "Acme Packet Net-Net session border controllers" (http://www.voiplogic.com/stuff/contentmgr/files/8a48b03d58b019a3836f9b7f1b4bd8e2/technical_specifications/ds_apkt_net_net_sbc.pdf). Acme Packet. .

External links

- (ITU-T | Kaleidoscope event 2010 | conference) (http://www.itu.int/dms_pub/itu-t/oth/29/04/T29040000030003PPTE.ppt)

Back-to-back user agent

A **back-to-back user agent** (**B2BUA**) is a logical network element in Session Initiation Protocol (SIP) applications.[1] SIP is a signaling protocol to manage multimedia Voice over Internet Protocol (VoIP) telephone calls. A back-to-back user agent operates between both end points of a phone call or communications session and divides the communication channel into two call legs and mediates all SIP signaling between both ends of the call, from call establishment to termination. As all control messages for each call flow through the B2BUA, a service provider may implement value-added features available during the call.

In the originating call leg the B2BUA acts as a *user agent server* (UAS) and processes the request as a *user agent client* (UAC) to the destination end, handling the signaling between end points back-to-back. A B2BUA maintains complete state for the calls it handles. Each side of a B2BUA operates as a standard SIP network element as specified in RFC 3261.

A B2BUA may provide the following functions:

- call management (billing, automatic call disconnection, call transfer, etc.)
- network interworking (perhaps with protocol adaptation)
- hiding of network internals (private addresses, network topology, etc.)

Often, B2BUAs are implemented in media gateways to also bridge the media streams for full control over the session.

A signaling gateway, part of a session border controller, is an example of a B2BUA.

References

[1] RFC 3261, *SIP: Session Initiation Protocol*, IETF, The Internet Society (2002)

External links

- Open Source Sippy SIP B2BUA (http://www.b2bua.org)

H.323

H.323 is a recommendation from the ITU Telecommunication Standardization Sector (ITU-T) that defines the protocols to provide audio-visual communication sessions on any packet network. The H.323 standard addresses call signaling and control, multimedia transport and control, and bandwidth control for point-to-point and multi-point conferences.[1]

It is widely implemented[2] by voice and videoconferencing equipment manufacturers, is used within various Internet real-time applications such as GnuGK and NetMeeting and is widely deployed worldwide by service providers and enterprises for both voice and video services over IP networks.

It is a part of the ITU-T H.32x series of protocols, which also address multimedia communications over ISDN, the PSTN or SS7, and 3G mobile networks.

H.323 call signaling is based on the ITU-T Recommendation Q.931 protocol and is suited for transmitting calls across networks using a mixture of IP, PSTN, ISDN, and QSIG over ISDN. A call model, similar to the ISDN call model, eases the introduction of IP telephony into existing networks of ISDN-based PBX systems, including transitions to IP-based PBXs.

Within the context of H.323, an IP-based PBX might be a gatekeeper or other call control element which provides service to telephones or videophones. Such a device may provide or facilitate both basic services and supplementary services, such as call transfer, park, pick-up, and hold.

History

The first version of H.323 was published by the ITU in November 1996[3] with an emphasis of enabling videoconferencing capabilities over a local area network (LAN), but was quickly adopted by the industry as a means of transmitting voice communication over a variety of IP networks, including WANs and the Internet (see VoIP).

Over the years, H.323 has been revised and re-published with enhancements necessary to better-enable both voice and video functionality over packet-switched networks, with each version being backward-compatible with the previous version.[4] Recognizing that H.323 was being used for communication, not only on LANs, but over WANs and within large carrier networks, the title of H.323 was changed when published in 1998.[5] The title, which has since remained unchanged, is "Packet-Based Multimedia Communications Systems." The current version of H.323 was approved in 2009.[6]

One strength of H.323 was the relatively early availability of a set of standards, not only defining the basic call model, but also the supplementary services needed to address business communication expectations.

H.323 was the first VoIP standard to adopt the Internet Engineering Task Force (IETF) standard Real-time Transport Protocol (RTP) to transport audio and video over IP networks.

Protocols

H.323 is a system specification that describes the use of several ITU-T and IETF protocols. The protocols that comprise the core of almost any H.323 system are:[7]

- H.225.0 Registration, Admission and Status (RAS), which is used between an H.323 endpoint and a Gatekeeper to provide address resolution and admission control services.
- H.225.0 Call Signaling, which is used between any two H.323 entities in order to establish communication.
- H.245 control protocol for multimedia communication, which describes the messages and procedures used for capability exchange, opening and closing logical channels for audio, video and data, control and indications.
- Real-time Transport Protocol (RTP), which is used for sending or receiving multimedia information (voice, video, or text) between any two entities.

Many H.323 systems also implement other protocols that are defined in various ITU-T Recommendations to provide supplementary services support or deliver other functionality to the user. Some of those Recommendations are:

- H.235 series describes security within H.323, including security for both signaling and media.
- H.239 describes dual stream use in videoconferencing, usually one for live video, the other for still images.
- H.450 series describes various supplementary services.
- H.460 series defines optional extensions that might be implemented by an endpoint or a Gatekeeper, including ITU-T Recommendations H.460.17, H.460.18, and H.460.19 for Network address translation (NAT) / Firewall (FW) traversal.

In addition to those ITU-T Recommendations, H.323 implements various IETF Request for Comments (RFCs) for media transport and media packetization, including the Real-time Transport Protocol (RTP).

Codecs

H.323 utilizes both ITU-defined codecs and codecs defined outside the ITU. Codecs that are widely implemented by H.323 equipment include:

- Audio codecs: G.711, G.729 (including G.729a), G.723.1, G.726, G.722, G.728, Speex, AAC-LD
- Text codecs: T.140
- Video codecs: H.261, H.263, H.264

All H.323 terminals providing video communications shall be capable of encoding and decoding video according to H.261 QCIF. All H.323 terminals shall have an audio codec and shall be capable of encoding and decoding speech according to ITU-T Rec. G.711. All terminals shall be capable of transmitting and receiving A-law and μ-law. Support for other audio and video codecs is optional.[6]

Architecture

The H.323 system defines several network elements that work together in order to deliver rich multimedia communication capabilities. Those elements are Terminals, Multipoint Control Units (MCUs), Gateways, Gatekeepers, and Border Elements. Collectively, terminals, multipoint control units and gateways are often referred to as **endpoints**.

While not all elements are required, at least two terminals are required in order to enable communication between two people. In most H.323 deployments, a gatekeeper is employed in order to, among other things, facilitate address resolution.

H.323 Network Elements

Terminals

Terminals in an H.323 network are the most fundamental elements in any H.323 system, as those are the devices that users would normally encounter. They might exist in the form of a simple IP phone or a powerful high-definition videoconferencing system.

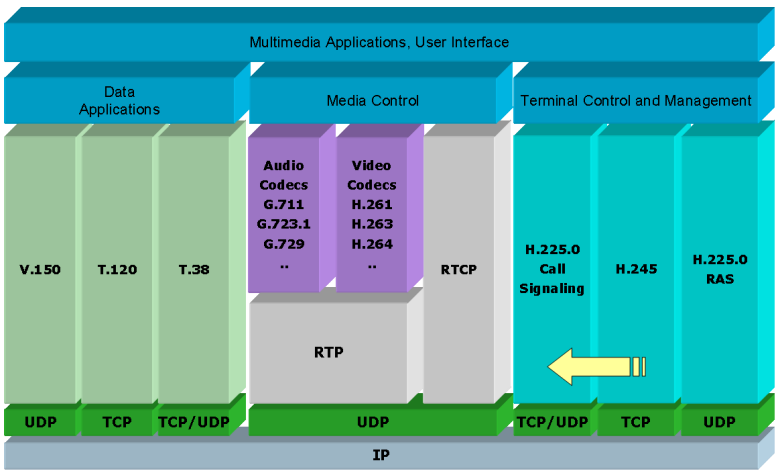


Figure 1 - A complete, sophisticated protocol stack

Inside an H.323 terminal is something referred to as a Protocol stack, which implements the functionality defined by the H.323 system. The protocol stack would include an implementation of the basic protocol defined in ITU-T Recommendation H.225.0 and H.245, as well as RTP or other protocols described above.

The diagram, figure 1, depicts a complete, sophisticated stack that provides support for voice, video, and various forms of data communication. In reality, most H.323 systems do not implement such a wide array of capabilities, but the logical arrangement is useful in understanding the relationships.

Multipoint Control Units

A Multipoint Control Unit (MCU) is responsible for managing multipoint conferences and is composed of two logical entities referred to as the Multipoint Controller (MC) and the Multipoint Processor (MP). In more practical terms, an MCU is a conference bridge not unlike the conference bridges used in the PSTN today. The most significant difference, however, is that H.323 MCUs might be capable of mixing or switching video, in addition to the normal audio mixing done by a traditional conference bridge. Some MCUs also provide multipoint data collaboration capabilities. What this means to the end user is that, by placing a video call into an H.323 MCU, the user might be able to see all of the other participants in the conference, not only hear their voices.

Gateways

Gateways are devices that enable communication between H.323 networks and other networks, such as PSTN or ISDN networks. If one party in a conversation is utilizing a terminal that is not an H.323 terminal, then the call must pass through a gateway in order to enable both parties to communicate.

Gateways are widely used today in order to enable the legacy PSTN phones to interconnect with the large, international H.323 networks that are presently deployed by services providers. Gateways are also used within the enterprise in order to enable enterprise IP phones to communicate through the service provider to users on the PSTN.

Gateways are also used in order to enable videoconferencing devices based on H.320 and H.324 to communicate with H.323 systems. Most of the third generation (3G) mobile networks deployed today utilize the H.324 protocol and are able to communicate with H.323-based terminals in corporate networks through such gateway devices.

Gatekeepers

A Gatekeeper is an optional component in the H.323 network that provides a number of services to terminals, gateways, and MCU devices. Those services include endpoint registration, address resolution, admission control, user authentication, and so forth. Of the various functions performed by the gatekeeper, address resolution is the most important as it enables two endpoints to contact each other without either endpoint having to know the IP address of the other endpoint.

Gatekeepers may be designed to operate in one of two signaling modes, namely "direct routed" and "gatekeeper routed" mode. Direct routed mode is the most efficient and most widely deployed mode. In this mode, endpoints utilize the RAS protocol in order to learn the IP address of the remote endpoint and a call is established directly with the remote device. In the gatekeeper routed mode, call signaling always passes through the gatekeeper. While the latter requires the gatekeeper to have more processing power, it also gives the gatekeeper complete control over the call and the ability to provide supplementary services on behalf of the endpoints.

H.323 endpoints use the RAS protocol to communicate with a gatekeeper. Likewise, gatekeepers use RAS to communicate with other gatekeepers.

A collection of endpoints that are registered to a single Gatekeeper in H.323 is referred to as a “zone”. This collection of devices does not necessarily have to have an associated physical topology. Rather, a zone may be entirely logical and is arbitrarily defined by the network administrator.

Gatekeepers have the ability to neighbor together so that call resolution can happen between zones. Neighboring facilitates the use of dial plans such as the Global Dialing Scheme. Dial plans facilitate “inter-zone” dialing so that two endpoints in separate zones can still communicate with each other.

Border Elements and Peer Elements

Border Elements and Peer Elements are optional entities similar to a Gatekeeper, but that do not manage endpoints directly and provide some services that are not described in the RAS protocol. The role of a border or peer element is understood via the definition of an "administrative domain".

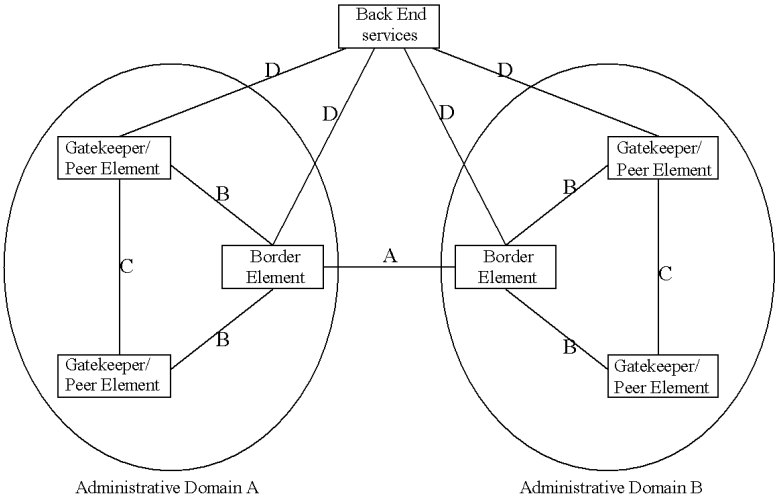


Figure 2 - An illustration of an administrative domain with border elements, peer elements, and gatekeepers

An administrative domain is the collection of all zones that are under the control of a single person or organization, such as a service provider. Within a service provider network there may be hundreds or thousands of gateway devices, telephones, video terminals, or other H.323 network elements. The service provider might arrange devices into "zones" that enable the service provider to best manage all of the devices under its control, such as logical arrangement by city. Taken together, all of the zones within the service provider network would appear to another service provider as an "administrative domain".

The border element is a signaling entity that generally sits at the edge of the administrative domain and communicates with another administrative domain. This communication might include such things as access authorization information; call pricing information; or other important data necessary to enable communication

between the two administrative domains.

Peer elements are entities within the administrative domain that, more or less, help to propagate information learned from the border elements throughout the administrative domain. Such architecture is intended to enable large-scale deployments within carrier networks and to enable services such as clearinghouses.

The diagram, figure 2, provides an illustration of an administrative domain with border elements, peer elements, and gatekeepers.

H.323 Network Signaling

H.323 is defined as a binary protocol, which allows for efficient message processing in network elements. The syntax of the protocol is defined in ASN.1 and uses the Packed Encoding Rules (PER) form of message encoding for efficient message encoding on the wire. Below is an overview of the various communication flows in H.323 systems.

H.225.0 Call Signaling

Once the address of the remote endpoint is resolved, the endpoint will use H.225.0 Call Signaling in order to establish communication with the remote entity. H.225.0 messages are:

- Setup and Setup acknowledge
- Call Proceeding
- Connect
- Alerting
- Information
- Release Complete
- Facility
- Progress
- Status and Status Inquiry
- Notify

In the simplest form, an H.323 call may be established as follows (figure 3):

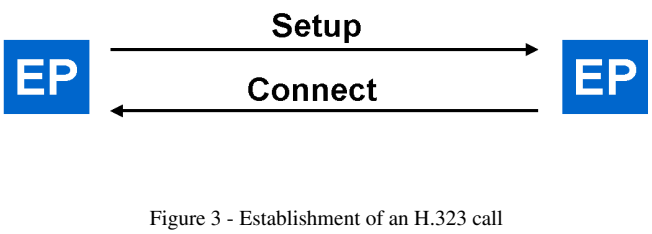


Figure 3 - Establishment of an H.323 call

In this example, the endpoint (EP) on the left initiated communication with the gateway on the right and the gateway connected the call with the called party. In reality, call flows are often more complex than the one shown, but most calls that utilize the Fast Connect procedures defined within H.323 can be established with as few as 2 or 3 messages. Endpoints must notify their gatekeeper (if gatekeepers are used) that they are in a call.

Once a call has concluded, a device will send a Release Complete message. Endpoints are then required to notify their gatekeeper (if gatekeepers are used) that the call has ended.

RAS Signaling

Endpoints use the RAS protocol in order to communicate with a gatekeeper. Likewise, gatekeepers use RAS to communicate with peer gatekeepers. RAS is a fairly simple protocol composed of just a few messages. Namely:

- Gatekeeper request, reject, and confirm messages (GRx)
- Registration request, reject, and confirm messages (RRx)
- Unregister request, reject, and confirm messages (URx)
- Admission request, reject, and confirm messages (ARx)
- Bandwidth request, reject, and confirm message (BRx)
- Disengage request, reject, and confirm (DRx)
- Location request, reject, and confirm messages (LRx)
- Info request, ack, nack, and response (IRx)
- Nonstandard message
- Unknown message response
- Request in progress (RIP)
- Resource availability indication and confirm (RAx)
- Service control indication and response (SCx)
- Admission confirm sequence (ACS)

When an endpoint is powered on, it will generally send a gatekeeper request (GRQ) message to "discover" gatekeepers that are willing to provide service. Gatekeepers will then respond with a gatekeeper confirm (GCF) and the endpoint will then select a gatekeeper to work with. Alternatively, it is possible that a gatekeeper has been predefined in the system's administrative setup so there is no need for the endpoint to discover one.

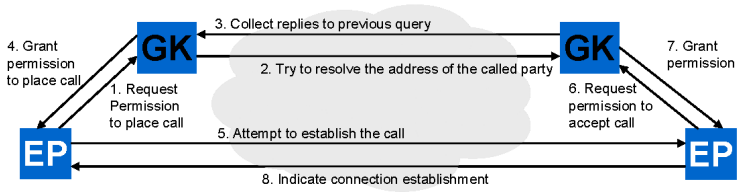


Figure 4 - A high-level communication exchange between two endpoints (EP) and two gatekeepers (GK)

Once the endpoint determines the gatekeeper to work with, it will try to register with the gatekeeper by sending a registration request (RRQ), to which the gatekeeper responds with a registration confirm (RCF). At this point, the endpoint is known to the network and can make and place calls.

When an endpoint wishes to place a call, it will send an admission request (ARQ) to the gatekeeper. The gatekeeper will then resolve the address (either locally, by consulting another gatekeeper, or by querying some other network service) and return the address of the remote endpoint in the admission confirm message (ACF). The endpoint can then place the call.

Upon receiving a call, a remote endpoint will also send an ARQ and receive an ACF in order to get permission to accept the incoming call. This is necessary, for example, to authenticate the calling device or to ensure that there is available bandwidth for the call.

Figure 4 depicts a high-level communication exchange between two endpoints (EP) and two gatekeepers (GK).

H.245 Call Control

Once a call has initiated (but not necessarily fully connected) endpoints may initiate H.245 call control signaling in order to provide more extensive control over the conference. H.245 is a rather voluminous specification with many procedures that fully enable multipoint communication, though in practice most implementations only implement the minimum necessary in order to enable point-to-point voice and video communication.

H.245 provides capabilities such as capability negotiation, master/slave determination, opening and closing of "logical channels" (i.e., audio and video flows), flow control, and conference control. It has support for both unicast and multicast communication, allowing the size of a conference to theoretically grow without bound.

Capability Negotiation

Of the functionality provided by H.245, capability negotiation is arguably the most important, as it enables devices to communicate without having prior knowledge of the capabilities of the remote entity. H.245 enables rich multimedia capabilities, including audio, video, text, and data communication. For transmission of audio, video, or text, H.323 devices utilize both ITU-defined codecs and codecs defined outside the ITU. Codecs that are widely implemented by H.323 equipment include:

- Video codecs: H.261, H.263, H.264
- Audio codecs: G.711, G.729, G.729a, G.723.1, G.726
- Text codecs: T.140

H.245 also enables real-time data conferencing capability through protocols like T.120. T.120-based applications generally operate in parallel with the H.323 system, but are integrated to provide the user with a seamless multimedia experience. T.120 provides such capabilities as application sharing T.128, electronic whiteboard T.126, file transfer T.127, and text chat T.134 within the context of the conference.

When an H.323 device initiates communication with a remote H.323 device and when H.245 communication is established between the two entities, the Terminal Capability Set (TCS) message is the first message transmitted to the other side.

Master/Slave Determination

After sending a TCS message, H.323 entities (through H.245 exchanges) will attempt to determine which device is the "master" and which is the "slave." This process, referred to as Master/Slave Determination (MSD), is important, as the master in a call settles all "disputes" between the two devices. For example, if both endpoints attempt to open incompatible media flows, it is the master who takes the action to reject the incompatible flow.

Logical Channel Signaling

Once capabilities are exchanged and master/slave determination steps have completed, devices may then open "logical channels" or media flows. This is done by simply sending an Open Logical Channel (OLC) message and receiving an acknowledgement message. Upon receipt of the acknowledgement message, an endpoint may then transmit audio or video to the remote endpoint.

Fast Connect

A typical H.245 exchange looks similar to figure 5:

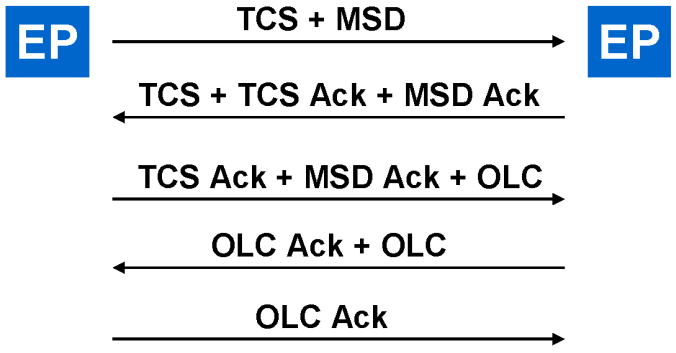


Figure 5 - A typical H.245 exchange

After this exchange of messages, the two endpoints (EP) in this figure would be transmitting audio in each direction. The number of message exchanges is numerous, each has an important purpose, but nonetheless takes time.

For this reason, H.323 version 2 (published in 1998) introduced a concept called Fast Connect, which enables a device to establish bi-directional media flows as part of the H.225.0 call establishment procedures. With Fast Connect, it is possible to establish a call with bi-directional media flowing with no more than two messages, like in figure 3.

Fast Connect is widely supported in the industry. Even so, most devices still implement the complete H.245 exchange as shown above and perform that message exchange in parallel to other activities, so there is no noticeable delay to the calling or called party.

Use cases

H.323 and Voice over IP services

Voice over Internet Protocol (VoIP) describes the transmission of voice using the Internet or other packet switched networks. ITU-T Recommendation H.323 is one of the standards used in VoIP. VoIP requires a connection to the Internet or another packet switched network, a subscription to a VoIP service provider and a client (an analogue telephone adapter (ATA), VoIP Phone or "soft phone"). The service provider offers the connection to other VoIP services or to the PSTN. Most service providers charge a monthly fee, then additional costs when calls are made. Using VoIP between two enterprise locations would not necessarily require a VoIP service provider, for example. H.323 has been widely deployed by companies who wish to interconnect remote locations over IP using a number of various wired and wireless technologies.

H.323 and Videoconference services

A videoconference, or videoteleconference (VTC), is a set of telecommunication technologies allowing two or more locations to interact via two-way video and audio transmissions simultaneously. There are basically two types of videoconferencing; dedicated VTC systems have all required components packaged into a single piece of equipment while desktop VTC systems are add-ons to normal PC's, transforming them into VTC devices. Simultaneous videoconferencing among three or more remote points is possible by means of a Multipoint Control Unit (MCU). There are MCU bridges for IP and ISDN-based videoconferencing. Due to the price point and proliferation of the Internet, and broadband in particular, there has been a strong spurt of growth and use of H.323-based IP videoconferencing. H.323 is accessible to anyone with a high speed Internet connection, such as DSL. Videoconferencing is utilized in various situations, for example; distance education, telemedicine, Video Relay Service, and business.

International Conferences

H.323 has been used in the industry to enable large-scale international video conferences that are significantly larger than the typical video conference. One of the most widely attended is an annual event called Megaconference.

Alternatives

The IETF produced a standard called the Session Initiation Protocol (SIP) that also enables voice and video communication over IP. There are also other ITU-T recommendations used for videoconferencing and videophone services - H.320 (using ISDN) and H.324 (using regular analog phone lines and 3G mobile phones). Some providers (such as Skype) also use their own closed, proprietary formats. Access Grid provides broadly similar functionality, with more emphasis on open-source and utilizing multicast. EVO also provides relatively open functionality via Java, and includes H.323 support.[8]

References

[1] Davidson, Jonathan; James Peters, Jim Peters, Brian Gracely. "H.323" (http://books.google.com/books?id=S5P7-Xtq7W8C&pg=PA229). *Voice over IP fundamentals*. Cisco Press. pp. 229–230. ISBN 978-1-57870-168-1. .

[2] H.323 Forum List of Products and Services (http://www.h323forum.org/products_services/)

[3] ITU-T Recommendation H.323 (11/1996) (http://www.itu.int/rec/T-REC-H.323-199611-S/en/), first version of H.323.

[4] ITU-T Recommendation H.323 (http://www.itu.int/rec/T-REC-H.323/en/), in force, superseded and withdrawn component.

[5] ITU-T Recommendation H.323 (02/1998) (http://www.itu.int/rec/T-REC-H.323-199802-S/en/), *Packet-based multimedia communications systems*.

[6] ITU-T Recommendation H.323 (12/2009) (http://www.itu.int/rec/T-REC-H.323), *Packet-based multimedia communications systems*.

[7] See ITU-T Recommendations of the H.323 System for a detailed list.

[8] "EVO The Collaboration Network" (http://evo.caltech.edu/evoGate). . Retrieved 2010-03-08.

ITU-T Recommendations of the H.323 System

ITU-T H.323 Core Recommendations

- ITU-T Recommendation H.323 (http://www.itu.int/rec/T-REC-H.323/en/), *Packet-based multimedia communications systems*.
- ITU-T Recommendation H.225.0 (http://www.itu.int/rec/T-REC-H.225.0/en/), *Call signalling protocols and media stream packetization for packet-based multimedia communication systems*.
- ITU-T Recommendation H.245 (http://www.itu.int/rec/T-REC-H.245/en/), *Control protocol for multimedia communication*.
- ITU-T Recommendation H.246 (http://www.itu.int/rec/T-REC-H.246/en/), *Interworking of H-series multimedia terminals with H-series multimedia terminals and voice/voiceband terminals on GSTN and ISDN*.
- ITU-T Recommendation H.283 (http://www.itu.int/rec/T-REC-H.283/en/), *Remote device control logical channel transport*.
- ITU-T Recommendation H.341 (http://www.itu.int/rec/T-REC-H.341/en/), *Multimedia management information base*.

ITU-T H.235 Series Recommendations

- ITU-T Recommendation H.235.1 (http://www.itu.int/rec/T-REC-H.235.1/en/), *H.323 security framework: Baseline security profile*.
- ITU-T Recommendation H.235.2 (http://www.itu.int/rec/T-REC-H.235.2/en/), *H.323 security framework: Signature security profile*.
- ITU-T Recommendation H.235.3 (http://www.itu.int/rec/T-REC-H.235.3/en/), *H.323 security: Hybrid security profile*.
- ITU-T Recommendation H.235.4 (http://www.itu.int/rec/T-REC-H.235.4/en/), *H.323 security: Direct and selective routed call security*.

- ITU-T Recommendation H.235.5 (http://www.itu.int/rec/T-REC-H.235.5/en/), *H.323 security: Framework for secure authentication in RAS using weak shared secrets*.
- ITU-T Recommendation H.235.6 (http://www.itu.int/rec/T-REC-H.235.6/en/), *H.323 security framework: Voice encryption profile with native H.235/H.245 key management*.
- ITU-T Recommendation H.235.7 (http://www.itu.int/rec/T-REC-H.235.7/en/), *H.323 security framework: Usage of the MIKEY key management protocol for the Secure Real Time Transport Protocol (SRTP) within H.235*.
- ITU-T Recommendation H.235.8 (http://www.itu.int/rec/T-REC-H.235.8/en/), *H.323 security: Key exchange for SRTP using secure signalling channels*.
- ITU-T Recommendation H.235.9 (http://www.itu.int/rec/T-REC-H.235.9/en/), *H.323 security: Security gateway support for H.323*.

ITU-T H.450 Series Recommendations

- ITU-T Recommendation H.450.1 (http://www.itu.int/rec/T-REC-H.450.1/en/), *Generic functional protocol for the support of supplementary services in H.323*.
- ITU-T Recommendation H.450.2 (http://www.itu.int/rec/T-REC-H.450.2/en/), *Call transfer supplementary service for H.323*.
- ITU-T Recommendation H.450.3 (http://www.itu.int/rec/T-REC-H.450.3/en/), *Call diversion supplementary service for H.323*.
- ITU-T Recommendation H.450.4 (http://www.itu.int/rec/T-REC-H.450.4/en/), *Call hold supplementary service for H.323*.
- ITU-T Recommendation H.450.5 (http://www.itu.int/rec/T-REC-H.450.5/en/), *Call park and call pickup supplementary service for H.323*.
- ITU-T Recommendation H.450.6 (http://www.itu.int/rec/T-REC-H.450.6/en/), *Call waiting supplementary service for H.323*.
- ITU-T Recommendation H.450.7 (http://www.itu.int/rec/T-REC-H.450.7/en/), *Message waiting indication supplementary service for H.323*.
- ITU-T Recommendation H.450.8 (http://www.itu.int/rec/T-REC-H.450.8/en/), *Name identification supplementary service for H.323*.
- ITU-T Recommendation H.450.9 (http://www.itu.int/rec/T-REC-H.450.9/en/), *Call completion supplementary service for H.323*.
- ITU-T Recommendation H.450.10 (http://www.itu.int/rec/T-REC-H.450.10/en/), *Call offering supplementary service for H.323*.
- ITU-T Recommendation H.450.11 (http://www.itu.int/rec/T-REC-H.450.11/en/), *Call intrusion supplementary service for H.323*.
- ITU-T Recommendation H.450.12 (http://www.itu.int/rec/T-REC-H.450.12/en/), *Common Information Additional Network Feature for H.323*.

ITU-T H.460 Series Recommendations

- ITU-T Recommendation H.460.1 (http://www.itu.int/rec/T-REC-H.460.1/en/), *Guidelines for the use of the generic extensible framework.*
- ITU-T Recommendation H.460.2 (http://www.itu.int/rec/T-REC-H.460.2/en/), *Number Portability interworking between H.323 and SCN networks*.
- ITU-T Recommendation H.460.3 (http://www.itu.int/rec/T-REC-H.460.3/en/), *Circuit maps within H.323 systems*.
- ITU-T Recommendation H.460.4 (http://www.itu.int/rec/T-REC-H.460.4/en/), *Call priority designation and country/international network of call origination identification for H.323 priority calls*.

- ITU-T Recommendation H.460.5 (http://www.itu.int/rec/T-REC-H.460.5/en/), *H.225.0 transport of multiple Q.931 information elements of the same type*.
- ITU-T Recommendation H.460.6 (http://www.itu.int/rec/T-REC-H.460.6/en/), *Extended Fast Connect feature*.
- ITU-T Recommendation H.460.7 (http://www.itu.int/rec/T-REC-H.460.7/en/), *Digit maps within H.323 systems*.
- ITU-T Recommendation H.460.8 (http://www.itu.int/rec/T-REC-H.460.8/en/), *Querying for alternate routes within H.323 systems*.
- ITU-T Recommendation H.460.9 (http://www.itu.int/rec/T-REC-H.460.9/en/), *Support for online QoS-monitoring reporting within H.323 systems*.
- ITU-T Recommendation H.460.10 (http://www.itu.int/rec/T-REC-H.460.10/en/), *Call party category within H.323 systems*.
- ITU-T Recommendation H.460.11 (http://www.itu.int/rec/T-REC-H.460.11/en/), *Delayed call establishment within H.323 systems*.
- ITU-T Recommendation H.460.12 (http://www.itu.int/rec/T-REC-H.460.12/en/), *Glare control indicator within H.323 systems*.
- ITU-T Recommendation H.460.13 (http://www.itu.int/rec/T-REC-H.460.13/en/), *Called user release control*.
- ITU-T Recommendation H.460.14 (http://www.itu.int/rec/T-REC-H.460.14/en/), *Support for Multi-Level Precedence and Preemption (MLPP) within H.323 systems*.
- ITU-T Recommendation H.460.15 (http://www.itu.int/rec/T-REC-H.460.15/en/), *Call signalling transport channel suspension and redirection within H.323 systems*.
- ITU-T Recommendation H.460.16 (http://www.itu.int/rec/T-REC-H.460.16/en/), *Multiple message release sequence capability*.
- ITU-T Recommendation H.460.17 (http://www.itu.int/rec/T-REC-H.460.17/en/), *Using H.225.0 call signalling connection as transport for H.323 RAS messages*.
- ITU-T Recommendation H.460.18 (http://www.itu.int/rec/T-REC-H.460.18/en/), *Traversal of H.323 signalling across network address translators and firewalls*.
- ITU-T Recommendation H.460.19 (http://www.itu.int/rec/T-REC-H.460.19/en/), *Traversal of H.323 media across network address translators and firewalls*.
- ITU-T Recommendation H.460.20 (http://www.itu.int/rec/T-REC-H.460.20/en/), *Location number within H.323 systems*.
- ITU-T Recommendation H.460.21 (http://www.itu.int/rec/T-REC-H.460.21/en/), *Message broadcast for H.323 systems*.
- ITU-T Recommendation H.460.22 (http://www.itu.int/rec/T-REC-H.460.22/en/), *Negotiation of security protocols to protect H.225.0 Call Signalling Messages*.

ITU-T H.500 Series Recommendations

- ITU-T Recommendation H.501 (http://www.itu.int/rec/T-REC-H.501/en/), *Protocol for mobility management and intra/inter-domain communication in multimedia systems*.
- ITU-T Recommendation H.510 (http://www.itu.int/rec/T-REC-H.510/en/), *Mobility for H.323 multimedia systems and services*.
- ITU-T Recommendation H.530 (http://www.itu.int/rec/T-REC-H.530/en/), *Symmetric security procedures for H.323 mobility in H.510*.

External links

General

- H.323 Definition and overview (http://www.iec.org/online/tutorials/h323/index.html)
- H.323 Forum (http://www.h323forum.org/)
- H.323 Information site (http://www.packetizer.com/ipmc/h323/)
- H.323 Tutorial and resources (http://www.telecomspace.com/vop-h323.html)
- Implementing H.323 (Zip) (http://hive.packetizer.com/users/h323forum/papers/basic_h323_von_99.zip)

Papers

- H.323 Protocol Overview (technical) (http://hive.packetizer.com/users/packetizer/papers/h323/h323_protocol_overview.pdf)
- H.323 Overview (less technical) (http://hive.packetizer.com/users/packetizer/papers/h323/overview_of_h323.pdf)
- H.323 Call flow covering H.225, Q.931, H.245, RTP and RTCP protocols (PDF) (http://www.eventhelix.com/RealtimeMantra/Telecom/h323_call_flow.pdf)
- H.323 Call flow (communication example) (http://www.en.voipforo.com/H323/H323_example.php)
- H.323 List of papers and presentations (http://www.h323forum.org/papers/)

Projects

- H.323 Plus open source H.323 project (http://www.h323plus.org/)
- Xmeeting for Mac OS X (http://xmeeting.sourceforge.net/)
- GNU (OpenSource) Gatekeeper (http://www.gnugk.org/)
- Ekiga: open source VoIP and video conferencing application for GNOME (http://www.ekiga.org/)

H.323 Gatekeeper

An **H.323 Gatekeeper** serves the purpose of Call Admission Control and translation services from E.164 IDs (commonly a phone number) to IP addresses in an H.323 telephony network. Gatekeepers can be combined with a gateway function to proxy H.323 calls and are sometimes referred to as Session Border Controllers. A gatekeeper can also deny access or limit the number of simultaneous connections to prevent network congestion.

H.323 endpoints are not required to register with a gatekeeper to be able to place point to point calls, but they are essential for any serious H.323 network to control call prefix routing and link capacities among other functions.

A typical H323 Gatekeeper call flow for a successful call may look like:

```
   | |                            |  |
Endpoint A                     Endpoint B
  1234                            1123
```

1. Endpoint A dials 1123 from the system.
2. Endpoint A sends ARQ (Admission Request) to the Gatekeeper.
3. Gatekeeper returns ACF (Admission Confirmation) with IP address of endpoint B.
4. Endpoint A sends Q.931 call setup messages to endpoint B.
5. Endpoint B sends the Gatekeeper an ARQ, asking if it can answer call.
6. Gatekeeper returns an ACF with IP address of endpoint A.
7. Endpoint B answers and sends Q.931 call setup messages to endpoint A.
8. IRR sent to Gatekeeper from both endpoints.
9. Either endpoint disconnects the call by sending a DRQ (Disconnect Request) to the Gatekeeper.
10. Gatekeeper sends a DCF (Disconnect Confirmation) to both endpoints.

The gatekeeper allows calls to be placed either: Directly between endpoints (Direct Endpoint Model), or Route the call signaling through itself (Gatekeeper Routed Model).

Application-level gateway

In the context of computer networking, an **application-level gateway**[1] (also known as **ALG** or **application layer gateway**) consists of a security component that augments a firewall or NAT employed in a computer network. It allows customized NAT traversal filters to be plugged into the gateway to support address and port translation for certain application layer "control/data" protocols such as FTP, BitTorrent, SIP, RTSP, file transfer in IM applications etc. In order for these protocols to work through NAT or a firewall, either the application has to know about an address/port number combination that allows incoming packets, or the NAT has to monitor the control traffic and open up port mappings (*firewall pinhole*) dynamically as required. Legitimate application data can thus be passed through the security checks of the firewall or NAT that would have otherwise restricted the traffic for not meeting its limited filter criteria.

Functions

An ALG may offer the following functions:

- allowing client applications to use dynamic ephemeral TCP/ UDP ports to communicate with the known ports used by the server applications, even though a firewall-configuration may allow only a limited number of known ports. In the absence of an ALG, either the ports would get blocked or the network administrator would need to explicitly open up a large number of ports in the firewall — rendering the network vulnerable to attacks on those ports.
- converting the network layer address information found inside an application payload between the addresses acceptable by the hosts on either side of the firewall/NAT. This aspect introduces the term 'gateway' for an ALG.
- recognizing application-specific commands and offering granular security controls over them
- synchronizing between multiple streams/sessions of data between two hosts exchanging data. For example, an FTP application may use separate connections for passing control commands and for exchanging data between the client and a remote server. During large file transfers, the control connection may remain idle. An ALG can prevent the control connection getting timed out by network devices before the lengthy file transfer completes.[2]

Deep packet-inspection of all the packets handled by ALGs over a given network makes this functionality possible. An ALG understands the protocol used by the specific applications that it supports.

For instance, for Session Initiation Protocol (SIP) Back-to-Back User agent (B2BUA), an ALG can allow firewall traversal with SIP. If the firewall has its SIP traffic terminated on an ALG then the responsibility for permitting SIP sessions passes to the ALG instead of the firewall. An ALG can solve another major SIP headache: NAT traversal. Basically a NAT with a builtin ALG can rewrite information within the SIP messages and can hold address bindings until the session terminates.

An ALG is very similar to a proxy server, as it sits between the client and real server, facilitating the exchange. There seems to be an industry convention that an ALG does its job without the application being configured to use it, by intercepting the messages. A proxy, on the other hand, usually needs to be configured in the client application. The client is then explicitly aware of the proxy and connects to it, rather than the real server.

ALG service in Microsoft Windows

The *Application Layer Gateway* service in Microsoft Windows provides support for third-party plugins that allow network protocols to pass through the Windows Firewall and work behind it and Internet Connection Sharing. ALG plugins can open ports and change data that is embedded in packets, such as ports and IP addresses. Windows Server 2003 also includes an ALG FTP plugin. The ALG FTP plugin is designed to support active FTP sessions through the NAT engine in Windows. To do this, the ALG FTP plugin redirects all traffic that passes through the NAT and that is destined for port 21 (FTP control port) to a private listening port in the 3000-5000 range on the Microsoft

loopback adapter. The ALG FTP plugin then monitors/updates traffic on the FTP control channel so that the FTP plugin can plumb port mappings through the NAT for the FTP data channels.

References

[1] RFC 2663 - ALG: official definition (refer section 2.9)

[2] *The File Transfer Protocol (FTP) and Your Firewall* (http://www.ncftp.com/ncftpd/doc/misc/ftp_and_firewalls.html)*/ Network Address Translation (NAT) Router / Load-Balancing Router.*

External links

- DNS Application Level Gateway (DNS_ALG) (http://tools.ietf.org/html/rfc2694)

Real-time Transport Protocol

The **Real-time Transport Protocol** (**RTP**) defines a standardized packet format for delivering audio and video over IP networks. RTP is used extensively in communication and entertainment systems that involve streaming media, such as telephony, video teleconference applications, television services and web-based push-to-talk features.

RTP is used in conjunction with the RTP Control Protocol (RTCP). While RTP carries the media streams (e.g., audio and video), RTCP is used to monitor transmission statistics and quality of service (QoS) and aids synchronization of multiple streams. RTP is originated and received on even port numbers and the associated RTCP communication uses the next higher odd port number.

RTP is one of the technical foundations of Voice over IP and in this context is often used in conjunction with a signaling protocol which assists in setting up connections across the network.

RTP was developed by the Audio-Video Transport Working Group of the Internet Engineering Task Force (IETF) and first published in 1996 as RFC 1889, superseded by RFC 3550 in 2003.

Overview

RTP is designed for end-to-end, real-time, transfer of stream data. The protocol provides facility for jitter compensation and detection of out of sequence arrival in data, that are common during transmissions on an IP network. RTP supports data transfer to multiple destinations through IP multicast.[1] RTP is regarded as the primary standard for audio/video transport in IP networks and is used with an associated profile and payload format.[2]

Real-time multimedia streaming applications require timely delivery of information and can tolerate some packet loss to achieve this goal. For example, loss of a packet in audio application may result in loss of a fraction of a second of audio data, which can be made unnoticeable with suitable error concealment algorithms.[3] The Transmission Control Protocol (TCP), although standardized for RTP use,[4] is not normally used in RTP application because TCP favors reliability over timeliness. Instead the majority of the RTP implementations are built on the User Datagram Protocol (UDP).[3] Other transport protocols specifically designed for multimedia sessions are SCTP[5] and DCCP, although, as of 2010, they are not in widespread use.[6]

RTP was developed by the Audio/Video Transport working group of the IETF standards organization. RTP is used in conjunction with other protocols such as H.323 and RTSP.[2] The RTP standard defines a pair of protocols, RTP and RTCP. RTP is used for transfer of multimedia data, and the RTCP is used to periodically send control information and QoS parameters.[7]

Protocol components

The RTP specification describes two sub-protocols:

- The data transfer protocol, RTP, which deals with the transfer of real-time data. Information provided by this protocol include timestamps (for synchronization), sequence numbers (for packet loss and reordering detection) and the payload format which indicates the encoded format of the data.[8]
- A control protocol, RTCP, is used to specify quality of service (QoS) feedback and synchronization between the media streams. The bandwidth of RTCP traffic compared to RTP is small, typically around 5%.[8][9]
- An optional signaling protocol such as H.323 or Session Initiation Protocol (SIP)
- An optional media description protocol such as Session Description Protocol

Sessions

An RTP Session is established for each multimedia stream. A session consists of an IP address with a pair of ports for RTP and RTCP. For example, audio and video streams will have separate RTP sessions, enabling a receiver to deselect a particular stream.[10] The ports which form a session are negotiated using other protocols such as RTSP (using SDP in the setup method)[11] and SIP. According to the specification, an RTP port should be even and the RTCP port is the next higher odd port number. RTP and RTCP typically use unprivileged UDP ports (1024 to 65535),[12] but may use other transport protocols (most notably, SCTP and DCCP) as well, as the protocol design is transport independent.

Profiles and Payload formats

One of the design considerations of RTP was to support a range of multimedia formats (such as H.264, MPEG-4, MJPEG, MPEG, etc.) and allow new formats to be added without revising the RTP standard. The design of RTP is based on the architectural principle known as application level framing (ALF). The information required by a specific application's needs is not included in the generic RTP header, but is instead provided through RTP profiles and payload formats.[7] For each class of application (e.g., audio, video), RTP defines a *profile* and one or more associated *payload formats*.[7] A complete specification of RTP for a particular application usage will require a profile and payload format specification(s).[13]:71

The *profile* defines the codecs used to encode the payload data and their mapping to payload format codes in the Payload Type (PT) field of the RTP header (see below). Each profile is accompanied by several payload format specifications, each of which describes the transport of a particular encoded data.[2] Some of the audio payload formats include: G.711, G.723, G.726, G.729, GSM, QCELP, MP3, DTMF etc., and some of the video payload formats include: H.261, H.263,[14] H.264, MPEG-4[14] etc.[15]

Examples of RTP Profiles include:

- The *RTP profile for Audio and video conferences with minimal control* (RFC 3551) defines a set of static payload type assignments, and a mechanism for mapping between a payload format, and a payload type identifier (in header) using Session Description Protocol (SDP).
- The Secure Real-time Transport Protocol (SRTP) (RFC 3711) defines a profile of RTP that provides cryptographic services for the transfer of payload data.[16]
- The experimental *Control Data Profile for RTP* (RTP/CDP[17]) for machine-to-machine communications.

Packet header

RTP packet header

bit offset	0-1	2	3	4-7	8	9-15	16-31		
0	Version	P	X	CC	M	PT	Sequence Number		
32	Timestamp								
64	SSRC identifier								
96	CSRC identifiers ...								
96+32×CC	Profile-specific extension header ID						Extension header length		
128+32×CC	Extension header ...								

The RTP header has a minimum size of 12 bytes. After the header, optional header extensions may be present. This is followed by the RTP payload, the format of which is determined by the particular class of application.[18] The fields in the header are as follows:

- **Version**: (2 bits) Indicates the version of the protocol. Current version is 2.[19]
- **P (Padding)**: (1 bit) Used to indicate if there are extra padding bytes at the end of the RTP packet. A padding might be used to fill up a block of certain size, for example as required by an encryption algorithm. The last byte of the padding contains the number of how many padding bytes were added (including itself).[19][13]:12
- **X (Extension)**: (1 bit) Indicates presence of an *Extension header* between standard header and payload data. This is application or profile specific.[19]
- **CC (CSRC Count)**: (4 bits) Contains the number of CSRC identifiers (defined below) that follow the fixed header.[13]:12
- **M (Marker)**: (1 bit) Used at the application level and defined by a profile. If it is set, it means that the current data has some special relevance for the application.[13]:13
- **PT (Payload Type)**: (7 bits) Indicates the format of the payload and determines its interpretation by the application. This is specified by an RTP profile. For example, see *RTP Profile for audio and video conferences with minimal control* (RFC 3551).[20]
- **Sequence Number**: (16 bits) The sequence number is incremented by one for each RTP data packet sent and is to be used by the receiver to detect packet loss and to restore packet sequence. The RTP does not specify any action on packet loss; it is left to the application to take appropriate action. For example, video applications may play the last known frame in place of the missing frame.[21] According to RFC 3550, the initial value of the sequence number should be random to make known-plaintext attacks on encryption more difficult.[13]:13 RTP provides no guarantee of delivery, but the presence of sequence numbers makes it possible to detect missing packets.[1]
- **Timestamp**: (32 bits) Used to enable the receiver to play back the received samples at appropriate intervals. When several media streams are present, the timestamps are independent in each stream, and may not be relied upon for media synchronization. The granularity of the timing is application specific. For example, an audio application that samples data once every 125 µs (8 kHz, a common sample rate in digital telephony) could use that value as its clock resolution. The clock granularity is one of the details that is specified in the RTP profile for an application.[21]
- **SSRC**: (32 bits) Synchronization source identifier uniquely identifies the source of a stream. The synchronization sources within the same RTP session will be unique.[13]:15
- **CSRC**: Contributing source IDs enumerate contributing sources to a stream which has been generated from multiple sources.[13]:15

- **Extension header**: (optional) The first 32-bit word contains a profile-specific identifier (16 bits) and a length specifier (16 bits) that indicates the length of the extension (EHL=extension header length) in 32-bit units, excluding the 32 bits of the extension header.[13]:17

RTP-based systems

A complete network based system will include other protocols and standards in conjunction with RTP. Protocols like SIP, RTSP, H.225 and H.245 are used for session initiation, control and termination. Other standards like H.264, MPEG, H.263 etc., are used to encode the payload data as specified via RTP Profile.[22]

An RTP sender captures the multimedia data, which is then encoded, framed and transmitted as RTP packets with appropriate timestamps and increasing sequence numbers. Depending on the RTP Profile in use, the *Payload Type* field is set. The RTP receiver captures the RTP packets, detects missing packets, and may perform reordering of packets. The frames are decoded depending on the payload format and presented to the end user.[22]

RFC references

- RFC 3550, Standard 64, RTP: A Transport Protocol for Real-Time Applications
- RFC 3551, Standard 65, RTP Profile for Audio and Video Conferences with Minimal Control
- RFC 6184, Proposed Standard, RTP Payload Format for H.264 Video
- RFC 3984, Obsolete, RTP Payload Format for H.264 Video
- RFC 4103, RTP Payload Format for Text Conversation
- RFC 3640, RTP Payload Format for Transport of MPEG-4 Elementary Streams
- RFC 3016, RTP Payload Format for MPEG-4 Audio/Visual Streams
- RFC 2250, Proposed Standard, RTP Payload Format for MPEG1/MPEG2 Video
- RFC 4175, RTP Payload Format for Uncompressed Video
- RFC 4695 (obsoleted by RFC 6295) RTP Payload Format for MIDI
- RFC 4696 An Implementation Guide for RTP MIDI

Notes

[1] Daniel Hardy (2002). *Network*. De Boeck Université. p. 298 (http://books.google.com/books?id=Oq8SEUW1wdQC&pg=PT320).

[2] Perkins 2003, p. 55

[3] Perkins 2003, p. 46

[4] RFC 4571

[5] Farrel, Adrian (2004). *The Internet and its protocols* (http://books.google.com/?id=LtBegQowqFsC&pg=PA363&dq=rtp+sctp). Morgan Kaufmann. p. 363. ISBN 978-1-55860-913-6. .

[6] Ozaktas, Haldun M.; Levent Onural (2007). *THREE-DIMENSIONAL TELEVISION* (http://books.google.com/?id=kQvCHpuXji8C&pg=PA366&dq=rtp+dccp). Springer. p. 366. ISBN 978-3-540-72531-2. .

[7] Larry L. Peterson (2007). *Computer Networks*. Morgan Kaufmann. p. 430 (http://books.google.com/books?id=zGVVuO-6w3IC&pg=PA430). ISBN 1-55860-832-X.

[8] Perkins 2003, p. 56

[9] Peterson 2007, p. 435

[10] Zurawski, Richard (2004). "RTP, RTCP and RTSP protocols" (http://books.google.com/?id=MwMDUBKZ3wwC). *The industrial information technology handbook*. CRC Press. pp. 28–7 (http://books.google.com/books?id=MwMDUBKZ3wwC&pg=PT225&dq=RTP+session). ISBN 978-0-8493-1985-3. .

[11] RFC 4566: *SDP: Session Description Protocol*, M. Handley, V. Jacobson, C. Perkins, IETF (July 2006)

[12] Collins, Daniel (2002). "Transporting Voice by using IP". *Carrier grade voice over IP*. McGraw-Hill Professional. pp. 47 (http://books.google.com/books?id=PVIuN9Y5FGMC&pg=PA47&dq=RTP+session). ISBN 0-07-136326-2.

[13] RFC 3550

[14] Chou, Philip A.; Mihaela van der Schaar (2007). *Multimedia over IP and wireless networks*. Academic Press. pp. 514 (http://books.google.com/books?id=zeLFs3GD0QQC&pg=PA514). ISBN 0-12-088480-1.

[15] Perkins 2003, p. 60

[16] Perkins 2003, p. 367

[17] Breese, Finley (2010). *Serial Communication over RTP/CDP*. BoD - Books on Demand. pp. (http://books.google.de/books?id=t18ehd_vM6wC&lpg=PP1&pg=PA9). ISBN 978-3-8391-8460-8.
[18] Peterson 2007, p. 430
[19] Peterson, p. 431 (http://books.google.com/books?id=zGVVuO-6w3IC&pg=PA431)
[20] Perkins 2003, p. 59
[21] Peterson, p. 432 (http://books.google.com/books?id=zGVVuO-6w3IC&pg=PA432)
[22] Perkins 2003, pp. 11–13

References

- Perkins, Colin (2003), *RTP* (http://books.google.com/?id=OM7YJAy9_m8C), Addison-Wesley, ISBN 978-0-672-32249-5
- Peterson, Larry L.; Bruce S. Davie (2007), *Computer Networks* (http://books.google.com/?id=zGVVuO-6w3IC) (4 ed.), Morgan Kaufmann, ISBN 978-0-12-374013-7
- "RTP" (http://books.google.com/books?id=D_GrQa2ZcLwC&pg=PA144). *Network Protocols Handbook*. Javvin Technologies. 2005. ISBN 978-0-9740945-2-6.
- "RTP" (http://www.youtube.com/watch?v=OaL2vVFbCG4&feature=channel_page). *Broadband Networks*. Ministry of Human resources, India. 2008.

External links

- oRTP, RTP library from Linphone written in C (http://www.linphone.org/eng/documentation/dev/ortp.html)
- Henning Schulzrinne's RTP page (http://www.cs.columbia.edu/~hgs/rtp) (including FAQ (http://www.cs.columbia.edu/~hgs/rtp/faq.html))
- GNU ccRTP (http://www.gnu.org/software/ccrtp/)
- JRTPLIB, a C++ RTP library (http://research.edm.uhasselt.be/~jori/page/index.php?n=CS.Jrtplib)
- RTPMobile .NET, an open source .NET RTP library (http://rtpmobile.sitesled.com)
- LScube project, providing a full streaming suite including experimental SCTP support (http://lscube.org)

RTP audio video profile

Real-time audio and video conferencing and communication applications that use the Real-time Transport Protocol (RTP) employ Session Description Protocol (SDP) to describe the media streams carried in a multi-media session. This description format specifies the technical parameters of the media streams. Such a set of RTP parameters of the media stream and its compression or encoding methods is known as a media *profile*, or **RTP audio video profile** (**RTP/AVP**). Each profile is identified by a standardized payload type identifier.[1]

RTP/AVP audio and video payload types

Payload type (PT)	Name	Type	No. of channels	Clock rate (Hz)	Description	References
0	PCMU	audio	1	8000	ITU-T G.711 PCM μ-Law Audio 64 kbit/s	RFC 3551
1	reserved (previously 1016)	audio	1	8000	reserved, previously CELP Audio 4.8 kbit/s	RFC 3551, previously RFC 1890
2	reserved (previously G721)	audio	1	8000	reserved, previously ITU-T G.721 ADPCM Audio 32 kbit/s	RFC 3551, previously RFC 1890
3	GSM	audio	1	8000	European GSM Full Rate Audio 13 kbit/s (GSM 06.10)	RFC 3551
4	G723	audio	1	8000	ITU-T G.723.1	RFC 3551
5	DVI4	audio	1	8000	IMA ADPCM Audio 32 kbit/s	RFC 3551
6	DVI4	audio	1	16000	IMA ADPCM 64 kbit/s	RFC 3551
7	LPC	audio	1	8000	Experimental Linear Predictive Coding Audio	RFC 3551
8	PCMA	audio	1	8000	ITU-T G.711 PCM A-Law Audio 64 kbit/s	RFC 3551
9	G722	audio	1	8000	ITU-T G.722 Audio	RFC 3551 - Page 14 [2]
10	L16	audio	2	44100	Linear PCM 16-bit Stereo Audio 1411.2 kbit/s,[3][4][5] uncompressed	RFC 3551, Page 27 [6]
11	L16	audio	1	44100	Linear PCM 16-bit Audio 705.6 kbit/s, uncompressed	RFC 3551, Page 27 [6]
12	QCELP	audio	1	8000	Qualcomm Code Excited Linear Prediction	RFC 2658, RFC 3551 [7]
13	CN	audio	1	8000	Comfort noise	RFC 3389
14	MPA	audio	1	90000	MPEG-1 or MPEG-2 Audio Only	RFC 3551, RFC 2250
15	G728	audio	1	8000	ITU-T G.728 Audio 16 kbit/s	RFC 3551
16	DVI4	audio	1	11025	IMA ADPCM	RFC 3551
17	DVI4	audio	1	22050	IMA ADPCM	RFC 3551
18	G729	audio	1	8000	ITU-T G.729 and G.729a	RFC 3551, Page 20 [8]

25	CELB	video	1	90000	Sun's CellB Video Encoding[9]	RFC 2029
26	JPEG	video	1	90000	JPEG Video	RFC 2435
28	NV	video	1	90000	Xerox PARC's Network Video (nv)[10]	RFC 3551, Page 32 [11]
31	H261	video	1	90000	ITU-T H.261 Video	RFC 4587
32	MPV	video	1	90000	MPEG-1 and MPEG-2 Video	RFC 2250
33	MP2T	audio/video	1	90000	MPEG-2 transport stream Video	RFC 2250
34	H263	video		90000	H.263 video, first version (1996)	RFC 3551, RFC 2190
dynamic	H263-1998	video		90000	H.263 video, second version (1998)	RFC 3551, RFC 4629, RFC 2190
dynamic	H263-2000	video		90000	H.263 video, third version (2000)	RFC 4629
dynamic (or profile)	H264	video		90000	H.264 video (MPEG-4 Part 10)	RFC 3984
dynamic (or profile)	theora	video		90000	Theora video	draft-barbato-avt-rtp-theora-01 [12]
dynamic	iLBC	audio	1	—	Internet low Bitrate Codec 13.33 or 15.2 kbit/s	RFC 3951
dynamic	PCMA-WB	audio		16000	ITU-T G.711.1, A-law	RFC 5391
dynamic	PCMU-WB	audio		16000	ITU-T G.711.1, μ-law	RFC 5391
dynamic	G718	audio		32000	ITU-T G.718	draft-ietf-avt-rtp-g718-03 [13]
dynamic	G719	audio	(various)	48000	ITU-T G.719	RFC 5404
dynamic	G7221	audio		16 or 32 kHz	ITU-T G.722.1	RFC 5577
dynamic	G726-16	audio	1	8000	ITU-T G.726 audio with 16 kbit/s	RFC 3551
dynamic	G726-24	audio	1	8000	ITU-T G.726 audio with 24 kbit/s	RFC 3551
dynamic	G726-32	audio	1	8000	ITU-T G.726 audio with 32 kbit/s	RFC 3551
dynamic	G726-40	audio	1	8000	ITU-T G.726 audio with 40 kbit/s	RFC 3551
dynamic	G729D	audio	1	8000	ITU-T G.729 Annex D	RFC 3551
dynamic	G729E	audio	1	8000	ITU-T G.729 Annex E	RFC 3551
dynamic	G7291	audio		(various)	ITU-T G.729.1	RFC 4749
dynamic	GSM-EFR	audio	1	8000	ITU-T GSM-EFR (GSM 06.60)	RFC 3551
dynamic	GSM-HR-08	audio	1	8000	ITU-T GSM-HR (GSM 06.20)	RFC 5993
dynamic (or profile)	AMR	audio	(various)	8000	Adaptive Multi-Rate audio	RFC 4867
dynamic (or profile)	AMR-WB	audio	(various)	16000	Adaptive Multi-Rate Wideband audio (ITU-T G.722.2)	RFC 4867
dynamic (or profile)	AMR-WB+	audio	1, 2 or omit	72000	Extended Adaptive Multi Rate – WideBand audio	RFC 4352

dynamic (or profile)	vorbis	audio	(various)	from 8 kHz to 192 kHz	RTP Payload Format for Vorbis Encoded Audio	RFC 5215
dynamic (or profile)	opus	audio	1, 2	48000 (the actual clock rate is signaled inside the payload)	RTP Payload Format for Opus Speech and Audio Codec	draft [14]
dynamic (or profile)	speex	audio	1	8000, 16000 or 32000	RTP Payload Format for the Speex Codec	RFC 5574
dynamic (96-127)	mpa-robust	audio		90000	A More Loss-Tolerant RTP Payload Format for MP3 Audio	RFC 5219
dynamic (or profile)	MP4A-LATM	audio		90000 or others	RTP Payload Format for MPEG-4 Audio	RFC 3016
dynamic (or profile)	MP4V-ES	video		90000 or others	RTP Payload Format for MPEG-4 Visual	RFC 3016
dynamic (or profile)	mpeg4-generic	audio/video		90000 or other	RTP Payload Format for Transport of MPEG-4 Elementary Streams	RFC 3640
dynamic	L8	audio	(various)	(various)	Linear PCM 8-bit audio with 128 offset	RFC 3551 Section 4.5.10 and Table 5
dynamic	DAT12	audio	(various)	8000, 11025, 16000, 22050, 24000, 32000, 44100, 48000 or others	IEC 61119 12-bit nonlinear audio	RFC 3190 Section 3
dynamic	L16	audio	(various)	8000, 11025, 16000, 22050, 24000, 32000, 44100, 48000 or others	Linear PCM 16-bit audio	RFC 3551 Section 4.5.11, RFC 2586
dynamic	L20	audio	(various)	8000, 11025, 16000, 22050, 24000, 32000, 44100, 48000 or others	Linear PCM 20-bit audio	RFC 3190 Section 4
dynamic	L24	audio	(various)	8000, 11025, 16000, 22050, 24000, 32000, 44100, 48000 or others	Linear PCM 24-bit audio	RFC 3190 Section 4

RFC 3551 lists details of the codec, or a reference for the details is provided. Payload identifiers 96–127 are reserved for payloads defined dynamically during a session. The minimum payload support is defined as 0 (PCMU) and 5 (DVI4). The document recommends dynamically assigned port numbers, although 5004 and 5005 have been registered for use of the profile and can be used instead. The standard also describes the process of registering new payload types with IANA.

References

[1] RFC 3551, *RTP Profile for Audio and Video Conferences with Minimal Control*, H. Schulzrinne, S. Casner, The Internet Society (July 2003).
[2] http://tools.ietf.org/html/rfc3551#page-14
[3] "RFC 2586 - The Audio/L16 MIME content type" (http://tools.ietf.org/html/rfc2586). 1999-05. . Retrieved 2010-03-16.
[4] "RFC 3108 - Conventions for the use of the Session Description Protocol (SDP) for ATM Bearer Connections" (http://tools.ietf.org/html/rfc3108#page-62). 2001-05. . Retrieved 2010-03-16.
[5] "RFC 4856 - Media Type Registration of Payload Formats in the RTP Profile for Audio and Video Conferences - Registration of Media Type audio/L16" (http://tools.ietf.org/html/rfc4856#page-18). 2007-03. . Retrieved 2010-03-16.
[6] http://tools.ietf.org/html/rfc3551#page-27
[7] http://tools.ietf.org/html/rfc3551#page-28
[8] http://tools.ietf.org/html/rfc3551#page-20
[9] SUN CellB Codec (http://docs.sun.com/app/docs/doc/802-5863/6i9jfuk3e?a=view), Retrieved on 2009-07-09.
[10] nv - network video on Henning Schulzrinne's website (http://www.cs.columbia.edu/~hgs/rtp/nv.html), Network Video on The University of Toronto's website (http://www.dgp.toronto.edu/tp/techdocs/NetVid.html), Retrieved on 2009-07-09.
[11] http://tools.ietf.org/html/rfc3551#page-32
[12] http://tools.ietf.org/html/draft-barbato-avt-rtp-theora-01
[13] http://tools.ietf.org/html/draft-ietf-avt-rtp-g718-03
[14] http://tools.ietf.org/html/draft-spittka-payload-rtp-opus-01

External links

- IANA assignments of Real-Time Transport Protocol (RTP) Parameters (http://www.iana.org/assignments/rtp-parameters)

Secure Real-time Transport Protocol

The **Secure Real-time Transport Protocol** (or **SRTP**) defines a profile of RTP (Real-time Transport Protocol), intended to provide encryption, message authentication and integrity, and replay protection to the RTP data in both unicast and multicast applications. It was developed by a small team of IP protocol and cryptographic experts from Cisco and Ericsson including David Oran, David McGrew, Mark Baugher, Mats Naslund, Elisabetta Carrara, James Black, Karl Norman, and Rolf Blom. It was first published by the IETF in March 2004 as RFC 3711.

Since RTP is closely related to RTCP (Real Time Control Protocol) which can be used to control the RTP session, SRTP also has a sister protocol, called **Secure RTCP** (or **SRTCP**); SRTCP provides the same security-related features to RTCP, as the ones provided by SRTP to RTP.

Utilization of SRTP or SRTCP is optional to the utilization of RTP or RTCP; but even if SRTP/SRTCP are used, all provided features (such as encryption and authentication) are optional and can be separately enabled or disabled. The only exception is the message authentication feature which is indispensably required when using SRTCP.

Data flow encryption

For encryption and decryption of the data flow (and hence for providing confidentiality of the data flow), SRTP (together with SRTCP) utilizes AES as the default cipher. There are two cipher modes defined which allow the original block cipher AES to be used as a stream cipher:

Segmented Integer Counter Mode

> A typical counter mode, which allows random access to any blocks, which is essential for RTP traffic running over unreliable network with possible loss of packets. In the general case, almost any function can be used in the role of "counter", assuming that this function does not repeat for a long number of iterations. But the standard for encryption of RTP data is just a usual integer incremental counter. AES running in this mode is the default encryption algorithm, with a default encryption key length of 128 bits and a default session salt key

length of 112 bits.

f8-mode

A variation of output feedback mode, enhanced to be seekable and with an altered initialization function. The default values of the encryption key and salt key are the same as for AES in Counter Mode. (AES running in this mode has been chosen to be used in UMTS 3G mobile networks.)

Besides the AES cipher, SRTP allows the ability to disable encryption outright, using the so called "NULL cipher", which can be assumed as the second supported cipher (or the third supported cipher mode in sum). In fact, the NULL cipher does not perform any encryption (i.e. the encryption algorithm functions as though the key stream contains only zeroes, and copies the input stream to the output stream without any changes). It is mandatory for this cipher mode to be implemented in any SRTP-compatible system. As such, it can be used when the confidentiality guarantees ensured by SRTP are not required, while other SRTP features (such authentication and message integrity) may be used.

Though technically SRTP can easily accommodate new encryption algorithms, the SRTP standard states that new encryption algorithms besides those described cannot simply be added in some implementation of SRTP protocol. The only legal way to add a new encryption algorithm, while still claiming the compatibility with SRTP standard, is to publish a new companion standard track RFC which must clearly define the new algorithm.

Authentication, integrity and replay protection

The above-listed encryption algorithms do not secure message integrity themselves, allowing the attacker to either forge the data or at least to replay previously transmitted data. Hence the SRTP standard also provides the means to secure the integrity of data and safety from replay.

To authenticate the message and protect its integrity, the HMAC-SHA1 algorithm (defined in RFC 2104) is used, which produces a 160-bit result, which is then truncated to 80 or 32 bits to become the authentication tag appended to the packet. The HMAC is calculated over the packet payload and material from the packet header, including the packet sequence number. To protect against replay attacks, the receiver maintains the indices of previously received messages, compares them with the index of each new received message and admits the new message only if it has not been played (i.e. sent) before. Such an approach heavily relies on the integrity protection being enabled (to make it impossible to spoof message indices).

Key Derivation

A key derivation function is used to derive the different keys used in a crypto context (SRTP and SRTCP encryption keys and salts, SRTP and SRTCP authentication keys) from one single *master key* in a cryptographically secure way. Thus, the key management protocol needs to exchange only one master key, all the necessary session keys are generated by applying the key derivation function.

Periodical application of the key derivation function will result in security benefits. It prevents an attacker from collecting large amounts of ciphertext encrypted with one single session key. Certain attacks are easier to carry out when a large amount of ciphertext is available. Furthermore, multiple applications of the key derivation function provides backwards and forward security in the sense that a compromised session key does not compromise other session keys derived from the same master key. This means that even if an attacker managed to recover a certain session key, he is not able to decrypt messages secured with previous and later session keys derived from the same master key. (Note that, of course, a leaked master key reveals all the session keys derived from it.)

SRTP relies on an external key management protocol to set up the initial master key. Two protocols specifically designed to be used with SRTP are ZRTP and MIKEY.

There are also other methods to negotiate the SRTP keys. There are several vendors which offer products that use the SDES key exchange method.

SRTP interoperability

See Comparison of VoIP software for phones, servers and applications supporting SRTP

External links

- RFCs
  - RFC 3711, Proposed Standard, The Secure Real-time Transport Protocol (SRTP)
  - RFC 4771, Proposed Standard, Integrity Transform Carrying Roll-Over Counter for the Secure Real-time Transport Protocol (SRTP)
  - RFC 3551, Standard 65, RTP Profile for Audio and Video Conferences with Minimal Control
  - RFC 3550, Standard 64, RTP: A Transport Protocol for Real-Time Applications
  - RFC 2104, Informational, HMAC: Keyed-Hashing for Message Authentication
- Entry for SRTP [1] in the voip-info.org-Wiki

References

[1] http://www.voip-info.org/wiki/view/SRTP

Real Time Streaming Protocol

The **Real Time Streaming Protocol** (**RTSP**) is a network control protocol designed for use in entertainment and communications systems to control streaming media servers. The protocol is used for establishing and controlling media sessions between end points. Clients of media servers issue VCR-like commands, such as *play* and *pause*, to facilitate real-time control of playback of media files from the server.

The transmission of streaming data itself is not a task of the RTSP protocol. Most RTSP servers use the Real-time Transport Protocol (RTP) in conjunction with Real-time Control Protocol (RTCP) for media stream delivery, however some vendors implement proprietary transport protocols. The RTSP server from RealNetworks, for example, also features RealNetworks' proprietary Real Data Transport (RDT).

RTSP was developed by the Multiparty Multimedia Session Control Working Group (MMUSIC WG) of the Internet Engineering Task Force (IETF) and published as RFC 2326 in 1998.[1]

RTSP using RTP and RTCP allows for the implementation of rate adaption.

Protocol directives

While similar in some ways to HTTP, RTSP defines control sequences useful in controlling multimedia playback. While HTTP is stateless, RTSP has state; an identifier is used when needed to track concurrent sessions. Like HTTP, RTSP uses TCP to maintain an end-to-end connection and, while most RTSP control messages are sent by the client to the server, some commands travel in the other direction (i.e. from server to client).

Presented here are the basic RTSP requests. Some typical HTTP requests, like the OPTIONS request, are also available. The default transport layer port number is 554.

OPTIONS

An OPTIONS request returns the request types the server will accept.

```
C->S:  OPTIONS rtsp://example.com/media.mp4 RTSP/1.0
       CSeq: 1
       Require: implicit-play
       Proxy-Require: gzipped-messages
```

```
S->C:  RTSP/1.0 200 OK
       CSeq: 1
       Public: DESCRIBE, SETUP, TEARDOWN, PLAY, PAUSE
```

DESCRIBE

A DESCRIBE request includes an RTSP URL (rtsp://...), and the type of reply data that can be handled. The default port for the RTSP protocol is 554 for both UDP (deprecated and very rarely used) and TCP transports. This reply includes the presentation description, typically in Session Description Protocol (SDP) format. Among other things, the presentation description lists the media streams controlled with the aggregate URL. In the typical case, there is one media stream each for audio and video.

```
C->S: DESCRIBE rtsp://example.com/media.mp4 RTSP/1.0
      CSeq: 2

S->C: RTSP/1.0 200 OK
      CSeq: 2
      Content-Base: rtsp://example.com/media.mp4
      Content-Type: application/sdp
      Content-Length: 460

      m=video 0 RTP/AVP 96
      a=control:streamid=0
      a=range:npt=0-7.741000
      a=length:npt=7.741000
      a=rtpmap:96 MP4V-ES/5544
      a=mimetype:string;"video/MP4V-ES"
      a=AvgBitRate:integer;304018
      a=StreamName:string;"hinted video track"
      m=audio 0 RTP/AVP 97
      a=control:streamid=1
      a=range:npt=0-7.712000
      a=length:npt=7.712000
      a=rtpmap:97 mpeg4-generic/32000/2
      a=mimetype:string;"audio/mpeg4-generic"
      a=AvgBitRate:integer;65790
      a=StreamName:string;"hinted audio track"
```

SETUP

A SETUP request specifies how a single media stream must be transported. This must be done before a PLAY request is sent. The request contains the media stream URL and a transport specifier. This specifier typically includes a local port for receiving RTP data (audio or video), and another for RTCP data (meta information). The server reply usually confirms the chosen parameters, and fills in the missing parts, such as the server's chosen ports. Each media stream must be configured using SETUP before an aggregate play request may be sent.

```
C->S: SETUP rtsp://example.com/media.mp4/streamid=0 RTSP/1.0
      CSeq: 3
      Transport: RTP/AVP;unicast;client_port=8000-8001
```

```
S->C: RTSP/1.0 200 OK
      CSeq: 3
      Transport: RTP/AVP;unicast;client_port=8000-8001;server_port=9000-9001
      Session: 12345678
```

PLAY

A PLAY request will cause one or all media streams to be played. Play requests can be stacked by sending multiple PLAY requests. The URL may be the aggregate URL (to play all media streams), or a single media stream URL (to play only that stream). A range can be specified. If no range is specified, the stream is played from the beginning and plays to the end, or, if the stream is paused, it is resumed at the point it was paused.

```
C->S: PLAY rtsp://example.com/media.mp4 RTSP/1.0
      CSeq: 4
      Range: npt=5-20
      Session: 12345678

S->C: RTSP/1.0 200 OK
      CSeq: 4
      Session: 12345678
      RTP-Info: url=rtsp://example.com/media.mp4/streamid=0;seq=9810092;rtptime=3450012
```

PAUSE

A PAUSE request temporarily halts one or all media streams, so it can later be resumed with a PLAY request. The request contains an aggregate or media stream URL. A range parameter on a PAUSE request specifies when to pause. When the range parameter is omitted, the pause occurs immediately and indefinitely.

```
C->S: PAUSE rtsp://example.com/media.mp4 RTSP/1.0
      CSeq: 5
      Session: 12345678

S->C: RTSP/1.0 200 OK
      CSeq: 5
      Session: 12345678
```

RECORD

This method initiates recording a range of media data according to the presentation description. The time stamp reflects start and end time(UTC). If no time range is given, use the start or end time provided in the presentation description. If the session has already started, commence recording immediately. The server decides whether to store the recorded data under the request URI or another URI. If the server does not use the request URI, the response should be 201 and contain an entity which describes the states of the request and refers to the new resource, and a Location header.

```
C->S: RECORD rtsp://example.com/media.mp4 RTSP/1.0
      CSeq: 6
      Session: 12345678

S->C: RTSP/1.0 200 OK
      CSeq: 6
      Session: 12345678
```

ANNOUNCE

The ANNOUNCE method serves two purposes:

When sent from client to server, ANNOUNCE posts the description of a presentation or media object identified by the request URL to a server. When sent from server to client, ANNOUNCE updates the session description in real-time. If a new media stream is added to a presentation (e.g., during a live presentation), the whole presentation description should be sent again, rather than just the additional components, so that components can be deleted.

```
C->S: ANNOUNCE rtsp:rtsp://example.com/media.mp4 RTSP/1.0
      CSeq: 7
      Date: 23 Jan 1997 15:35:06 GMT
      Session: 12345678
      Content-Type: application/sdp
      Content-Length: 332

      v=0
      o=mhandley 2890844526 2890845468 IN IP4 126.16.64.4
      s=SDP Seminar
      i=A Seminar on the session description protocol
      u=http://www.cs.ucl.ac.uk/staff/M.Handley/sdp.03.ps
      e=mjh@isi.edu (Mark Handley)
      c=IN IP4 224.2.17.12/127
      t=2873397496 2873404696
      a=recvonly
      m=audio 3456 RTP/AVP 0
      m=video 2232 RTP/AVP 31

S->C: RTSP/1.0 200 OK
      CSeq: 7
```

TEARDOWN

A TEARDOWN request is used to terminate the session. It stops all media streams and frees all session related data on the server.

```
C->S: TEARDOWN rtsp://example.com/media.mp4 RTSP/1.0
      CSeq: 8
      Session: 12345678

S->C: RTSP/1.0 200 OK
      CSeq: 8
```

GET_PARAMETER

The GET_PARAMETER request retrieves the value of a parameter of a presentation or stream specified in the URI. The content of the reply and response is left to the implementation. GET_PARAMETER with no entity body may be used to test client or server liveness ("ping").

```
S->C: GET_PARAMETER rtsp://example.com/media.mp4 RTSP/1.0
      CSeq: 9
      Content-Type: text/parameters
```

```
      Session: 12345678
      Content-Length: 15

      packets_received
      jitter

C->S: RTSP/1.0 200 OK
      CSeq: 9
      Content-Length: 46
      Content-Type: text/parameters

      packets_received: 10
      jitter: 0.3838
```

SET_PARAMETER

This method requests to set the value of a parameter for a presentation or stream specified by the URI.

```
C->S: SET_PARAMETER rtsp://example.com/media.mp4 RTSP/1.0
      CSeq: 10
      Content-length: 20
      Content-type: text/parameters

      barparam: barstuff

S->C: RTSP/1.0 451 Invalid Parameter
      CSeq: 10
      Content-length: 10
      Content-type: text/parameters

      barparam
```

REDIRECT

：A redirect request informs the client that it must connect to another server location. It contains the mandatory header Location, which indicates that the client should issue requests for that URL. It may contain the parameter Range, which indicates when the redirection takes effect. If the client wants to continue to send or receive media for this URI, the client MUST issue a TEARDOWN request for the current session and a SETUP for the new session at the designated host.

```
S->C: REDIRECT rtsp://example.com/media.mp4 RTSP/1.0
      CSeq: 11
      Location: rtsp://bigserver.com:8001
      Range: clock=19960213T143205Z-
```

Embedded (Interleaved) Binary Data

Certain firewall designs and other circumstances may force a server to interleave RTSP methods and stream data. This interleaving should generally be avoided unless necessary since it complicates client and server operation and imposes additional overhead. Interleaved binary data SHOULD only be used if RTSP is carried over TCP.Stream data such as RTP packets is encapsulated by an ASCII dollar sign (24 hexadecimal), followed by a one-byte channel identifier, followed by the length of the encapsulated binary data as a binary,

two-byte integer in network byte order. The stream data follows immediately afterwards, without a CRLF, but including the upper-layer protocol headers. Each $ block contains exactly one upper-layer protocol data unit, e.g., one RTP packet.

```
C->S: SETUP rtsp://example.com/media.mp4 RTSP/1.0
      CSeq: 3
      Transport: RTP/AVP/TCP;interleaved=0-1

S->C: RTSP/1.0 200 OK
      CSeq: 3
      Date: 05 Jun 1997 18:57:18 GMT
      Transport: RTP/AVP/TCP;interleaved=0-1
      Session: 12345678

C->S: PLAY rtsp://example.com/media.mp4 RTSP/1.0
      CSeq: 4
      Session: 12345678

S->C: RTSP/1.0 200 OK
      CSeq: 4
      Session: 12345678
      Date: 05 Jun 1997 18:59:15 GMT
      RTP-Info: url=rtsp://example.com/media.mp4;
      seq=232433;rtptime=972948234

     S->C: $\000{2 byte length}{"length" bytes data, w/RTP header}
     S->C: $\000{2 byte length}{"length" bytes data, w/RTP header}
     S->C: $\001{2 byte length}{"length" bytes  RTCP packet}
```

Implementations

Server

- QuickTime Streaming Server: Apple's closed-source streaming server that ships with Mac OS X Server.
- Darwin Streaming Server: Open-sourced version of QuickTime Streaming Server maintained by Apple.
- pvServer: Formerly called PacketVideo Streaming Server, this is Alcatel-Lucent's streaming server product.
- Helix Universal Server: RealNetworks commercial streaming server for RTSP, RTMP, iOS, Silverlight and HTTP streaming media clients
- Helix DNA Server: RealNetworks' streaming server. Comes in both open-source and proprietary flavors.
- LIVE555: Open source C++ server and client libraries used in well known clients like VLC and mplayer.
- Feng: Lean and mean streaming server with focus with rfc compliance.
- VideoLAN: Open source media player and streaming server.
- Windows Media Services: Microsoft's streaming server included with Windows Server.
- VX30: Streaming video server and embedded JAVA client from Maui X-Stream.
- Xenon Streaming Server: Mobile streaming server from Vidiator Technology (US) Inc.
- RtpRtspStack[2]: Streaming server which is designed for low footprint and high performance applications.
- Gstreamer based RTSP Server and client.
- FFmpeg: includes ffserver a GPL or LGPL RTSP streaming server.

- Erlyvideo[3] has RTSP client and can restream video to other protocols.
- ViaMotion : integrated RTSP server for Video On Demand by Anevia

Client

- cURL (beginning with version 7.20.0—9 February 2010[4])
- FFmpeg[5]
- GStreamer
- Media Player Classic
- MPlayer
- MythTV via Freebox
- QuickTime
- RealPlayer
- Skype
- Spotify
- VLC media player
- Winamp
- Windows Media Player
- xine
- JetAudio

References

[1] RFC 2326, *Real Time Streaming Protocol (RTSP)*, IETF, 1998

[2] EInfochips RtpRtspStack - RTSP/RTP Server and RTSP/RTP client (http://einfochips.com/ips-frameworks/rtp-rtsp-server-client-stacks.php)

[3] erlyvideo website (http://erlyvideo.org/)

[4] cURL - Changes (http://curl.haxx.se/changes.html)

[5] "FFmpeg Documentation" (http://ffmpeg.org/ffmpeg.html#rtsp). The FFmpeg project. September 11, 2012. Section 20.19. . Retrieved 2012-09-11.

External links

- "Real Time Streaming Protocol Information and Updates" (http://web.archive.org/web/20070306002838/http://www.rtsp.org/). Archived from the original (http://www.rtsp.org) on 2007-03-06., a central information repository about RTSP.
- Tunnelling RTSP and RTP through HTTP (http://developer.apple.com/quicktime/icefloe/dispatch028.html), A standard solution to help RTSP work through firewalls and web proxies

G.711

G.711 is an ITU-T standard for audio companding. It is primarily used in telephony. The standard was released for usage in 1972. Its formal name is *Pulse code modulation (PCM) of voice frequencies*. It is required standard in many technologies, for example in H.320 and H.323 specifications. It can also be used for fax communication over IP networks (as defined in T.38 specification). G.711, also known as Pulse Code Modulation (PCM), is a very commonly used waveform codec. G.711 uses a sampling rate of 8,000 samples per second, with the tolerance on that rate 50 parts per million (ppm). Non-uniform (logarithmic) quantization with 8 bits is used to represent each sample, resulting in a 64 kbit/s bit rate. There are two slightly different versions; μ-law, which is used primarily in North America, and A-law, which is in use in most other countries outside North America.

Two enhancements to G.711 have been published: **G.711.0** utilizes lossless data compression to reduce the bandwidth usage and **G.711.1** increases audio quality by increasing bandwidth.

Features

- Sampling frequency 8 kHz
- 64 kbit/s bitrate (8 kHz sampling frequency x 8 bits per sample)
- Typical algorithmic delay is 0.125 ms, with no look-ahead delay
- G.711 is a waveform speech coder
- G.711 Appendix I defines a Packet Loss Concealment (PLC) algorithm to help hide transmission losses in a packetized network
- G.711 Appendix II defines a Discontinuous Transmission (DTX) algorithm which uses Voice Activity Detection (VAD) and Comfort Noise Generation (CNG) to reduce bandwidth usage during silence periods
- PSQM testing under ideal conditions yields Mean Opinion Scores of 4.45 for G.711 μ-law, 4.45 for G.711 A-law
- PSQM testing under network stress yields Mean Opinion Scores of 4.13 for G.711 μ-law, 4.11 for G.711 A-law

Types

G.711 defines two main compression algorithms, the μ-law algorithm (used in North America & Japan) and A-law algorithm (used in Europe and the rest of the world). Both are logarithmic, but A-law was specifically designed to be simpler for a computer to process. The standard also defines a sequence of repeating code values which defines the power level of 0 dB.

The μ-law and A-law algorithms encode 14-bit and 13-bit signed linear PCM samples (respectively) to logarithmic 8-bit samples. Thus, the G.711 encoder will create a 64 kbit/s bitstream for a signal sampled at 8 kHz.[1]

G.711 μ-law tends to give more resolution to higher range signals while G.711 A-law provides more quantization levels at lower signal levels.

A-Law

A-law encoding thus takes a 13-bit signed linear audio sample as input and converts it to an 8 bit value as follows:

Linear input code	Compressed code
s0000000wxyz`a	s000wxyz
s0000001wxyz`a	s001wxyz
s000001wxyz`ab	s010wxyz
s00001wxyz`abc	s011wxyz
s0001wxyz`abcd	s100wxyz
s001wxyz`abcde	s101wxyz
s01wxyz`abcdef	s110wxyz
s1wxyz`abcdefg	s111wxyz

Where s is the sign bit, and bits after the backtick mark ` are discarded. So for example, 1'0000'0001'0101 maps to 1000'1010 (according to the first row of the table), and 0'0000'0011'0101 maps to 0001'1010 (according to the second).

This can be seen as a floating point number with 4 bits of mantissa and 3 bits of exponent.

In addition, the standard specifies that all resulting even bits are inverted before the octet is transmitted. This is to provide plenty of 0/1 transitions to facilitate the clock recovery process in the PCM receivers. Thus, a silent A-law encoded PCM channel has the 8 bit samples coded 0x55 instead of 0x00 in the octets (or 0xD5 if the sign bit happens to be set).

Note that the ITU define bit 1 to have the value 128 and bit 8 to have the value 1.

The more widely accepted convention has bit 7 = 128 and bit 0 = 1.

Note that when data is sent over E0 (G.703), MSB (signbit) is sent first and LSB is sent last.

μ-Law

μ-law encoding takes a 14-bit signed linear audio sample as input, increases the magnitude by 32 (binary 10000), and converts it to an 8 bit value as follows:

Linear input code	Compressed code
s00000001wxyza	s000wxyz
s0000001wxyzab	s001wxyz
s000001wxyzabc	s010wxyz
s00001wxyzabcd	s011wxyz
s0001wxyzabcde	s100wxyz
s001wxyzabcdef	s101wxyz
s01wxyzabcdefg	s110wxyz
s1wxyzabcdefgh	s111wxyz

Where *s* is the sign bit.

In addition, the standard specifies that all result bits are inverted before the octet is transmitted. Thus, a silent μ-law encoded PCM channel has the 8 bit samples coded 0xFF instead of 0x00 in the octets.

Also the "trick" of adding 32 means μ-law does not encode all 14-bit values; inputs must be within ±8159.

G.711.0

G.711.0, also known as G.711 LLC, utilizes lossless data compression to reduce the bandwidth usage by as much as 50 percent.[2] The *Lossless compression of G.711 pulse code modulation* standard was approved by ITU-T in September 2009.[3][4]

G.711.1

G.711.1 is an extension to G.711, published as ITU-T Recommendation G.711.1 in March 2008. Its formal name is *Wideband embedded extension for G.711 pulse code modulation*.[4][5]

G.711.1, allows the addition of narrowband and/or wideband (16000 samples/s) enhancements, each at 25 % of the bitrate of the (included) base G.711 bitstream, leading to data rates of 64, 80 or 96 kbit/s.

G.711.1 is compatible with G.711 at 64 kbit/s, hence an efficient deployment in existing G.711-based voice over IP (VoIP) infrastructures is foreseen. The G.711.1 coder can encode signals at 16 kHz with a bandwidth of 50–7000 Hz at 80 and 96 kbit/s, and for 8-kHz sampling the output may produce signals with a bandwidth ranging from 50 up to 4000 Hz, operating at 64 and 80 kbit/s.[5]

The G.711.1 encoder creates embedded bitstream structured in three layers corresponding to three available bit rates: 64, 80 and 96 kbit/s. The bitstream does not contain any information on which layers are contained, an implementation would require outband signalling on which layers are available. The three G.711.1 layers are: log companded pulse code modulation (PCM) of the lower band including noise feedback, embedded PCM extension with adaptive bit allocation for enhancing the quality of the base layer in the lower band, and weighted vector quantization coding of the higher band based on modified discrete cosine transformation (MDCT).[5]

Two extensions for G.711.1 are planned in 2010: superwideband extension (bandwidth to 14000 Hz) and lossless bitstream compression.[6]

Licensing

Since G.711 was released in 1972 its patents have long since expired, so it is freely available.[7]

References

[1] G.711 : Pulse code modulation (PCM) of voice frequencies; ITU-T Recommendation (11/1988) (http://www.itu.int/rec/T-REC-G.711/), Retrieved on 2009-07-08

[2] ITU-T (2009-07-17). "ITU-T Newslog - Voice codec gets new lossless compression" (http://www.itu.int/ITU-T/newslog/Voice+Codec+Gets+New+Lossless+Compression.aspx). . Retrieved 2010-02-28.

[3] ITU-T. "G.711.0 : Lossless compression of G.711 pulse code modulation" (http://www.itu.int/rec/T-REC-G.711.0-200909-P/en). . Retrieved 2010-02-28.

[4] *Recent Audio/Speech Coding Developments in ITU-T and future trends* (http://www.eurasip.org/Proceedings/Eusipco/Eusipco2008/plenaries/lamblin.pdf), 2008-08, , retrieved 2010-02-28

[5] ITU-T (2008) G.711.1 : Wideband embedded extension for G.711 pulse code modulation (http://www.itu.int/rec/T-REC-G.711.1/en) Retrieved on 2009-06-19

[6] Nokia Research Center (2009-04-06), *Coding standards* (http://www.ficora.fi/attachments/suomiry/5lN8SYM14/Stand09-Hagqvist.pdf), , retrieved 2010-03-01

[7] "G711 Spec" (http://www.itu.int/ITU-T/recommendations/related_ps.aspx?id_prod=911). . Retrieved 2011-07-05.

External links

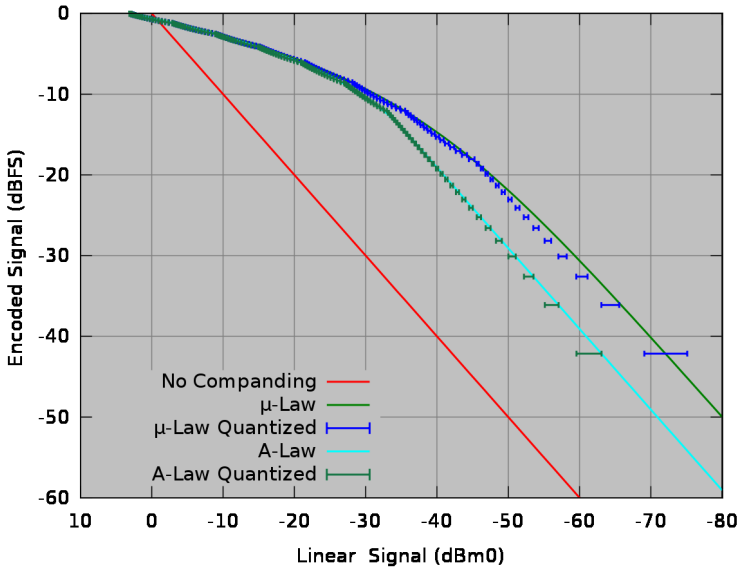
- ITU-T Recommendation G.711 (http://www.itu.int/rec/dologin_pub.asp?lang=e&id=T-REC-G.711-198811-I!!PDF-E&type=items) - (STD.ITU-T RECMN G.711-ENGL 1989)
- ITU-T G.711 page (http://www.itu.int/rec/recommendation.asp?type=folders&lang=e&parent=T-REC-G.711)
- ITU-T G.191 software tools for speech and audio coding, including G.711 C code (http://www.itu.int/rec/T-REC-G.191/en)
- Code Project C# implementation of G.711 with source code (http://www.codeproject.com/csharp/g711audio.asp)
- RFC 3551 - RTP Profile for Audio and Video Conferences with Minimal Control (http://tools.ietf.org/html/rfc3551#page-28) - G.711 - PCMA and PCMU definition.
- RFC 4856 - Registration of Media Type audio/PCMA and audio/PCMU (http://tools.ietf.org/html/rfc4856#page-21)
- RFC 5391 - RTP Payload Format for ITU-T Recommendation G.711.1 (PCMA-WB and PCMU-WB)

A-law algorithm

An **A-law algorithm** is a standard companding algorithm, used in European digital communications systems to optimize, *i.e.,* modify, the dynamic range of an analog signal for digitizing.

It is similar to the μ-law algorithm used in North America and Japan.

For a given input *x*, the equation for A-law encoding is as follows,



Graph of μ-law & A-law algorithms

$$F(x) = \operatorname{sgn}(x) \begin{cases} \frac{A|x|}{1+\ln(A)}, & |x| < \frac{1}{A} \\ \frac{1+\ln(A|x|)}{1+\ln(A)}, & \frac{1}{A} \le |x| \le 1, \end{cases}$$

where *A* is the compression parameter. In Europe, $A = 87.7$; the value 87.6 is also used.

A-law expansion is given by the inverse function,

$$F^{-1}(y) = \operatorname{sgn}(y) \begin{cases} \frac{|y|(1+\ln(A))}{A}, & |y| < \frac{1}{1+\ln(A)} \\ \frac{\exp(|y|(1+\ln(A))-1)}{A}, & \frac{1}{1+\ln(A)} \le |y| < 1. \end{cases}$$

The reason for this encoding is that the wide dynamic range of speech does not lend itself well to efficient linear digital encoding. A-law encoding effectively reduces the dynamic range of the signal, thereby increasing the coding efficiency and resulting in a signal-to-distortion ratio that is superior to that obtained by linear encoding for a given

number of bits.

Comparison to μ-law

The μ-law algorithm provides a slightly larger dynamic range than the A-law at the cost of worse proportional distortion for small signals. By convention, A-law is used for an international connection if at least one country uses it.

External links

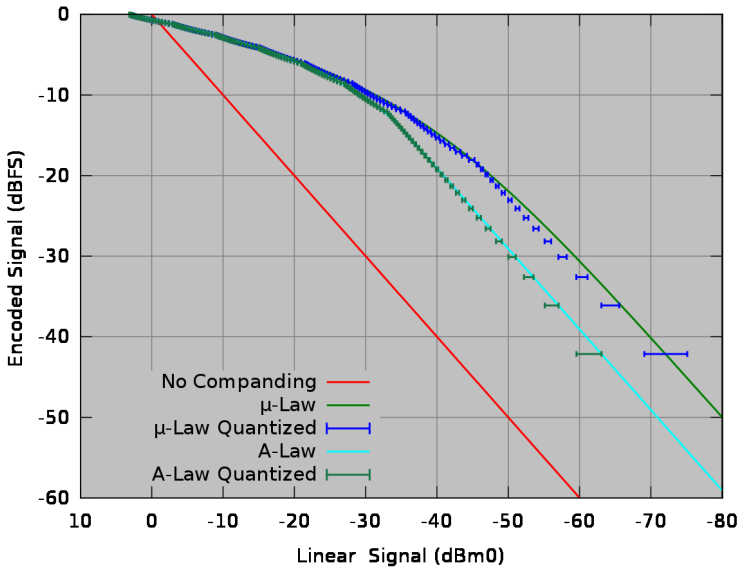
- Waveform Coding Techniques [1] - Has details of implementation (but note that the A-law equation is incorrect)
- A-Law and μ-law Companding Implementations Using the TMS320C54x [2] (PDF)
- A-law and μ-law realisation (in C) [3]

References

[1] http://www.cisco.com/en/US/tech/tk1077/technologies_tech_note09186a00801149b3.shtml
[2] http://www.eettaiwan.com/ARTICLES/2001MAY/PDF1/2001MAY02_NTEK_DSP_AN1135.PDF
[3] http://hazelware.luggle.com/tutorials/mulawcompression.html

μ-law algorithm

The **μ-law algorithm** (sometimes written "mu-law", or misspelled "u-law") is a companding algorithm, primarily used in the digital telecommunication systems of North America and Japan. Companding algorithms reduce the dynamic range of an audio signal. In analog systems, this can increase the signal-to-noise ratio (SNR) achieved during transmission, and in the digital domain, it can reduce the quantization error (hence increasing signal to quantization noise ratio). These SNR increases can be traded instead for reduced bandwidth for equivalent SNR.



Companding of μ-law and A-law algorithms

It is similar to the A-law algorithm used in regions where digital telecommunication signals are carried on E-1 circuits, e.g. Europe.

Algorithm types

There are two forms of this algorithm—an analog version, and a quantized digital version.

Continuous

For a given input x, the equation for μ-law encoding is[1]

$$F(x) = \operatorname{sgn}(x)\frac{\ln(1+\mu|x|)}{\ln(1+\mu)} \qquad -1 \le x \le 1,$$

where μ = 255 (8 bits) in the North American and Japanese standards. It is important to note that the range of this function is −1 to 1.

μ-law expansion is then given by the inverse equation:

$$F^{-1}(y) = \operatorname{sgn}(y)(1/\mu)((1+\mu)^{|y|}-1) \qquad -1 \le y \le 1$$

The equations are culled from Cisco's Waveform Coding Techniques [1].

Discrete

This is defined in ITU-T Recommendation G.711.[2]

G.711 is unclear about what the values at the limit of a range code up as. (e.g. whether +31 codes to 0xEF or 0xF0). However G.191 provides example C code for a μ-law encoder which gives the following encoding. Note the difference between the positive and negative ranges. e.g. the negative range corresponding to +30 to +1 is −31 to −2. This is accounted for by the use of a 1's complement (simple bit inversion) rather than 2's complement to convert a negative value to a positive value during encoding.

Quantized μ-law algorithm

14 bit Binary Linear input code	8 bit Compressed code
+8158 to +4063 in 16 intervals of 256	0x80 + interval number
+4062 to +2015 in 16 intervals of 128	0x90 + interval number
+2014 to +991 in 16 intervals of 64	0xA0 + interval number
+990 to +479 in 16 intervals of 32	0xB0 + interval number
+478 to +223 in 16 intervals of 16	0xC0 + interval number
+222 to +95 in 16 intervals of 8	0xD0 + interval number
+94 to +31 in 16 intervals of 4	0xE0 + interval number
+30 to +1 in 15 intervals of 2	0xF0 + interval number
0	0xFF
−1	0x7F
−31 to −2 in 15 intervals of 2	0x70 + interval number
−95 to −32 in 16 intervals of 4	0x60 + interval number
−223 to −96 in 16 intervals of 8	0x50 + interval number
−479 to −224 in 16 intervals of 16	0x40 + interval number
−991 to −480 in 16 intervals of 32	0x30 + interval number
−2015 to −992 in 16 intervals of 64	0x20 + interval number
−4063 to −2016 in 16 intervals of 128	0x10 + interval number
−8159 to −4064 in 16 intervals of 256	0x00 + interval number

Implementation

There are three ways of implementing a μ-law algorithm:

Analog

Use an amplifier with non-linear gain to achieve companding entirely in the analog domain.

Non-linear ADC

Use an Analog to Digital Converter with quantization levels which are unequally spaced to match the μ-law algorithm.

Digital

Use the quantized digital version of the μ-law algorithm to convert data once it is in the digital domain.

Usage justification

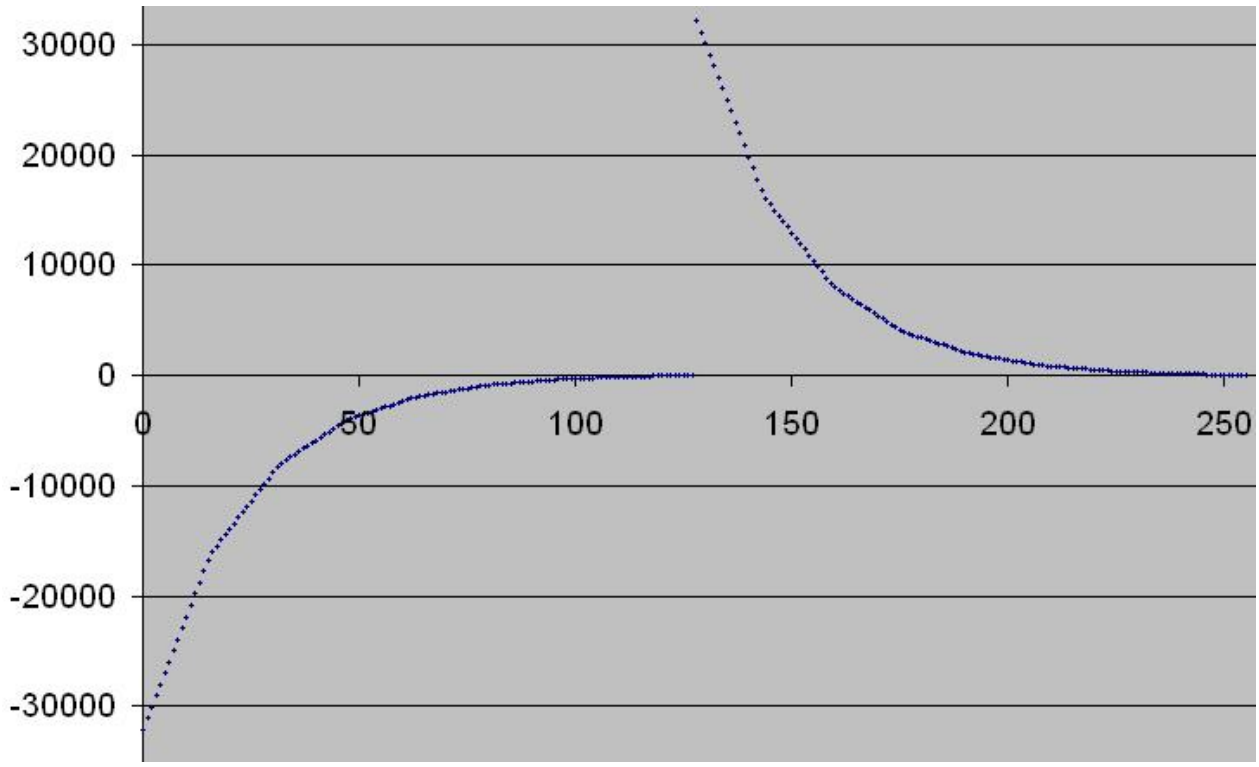
This encoding is used because speech has a wide dynamic range. In the analog world, when mixed with a relatively constant background noise source, the finer detail is lost. Given that the precision of the detail is compromised anyway, and assuming that the signal is to be perceived as audio by a human, one can take advantage of the fact that perceived intensity (loudness) is logarithmic[3] by compressing the signal using a logarithmic-response op-amp. In telco circuits, most of the noise is injected on the lines, thus after the compressor, the intended signal will be perceived as significantly louder than the static, compared to an un-compressed source. This became a common telco solution, and thus, prior to common digital usage, the μ-law specification was developed to define an inter-compatible standard.

As the digital age dawned, it was noted that this pre-existing algorithm had the effect of significantly reducing the number of bits needed to encode recognizable human voice. Using μ-law, a sample could be effectively encoded in as few as 8 bits, a sample size that conveniently matched the symbol size of most standard computers.

μ-law encoding effectively reduced the dynamic range of the signal, thereby increasing the coding efficiency while biasing the signal in a way that results in a signal-to-distortion ratio that is greater than that obtained by linear encoding for a given number of bits. This is an early form of perceptual audio encoding.

The μ-law algorithm is also used in the .au format, which dates back at least to the SPARCstation 1 as the native method used by Sun's /dev/audio interface, widely used as a de facto standard for Unix sound. The .au format is also used in various common audio APIs such as the classes in the `sun.audio` Java package in Java 1.1 and in some C# methods.

This plot illustrates how μ-law concentrates sampling in the smaller (softer) values. The values of a μ-law byte 0-255 are the horizontal axis, the vertical axis is the 16 bit linear decoded value. This image was generated with the Sun Microsystems c routine g711.c commonly available on the Internet.



Comparison with A-law

The μ-law algorithm provides a slightly larger dynamic range than the A-law at the cost of worse proportional distortion for small signals. By convention, A-law is used for an international connection if at least one country uses it.

References

This article incorporates public domain material from the General Services Administration document "Federal Standard 1037C" [4].

[1] "Cisco - Waveform Coding Techniques" (http://www.cisco.com/en/US/tech/tk1077/technologies_tech_note09186a00801149b3.shtml). . Retrieved 2008-07-29.
[2] "ITU-T Recommendation G.711" (http://www.itu.int/rec/dologin_pub.asp?lang=e&id=T-REC-G.711-198811-I!!PDF-E&type=items). .
[3] Wikipedia on sound
[4] http://www.its.bldrdoc.gov/fs-1037/fs-1037c.htm

External links

- Waveform Coding Techniques (http://www.cisco.com/en/US/tech/tk1077/technologies_tech_note09186a00801149b3.shtml) – details of implementation
- A-Law and mu-Law Companding Implementations Using the TMS320C54x (http://focus.ti.com/lit/an/spra163a/spra163a.pdf) (PDF)
- TMS320C6000 μ-Law and A-Law Companding with Software or the McBSP (http://focus.ti.com/lit/an/spra634/spra634.pdf) (PDF)
- A-law and μ-law realisation (in C) (http://hazelware.luggle.com/tutorials/mulawcompression.html) (ctrl-a "highlight all" to see linked black-on-black text).

G.729

G.729 is an audio data compression algorithm for voice that compresses digital voice in packets of 10 milliseconds duration. It is officially described as *Coding of speech at 8 kbit/s using code-excited linear prediction* speech coding (CS-ACELP).[1]

Because of its low bandwidth requirements, G.729 is mostly used in Voice over Internet Protocol (VoIP) applications where bandwidth must be conserved, such as conference calls. Standard G.729 operates at a bit rate of 8 kbit/s, but there are extensions, which provide rates of 6.4 kbit/s (Annex D, F, H, I, C+) and 11.8 kbit/s (Annex E, G, H, I, C+) for worse and better speech quality, respectively.

G.729 has been extended with various features, commonly designated as G.729a and G.729b.

Dual-tone multi-frequency signaling (DTMF), fax transmissions, and high-quality audio cannot be transported reliably with this codec. DTMF requires the use of the RTP Payload for DTMF Digits, Telephony Tones, and Telephony Signals as specified in RFC 2833.

G.729 Annexes

G.729 Annexes [2]

Functionality	-	A	B	C	D	E	F	G	H	I	C+	J
Low complexity		X	X									
Fixed-point	X	X	X		X	X	X	X	X	X		X
Floating-point				X							X	
8 kbit/s	X	X	X	X	X	X	X	X	X	X	X	X
6.4 kbit/s					X		X		X	X	X	
11.8 kbit/s						X		X	X	X	X	
DTX			X				X	X		X	X	
Embedded variable bit rate, wideband												X

G.729 Annex A

G.729a is a compatible extension of G.729, but requires less computational power. This lower complexity, however, bears the cost of marginally reduced speech quality.

G.729a was developed by a consortium of organizations: France Telecom, Mitsubishi Electric Corporation, Nippon Telegraph and Telephone Corporation (NTT)

The features of G.729a are:

- Sampling frequency 8 kHz/16-bit (80 samples for 10 ms frames)
- Fixed bit rate (8 kbit/s 10 ms frames)
- Fixed frame size (10 bytes for 10 ms frame)
- Algorithmic delay is 15 ms per frame, with 5 ms look-ahead delay
- G.729a is a hybrid speech coder which uses Algebraic Code Excited Linear Prediction (ACELP)
- The complexity of the algorithm is rated at 15, using a relative scale where G.711 is 1 and G.723.1 is 25.
- PSQM testing under ideal conditions yields Mean Opinion Scores of 4.04 for G.729a, compared to 4.45 for G.711 (μ-law)

- PSQM testing under network stress yields Mean Opinion Scores of 3.51 for G.729a, compared to 4.13 for G.711 (μ-law)

G.729 Annex B

G.729 has been extended in Annex B (G.729b) which provides a silence compression method that enables a voice activity detection (VAD) module. It is used to detect voice activity in the signal. It also includes a discontinuous transmission (DTX) module which decides on updating the background noise parameters for non speech (noisy frames). It uses 2-byte Silence Insertion Descriptor (SID) frames transmitted to initiate comfort noise generation (CNG). If transmission is stopped, and the link goes quiet because of no speech, the receiving side might assume that the link has been cut. By inserting comfort noise, analog hiss is simulated digitally during silence to assure the receiver that the link is active and operational.

Other extensions

Recently, G.729 has been extended (with Annex J) to provide support for wideband speech and audio coding, i.e., the transmitted acoustic frequency range is extended to 50 Hz - 7 kHz. The respective extension to G.729 is referred to as G.729.1. The G.729.1 codec is hierarchically organized: Its bit rate and the obtained quality are adjustable by simple bitstream truncation.

Licensing

G.729 includes patents from several companies and is licensed by Sipro Lab Telecom. Sipro Lab Telecom is the authorized Intellectual Property Licensing Administrator for G.729 technology and patent pool.[3][4][5][6] In a number of countries, the use of G.729 may require a license fee and/or royalty fee.[5]

References

[1] International Telecommunications Union, Standardization Sector (ITU-T), Study Group 15 (1993-1996), *Recommendation G.729*, March 1996.

[2] ITU-T (2007-01) (PDF). *G.729 : Coding of speech at 8 kbit/s using conjugate-structure algebraic-code-excited linear prediction (CS-ACELP)* (http://www.itu.int/rec/dologin_pub.asp?lang=e&id=T-REC-G.729-200701-I!!SOFT-ZST-E&type=items). p. i. . Retrieved 2009-07-21.

[3] Sipro Lab Telecom Website (http://www.sipro.com)

[4] VoiceAge Corporation (2007-10-14). "G.729 Licensing" (http://web.archive.org/web/20071014162058/http://www.voiceage.com/licg729.php). Archived from the original (http://www.voiceage.com/licg729.php) on 2007-10-14. . Retrieved 2009-09-17.

[5] Sipro Lab Telecom (2007-10-25). "FAQ G.729 and G.723.1" (http://web.archive.org/web/20071025051836/http://www.sipro.com/faq.php). Archived from the original (http://www.sipro.com/faq.php) on 2007-10-25. . Retrieved 2009-09-17.

[6] Sipro Lab Telecom (2006-10-29). "G.729 IPR Pool" (http://web.archive.org/web/20061029005724/http://www.sipro.com/g729onestop.php). Archived from the original (http://www.sipro.com/g729onestop.php) on 2006-10-29. . Retrieved 2009-09-17.

External links

- ITU-T Recommendation G.729 (http://www.itu.int/rec/recommendation.asp?type=folders&lang=e&parent=T-REC-G.729) - technical specification
- G.729 Error Recovery for Internet Telephony (http://www.cs.columbia.edu/techreports/cucs-016-01.pdf#search="g.729 error recovery for internet telephony")
- ITU Patent database (http://www.itu.int/ITU-T/dbase/patent/index.html)
- Sipro Lab Telecom (administers the patent pools for G.723.1 and G.729) (http://www.sipro.com/)
- Voiceage's free G.729 implementation (http://www.voiceage.com/openinit_g729.php)

G.722

G.722[1] is a ITU-T standard 7 kHz wideband speech codec operating at 48, 56 and 64 kbit/s. It was approved by ITU-T in November 1988. Technology of the codec is based on sub-band ADPCM (SB-ADPCM).

G.722 sample audio data at a rate of 16 kHz (using 14 bits), double that of traditional telephony interfaces, which results in superior audio quality and clarity.

Other ITU-T 7 kHz wideband codecs include G.722.1 and G.722.2. These codecs are not variants of G.722 and they use different patented compression technologies. G.722.1 is based on Siren codecs and offers lower bit-rate compressions. A more recent G.722.2, also known as AMR-WB ("Adaptive Multirate Wideband") is based on ACELP and offers even lower bit-rate compressions, as well as the ability to quickly adapt to varying compressions as the network topography mutates. In the latter case, bandwidth is automatically conserved when network congestion is high. When congestion returns to a normal level, a lower-compression, higher-quality bitrate is restored.

Applications

G.722 is an ITU standard codec that provides 7 kHz wideband audio at data rates from 48, 56 and 64 kbit/s. This is useful for voice over IP applications, such as on a local area network where network bandwidth is readily available, and offers a significant improvement in speech quality over older narrowband codecs such as G.711, without an excessive increase in implementation complexity. Environments where bandwidth is more constrained may prefer one of the more bit-efficient codecs, such as G.722.1 (Siren7) or G.722.2 (AMR-WB).

G.722 has also been widely used by radio broadcasters for sending commentary grade audio over a single 56 or 64 kbit/s ISDN B-channel (the least significant bit is dropped on 56kb circuits).

RTP encapsulation

G.722 VoIP is typically carried in RTP payload type 9.[2] Note that IANA records the clock rate for type 9 G.722 as 8 kHz (instead of 16 kHz), RFC3551[3] clarifies that this is due to a historical error and is retained in order to maintain backward compatibility. Consequently correct implementations represent the value 8,000 where required but encode and decode audio at 16 kHz.

Whilst G.722 allows for bitrates of 64, 56 and 48 kbit/s, in practice, data is encoded at 64 kbit/s, with bits from the lower sub-band being used to encode auxiliary data. The greater the number of bits allocated to aux data, the lower the bit rate.

Licensing

G.722 patents have expired, so it is freely available.

G722 specification [4]

References

[1] ITU-T G.722 page (http://www.itu.int/rec/recommendation.asp?type=folders&lang=e&parent=T-REC-G.722) ITU-T Recommendation G.722 (11/88), "7 kHz audio-coding within 64 kbit/s"

[2] IANA: Authoritative repository for RTP Parameters (http://www.iana.org/assignments/rtp-parameters)

[3] RFC 3551 (http://tools.ietf.org/html/rfc3551)Request For Comments 3551: RTP Profile for Audio and Video Conferences with Minimal Control. Schulzrinne & Casener, July 2003. Also Internet Standard 65.

[4] http://www.itu.int/rec/dologin_pub.asp?lang=e&id=T-REC-G.722-198811-I!!PDF-E&type=items

G.726

G.726 is an ITU-T ADPCM speech codec standard covering the transmission of voice at rates of 16, 24, 32, and 40 kbit/s. It was introduced to supersede both G.721, which covered ADPCM at 32 kbit/s, and G.723, which described ADPCM for 24 and 40 kbit/s. G.726 also introduced a new 16 kbit/s rate. The four bit rates associated with G.726 are often referred to by the bit size of a sample, which are 2-bits, 3-bits, 4-bits, and 5-bits respectively.

The most commonly used mode is 32 kbit/s, which doubles the usable network capacity by using half the rate of G.711. It is primarily used on international trunks in the phone network and is the standard codec used in DECT wireless phone systems. The principal application of 24 and 16 kbit/s channels is for overload channels carrying voice in digital circuit multiplication equipment (DCME). The principal application of 40 kbit/s channels is to carry data modem signals in DCME, especially for modems operating at greater than 4800 kbit/s.

History

G.721 was introduced in 1984, while G.723 was introduced in 1988. They were folded into G.726 in 1990.

G.727 was introduced at the same time as G.726, and includes the same bit rates, but is optimized for packet circuit multiplex equipment (PCME) environment. This is achieved by embedding 2-bit quantizer to 3-bit quantizer and same for the higher modes. This allows dropping of the least significant bit from the bit stream without adverse effects on speech signal.

Features

- Sampling frequency 8 kHz
- 16 kbit/s, 24 kbit/s, 32 kbit/s, 40 kbit/s bit rates available
- Generates a bitstream, therefore frame length is determined by packetization (typically 80 samples for 10 ms frame size)
- Typical algorithmic delay is 0.125 ms, with no look-ahead delay
- G.726 is a waveform speech coder which uses Adaptive Differential Pulse Code Modulation (ADPCM)
- PSQM testing under ideal conditions yields Mean Opinion Scores of 4.30 for G.726 (32 kbit/s), compared to 4.45 for G.711 (μ-law)
- PSQM testing under network stress yields Mean Opinion Scores of 3.79 for G.726 (32 kbit/s), compared to 4.13 for G.711 (μ-law)
- 40 kbit/s G.726 can carry 12000 bit/s and slower modem signals, while 32 kbit/s G.726 can carry 2400 bit/s and slower modem signals well and 4800 bit/s with some more degradation than clear channel codecs.

External links

- ITU-T G.726 page [1]
- ITU-T G.191 software tools for speech and audio coding, including G.726 C code [2]
- RFC 3551 - RTP Profile for Audio and Video Conferences with Minimal Control, G726-40, G726-32, G726-24, and G726-16 [3]

References

[1] http://www.itu.int/rec/recommendation.asp?type=folders&lang=e&parent=T-REC-G.726
[2] http://www.itu.int/rec/T-REC-G.191/en
[3] http://tools.ietf.org/html/rfc3551#page-18

Network address translation

In computer networking, **network address translation** (**NAT**) is the process of modifying IP address information in IP packet headers while in transit across a traffic routing device.

The simplest type of NAT provides a one-to-one translation of IP addresses. RFC 2663 refers to this type of NAT as **basic NAT**. It is often also referred to as **one-to-one NAT**. In this type of NAT only the IP addresses, IP header checksum and any higher level checksums that include the IP address need to be changed. The rest of the packet can be left untouched (at least for basic TCP/UDP functionality, some higher level protocols may need further translation). Basic NATs can be used when there is a requirement to interconnect two IP networks with incompatible addressing.

However it is common to hide an entire IP address space, usually consisting of private IP addresses, behind a single IP address (or in some cases a small group of IP addresses) in another (usually public) address space. To avoid ambiguity in the handling of returned packets, a one-to-many NAT must alter higher level information such as TCP/UDP ports in outgoing communications and must maintain a translation table so that return packets can be correctly translated back. RFC 2663 uses the term **NAPT** (**network address and port translation**) for this type of NAT. Other names include **PAT** (**port address translation**), **IP masquerading**, **NAT Overload** and **many-to-one NAT**. Since this is the most common type of NAT it is often referred to simply as NAT.

As described, the method enables communication through the router only when the conversation originates in the masqueraded network, since this establishes the translation tables. For example, a web browser in the masqueraded network can browse a website outside, but a web browser outside could not browse a web site in the masqueraded network. However, most NAT devices today allow the network administrator to configure translation table entries for permanent use. This feature is often referred to as "static NAT" or port forwarding and allows traffic originating in the "outside" network to reach designated hosts in the masqueraded network.

In the mid-1990s NAT became a popular tool for alleviating the consequences of IPv4 address exhaustion.[1] It has become a common, indispensable feature in routers for home and small-office Internet connections. Most systems using NAT do so in order to enable multiple hosts on a private network to access the Internet using a single public IP address.

Network address translation has serious drawbacks on the quality of Internet connectivity and requires careful attention to the details of its implementation. In particular, all types of NAT break the originally envisioned model of IP end-to-end connectivity across the Internet and NAPT makes it difficult for systems behind a NAT to accept incoming communications. As a result, NAT traversal methods have been devised to alleviate the issues encountered.

One-to-many NATs

The majority of NATs map multiple private hosts to one publicly exposed IP address. In a typical configuration, a local network uses one of the designated "private" IP address subnets (RFC 1918). A router on that network has a private address in that address space. The router is also connected to the Internet with a "public" address assigned by an Internet service provider. As traffic passes from the local network to the Internet, the source address in each packet is translated on the fly from a private address to the public address. The router tracks basic data about each active connection (particularly the destination address and port). When a reply returns to the router, it uses the connection tracking data it stored during the outbound phase to determine the private address on the internal network to which to forward the reply.

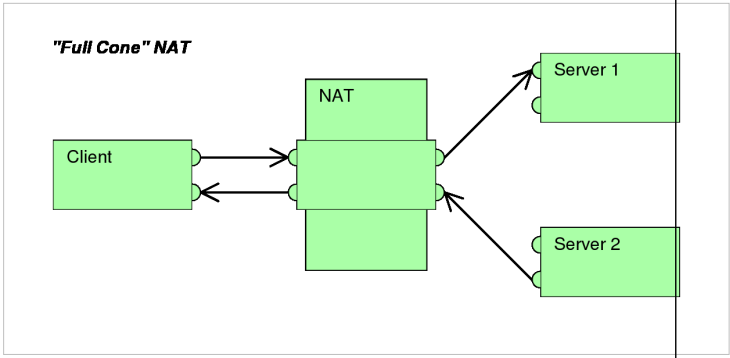
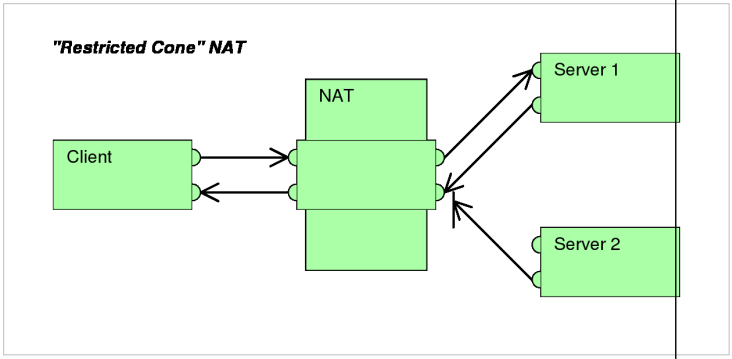
All Internet packets have a source IP address and a destination IP address. Typically packets passing from the private network to the public network will have their source address modified while packets passing from the public network back to the private network will have their destination address modified. More complex configurations are also

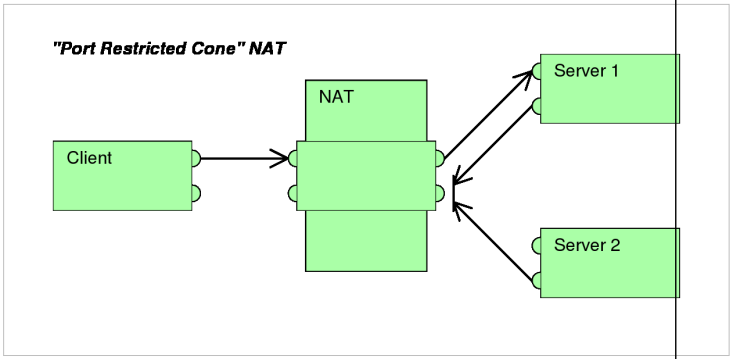
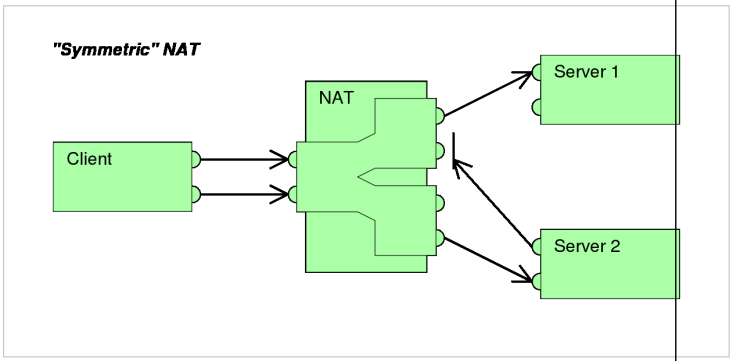
possible.

To avoid ambiguity in how to translate returned packets, further modifications to the packets are required. The vast bulk of Internet traffic is TCP and UDP packets, and for these protocols the port numbers are changed so that the combination of IP and port information on the returned packet can be unambiguously mapped to the corresponding private address and port information. Protocols not based on TCP or UDP require other translation techniques. ICMP packets typically relate to an existing connection and need to be mapped using the same IP and port mappings as that connection.

Methods of Port translation

There are several ways of implementing network address and port translation. In some application protocols that use IP address information, the application running on a node in the masqueraded network needs to determine the external address of the NAT, i.e., the address that its communication peers detect, and, furthermore, often needs to examine and categorize the type of mapping in use. Usually this is done because it is desired to set up a direct communications path (either to save the cost of taking the data via a server or to improve performance) between two clients both of which are behind separate NATs. For this purpose, the Simple traversal of UDP over NATs (STUN) protocol was developed (RFC 3489, March 2003). It classified NAT implementation as *full cone NAT*, *(address) restricted cone NAT*, *port restricted cone NAT* or *symmetric NAT* and proposed a methodology for testing a device accordingly. However, these procedures have since been deprecated from standards status, as the methods have proven faulty and inadequate to correctly assess many devices. New methods have been standardized in RFC 5389 (October 2008) and the STUN acronym now represents the new title of the specification: *Session Traversal Utilities for NAT*.

Full-cone NAT, also known as *one-to-one NAT* • Once an internal address (iAddr:iPort) is mapped to an external address (eAddr:ePort), any packets from iAddr:iPort will be sent through eAddr:ePort. • *Any external host* can send packets to iAddr:iPort by sending packets to eAddr:ePort.	
(Address) restricted cone NAT • Once an internal address (iAddr:iPort) is mapped to an external address (eAddr:ePort), any packets from iAddr:iPort will be sent through eAddr:ePort. • An external host (*hAddr:any*) can send packets to iAddr:iPort by sending packets to eAddr:ePort only if iAddr:iPort has previously sent a packet to hAddr:*any*. "Any" means the port number doesn't matter.	

Port-restricted cone NAT Like an address restricted cone NAT, but the restriction includes port numbers. • Once an internal address (iAddr:iPort) is mapped to an external address (eAddr:ePort), any packets from iAddr:iPort will be sent through eAddr:ePort. • An external host (*hAddr:hPort*) can send packets to iAddr:iPort by sending packets to eAddr:ePort only if iAddr:iPort has previously sent a packet to hAddr:hPort.	
Symmetric NAT • Each request from the same internal IP address and port to a specific destination IP address and port is mapped to a unique external source IP address and port, if the same internal host sends a packet even with the same source address and port but to a different destination, a different mapping is used. • Only an external host that receives a packet from an internal host can send a packet back.	

This terminology has been the source of much confusion, as it has proven inadequate at describing real-life NAT behavior.[2] Many NAT implementations combine these types, and it is therefore better to refer to specific individual NAT behaviors instead of using the Cone/Symmetric terminology. Especially, most NAT translators combine *symmetric NAT* for outgoing connections with *static port mapping*, where incoming packets to the external address and port are redirected to a specific internal address and port. Some products can redirect packets to several internal hosts, e.g. to divide the load between a few servers. However, this introduces problems with more sophisticated communications that have many interconnected packets, and thus is rarely used.

Type of NAT and NAT Traversal

The NAT traversal problem arises when two peers behind distinct NAT try to communicate. One way to solve this problem is to use port forwarding, another way is to use various NAT traversal techniques. The most popular technique for TCP NAT traversal is TCP hole punching, which requires the NAT to follow the *port preservation* design for TCP, as explained below.

Many NAT implementations follow the *port preservation* design especially for TCP, which is to say that they use the same values as internal and external port numbers. NAT *port preservation* for outgoing TCP connections is especially important for TCP NAT traversal, because programs usually bind distinct TCP sockets to ephemeral ports for distinct TCP connections, rendering NAT port prediction impossible for TCP.

On the other hand, for UDP, NATs do not need to have *port preservation* because applications usually reuse the same UDP socket to send packets to distinct hosts, making port prediction straightforward, as it is the same source port for each packet.

Furthermore, *port preservation* in NAT for TCP allows P2P protocols to offer less complexity and less latency because there is no need to use a third party to discover the NAT port since the application already knows the NAT port.[3]

However, if two internal hosts attempt to communicate with the same external host using the same port number, the external port number used by the second host will be chosen at random. Such NAT will be sometimes perceived as *(address) restricted cone NAT* and other times as *symmetric NAT*.

Recent studies have shown that roughly 70% of clients in P2P networks employ some form of NAT.[4]

Implementation

Establishing Two-Way Communication

Every TCP and UDP packet contains both a source IP address and source port number as well as a destination IP address and destination port number. The port address/IP address pair forms a socket. In particular, the source port address and source IP address form the source socket.

For publicly accessible services such as web servers and mail servers the port number is important. For example, port 80 connects to the web server software and port 25 to a mail server's SMTP daemon. The IP address of a public server is also important, similar in global uniqueness to a postal address or telephone number. Both IP address and port must be correctly known by all hosts wishing to successfully communicate.

Private IP addresses as described in RFC 1918 are significant only on private networks where they are used, which is also true for host ports. Ports are unique endpoints of communication on a host, so a connection through the NAT device is maintained by the combined mapping of port and IP address.

PAT (Port Address Translation) resolves conflicts that would arise through two different hosts using the same source port number to establish unique connections at the same time.

An Analogy

A NAT device is similar to a phone system at an office that has one public telephone number and multiple extensions. Outbound phone calls made from the office all appear to come from the same telephone number. However, an incoming call that does not specify an extension cannot be transferred to an individual inside the office. In this scenario, the office is a private LAN, the main phone number is the public IP address, and the individual extensions are unique port numbers.[5]

Translation of the Endpoint

With NAT, all communication sent to external hosts actually contain the *external* IP address and port information of the NAT device instead of internal host IPs or port numbers.

- When a computer on the private (internal) network sends a packet to the external network, the NAT device replaces the internal IP address in the source field of the packet header (*sender's address*) with the external IP address of the NAT device. PAT may then assign the connection a port number from a pool of available ports, inserting this port number in the source port field (much like the *post office box number*), and forwards the packet to the external network. The NAT device then makes an entry in a translation table containing the internal IP address, original source port, and the translated source port. Subsequent packets from the same connection are translated to the same port number.
- The computer receiving a packet that has undergone NAT establishes a connection to the port and IP address specified in the altered packet, oblivious to the fact that the supplied address is being translated (analogous to using a *post office box number*).
- A packet coming from the external network is mapped to a corresponding internal IP address and port number from the translation table, replacing the external IP address and port number in the incoming packet header (similar to the translation from *post office box number* to *street address*). The packet is then forwarded over the inside network. Otherwise, if the destination port number of the incoming packet is not found in the translation table, the packet is dropped or rejected because the PAT device doesn't know where to send it.

NAT will only translate IP addresses and ports of its internal hosts, hiding the true endpoint of an internal host on a private network.

Visibility of Operation

NAT operation is typically transparent to both the internal and external hosts.

Typically the internal host is aware of the true IP address and TCP or UDP port of the external host. Typically the NAT device may function as the default gateway for the internal host. However the external host is only aware of the public IP address for the NAT device and the particular port being used to communicate on behalf of a specific internal host.

NAT and TCP/UDP

"Pure NAT", operating on IP alone, may or may not correctly parse protocols that are totally concerned with IP information, such as ICMP, depending on whether the payload is interpreted by a host on the "inside" or "outside" of translation. As soon as the protocol stack is traversed, even with such basic protocols as TCP and UDP, the protocols will break unless NAT takes action beyond the network layer.

IP packets have a checksum in each packet header, which provides error detection only for the header. IP datagrams may become fragmented and it is necessary for a NAT to reassemble these fragments to allow correct recalculation of higher-level checksums and correct tracking of which packets belong to which connection.

The major transport layer protocols, TCP and UDP, have a checksum that covers all the data they carry, as well as the TCP/UDP header, plus a "pseudo-header" that contains the source and destination IP addresses of the packet carrying the TCP/UDP header. For an originating NAT to pass TCP or UDP successfully, it must recompute the TCP/UDP header checksum based on the translated IP addresses, not the original ones, and put that checksum into the TCP/UDP header of the first packet of the fragmented set of packets. The receiving NAT must recompute the IP checksum on every packet it passes to the destination host, and also recognize and recompute the TCP/UDP header using the retranslated addresses and pseudo-header. This is not a completely solved problem. One solution is for the receiving NAT to reassemble the entire segment and then recompute a checksum calculated across all packets.

The originating host may perform Maximum transmission unit (MTU) path discovery to determine the packet size that can be transmitted without fragmentation, and then set the *don't fragment* (DF) bit in the appropriate packet header field.

Destination network address translation (DNAT)

DNAT is a technique for transparently changing the destination IP address of an en-route packet and performing the inverse function for any replies. Any router situated between two endpoints can perform this transformation of the packet.

DNAT is commonly used to publish a service located in a private network on a publicly accessible IP address. This use of DNAT is also called port forwarding, or DMZ when used on an entire server, which becomes exposed to the WAN, becoming analogous to an undefended military demilitarised zone (DMZ).

SNAT

The meaning of the term *SNAT* varies by vendor. Many vendors have proprietary definitions for *SNAT*. A common expansion is *source NAT*, the counterpart of *destination NAT* (*DNAT*). Microsoft uses the acronym for *Secure NAT*, in regard to the ISA Server. For Cisco Systems, *SNAT* means *stateful NAT*. For Watchguard Systems, *SNAT* means *static NAT*.

Microsoft's Secure network address translation (SNAT) is part of Microsoft's Internet Security and Acceleration Server and is an extension to the NAT driver built into Microsoft Windows Server. It provides connection tracking and filtering for the additional network connections needed for the FTP, ICMP, H.323, and PPTP protocols as well as the ability to configure a transparent HTTP proxy.

Secure network address translation

In computer networking, the process of network address translation done in a secure way involves rewriting the source and/or destination addresses of IP packets as they pass through a router or firewall.

Dynamic network address translation

Dynamic NAT, just like static NAT, is not common in smaller networks but is found within larger corporations with complex networks. The way dynamic NAT differs from static NAT is that where static NAT provides a one-to-one internal to public static IP address mapping, dynamic NAT doesn't make the mapping to the public IP address static and usually uses a group of available public IP addresses.

Applications affected by NAT

Some Application Layer protocols (such as FTP and SIP) send explicit network addresses within their application data. FTP in active mode, for example, uses separate connections for control traffic (commands) and for data traffic (file contents). When requesting a file transfer, the host making the request identifies the corresponding data connection by its network layer and transport layer addresses. If the host making the request lies behind a simple NAT firewall, the translation of the IP address and/or TCP port number makes the information received by the server invalid. The Session Initiation Protocol (SIP) controls many Voice over IP (VoIP) calls, and suffers the same problem. SIP and SDP may use multiple ports to set up a connection and transmit voice stream via RTP. IP addresses and port numbers are encoded in the payload data and must be known prior to the traversal of NATs. Without special techniques, such as STUN, NAT behavior is unpredictable and communications may fail.

Application layer gateway (ALG) software or hardware may correct these problems. An ALG software module running on a NAT firewall device updates any payload data made invalid by address translation. ALGs obviously need to understand the higher-layer protocol that they need to fix, and so each protocol with this problem requires a separate ALG. For example, on many Linux systems, there are kernel modules called *connection trackers* which serve to implement ALGs. However, ALG does not work if the control channel is encrypted (e.g. FTPS).

Another possible solution to this problem is to use NAT traversal techniques using protocols such as STUN or ICE, or proprietary approaches in a session border controller. NAT traversal is possible in both TCP- and UDP-based applications, but the UDP-based technique is simpler, more widely understood, and more compatible with legacy NATs. In either case, the high level protocol must be designed with NAT traversal in mind, and it does not work reliably across symmetric NATs or other poorly behaved legacy NATs.

Other possibilities are UPnP (Universal Plug and Play) or NAT-PMP (NAT Port Mapping Protocol), but these require the cooperation of the NAT device.

Most traditional client-server protocols (FTP being the main exception), however, do not send layer 3 contact information and therefore do not require any special treatment by NATs. In fact, avoiding NAT complications is

practically a requirement when designing new higher-layer protocols today (e.g. the use of SFTP instead of FTP).

NATs can also cause problems where IPsec encryption is applied and in cases where multiple devices such as SIP phones are located behind a NAT. Phones which encrypt their signaling with IPsec encapsulate the port information within an encrypted packet, meaning that NA(P)T devices cannot access and translate the port. In these cases the NA(P)T devices revert to simple NAT operation. This means that all traffic returning to the NAT will be mapped onto one client causing service to more than one client "behind" the NAT to fail. There are a couple of solutions to this problem: one is to use TLS, which operates at level 4 in the OSI Reference Model and therefore does not mask the port number; another is to encapsulate the IPsec within UDP - the latter being the solution chosen by TISPAN to achieve secure NAT traversal.

The DNS protocol vulnerability announced by Dan Kaminsky on July 8, 2008 is indirectly affected by NAT port mapping. To avoid DNS server cache poisoning, it is highly desirable to not translate UDP source port numbers of outgoing DNS requests from a DNS server which is behind a firewall which implements NAT. The recommended work-around for the DNS vulnerability is to make all caching DNS servers use randomized UDP source ports. If the NAT function de-randomizes the UDP source ports, the DNS server will be made vulnerable.

Advantages of PAT

In addition to the advantages provided by NAT:

- PAT (Port Address Translation) allows many internal hosts to share a single external IP address.
- Users who do not require support for inbound connections do not consume public IP addresses.

Drawbacks

The primary purpose of IP-masquerading NAT is that it has been a practical solution to the impending exhaustion of IPv4 address space. Even large networks can be connected to the Internet with as little as a single IP address. The more common arrangement is having machines that require end-to-end connectivity supplied with a routable IP address, while having machines that do not provide services to outside users behind NAT with only a few IP addresses used to enable Internet access, however, this brings some problems, outlined below.

Some[6] have also called this exact feature a major drawback, since it delays the need for the implementation of IPv6:

> "[...] it is possible that its [NAT's] widespread use will significantly delay the need to deploy IPv6. [...] It is probably safe to say that networks would be better off without NAT [...]"

Hosts behind NAT-enabled routers do not have end-to-end connectivity and cannot participate in some Internet protocols. Services that require the initiation of TCP connections from the outside network, or stateless protocols such as those using UDP, can be disrupted. Unless the NAT router makes a specific effort to support such protocols, incoming packets cannot reach their destination. Some protocols can accommodate one instance of NAT between participating hosts ("passive mode" FTP, for example), sometimes with the assistance of an application-level gateway (see below), but fail when both systems are separated from the Internet by NAT. Use of NAT also complicates tunneling protocols such as IPsec because NAT modifies values in the headers which interfere with the integrity checks done by IPsec and other tunneling protocols.

End-to-end connectivity has been a core principle of the Internet, supported for example by the Internet Architecture Board. Current Internet architectural documents observe that NAT is a violation of the End-to-End Principle, but that NAT does have a valid role in careful design.[7] There is considerably more concern with the use of IPv6 NAT, and many IPv6 architects believe IPv6 was intended to remove the need for NAT.[8]

Because of the short-lived nature of the stateful translation tables in NAT routers, devices on the internal network lose IP connectivity typically within a very short period of time unless they implement NAT keep-alive mechanisms by frequently accessing outside hosts. This dramatically shortens the power reserves on battery-operated hand-held

devices and has thwarted more widespread deployment of such IP-native Internet-enabled devices.

Some Internet service providers (ISPs), especially in India, Russia, parts of Asia and other "developing" regions provide their customers only with "local" IP addresses, due to a limited number of external IP addresses allocated to those entities. Thus, these customers must access services external to the ISP's network through NAT. As a result, the customers cannot achieve true end-to-end connectivity, in violation of the core principles of the Internet as laid out by the Internet Architecture Board.

- Scalability - An implementation that only tracks ports can be quickly depleted by internal applications that use multiple simultaneous connections (such as an HTTP request for a web page with many embedded objects). This problem can be mitigated by tracking the destination IP address in addition to the port (thus sharing a single local port with many remote hosts), at the expense of implementation complexity and CPU/memory resources of the translation device.
- Firewall complexity - Because the internal addresses are all disguised behind one publicly accessible address, it is impossible for external hosts to initiate a connection to a particular internal host without special configuration on the firewall to forward connections to a particular port. Applications such as VOIP, videoconferencing, and other peer-to-peer applications must use NAT traversal techniques to function.

Specifications

IEEE[9] Reverse Address and Port Translation (RAPT, or RAT) allows a host whose real IP address is changing from time to time to remain reachable as a server via a fixed home IP address. In principle, this should allow setting up servers on DHCP-run networks. While not a perfect mobility solution, RAPT together with upcoming protocols like DHCP-DDNS, it may end up becoming another useful tool in the network admin's arsenal.

IETF[10] *RAPT* (IP Reachability Using Twice Network Address and Port Translation) The RAT device maps an IP datagram to its associated CN and 0MN by using three additional fields: the IP protocol type number and the transport layer source and destination connection identifiers (e.g. TCP port number or ICMP echo request/reply ID field).

Cisco *RAPT* implementation is PAT (Port Address Translation) or NAT overloading, and maps multiple private IP addresses to a single public IP address. Multiple addresses can be mapped to a single address because each private address is tracked by a port number. PAT uses unique source port numbers on the inside global IP address to distinguish between translations. The port number is encoded in 16 bits. The total number of internal addresses that can be translated to one external address could theoretically be as high as 65,536 per IP address. Realistically, the number of ports that can be assigned a single IP address is around 4000. PAT will attempt to preserve the original source port. If this source port is already used, PAT will assign the first available port number starting from the beginning of the appropriate port group 0-511, 512-1023, or 1024-65535. When there are no more ports available and there is more than one external IP address configured, PAT moves to the next IP address to try to allocate the original source port again. This process continues until it runs out of available ports and external IP addresses.

3COM **U.S. Patent 6055236** [11] (Method and system for locating network services with distributed network address translation) Methods and system for locating network services with distributed network address translation. Digital certificates are created that allow an external network device on an external network, such as the Internet, to request a service from an internal network device on an internal distributed network address translation network, such as a stub local area network. The digital certificates include information obtained with a Port Allocation Protocol used for distributed network address translation. The digital certificates are published on the internal network so they are accessible to external network devices. An external network device retrieves a digital certificate, extracts appropriate information, and sends a service request packet to an internal network device on an internal distributed network address translation network. The external network device is able to locate and request a service from an internal network device. An external network device can also request a security service, such as an Internet Protocol security ("IPsec") service from an internal network device. The external network device and the internal network device can

establish a security service (e.g., Internet Key Exchange protocol service). The internal network device and external network device can then establish a Security Association using Security Parameter Indexes ("SPI") obtained using a distributed network address translation protocol. External network devices can request services, and security services on internal network devices on an internal distribute network address translation network that were previously unknown and unavailable to the external network devices.

Examples of NAT software

- Internet Connection Sharing (ICS): Windows NAT+DHCP since W98SE
- WinGate: like ICS plus lots of control
- iptables: the Linux packet filter and NAT (interface for NetFilter)
- IPFilter: Solaris, NetBSD, FreeBSD, xMach.
- PF (firewall): The OpenBSD Packet Filter.
- Netfilter Linux packet filter framework

References

[1] www.tcpipguide.com/free/t_IPNetworkAddressTranslationNATProtocol.htm (http://www.tcpipguide.com/free/t_IPNetworkAddressTranslationNATProtocol.htm)

[2] François Audet; and Cullen Jennings (January 2007) (text). *RFC 4787 Network Address Translation (NAT) Behavioral Requirements for Unicast UDP* (http://www.ietf.org/rfc/rfc4787.txt). IETF. . Retrieved 2007-08-29.

[3] "Characterization and Measurement of TCP Traversal through NATs and Firewalls" (http://nutss.gforge.cis.cornell.edu/pub/imc05-tcpnat/). December 2006. .

[4] "Illuminating the shadows: Opportunistic network and web measurement" (http://illuminati.coralcdn.org/stats/). December 2006. .

[5] "The Audio over IP Instant Expert Guide" (http://www.tieline.com/Downloads/Audio-over-IP-Instant-Expert-Guide-v1.pdf). Tieline. January 2010. . Retrieved 2011-08-19.

[6] Larry L. Peterson; and Bruce S. Davie; *Computer Networks: A Systems Approach*, Morgan Kaufmann, 2003, pp. 328-330, ISBN 1-55860-832-X

[7] R. Bush; and D. Meyer; RFC 3439, *Some Internet Architectural Guidelines and Philosophy* (http://www.ietf.org/rfc/rfc3439.txt), December 2002

[8] G. Van de Velde *et al.*; RFC 4864, *Local Network Protection for IPv6* (http://tools.ietf.org/rfc/rfc4864.txt), May 2007

[9] http://ieeexplore.ieee.org/iel4/6056/16183/00749275.pdf

[10] http://www3.ietf.org/proceedings/99nov/I-D/draft-ietf-nat-rnat-00.txt

[11] http://www.google.com/patents?vid=6055236

External links

- NAT-Traversal Test and results (http://nattest.net.in.tum.de)
- Characterization of different TCP NATs (http://nutss.net/pub/imc05-tcpnat/) – Paper discussing the different types of NAT
- Anatomy: A Look Inside Network Address Translators – Volume 7, Issue 3, September 2004 (http://www.cisco.com/en/US/about/ac123/ac147/archived_issues/ipj_7-3/anatomy.html)
- Jeff Tyson, HowStuffWorks: *How Network Address Translation Works* (http://computer.howstuffworks.com/nat.htm/printable)
- NAT traversal techniques in multimedia Networks (http://www.newport-networks.com/whitepapers/nat-traversal1.html) – White Paper from Newport Networks
- NAT traversal for IP Communications (http://www.voiptraversal.com/EyeballAnyfirewallWhitePaper.pdf) – White Paper from Eyeball Networks
- Peer-to-Peer Communication Across Network Address Translators (http://www.brynosaurus.com/pub/net/p2pnat/) (PDF) (http://www.brynosaurus.com/pub/net/p2pnat.pdf) – NAT traversal techniques for UDP and TCP
- http://www.zdnetasia.com/insight/network/0,39044847,39050002,00.htm

- RFCs
  - RFC 1631 (Status: Obsolete) - The IP Network Address Translator (NAT)
  - RFC 1918 - Address Allocation for Private Internets
  - RFC 3022 (Status: Informational) – Traditional IP Network Address Translator (Traditional NAT)
  - RFC 4008 (Status: Standards Track) – Definitions of Managed Objects for Network Address Translators (NAT)
  - RFC 5128 (Status: Informational) - State of Peer-to-Peer (P2P) Communications across Network Address Translators (NATs)
  - RFC 4966 (Status: Informational) - Reasons to Move the Network Address Translator - Protocol Translator (NAT-PT) to Historic Status
- *Speak Freely* End of Life Announcement (http://www.fourmilab.ch/speakfree/unix/) – John Walker's discussion of why he stopped developing a famous program for free Internet communication, part of which is directly related to NAT
- natd (http://www.freebsd.org/doc/en_US.ISO8859-1/books/handbook/network-natd.html)
- SNAT, DNAT and OCS2007R2 (http://www.cainetworks.com/support/training/snat-dnat-ocs.html) – discussing the SNAT in Microsoft OCS 2007R2
- Alternative Taxonomy (Part of the documentation for the IBM iSeries)
  - Static NAT (http://publib.boulder.ibm.com/infocenter/iseries/v5r3/index.jsp?topic=/rzajw/rzajwstatic.htm)
  - Dynamic NAT (http://publib.boulder.ibm.com/infocenter/iseries/v5r3/index.jsp?topic=/rzajw/rzajwdynamic.htm)
  - Masquerade NAT (http://publib.boulder.ibm.com/infocenter/iseries/v5r3/index.jsp?topic=/rzajw/rzajwaddmasq.htm)
- Network Address Translation - NAT (http://blog.ipexpert.com/2009/09/07/network-address-translation-nat/)
- Cisco Systems
  - Document ID 6450: How NAT Works (http://www.cisco.com/en/US/tech/tk648/tk361/technologies_tech_note09186a0080094831.shtml)
  - Document ID 26704: Network Address Translation (NAT) FAQ (http://www.cisco.com/en/US/tech/tk648/tk361/technologies_q_and_a_item09186a00800e523b.shtml)
  - White Paper: Cisco IOS Network Address Translation Overview (http://www.cisco.com/en/US/technologies/tk648/tk361/tk438/technologies_white_paper09186a0080091cb9.html)
  - Cisco IOS NAT Commands Cisco IOS commands (http://www.cisco.com/univercd/cc/td/doc/product/software/ios113ed/cs/csprtd/csprtd11/csnat.htm)
  - Animation Cisco NAT sample (http://www.cisco.com/image/gif/paws/6450/nat.swf)

NAT traversal

NAT traversal is a general term for techniques that establish and maintain Internet protocol connections traversing network address translation (NAT) gateways. Network address translation breaks end-to-end connectivity. Intercepting and modifying traffic can only be performed transparently in the absence of secure encryption and authentication. NAT traversal techniques are typically required for client-to-client networking applications, especially peer-to-peer and Voice over IP (VoIP) deployments. Many techniques exist, but no single method works in every situation since NAT behavior is not standardized. Many NAT traversal techniques require assistance from a server at a publicly routable IP address. Some methods use the server only when establishing the connection, while others are based on relaying all data through it, which adds bandwidth costs and increases latency, detrimental to real-time voice and video communications.

Most NAT behavior-based techniques bypass enterprise security policies. Enterprise security experts prefer techniques that explicitly cooperate with NAT and firewalls, allowing NAT traversal while still enabling marshalling at the NAT to enforce enterprise security policies. From this point of view, the most promising IETF standards are Realm-Specific IP (RSIP) and Middlebox Communications (MIDCOM).

SOCKS, the oldest NAT traversal protocol, is still widely available. In home or small office settings, Universal Plug and Play (UPnP) is supported by most small NAT gateways. NAT-T is commonly used by IPsec virtual private network clients in order to have Encapsulating Security Payload packets traverse NAT.

The NAT traversal problem

NAT devices are commonly used to alleviate IPv4 address exhaustion by allowing the use of private IP addresses on home and corporate networks behind routers with a single public IP address facing the public Internet. The internal network devices communicate with hosts on the external network by changing the source address of outgoing requests to that of the NAT device and relaying replies back to the originating device. This leaves the internal network ill-suited to host servers, as the NAT device has no automatic method of determining the internal host for which incoming packets are destined. This is not a problem for home users behind NAT devices doing general web access and e-mail. However, applications such as peer-to-peer file sharing, VoIP services and the online services of current generation video game consoles require clients to be servers as well, thereby posing a problem for users behind NAT devices, as incoming requests cannot be easily correlated to the proper internal host. Furthermore many of these types of services carry IP address and port number information in the application data, potentially requiring substitution or special traversal techniques for NAT traversal.

NAT traversal and IPsec

In order for IPsec to work through a NAT, the following protocols need to be allowed through the NAT interface(s), e.g. the LAN router:

- Internet Key Exchange (IKE) - User Datagram Protocol (UDP) port 500
- Encapsulating Security Payload (ESP) - IP protocol number 50
- Authentication Header (AH) - IP protocol number 51

or, in case of NAT-T:

- IKE - UDP port 500
- IPsec NAT-T - UDP port 4500

Often this is accomplished on home routers by enabling "IPsec Passthrough".

In Windows XP, NAT-T is enabled by default, but in XP with SP2, has been disabled by default for the case when the VPN server is also behind a NAT device, because of a rare and controversial security issue.[1] IPsec NAT-T

patches are also available for Windows 2000, Windows NT and Windows 98.

One usage of NAT-T and IPsec is to enable opportunistic encryption between systems. NAT-T allows systems behind NATs to request and establish secure connections on demand.

IETF references

- RFC 1579 - Firewall Friendly FTP
- RFC 2663 - IP Network Address Translator (NAT) Terminology and Considerations
- RFC 2709 - Security Model with Tunnel-mode IPsec for NAT Domains
- RFC 2993 - Architectural Implications of NAT
- RFC 3022 - Traditional IP Network Address Translator (Traditional NAT)
- RFC 3027 - Protocol Complications with the IP Network Address Translator (NAT)
- RFC 3235 - Network Address Translator (NAT)-Friendly Application Design Guidelines
- RFC 3715 - IPsec-Network Address Translation (NAT) Compatibility
- RFC 3947 - Negotiation of NAT-Traversal in the IKE
- RFC 5128 - State of Peer-to-Peer (P2P) Communication across Network Address Translators (NATs)

References

[1] "IPSec NAT-T is not recommended for Windows Server 2003 computers that are behind network address translators" (http://support.microsoft.com/kb/885348/en-us). Microsoft knowledge base #885348. .

External links

- NAT-Traversal Test (http://nattest.net.in.tum.de)
- How Skype & Co. get round firewalls (http://www.heise-online.co.uk/security/How-Skype-Co-get-round-firewalls--/features/82481)
- NAT Traversal in IPSec - an article by Rami Rosen (http://www.linuxfoundation.org/collaborate/workgroups/networking/nat-traversal-ipsec)

STUN

"stun" redirects here. For other topics including this term, see stun (disambiguation).

STUN is a standardized set of methods, including a network protocol, used in NAT traversal for applications of real-time voice, video, messaging, and other interactive IP communications. STUN is an acronym for **Session Traversal Utilities for NAT**, and is documented in RFC 5389.[1] RFC 5389 obsoletes the previous specification, entitled **Simple Traversal of User Datagram Protocol (UDP) through Network Address Translators (NATs)**, documented in RFC 3489.[2] The obsolete version of STUN, sometimes referred to as Classic STUN, was intended as a complete solution for NAT traversal, and featured an algorithm to allow endpoints to determine NAT behaviour. The current version of STUN is presented as a tool to be used by other protocols, such as ICE. STUN removes the NAT classification algorithm and defines an extensible packet format.

The STUN protocol allows applications operating behind a network address translator (NAT) to discover the presence of the network address translator and to obtain the mapped (public) IP address (NAT address) and port number that the NAT has allocated for the application's User Datagram Protocol (UDP) connections to remote hosts. The protocol requires assistance from a third-party network server (STUN server) located on the opposing (public) side of the NAT, usually the public Internet. The original version of the protocol also specified methods to ascertain the specific type of NAT, but those methods have been deprecated in the newer specification, because of the plethora of specific NAT implementation behavior in various networking equipment and the resulting intractability of the problem and the deficiencies of the method used.

NAT traversal solutions

Network address translation is implemented via a number of different address and port mapping schemes, none of which are standardized.

STUN is not a self-contained NAT traversal solution applicable in all NAT deployment scenarios and does not work correctly with all of them. It is a tool among other methods and it is a tool for other protocols in dealing with NAT traversal, most notably Traversal Using Relay NAT (TURN) and Interactive Connectivity Establishment (ICE).[1]

STUN does work with primarily three types: full cone NAT, restricted cone NAT, and port restricted cone NAT. In the cases of restricted cone or port restricted cone NATs, the client must send out a packet to the endpoint before the NAT will allow packets from the endpoint through to the client. STUN does not work with symmetric NAT (also known as bi-directional NAT) which is often found in the networks of large companies. Since the IP address of the STUN server is different from that of the endpoint, in the symmetric NAT case, the NAT mapping will be different for the STUN server than for an endpoint. TURN offers better results with symmetric NAT.

Protocol overview

STUN is a lightweight client-server protocol requiring only simple query and response. The client side is implemented in the user's communications application, such as a voice over Internet Protocol (VoIP) phone or instant messaging client.

The base protocol operates essentially as follows. The client, often operating inside a private network, sends a *binding request* to a STUN server on the public Internet. The STUN server sends a *success response* that contains the IP address and port as observed from its perspective. The result is usually XOR mapped to avoid translation of packet contents.

STUN usually operates on a User Datagram Protocol (UDP) messaging transport. Since UDP does not provide reliable transport guarantees, reliability is achieved by application-controlled retransmissions of the STUN requests. STUN servers do not implement any reliability mechanism for their responses.[1] When reliability is mandatory, the

Transmission Control Protocol (TCP) may be used, but induces extra networking overhead. In security-sensitive applications, STUN may be transported and encrypted by Transport Layer Security (TLS).

An application may automatically determine a suitable STUN server for communications with a particular peer by querying the Domain Name System (DNS) for the `stun` (for UDP) or `stuns` (for TCP/TLS) server record (SRV) resource record, e.g., _stun._udp.example.com. The standard listening port number for a STUN server is 3478 for UDP and TCP, and 5349 for TLS. Alternatively, TLS may also be run on the TCP port if the server implementation can de-multiplex TLS and STUN packets. In case no STUN server is found using DNS lookups, the standard recommends that the destination domain name should be queried for address records (A or AAAA) which would be used with the default port numbers.

In addition to using protocol encryption via TLS, STUN also has built-in authentication and message-integrity mechanisms via specialized STUN packet types.

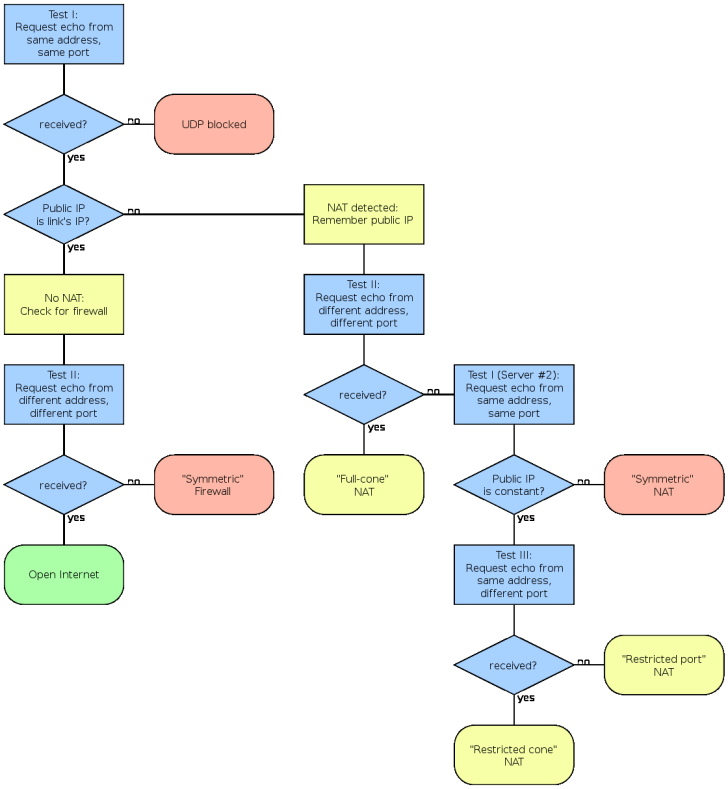
When a client has discovered its external address, it can use this as a candidate for communicating with peers by sharing the external NAT address rather than the private address (which is, by definition, not reachable from peers on the public network).

If both peers are located in different private networks behind a NAT, the peers must coordinate to determine the best communication path between them. Some NAT behavior may restrict peer connectivity even when the public binding is known. The Interactive Connectivity Establishment (ICE) protocol provides a structured mechanism to determine the optimal communication path between two peers. Session Initiation Protocol (SIP) extensions are defined to enable the use of ICE when setting up a call between two hosts.

Classic STUN NAT characterization algorithm

Classic STUN specified an algorithm to characterize NAT behavior according to the address and port mapping behavior. This algorithm is not reliably successful and only applicable to a subset of NAT devices deployed.

The algorithm consists of a series of tests to be performed by an application. When the path through the diagram ends in a red box, UDP communication is not possible and when the path ends in a yellow or green box, communication is possible.



References

[1] RFC 5389, *Session Traversal Utilities for NAT (STUN)*, J. Rosenberg, R. Mahy, P. Matthews, D. Wing, The Internet Society (October 2008)

[2] RFC 3489, *STUN - Simple Traversal of User Datagram Protocol (UDP) Through Network Address Translators (NATs)*, J. Rosenberg, J. Weinberger, C. Huitema, R. Mahy, The Internet Society (March 2003)

External links

- STUNTMAN - Open source STUN server code for RFC 5389 and RFC 3489 (http://www.stunprotocol.org)
- STUNT (http://nutss.gforge.cis.cornell.edu/stunt.php) - "STUN and TCP too", which extends STUN to include TCP functionality
- Yahoo! - Director of Engineering explaining STUN and TURN (Video) (http://www.youtube.com/watch?v=9MWYw0fltr0&eurl=http://www.voip-news.com/feature/top-voip-videos-051707/)
- STUN Client and Server library (http://sourceforge.net/projects/stun/)
- JSTUN - A Java STUN implementation (http://jstun.javawi.de/)
- ICE4J- A Java ICE, STUN and TURN library (http://code.google.com/p/ice4j/)

Traversal Using Relays around NAT

Traversal Using Relays around NAT (**TURN**) is a protocol that allows for an element behind a network address translator (NAT) or firewall to receive incoming data over TCP or UDP connections. It is most useful for elements behind symmetric NATs or firewalls that wish to be on the receiving end of a connection to a single peer. TURN does not allow for users to run servers on well known ports if they are behind a NAT; it supports the connection of a user behind a NAT to only a single peer. In that regard, its role is to provide the same security functions provided by symmetric NATs and firewalls, but to *turn* the tables so that the element on the inside can be on the receiving end, rather than the sending end, of a connection that is requested by the client.

TURN is specified by RFC 5766. An update to TURN for IPv6 is specified in RFC 6156.

Introduction

NATs, while providing many benefits, also come with many drawbacks. The most troublesome of those drawbacks is the fact that they break many existing IP applications, and make it difficult to deploy new ones. Guidelines have been developed that describe how to build "NAT friendly" protocols, but many protocols simply cannot be constructed according to those guidelines. Examples of such protocols include multimedia applications and file sharing.

Session Traversal Utilities for NAT (STUN) provides one means for an application to traverse a NAT. STUN allows a client to obtain a transport address (an IP address and port) which may be useful for receiving packets from a peer. However, addresses obtained by STUN may not be usable by all peers. Those addresses work depending on the topological conditions of the network. Therefore, STUN by itself cannot provide a complete solution for NAT traversal.

A complete solution requires a means by which a client can obtain a transport address from which it can receive media from any peer which can send packets to the public Internet. This can only be accomplished by relaying data through a server that resides on the public Internet. This specification describes Traversal Using Relay NAT (TURN), a protocol that allows a client to obtain IP addresses and ports from such a relay.

Although TURN will almost always provide connectivity to a client, it comes at high cost to the provider of the TURN server. It is therefore desirable to use TURN as a last resort only, preferring other mechanisms (such as STUN or direct connectivity) when possible. To accomplish that, the Interactive Connectivity Establishment (ICE) methodology can be used to discover the optimal means of connectivity.

External links

- Traversal Using Relays around NAT (TURN): RFC5766 [1]
- Traversal Using Relays around NAT (TURN) Extension for IPv6: RFC5766 [2]
- Yahoo! - Director of Engineering explaining STUN and TURN (Video) [3]

Implementations

- Restund [4] OpenSource Modular STUN/TURN Server (BSD License)
- Numb [5] is a free STUN/TURN server.
- TurnServer [6] - OpenSource TURN server.
- reTurn [7] - opensource STUN/TURN server and client library (C++)
- AnyFirewall [8] - STUN, TURN & ICE library.

References

[1] http://tools.ietf.org/html/rfc5766
[2] http://tools.ietf.org/html/rfc6156
[3] http://www.youtube.com/watch?v=9MWYw0fltr0&
eurl=http%3A%2F%2Fwww%2Evoip%2Dnews%2Ecom%2Ffeature%2Ftop%2Dvoip%2Dvideos%2D051707%2F
[4] http://www.creytiv.com/restund.html
[5] http://numb.viagenie.ca/
[6] http://www.turnserver.org/
[7] http://www.resiprocate.org/reTurn_Overview
[8] http://developer.anyfirewall.com/

Interactive Connectivity Establishment

Interactive Connectivity Establishment (ICE) is a technique used in computer networking involving network address translators (NATs) in Internet applications of Voice over Internet Protocol (VoIP), peer-to-peer communications, video, instant messaging and other interactive media. In such applications, NAT traversal is an important component to facilitate communications involving hosts on private network installations, often located behind firewalls.

ICE is developed by the Internet Engineering Task Force MMUSIC working group and is published as RFC 5245[1], which has obsoleted RFC 4091[2].

Overview

Since the number of IPv4 addresses are limited to their 32-bit representation, not every network enabled device can have a unique public IP with which to be visible on the Internet. Network Address Translators (NAT) work by changing a private address into a public one when an outbound request passes through them. When a client establishes TCP connections through SYN packets, the NAT updates an internal table with each entry creating a mapping between an internal, private IP to a public one[3]. Many applications run into problems when put in this situation with one example being VoIP traffic where a client needs to register with a unique address to a SIP proxy. Another problem relates to firewalls which might block VoIP traffic completely. ICE provides a framework for dealing with these problems.

Session Traversal Utilities for NAT (STUN) is a client server protocol returning the public IP to a client together with information from which the client can infer the type of NAT it sits behind while Traversal Using Relays around NAT (TURN) places a third party server to relay messages between two clients where peer to peer media traffic is not allowed by a firewall.

IETF Specifications

- RFC 5389: Session Traversal Utilities for NAT (STUN).
- RFC 5766: Traversal Using Relays around NAT (TURN): Relay Extensions to STUN.
- RFC 5245: Interactive Connectivity Establishment (ICE): A Protocol for NAT Traversal for Offer/Answer Protocols.

References

[1] RFC 5245, *Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal for Offer/Answer Protocols*, J. Rosenberg (April 2010)

[2] RFC 4091, *The Alternative Network Address Types (ANAT) Semantics for the Session Description Protocol (SDP) Grouping Framework*, G. Camarillo, J. Rosenberg (June 2005)

[3] Müller A, Carl (2008) Behavior and Classification of NAT Devices and Implications for NAT Traversal IEEE Network September/October 2008. Available from: http://ieeexplore.ieee.org.ezproxy.liv.ac.uk/stamp/stamp.jsp?tp=&arnumber=4626227 [Accessed at: 2 April 2011]

External links

- IETF Journal article on ICE (https://www.internetsociety.org/articles/interactive-connectivity-establishment) - read first
- ICE Tutorial (http://www.jdrosen.net/papers/ice-basic-tutorial.pdf)
- MMUSIC working group (https://datatracker.ietf.org/wg/mmusic/charter/)
- BEHAVE working group (https://datatracker.ietf.org/wg/behave/charter/)
- PJNATH - Open Source ICE, STUN, and TURN Library (http://www.pjsip.org/pjnath/docs/html/index.htm)
- libnice: GLib ICE library (http://nice.freedesktop.org/wiki/)

Media Gateway Control Protocol

Media Gateway Control Protocol also known as **MGCP** is one of the implementation of the **Media Gateway Control Protocol Architecture**[1] for controlling media gateways on Internet Protocol (IP) networks and the public switched telephone network (PSTN). The general base architecture and programming interface is described in RFC 2805 and the current specific MGCP definition is RFC 3435 (obsoleted RFC 2705). It is a successor to the Simple Gateway Control Protocol (SGCP) which was developed by Bellcore and Cisco. In November 1998, Simple Gateway Control Protocol (SGCP) was combined Level 3 Communications Internet Protocol Device Control (IPDC), to form Media Gateway Control Protocol MGCP.[2]

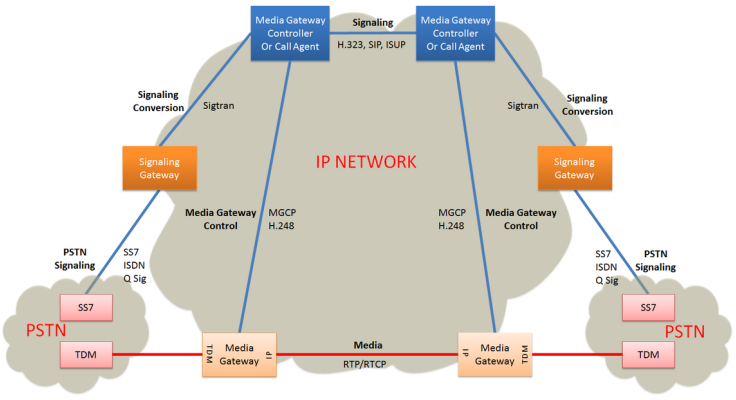
MGCP is a signalling and call control protocol used within Voice over IP (VoIP) systems that typically inter-operate with the public switched telephone network (PSTN). As such it implements a PSTN-over-IP model with the power of the network residing in a call control center (softswitch, similar to the central office of the PSTN) and the endpoints being "low-intelligence" devices, mostly simply executing control commands. The protocol represents a decomposition of other VoIP models, such as H.323, in which the media gateways (e.g., H.323's gatekeeper) have higher levels of signalling intelligence.

MGCP uses the Session Description Protocol (SDP) for specifying and negotiating the media streams to be transmitted in a call session and the Real-time Transport Protocol (RTP) for framing of the media streams.

Another implementation of the Media Gateway Control Protocol Architecture exists is H.248/Megaco protocol, a collaboration of the Internet Engineering Task Force (RFC 3525) and International Telecommunication Union (Recommendation H.248.1). Both protocols follow the guidelines of the API *Media Gateway Control Protocol Architecture and Requirements* in RFC 2805. However, the protocols are incompatible due to differences in protocol syntax and underlying connection model.

Architecture

The Media Gateway Control Protocol Architecture and its methodologies and programming interfaces are described in RFC 2805.[3]



Gateway Control Protocol Relationship

MGCP is a master/slave protocol that allows a call control device such as Call Agent to take control of a specific port on a Media Gateway. In MGCP context Media Gateway Controller is refereed to as Call Agent. This has the advantage of centralized gateway administration and provides for largely scalable IP Telephony solutions. The distributed system is composed of a Call Agent], at least one Media Gateway (MG) that performs the conversion of media signals between circuits and packets switched networks, and at least one Signaling gateway (SG) when connected to the PSTN (conversion from TDM voice to Voice over IP).

MGCP assumes a call control architecture where there is limited intelligence at the edge (endpoints, Media Gateways) and intelligence at the core Call Agent. The MGCP assumes that Call Agents, will synchronize with each other to send coherent commands and responses to the gateways under their control.

The Call Agent uses MGCP to tell the Media Gateway:

- what events should be reported to the Call Agent
- how endpoints should be connected together
- what signals should be played on endpoints.

MGCP also allows the Call Agent to audit the current state of endpoints on a Media Gateway.

The Media Gateway uses MGCP to report events (such as off-hook, or dialed digits) to the Call Agent.

(While any Signaling Gateway is usually on the same physical switch as a Media Gateway, this needn't be so. The Call Agent does not use MGCP to control the Signaling Gateway; rather, SIGTRAN protocols are used to backhaul signaling between the Signaling Gateway and Call Agent).

Multiple call agents

Typically, a Media Gateway is configured with a list of Call Agents from which it may accept programming (where that list normally comprises only one or two Call Agents).

In principle, event notifications may be sent to different Call Agents for each endpoint on the gateway (as programmed by the Call Agents, by setting the NotifiedEntity parameter). In practice, however, it is usually desirable that at any given moment all endpoints on a gateway should be controlled by the same Call Agent; other Call Agents are available only to provide redundancy in the event that the primary Call Agent fails, or loses contact with the Media Gateway. In the event of such a failure it is the backup Call Agent's responsibility to reprogram the MG so that the gateway comes under the control of the backup Call Agent. Care is needed in such cases; two Call Agents may know that they have lost contact with one another, but this does not guarantee that they are not both attempting to control the same gateway. The ability to audit the gateway to determine which Call Agent is currently controlling can be used to resolve such conflicts.

MGCP assumes that the multiple Call Agents will maintain knowledge of device state among themselves (presumably with an unspecified protocol) or rebuild it if necessary (in the face of catastrophic failure). Its failover

features take into account both planned and unplanned outages.

Protocol overview

MGCP packets are unlike those generated by many other protocols. Usually wrapped in UDP port 2427, the MGCP datagrams are formatted with whitespace, much like you would expect to find in TCP protocols.

An MGCP packet is either a command or a response. Every issued MGCP command has a transaction ID and receives a response. Commands begin with a four-letter verb. Responses begin with a three number response code.

There are nine (9) command verbs:

```
AUEP, AUCX, CRCX, DLCX, EPCF, MDCX, NTFY, RQNT, RSIP
```

Two verbs are used by a Call Agent to query (the state of) a Media Gateway:

```
AUEP – Audit Endpoint
AUCX – Audit Connection
```

Three verbs are used by a Call Agent to manage an RTP connection on a Media Gateway (a Media Gateway can also send a DLCX when it needs to delete a connection for its self-management):

```
CRCX – Create Connection
DLCX – Delete Connection
MDCX – Modify Connection
```

One verb is used by a Call Agent to request notification of events on the Media Gateway, and to request a Media Gateway to apply signals:

```
RQNT – Request for Notification
```

One verb is used by a Call Agent to modify coding characteristics expected by the "line-side" on the Media Gateway:

```
EPCF – Endpoint Configuration
```

One verb is used by a Media Gateway to indicate to the Call Agent that it has detected an event for which the Call Agent had previously requested notification of (via the RQNT command verb):

```
NTFY – Notify
```

One verb is used by a Media Gateway to indicate to the Call Agent that it is in the process of restarting:

```
RSIP – Restart In Progress
```

Implementations

Two implementations of the Media Gateway Control Protocol are in common use. The names of both are abbreviations of the protocol group:

- MGCP is described in RFC 3435.[4]
- H.248/Megaco is described in RFC 3525.[5] [Obsoleted by: RFC 5125.[6]]

Although similar in architecture, MGCP and H.248/Megaco are distinctly different protocols and are not interoperable. H.248/Megaco and MGCP protocols are complementary to H.323 and Session Initiation Protocol. Both H.323 and SIP can be referred to as "intelligent endpoint protocols". H.248/Megaco and MGCP can be referred to as "device control protocols".[7] [8]

RFCs

- RFC 3435 - Media Gateway Control Protocol (MGCP) Version 1.0 (this supersedes RFC 2705)
- RFC 3660 - Basic Media Gateway Control Protocol (MGCP) Packages (informational)
- RFC 3661 - Media Gateway Control Protocol (MGCP) Return Code Usage
- RFC 3064 - MGCP CAS Packages
- RFC 3149 - MGCP Business Phone Packages
- RFC 3991 - Media Gateway Control Protocol (MGCP) Redirect and Reset Package
- RFC 3992 - Media Gateway Control Protocol (MGCP) Lockstep State Reporting Mechanism (informational)
- RFC 2805 - Media Gateway Control Protocol Architecture and Requirements
- RFC 2897 - Proposal for an MGCP Advanced Audio Package

References

[1] RFC 2805, Media Gateway Control Protocol Architecture and Requirements, N. Greene, M. Ramalho, B. Rosen, IETF, April 2000

[2] "Level 3 Communications, Bellcore Announce Merger of Protocol Specifications for Voice Over IPe" (http://level3.mediaroom.com/index.php?s=23600&item=65733). Level 3 Communications. . Retrieved 08 June 2012.

[3] RFC 2805, *Media Gateway Control Protocol Architecture and Requirements*, N. Greene, M. Ramalho, B. Rosen, The Internet Society (April 2000)

[4] RFC 3435, *Media Gateway Control Protocol (MGCP) Version 1.0*, F. Andreasen, B. Foster, The Internet Society (January 2003)

[5] RFC 3525, *Gateway Control Protocol Version 1*, C. Groves, M. Pantaleo, T. Anderson, T. Taylor (editors), The Internet Society (June 2003)

[6] RFC 5125, *Reclassification of RFC 3525 to Historic*, T. Taylor, The IETF Trust (February 2008)

[7] |title=Use of MEGACO vis-à-vis MGCP to build a Gateway Solution (http://hive1.hive.packetizer.com/users/packetizer/papers/ipmc/MEGACOvsMGCP_v3.pdf)

[8] "SIP core working group charter h2.48 history" (http://www.packetizer.com/ipmc/h248/history.html). packetizer.comg. . Retrieved 2012-06-07.

External links

- MGCP Information Site (http://www.packetizer.com/voip/mgcp) Information related to MGCP
- H.248 Information Site (http://www.packetizer.com/voip/h248/) Information related to H.248/Megaco, including pointers to standards and draft specifications

Article Sources and Contributors

Voice over IP *Source*: http://en.wikipedia.org/w/index.php?oldid=513492100 *Contributors*: .::Arbitrary::., 123microsoft123, 159753, 16@r, 23skidoo, 28421u2232nfenfcenc, 6birc, A. B., A.Ward, A13ean, A876, AJR, AVMsoftware, Abesford, Academic Challenger, Acalamari, Acce245, Accountwiki2, Acidburn24m, Adam78, Adclark88, Addihockey10, Adjektiv, Adoresoftphone, Aelman, Agnvoip, Ahmadalweshah, Ahoerstemeier, Ajrichens, Akram1997, Al E., Alan Liefting, Alansohn, Ale jrb, Alecv, Alejo2083, Alex de carvalho, Alexander Straub, AlistairMcMillan, Alokagga, Alphachimp, Altesys, Alw75, Alxeedo, Amalia07, Amandavpi, Amescreative, Ananth t, Andre nsi, AndrewKay, Andromeda451, Andy Dingley, Andypandy.UK, Angr, Anir1uph, Anon2, Antonio Lopez, Apapadop, Apyule, Archelon, Aresnika, Argon233, ArkansasHistory, Ashton1983, Ashumit02, Asimzaidi123, Atlant, Atlasvoice, Ausinha, Avi.dorfman, Avé, Azazello, BRUTE, Babajobu, Banej, Banes, Bansaribarot, Beboy10, Beetstra, Beland, Ben.c.roberts, BenFrantzDale, Bender235, Beno1000, Bevo, Bfax, Bfurst, Biggkid5, Bigtex 1, Billhunt, Billkor, Binksternet, Biot, Bluerasberry, Bluezy, Bobblewik, Bobo192, Boffin, Bogdanwiki, Bonius, Boogachamp, Bowensrun, Bowmanjj, Box News, Bozsingdani, Bphenix, Bprincipi, Bradleyyard, Braeside, Brandon, Brest, Brocha, BrokenSegue, Brycen, By97aa, C172rg, CTZMSC3, Cabiria, CalamityGNDM, Caliqt84, Calltech, Caltas, CambridgeBayWeather, Camie333, Camptown, Can't sleep, clown will eat me, CanisRufus, Canthusus, CapitalR, Capricorn42, CardinalDan, CaribDigita, Cbarbry, Ccowling, CesarB, Chaldor, Charles Matthews, Chealer, ChrisJMoor, Chrislk02, Christopher Mahan, Churb75, Cielomobile, Cindyt27, Civil Engineer III, Ckatz, Clairecare, Cliconnect, Clq, Cluth, Coffee, Colin Keigher, Colonies Chris, Combes, CommEvo, Computerjoe, Connelly, Coolmac860, Coreyxbs, Cre8d, Crosbiesmith, Crutter, CsamFord, Cst17, Cweath7, Cyberdyme, Cyrilfr, Cyrius, D.c.camero, DDeckert, DaMenace123, Daev, Daikiri, Dan100, DanielCD, DanielVonEhren, Danr2k6, Dapps007, Daquell, Dask137, DataHead2112, DaveSymonds, Daven200520, Daveslomo, Davfox, David Johnson, David Sorf, David.a.gelman, DavidStainberg, Dawnseeker2000, Db11710, Dbollard99, DeadEyeArrow, Deeahbz, Deeprig9, Deineka, Dennis70, DennisDaniels, Depill.is, DerHexer, Desecrator, Deserthawks, Dgtsyb, Diberri, Dicklyon, Diego.viola, Diverman, Dlsieradzki, Dlyons493, Dmar198, Dobregon, Dobrien, DocWatson42, Docboat, Donama, Doretel, DoretelCom, Double DKool, Dpocock, DrDorkus, Dreadstar, Dreamafter, Drinking.coffee, Dspanalyst, Duckdad, Ed Poor, Edgeley, Edtealdi, Efrondozo, Egstcm, Ejwma, ElBenevolente, Elbperle, Electriccatfish2, EncMstr, Enochlau, Enric Naval, Eorlov, Epastore, Epbr123, Epl18, Erianna, Ethan01, Europrobe, Evanwolf, Everyking, Evice, Expertu, F, F Notebook, FT2, Falcorian, Faradayplank, Fastfission, Fayedizard, FayssalF, Fennelcm, Firetrap9254, Flambib, Flod logic, Fluffernutter, Foodeei, Foofy, Forkqueue, Fraggle81, Fram, Francis Good, Francs2000, Fred Bradstadt, Fred Gandt, Free Bear, Freeness, Fresheneesz, Freshtel, Freyr, Frizzle, Funkyguy, Fvp, Fyrael, GRD, GULFSIP.Engineer, GULFSIP.voip.engineer, Gabrielyan, Gadfium, Gail, Galoubet, Garvanet, Gbleem, Ged Davies, Ged UK, Geekking, Geep23, Gfoley4, Ghadsall, Giftlite, Gilliam, Gimpmask, Glendonflowers, Glenitec610, GlobalEdge 2010, Globaltransmitter, Gobbagoo, Goblin, Gogo Dodo, GoingBatty, Gopalkrishnan83, Goplett, GraemeL, Graham87, GregA, GrooveDog, Guaka, Gubbay, Gurchzilla, Gwernol, Gwicke, HUnger, HWSager, Haakon, Hack-Man, Hadal, Hairy Dude, HappyInGeneral, Harryzilber, Headbomb, Henrygb, Hetar, Hgarciagg, Hipvoice, Hnram96, HockeyInJune, Holme053, Hooiemajoris, Hooman.marandi, Hooperbloob, Hope work, Hopper96, Hrbrendan, Hugh168, Hughzhu, Husond, Hydrargyrum, IMac4ME, Ibanix, Ignorance is strength, Ikanreed, Imcdnzl, InShaneee, Incnis Mrsi, Indangermouse, Infrogmation, Inotbf, Insanephantom, Intelafone, Interchangegroup, Intercompages, Interiot, Ipatrol, IronGargoyle, IronHelix, Iskitoofast4u, It Is Me Here, Itsme, Itusg15q4user, Ixfd64, J Milburn, J Santoso, J.delanoy, JHunterJ, JSsoftware, Jadams76, Jafeluv, Jahoe, Jalal0, James086, JanCeuleers, Jana Grare, Janahan, Jasenlee, Jasonon, Jasper Deng, Javacava, Javamen, Jbrock11, Jcline0, Jcyrus80, Jebba, Jeffholton, Jeffman78, Jeffq, Jehochman, JeremyR, Jerome Charles Potts, JetLover, Jezaraknid, Jhessela, Jhknight, Jidanni, Jignesh.4soni, Jim.henderson, Jim1138, Jimp, Jimthing, Jjspirko, Jmabel, Jmediate88, Jnavas, Joe1945-01, JoeSmack, John Doe or Jane Doe, John Vandenberg, Johnflan, Johnleemk, Johnnie ong, JohnnyB, JonHarder, JonTeh, Jone5050, Jongrant, Jonolumb, Jonuday, Joseph Solis in Australia, Joy, Jptdrake, Jreconomy, Jroddi, Jrtayloriv, KF, KVDP, Kaare, Kamaniskeg, Karada, Karl-Henner, Karl2620, Karn, Katalaveno, Katiker, Kazvorpal, Kbrose, KelleyCook, Ken Olsen, Kevincw01, Kevinmon, Key134, Kgfleischmann, Kglavin, Kgrr, Kikos, King adan, King of Hearts, Kingkobus, Kingpin13, Kiscica, Kku, Klapouchy, Koavf, Kozuch, KrakatoaKatie, Krallja, Kristof vt, Kryn1030, Kushal one, Kusma, Kvng, Kw27, Kylu, Kyng, LOL, LP-mn, Landroni, Lankiveil, Lansal, Larry2342, Lazydaisy, LeaveSleaves, Leeyc0, Lerdsuwa, Lethe, Leuk he, Linkspamremover, Linuxerist, Liquid Chrome1, LittleOldMe, Lolwarrior, Lootzy, LorenzoB, Lotje, Louietyj, Lozza, Lrlfrance, LtNOWIS, Luke Bales, Luna Santin, Lupo, MAlvis, MBK004, MER-C, MLD7865 Auto, Mac100, MacGyver07, Macdmq, Mad Carew, Maggu, Makefreecalls, MamboJambo, Manickaselvam, MansonP, Manticore, Manuel Anastácio, Manuelamsp, Marc Lacoste, Marco79, MarcoAurelio, Maris Jansons, Mark v1.0, MartinHarper, MartrtinS, MartynDavies, MasterfulTroll, Matanya (renamed), Matthew Yeager, Matthewbulat, MattieTK, MattisManzel, Mattsday, Mauls, MaxDowney, Mcfutech, Mclowes, Mcsboulder, Meaghan, Mehrdadd, Mendel, Merkel, Michael Hardy, Miguelaustro, Mike Gale, Mikeblas, Miken2005, Mikez711, Miladchik, Miltiades490bc, Mindmatrix, Mingthemad, Misterbigbrain, Mkzz, Mmccalpin, Mobidick0, Mobutu LM2006, Mohammed 77b, MonoAV, Monsieur champs, Mormegil, Morruga, Morte, Mote, Motoroflife, Mr.chetanladdha, Muhandes, Mukat, Mulligatawny, Murdamw, Muru, MusicTree3, Mws1979, Mxn, Myleslong, Mylivetech, Mysdaao, Mysekurity, Mysidia, N Yo FACE, NTK, Nageh, Nandini, Natl1, Naudefj, Navstar, NawlinWiki, Neilc, NerdyNSK, Nethgirb, Neustradamus, New2way, Newton2, Niazza, NickW557, Nikzbitz, Nishkid64, Nitinkrgupta, Nloth, Nneonneo, Noel Streatfield, Nono64, Nonodename, Notwist, Novum, Nubiatech, Nww mag, Oblivious, Ochib, Ohnoitsjamie, Oli Filth, Oliver.Shepherd, Oliver202, Olivier, Omegatron, OverlordQ, Ozeki, Paidup, ParticleMan, Paul August, Pchov, PdDemeter, Peruvianllama, Pessi, Peter Hitchmough, Petrb, Pfwebadmin, Pgilman, Phatom87, Philip Trueman, Phoenix-forgotten, Phyzome, Piano non troppo, Picaroon, Pinnocchio, Pjbrockmann, Pjrich, Pnm, Poeloq, Pogostokje, Pol098, Polscifreak, Ponyo, Poorleno, Postdlf, Poweroid, Pparazorback, ProblemAssassin, Prunk, Pseudomonas, Pshibles, Pugglewuggle, Pugliavi, Quiensabe, Quintin3265, Qwyrxian, R'n'B, R3m0t, RadioKirk, Radioraiders, Rafehh, Rainman74, Rajarshic02, Ramshi1, Randy Johnston, Rapido, Rasmus Faber, RasputinAXP, Rchamberlain, Rearden9, Reedy, Reffidini, Remember the dot, Remobasith, Renffeh, Respawn2003, RexNL, Rhobite, Rich Farmbrough, Richardjohnstone, Rick7425, Rillian, Ringbang, Rishishringan, Ritterrat, Rjwilmsi, Robertbricker, Robertvan1, RoboAction, RoccoKIT, RockMFR, Ronz, Rose Booth, Rrburke, Rrw, Rubgwobouhgrwoosf, Rudykog, Ruhrfisch, Rupertb, S.K., S3000, SJP, SMC, SUparJErk, Saaga, Sadads, Samw, Sandeep kumar pal, Sangyup81, SasiSasi, Schmidtja, SchuminWeb, Scott0485, Scottperry, Seajay1979, Seajay19791, Sean Whitton, Secretflower, Seisatsu, Senator2029, Shadowjams, Shandris, Shara77, Sharonpfingo, Sharonxiv, Shebazahmed, Shii, Shinjodenn, Shizane, Signalhead, Silverfox196, Simeondahl, SimonInns, SimonP, Simsim4, Skarmachild, Skinkie, Slakr, Slon02, Smalljim, Smasafy, Smitty, Smostowfi, Smyth, Soptep, Sosodank, Spacestationshuttle, Sparklyfeet, Spearhawk, Spencer.fry@yale.edu, Spiko-carpediem, Splatg, Splendour, SqueakBox, Squideshi, Squids and Chips, Squiquifox, Squire of the Infernal Knight-Lord of Penguins, Srleffler, StaticGull, Stefano85, Stepa, Stephan Leeds, Stephen Lange Ranzini, Steven312, Stevertigo, Stevevogelnu, Stilldavid, Strait, Strib, Stuart Ward UK, Stupid2, Stwalkerster, Sudharana, Sumantu.mittal, Sun Creator, Superm401, Surge79uwf, Susyr, Sweetpoet, Sylverboss, Symbianguru, Synchrite, Sywyk, T2X, TJJFV, Tabletop, Tarquin, Tarret, Tassedethe, TastyPoutine, Tchin-NJITWILL, Tckma, Tcncv, Techman224, Tecpc2001, Teemuk, Telecom.portal, Tenbaset, Terinavy, Tegairn, Tglovoip, Thakkar v, Thane Eichenauer, The Anome, The Belgain, The Rambling Man, The Storm Surfer, The Thin Man Who Never Leaves, The bellman, TheKMan, Thedp, Theh1982, Therebelcountry, Thingg, Thinkingphones, Thiseye, Thomas Blomberg, Thompsontough, Tim Ivorson, Timrem, Tizio, Tkoeppe, Tliberty10, Tmaufer, Tmnmt, ToP Racer, Tom k&e, Tombomp, Tomjenkins52, Tomwalden, Toniher, Tony1, Towel401, Toytoy, Tran Quoc123, Trevor MacInnis, Tslocum, Twizler3, Tylko, Ulf Abrahamsson, Ulric1313, UltraMagnus, Unixer, Unused0030, Utcursch, VCA, VSteiger, Val42, Vanished user 39948282, Vaquerito, VegaDark, Vegaswikian, Veinor, Veledan, Veromark, Versageek, Vespristiano, Vickygarg, Villarinho, Vina, Vinnythejinny, Vinogradovisoleksii, ViperSnake151, Vipinhari, Vmart, VoIPExpert, Voice99, VoiceNet, Voicepulse, Voidvector, Voip-solution, Voip.world, Voipfone, Voipphoneservice, Voiptopia, Voixium, Volker, Voskypro, Votingontheinternet, Voxbone, Vrenator, Vy0123, W Nowicki, WR:ichibantel, Wafulz, Waggers, Wannabav8r, Wavelength, Welsh, Weregerbil, Werelnon, West.andrew.g, WhaleyTim, Whicky1978, Whiskers9, WhiteBoy, Who, Who123, Wholived, WibblyLeMoende, WikHead, Wiki alf, WikiBlogger61, Wikidmoz, Wikieditor06, Wikivoip, Wile E. Heresiarch, WilliamKF, Wimt, Windharp, Wlpekar, Wmahan, Wongm, Woohookitty, Wtmitchell, Wtshymanski, Wwwqq, Xdenizen, Yahoolian, Yamamoto Ichiro, Yaronf, Yfrwlf, YongKim, Yoshboy, Yueni, Yurik, Zachripples, Zervaman, ZimZalaBim, Zimbabweed, Zondor, Zwz8406, Zzuuzz, Zzyzx11, ZzzBrett, आशीष भटनागर, පසිඳු කාවින්ද, 2226 anonymous edits

Global Dialing Scheme *Source*: http://en.wikipedia.org/w/index.php?oldid=513497201 *Contributors*: Bwpach, D6, DGG, ErrantX, Gowersteve, LunchBox45, Mogism, Pnm, Troyer59, Vassyana, 4 anonymous edits

Session Description Protocol *Source*: http://en.wikipedia.org/w/index.php?oldid=513495289 *Contributors*: Aldie, AvicAWB, Birczanin, Conversion script, Dokaspar, Doug Bell, Fsiyavud, Giwa-jp, Hamilkar, Hefo, Hu12, Inkling, JTN, JesperANielsen, Kate, Kbrose, Kgfleischmann, Kinema, Kvng, Mange01, Mark McLoughlin, Maximaximax, Melamed katz, Mgream, Otherox, Raanoo, RedWolf, Rich Farmbrough, RobLa, Robertabela, Ryans.ryu, Shenshu, SixSix, Stuuf, SystemBuilder, Wrs1864, 44 anonymous edits

Session Initiation Protocol *Source*: http://en.wikipedia.org/w/index.php?oldid=508273645 *Contributors*: 2001:67C:1A00:4:0:0:0:1972, A.M., AIMSzpc, Adem reid, Agnvoip, Alan Smithee, Alansohn, Alf Boggis, Algocu, AlistairMcMillan, Altesys, Amirshaheen, Amniarix, Andyparkins, Antzervos, Armando, Ashleyarmitt, AspieMind, Ausinha, Avi.dorfman, Azurepalm, B timmins, B1ll1ngsl34, Batsonjay, Bazza37, Bchatelet, Bender235, Biot, Blatkinson, Bluezy, Bobet, Bpadinha, Brandon, Brazil4Linux, Breno, Brest, Bubuka, Calltech, Can't sleep, clown will eat me, Carlos-alberto-teixeira, Carre, Cburnett, Cfeet77, Chris Roy, Christopher Mahan, Codecmdr, Conversion script, Cub001, Curtis Newton, Cybercobra, Cyrius, DabMachine, Daniel.Cardenas, Dariuspomaha, DataSurfer, David Johnson, DavidAyers, DeadEyeArrow, Dewikipeder, Dgtsyb, Dhughes, Donsez, Doug Bell, DrMac, EagleEye96, Ed Poor, Edcolins, Edward Waverley, Edwilson97, Ehjelmeland, Ejabberd, Elgaard, Emvee, Enjoi4586, Ennustaja, Eproulx, Equendil, Eric boutilier, Erikgje, Eshouthe, Etu, Excirial, Faico, Fanf, Flambib, Franl, Frap, Fred Gandt, GDallimore, GentlemanGhost, Geomaster1, Gogo Dodo, Graham87, GreyCat, Gronky, Gshaham, Gtamas, Guaka, Guillaume.steinmetz, Hairy Dude, Harryzilber, Hopper96, Hu12, Hujaza, I don't remember my username, I'm not human, IMSoP, IRedRat, Ibc wiki, Int21h, Isnow, Itai, Jadahl, James nits, James.hamlin, Jay.Here, Jbonocore, Jcgriffiths, Jdi153, Jeffme, Jehochman, JesperANielsen, Jfayel, Jharrell, Jniemi, Joehamiltonjr, Joevf, Johnpseudo, Johnthesmith, JonTeh, Jpdemont, Judzillah, Julesd, Kasperd, Kat, Kbrose, KenCFTeam, Kgfleischmann, Kinema, Kjtobo, Kkkdddiii, Kmorozov, Koavf, Kvng, Kyng, Laurie G, Layer, Leandrod, Liftarn, Lincoln Josh, Linkminer, Lion789, Logixoul, Lradrama, Lupin, Luser, Lzur, MMuzammils, MSGJ, Magnus Manske, Malcolmredheron, Mange01, MartynDavies, Master of Puppets, MathsPoetry, Matt Darby, Matt tw, Matthi2, Maury Markowitz, Mboedick, McCarthur, Meand, Metamagician3000, Miguel.lima, Mindmatrix, Moemin05, Molestash, Monk127, Morte, Mpm777, MrJones, MrOllie, Mram80, Mranga, MrsJonesical, Mulad, Mulligatawny, Mvineetmenon, Neustradamus, Nexus501, Noahspurrier, Notheruser, Now3d, Nubiatech, Nyco, Odd bloke, OldMan, Olle.johansson, Oxymoron83, PM800, Pagingmrherman, ParadialEditor, Patcito, Pbranfield, PeterB, Pgk, Phatom87, Pinkunicorn, Pmsyyz, Pnetz, Pugliavi, R'n'B, Raanoo, Rathee, Rchamberlain, RedWolf, Requestion, RichiH, Rick Block, Riffic, Ringbang, Rjwilmsi, RobLa, Russellbryant, Rzelnik, Salmar, Satch69, Schweini, Scienceguy8m, Sdrtirs, SeanLegassick, Seav, Serych, Seyhan Aydin, Sgodin, ShaunMacPherson, Sietse Snel, Silverfox196, Sim, Sipexpert, Skowa, Smyth, SpaceFlight89, Strait, Strongsauce, SudoMonas, Surf08008, Suruena, Taed, Tanovic, Teemuk, Tekalpha, Thane Eichenauer, TheMandarin, Tietew, Tliberty10, Tom Morris, Torzsmokus, Touisiau, Towel401, Ttlkr, Ttwaring, Tuxa, Tzury, UncleBubba, Utcursch, Verdatum, Voidxor, Voip81, Voixium, WR:ichibantel, Wensong, Widefox, Will Beback Auto, Wk muriithi, Wongm, Wormhole80, Wrs1864, Yaronf, YordanGeorgiev, Yun-Yuuzhan (lost password), Zach Vega, Zhong.xie, Zoicon5, 544 anonymous edits

List of SIP response codes *Source*: http://en.wikipedia.org/w/index.php?oldid=506206006 *Contributors*: BW, Brandon, ConCompS, Cristianocosta, DiarmuidLeonard, Docpauluk, Franklutz, Kalamanda, Kenyon, Kgfleischmann, Kvng, Meand, Mikanas, Praveen.mundlapati, Rohini kamath, Serban.nistor, Slepp, Ulf Abrahamsson, Verdatum, Ville Lehtonen, VitriolUK, 48 anonymous edits

SIP Trunking *Source*: http://en.wikipedia.org/w/index.php?oldid=507163463 *Contributors*: Chris the speller, Kbrose, Kjtobo, McSly, Natttam, Pugliavi, Ricksantangelo, Srleffler, Tgeairn, Visik, 9 anonymous edits

Back-to-back user agent *Source*: http://en.wikipedia.org/w/index.php?oldid=379773947 *Contributors*: Bala neelakantan, Billrrichter, Bluezy, Brandon, Diana cionoiu, JonHarder, Kbrose, Kgfleischmann, PPBlais, Sobomax, TJJFV, 12 anonymous edits

H.323 *Source*: http://en.wikipedia.org/w/index.php?oldid=508266139 *Contributors*: Aegis Maelstrom, Aesopos, Alecv, Ali@gwc.org.uk, Andreas Kaufmann, Arteitle, Avi.dorfman, Bedel23, Beland, Blehfu, Bluezy, Bobet, Bontenbal, Brandon, Calltech, Cbarbry, Ceros, CosineKitty, Cwhuang, Cybercobra, DaveHowe, Deeahbz, Dgtsyb, Donnyw, DopefishJustin, EdgeOfEpsilon, ElTyrant, Ernestvoice, Fleminra, FoeNyx, Folajimi, GChriss, Giraffedata, Graham Rule, Guaka, Gurch, Haakon, Harryzilber, Hu12, Hvtuananh, ICU Global, ITU-T, Igoldste, Ilbc, JanCeuleers, Jdiegmueller, Johnny.cespedes, Jose.anda, Kbrose, Kcordina, KentTong, Kenyon, Kgfleischmann, KillerChihuahua, Konrads, Kvng, Lupo, MER-C, Marc Mongenet, MarkWahl, Meand, Michaelmogessie, Mikecron, Mikewax, Mindmatrix, Miraculix, Mulligatawny, Munford, NMChico24, Nasa-verve, Nayuki, Nimeshc, Nit-RAM, Nunners, Onewhohelps, PanagosTheOther, Paulej, Phatom87, Phoenix-forgotten, R'n'B, Razorflame, Rchandra, Rg3, Rjwilmsi, Rorymcd, Royk, Sergey999, Slashdevslashtty, SudoMonas, Tagishsimon, TheMandarin, Tipiac, Tonkie, WR:ichibantel, Widefox, YordanGeorgiev, Yurik, 164 anonymous edits

H.323 Gatekeeper *Source*: http://en.wikipedia.org/w/index.php?oldid=484413206 *Contributors*: Andreas Kaufmann, Bouktin, Cwhuang, EagleOne, Epolk, Folajimi, Harryzilber, James Flockton, Kcordina, MarSch, MarcelASAB, Mayank431, PDH, Splatonline, TheParanoidOne, 9 anonymous edits

Application-level gateway *Source*: http://en.wikipedia.org/w/index.php?oldid=505326415 *Contributors*: Dlrohrer2003, Hairy Dude, Kvng, Pedant17, Pnm, PrestonH, Qxz, Raanoo, Rchandra, Teles, Warren, Wrs1864, مونا بشيري, 26 anonymous edits

Real-time Transport Protocol *Source*: http://en.wikipedia.org/w/index.php?oldid=511853654 *Contributors*: Alerante, Allefant, Andy pryor, Antoncampos, Arunsjamwal, Avi.dorfman, Bobo192, Bodnotbod, Brandon, Brigman, Cedars, CesarB, Ceyockey, Chaotic Mind, Conversion script, Cory Donnelly, Csabka, Cschim, DKEdwards, Dac04, Daniel.Cardenas, Dcoetzee, Dgtsyb, Diana cionoiu, Dokaspar, Doncrawley, DrMac, E n sh, Echoray, Enjoi4586, Fleminra, FrankTobia, Gaius Cornelius, Gduffy, Glenn, Hairy Dude, Honeyman, Hu12, Isheden, Itai, Itsgeneb, JTN, JirkaHnidek, Joaopais, John smith4092, JohnOwens, Josemariasaldana, Juliancolton, Karn, Kbrose, Keancaptinh, Kenyon, Kgashok, Kgfleischmann, Kinema, Kkailas, Ko-, Koavf, Kurienmathew, Kvng, Ling.Nut, Lzur, MMuzammils, Malcolmst, Mange01, Marlogger, Mazevedo, MementoVivere, Mezzaluna, Mgream, Michael Devore, Mikewax, Mild Bill Hiccup, Muriel Gottrop, Mwtoews, Nixdorf, Pagingmrherman, Prasado, Puckly, RN1970, RTP123, Rapsar, Sceptre, SciComTech, Sdrtirs, Smartyepp, Snuffkin, Soumyasch, Stan3, Strait, Suruena, Susantab, TechMaker, Teemuk, Tero, Tfelber, Thakkar v, TheKMan, TheMandarin, Thelittlemouse, Tresiden, TwilligToves, Tzartzam, Vadmium, Victor Trac, Vina, Wikisierracharlie, Woolf44, Wrs1864, Yaronf, YordanGeorgiev, Yurik, 178 anonymous edits

RTP audio video profile *Source*: http://en.wikipedia.org/w/index.php?oldid=513502657 *Contributors*: Callidior, Dgtsyb, Fabrictramp, Fbdave, FrankT, Isheden, Kbrose, Kvng, Leolaursen, Lightmouse, Sergeymasushko, Stan3, 44 anonymous edits

Secure Real-time Transport Protocol *Source*: http://en.wikipedia.org/w/index.php?oldid=501149452 *Contributors*: Ahoerstemeier, Apyule, Bporopat, DKEdwards, Davidoran, Honeyman, Hu12, IRedRat, Kgfleischmann, Kvng, Mapet, Matt Crypto, Mendaliv, Mgream, Michael Rogers, Mm 202, Nageh, Nneuman, Ntsimp, Nyco, Odlund, Pdelong, Phatom87, PowerKong, Radagast83, Rearden9, RobertHannah89, Robertabela, Tfl, Timrem, Yamamoto Ichiro, Yaronf, 51 anonymous edits

Real Time Streaming Protocol *Source*: http://en.wikipedia.org/w/index.php?oldid=511947931 *Contributors*: 6birc, Adeian, Aldie, Alerante, Alexh19740110, Alone Coder, Andrew1214, Andthenandthenandthen, Armando, Barek, Breno, Ceefour, CesarB, Chenzw, Conversion script, Daniel.Cardenas, Davidjsmith67, Dcljr, DennisRobinson, Developer anu, Diberri, DonDiego, Dplore, Enjoi4586, Fabriciodosanjossilva, Facorread, Fudoreaper, Guaka, Hatsandcats, Hello32020, Hummerim, Hymek, Ice Ardor, Int21h, J. M., John smith4092, Johnny shaw, Kate, Kbrose, Koenige, Kostmo, Kuldip.sagar, Kvng, Lee Carre, Login to my passion, Mamaberry11, Mange01, Maxlapshin, Mishka.medvezhonok, Mithaca, Mulad, NeilFraser, NellieBly, Now3d, Ottexpert, Phatom87, Pravisas3, Purplefeltangel, Pxma, ReCover, Rentier, Rezd, RoyBoy, Roychai, Rrjanbiah, Sampart, Sapints, Sdrtirs, SpaceFlight89, Suruena, TRiG, Technobadger, Telecomtom, TheMandarin, Tobias Hoevekamp, UncleBubba, Venkytv, Vina, WikHead, Ykanada, Zanetu, 217 anonymous edits

G.711 *Source*: http://en.wikipedia.org/w/index.php?oldid=503963038 *Contributors*: 123465421jhytwretpo98721654, Aldie, AlexPlank, Alf Boggis, Apoc2400, Benhoyt, Bobblewik, Bryan Derksen, Dgtsyb, Digital Brains, DmitriyV, Dpupkov, Ferdinand Pienaar, Fonetikli, Fudoreaper, Grafen, Jamelan, JanCeuleers, Jim.henderson, Kanata14228, Kbrose, Kvng, Lawrence Cohen, Lightmouse, Lzur, MarkWahl, MathsPoetry, Mavros, Michael Hardy, Nick, Nn123645, OverlordQ, PaulWay, Polluks, Prari, RenesisX, Requestion, Rettetast, RevRagnarok, Rrw, Sashman, Saxbryn, Sdemjanenko, Sykemyke, Walkranrunning, 112 anonymous edits

A-law algorithm *Source*: http://en.wikipedia.org/w/index.php?oldid=469447145 *Contributors*: Abdul raja, Abdull, Bjh21, Bryan Derksen, CALR, Damian Yerrick, Dicklyon, Divide, DmitriyV, Foobaz, Giftlite, Imran, Inkling, ManN, Mavros, Mgimpel, Nick, Oarih, Omegatron, OverlordQ, Ozhiker, Ray Van De Walker, Ryan Roos, Speck-Made, Tardis, 21 anonymous edits

μ-law algorithm *Source*: http://en.wikipedia.org/w/index.php?oldid=503944708 *Contributors*: Abdul raja, Abdull, Attys, Bjh21, Brandon, Bryan Derksen, CALR, Callidior, CanisRufus, CyberSkull, Damian Yerrick, DavesPlanet, Dbenbenn, Deville, Diza, DmitriyV, Doug Bell, EAi, Evice, Furrykef, Gyro Copter, Handyhuy, Inkling, Jamelan, Jmgonzalez, Joeinwap, Karada, Kbrose, Kvng, Kwamikagami, Lmatt, Lovibond, Magioladitis, ManN, Mgimpel, NetJohn, Oarih, Omegatron, OverlordQ, Ozhiker, Pjacobi, Ray Van De Walker, Robwentworth, RokerHRO, Ryan Roos, Speck-Made, TimMorley, Tombomp, Torc2, Welsh, X-Fi6, 42 anonymous edits

G.729 *Source*: http://en.wikipedia.org/w/index.php?oldid=503963298 *Contributors*: Alexburke, Alsundma, Alxr101, Andromeda451, Arya6000, Bkervaski, Bloodshedder, Calltech, Cebra, Crazycomputers, Cryerson, Daf, DanMS, Digital Brains, DmitriyV, Erast, Eug, Glenno, Grendelkhan, JamesHenstridge, Jic, Jmvalin, Kbrose, Lightmouse, Lmatt, Lzur, ManiacK, MikeCapone, Miracle Pen, Nick, OverlordQ, PPBBB, Pawnbroker, Requestion, Sebesta, Shaddack, Sietse Snel, Terrible Tim, Tex23, TheParanoidOne, Towel401, Whaa?, WinTakeAll, 76 anonymous edits

G.722 *Source*: http://en.wikipedia.org/w/index.php?oldid=499378372 *Contributors*: AdamRoach, Behind The Wall Of Sleep, ChrisRing, Danny, Dicklyon, DmitriyV, Fonetikli, Furrykef, Jamelan, Kvng, Lightmouse, MarkWahl, Mindpimp, Nick, OverlordQ, Rcowdam, Requestion, Signalhead, Villarinho, Walkranrunning, 25 anonymous edits

G.726 *Source*: http://en.wikipedia.org/w/index.php?oldid=505867741 *Contributors*: A5b, Bobblewik, Callidior, DH85868993, Dgtsyb, DmitriyV, Editore99, Edokter, Furrykef, Hut 6.5, Intractable, Jamelan, Kanata14228, Ketiltrout, Kinema, Krille, Kvng, Lkinkade, Lovibond, Lupin, Lzur, MarkWahl, Mboverload, Mihaychuk, Mrand, Nick, OverlordQ, Pcb21, Requestion, Svick, Timo Honkasalo, 22 anonymous edits

Network address translation *Source*: http://en.wikipedia.org/w/index.php?oldid=513615056 *Contributors*: (, 2001:470:1F09:10D6:215:FF:FE77:FD85, 65.29.90.xxx, Aapo Laitinen, Aawc, Aelantha, Aitias, Ajo Mama, Alan U. Kennington, Alansohn, Aldie, Alex Smotrov, Alex.atkins, Alex.zeffertt, Alexhixon, AlistairMcMillan, Althena, Altrn8r, Andrew Hampe, Andrewpmk, Andrewriddell2, Aneah, Angela, Ap, ArsénireDeGallium, Ashwin, Asymmetric, Balajisarathi, Barek, Bbpen, Benoit rigaut, Bevo, Bos-Herz edit acct, Brion VIBBER, Brynosaurus, Cate, Cbarbry, Cburnett, CesarB, Cf. Hay, Cheung1303, Chowbok, Christian75, CommonsDelinker, Conversion script, Copsewood, Cotoco, Cpartsenidis, Crazycomputers, Crispmuncher, CrucifiedChrist, CyberSkull, Cybjit, D235, DARTH SIDIOUS 2, Daf, Damienivan, DanielEng, Daveg1k, Daveofthenewcity, Dawnseeker2000, Dcoetzee, DevastatorIIC, Dgtsyb, DiGiT, DisillusionedBitterAndKnackered, Droob, Drpixie, Drumzandspace2000, Dspradau, Dysprosia, EH74DK, Edcolins, Eddy264, Edward, Eolsson, Equendil, Ergy, Everyking, Evil Monkey, Excirial, Felipe1982, Fenix*NBK*, Fresheneesz, Gandaliter, Gareth Owen, Gary King, Garyvdm, Giftlite, Giraffedata, Glenn, Goatasaur, Golbez, GorillaWarfare, Gracefool, Graham87, Grimmfarmer, Guiltyspark, Guitargod2323, Hairy Dude, HarlandQPitt, Harrymcfogs, Hcberkowitz, Helix84, Hovden, Hydrargyrum, Icairns, Imcdnzl, Indrian, Iranway, Ivan Pozdeev, Ivan.Lt, J.delanoy, JHunterJ, JTN, Jan Kunder, Jasper Deng, Jengelh, Jez9999, Jhbdel, JidGom, Johnuniq, Jokerspuppet, JonDePlume, JonHarder, Jondel, Jonshea, Josh Parris, Joshf, Joy, Jpbowen, Jsnx, Just Another Dan, Jyoti.mickey, KD5TVI, Karada, Karstbj, Kbdank71, Kbrose, Kenyon, Keycard, Kgfleischmann, Kristof vt, Ksn, Kvng, Kwi, Kzollman, Laserbrian, Lavenderbunny, Leuk he, LiDaobing, Lightdarkness, LittleOldMe, LobStoR, Lspo99, M gol, MARQUIS111, MER-C, Magnus Manske, Mallow40, Maltest, Mandarax, Mannafredo, Manop, Marchash, MarcoTolo, Mav, Mditto, Mercury543210, Mindmatrix, Mintguy, Misza13, Mmmeg, Murjek, Mygerardromance, Naniwako, Nazli, Nealmcb, Nilmerg, Nimiew, Nixdorf, Nkansahrexford, Nubiatech, Nurg, Nyttend, Oalbacha, Oystein, PPBlais, Para, Pash, Pde, Pdelong, Peyre, Phatom87, Phenry, Philadams, Philbert2.71828, Piano non troppo, PierreAbbat, Pinkadelica, Plugwash, Pmsyyz, Pol098, Profchakraborty iitkanpur, Psychocim, Quarl, Quest for Truth, Rabarberski, Ramsey585, Rchandra, RedWolf, Rick Sidwell, Rob.bosch, Robert Brockway, Rohitthakral, Ross Fraser, Rrburke, Rushtoshankar, Ryan Roos, SF007, SLATE, SQL, SalineBrain, SaulPerdomo, Sbmehta, Seikku Kaita, Shahid789, Shirimasen, Shiro jdn, Sideswipe091976, Siipikarja, Simetrical, Simon South, SimonEast, SimonSellick, Slackerhobo, Smalljim, SoWhy, Sollosonic, SpyMagician, Steelmans1980, Stephan Leeds, Stephenb, Steven Zhang, Sujirou, Sun Creator, Svetovid, Syndicate, Taestell, Tagishsimon, TakuyaMurata, Teles, The Anome, The Inedible Bulk, Tiddly Tom, Tide rolls, Tobias Bergemann, Tommy2010, TonyHagale, Tresiden, Tristanb, Truthanado, Tsunanet, Tverbeek, Twasono, Twilsonb, Ulric1313, UncleBubba, Urhixidur, Vanished user 5zariu3jisj0j4irj, VanishingUser, Wavelength, Wernher, Wiki alf, Wikipelli, Wimblykit, Winterheart, WithGLEE, WojPob, Wolf0403, Wolfkeeper, Wolfrock, Woohookitty, Wrs1864, Xmm0, Xpanzion, Yk4ever, YordanGeorgiev, Zap Rowsdower, Zbxgscqf, Zhlmmc, Zondor, Zundark, 630 anonymous edits

NAT traversal *Source*: http://en.wikipedia.org/w/index.php?oldid=501601421 *Contributors*: A3 nm, Avenged Eightfold, Bellisman, Brynosaurus, Cab.jones, Calltech, Chris the speller, ChrisGualtieri, Cwolfsheep, Darrell Greenwood, Daveg1k, Diamondland, Gtg204y, Haakon, Harryboyles, Heron, Hu12, Iain Cheyne, Indefatigable, Int21h, Invertebra, Iridescent, Jasper Deng, JidGom, Kanezor, Kbrose, Kgfleischmann, Kurapix, Kvng, Lbecque, Lujianxiong, MacFreek, Mground, Mmom, Mr. Flibble, Petri Krohn, Phatom87, Plustgarten, Press@eyeball.com, QUILzhunter931, Rdenis, Rearden9, RossPatterson, Saxifrage, Sega381, Shadow1, Shawn in Montreal, SteinbDJ, TJJFV, Toh, Uiteoi, Wikiborg, Yarnover, Zconf, 103 anonymous edits

STUN *Source*: http://en.wikipedia.org/w/index.php?oldid=500824823 *Contributors*: Alvin-cs, Asymmetric, Barcex, Bpringlemeir, Brandon, Briankellis, Brighterorange, Byrial, Calltech, Camelek, CanisRufus, Cbarbry, CyberShadow, D235, Daf, Danroa, Danwarne, Dima125, Divyangmithaiwala, Doradus, ElTyrant, Elronxenu, Ettrig, Fieldday-sunday, GChriss, Gaius Cornelius, Genghiskhanviet, Hhielscher, Hu12, IOLJeff, Initialdp, Jeremy Visser, Karstbj, Kbrose, Kgfleischmann, Kvng, Lance808, Lujianxiong, MMuzammils, Malik Shabazz, Martin von Gagern, Mmxx, Nkour, Nomis80, Nyco, Ping6, Plugwash, Press@eyeball.com, Rjb1000, Robertabela, Rogerdpack, Samdutton, Sdstrowes, Sigmafactor, Siipikarja, Sklenny, Spearhead, Speck-Made, Splintax, Stephan Leeds, Suruena, Tbutter, The Anome, Thumperward, ToastieIL, UncleBubba, Warren, Wdyoung, Wikiskipper1789, ZeroOne, Zik-Zak, 96 anonymous edits

Traversal Using Relays around NAT *Source*: http://en.wikipedia.org/w/index.php?oldid=487014408 *Contributors*: Alvin-cs, Bloodshedder, Briankellis, CharlesC, Deville, EagleOne, Frap, Gardar Rurak, Hhielscher, Josh Parris, Kvng, MMuzammils, Nomis80, Ping6, Rchandra, Sesshomaru, Shashark, Sranil, Tonyhansen, Widefox, 23 anonymous edits

Interactive Connectivity Establishment *Source*: http://en.wikipedia.org/w/index.php?oldid=498057273 *Contributors*: Aavella77, Alfredhe, Brynosaurus, Calltech, CarolGray, Daf, DrPizza, Fenixlemus, Hairy Dude, Hhielscher, Irishguy, JamesHenstridge, JonHarder, Kbrose, Kgfleischmann, Kvng, Maggu, Mmick66, Nabla, Neustradamus, Ping6, Porttikivi, Press@eyeball.com, Rich Farmbrough, Saxifrage, Sdstrowes, Tfl, The Belgain, Tothwolf, Uiteoi, 37 anonymous edits

Media Gateway Control Protocol *Source*: http://en.wikipedia.org/w/index.php?oldid=501377493 *Contributors*: Abasota, Abhayani, Angela, Anonymous Dissident, Anthony Appleyard, Arcandam, Archer4523, Armfield, Ashok srinath, Avi.dorfman, Bluezy, Byrial, Bytemaster, Calltech, Cje, CultureDrone, Guaka, Haakon, JTN, Kagato, Kbrose, Kgfleischmann, Kinema, Kjboyleii, Kvng, MMuzammils, MathsPoetry, Mattingly23, Miracle Pen, Rchandra, RedWolf, Teemuk, Template namespace initialisation script, Thumperward, Tmh, Unixer, Wpjonathan, Wrs1864, Yaronf, 73 anonymous edits

Image Sources, Licenses and Contributors

File:Voip-typical.gif *Source*: http://en.wikipedia.org/w/index.php?title=File:Voip-typical.gif *License*: Creative Commons Attribution *Contributors*: FSII, JMPerez, מבעת-זרם

File:SIP signaling.png *Source*: http://en.wikipedia.org/w/index.php?title=File:SIP_signaling.png *License*: Public Domain *Contributors*: Original uploader was Dewikipeder at en.wikipedia. Later version(s) were uploaded by Noir at en.wikipedia.

File:Typical H.323 Stack.png *Source*: http://en.wikipedia.org/w/index.php?title=File:Typical_H.323_Stack.png *License*: Public Domain *Contributors*: ITU-T (talk). Original uploader was ITU-T at en.wikipedia

File:H.323 Architecture.png *Source*: http://en.wikipedia.org/w/index.php?title=File:H.323_Architecture.png *License*: Public Domain *Contributors*: ITU-T (talk)

File:Establishment of an H.323 call.png *Source*: http://en.wikipedia.org/w/index.php?title=File:Establishment_of_an_H.323_call.png *License*: Public Domain *Contributors*: ITU-T (talk)

File:H.323 High-level call flow.png *Source*: http://en.wikipedia.org/w/index.php?title=File:H.323_High-level_call_flow.png *License*: Public Domain *Contributors*: ITU-T (talk)

File:A typical H.245 exchange.png *Source*: http://en.wikipedia.org/w/index.php?title=File:A_typical_H.245_exchange.png *License*: Public Domain *Contributors*: ITU-T (talk)

Image:Ulaw alaw db.svg *Source*: http://en.wikipedia.org/w/index.php?title=File:Ulaw_alaw_db.svg *License*: Creative Commons Attribution-Sharealike 3.0 *Contributors*: Ozhiker

Image:Ulaw.JPG *Source*: http://en.wikipedia.org/w/index.php?title=File:Ulaw.JPG *License*: Public Domain *Contributors*: DavesPlanet, Keenan Pepper

Image:PD-icon.svg *Source*: http://en.wikipedia.org/w/index.php?title=File:PD-icon.svg *License*: Public Domain *Contributors*: Alex.muller, Anomie, Anonymous Dissident, CBM, MBisanz, PBS, Quadell, Rocket000, Strangerer, Timotheus Canens, 1 anonymous edits

Image:Full Cone NAT.svg *Source*: http://en.wikipedia.org/w/index.php?title=File:Full_Cone_NAT.svg *License*: Creative Commons Attribution-ShareAlike 3.0 Unported *Contributors*: Christoph Sommer

Image:Restricted Cone NAT.svg *Source*: http://en.wikipedia.org/w/index.php?title=File:Restricted_Cone_NAT.svg *License*: Creative Commons Attribution-ShareAlike 3.0 Unported *Contributors*: Christoph Sommer

Image:Port Restricted Cone NAT.svg *Source*: http://en.wikipedia.org/w/index.php?title=File:Port_Restricted_Cone_NAT.svg *License*: Creative Commons Attribution-ShareAlike 3.0 Unported *Contributors*: Christoph Sommer

Image:Symmetric NAT.svg *Source*: http://en.wikipedia.org/w/index.php?title=File:Symmetric_NAT.svg *License*: Creative Commons Attribution-ShareAlike 3.0 Unported *Contributors*: Christoph Sommer

Image:STUN Algorithm3.svg *Source*: http://en.wikipedia.org/w/index.php?title=File:STUN_Algorithm3.svg *License*: Creative Commons Attribution-ShareAlike 3.0 Unported *Contributors*: Christoph Sommer

File:Converged_Network_Architecture.png *Source*: http://en.wikipedia.org/w/index.php?title=File:Converged_Network_Architecture.png *License*: GNU Free Documentation License *Contributors*: Archer4523

License