Helsinki University of Technology Networking Laboratory
Teknillinen Korkeakoulu Tietoverkkolaboratorio
Espoo 2001

Report 2/2001

IP TELEPHONY PROTOCOLS, ARCHITECTURES AND ISSUES

Raimo Kantola (editor)

TEKNILLINEN KORKEAKOULU
TEKNISKA HÖGSKOLAN
HELSINKIN UNIVERSITY OF TECHNOLOGY

Helsinki University of Technology Networking Laboratory
Teknillinen Korkeakoulu Tietoverkkolaboratorio
Espoo 2001

Report 2/2001

IP TELEPHONY PROTOCOLS, ARCHITECTURES AND ISSUES

Raimo Kantola (editor)

Helsinki University of Technology

Department of Electrical and Communications Engineering
Networking Laboratory

Teknillinen Korkeakoulu
Sähkö- ja tietoliikennetekniikan osasto
Tietoverkkolaboratorio

Distributor:
Helsinki University of Technology
Networking Laboratory
P.O.Box 3000
FIN-02015 HUT
Tel. +358-0-451 2461
Fax +358-0-451 2474

ISBN 951-22-5452-2
ISSN 1458-0322

Otamedia Oy
Espoo 2001

Abstract

This report is a result of a post graduate seminar on IP Telephony (Course S38.130 Spring 2001). The report first gives an overview of IP Telephony and then proceeds to discuss quality of voice in an IP network, voice coding, the IP Telephony protocols, the service arcitecture and the potential service technologies. The protocols include signaling, transport and routing information protocols. The papers are based on literature study including materials of the 3GPP, students own research work and student's own measurements. Of particular interest are issues in applying IP Telephony in the 3rd generation mobile networks. These issues are discussed in several papers. Of particular interest is also the paper on choosing the transport protocol for SIP – it contains ideas that, we believe, have not been published before.

115 pgs

Preface

This report is a result of a post graduate seminar on IP Telephony. The papers appearing in the report were mainly prepared by the students during the Spring term 2001 and presented in the seminar itself that took place in Otaniemi, Espoo, Finland on April 6-7, 2001. After the seminar, based on comments, the students continued to improve their papers and finally the qualifying papers were selected by the editor.

The seminar was structured around Hersent, et al book on IP Telephony. Additional material includes RFCs and Internet Drafts related to IP Telephony and IP Voice and the materials of the 3G Partnership Project. List of references is at the end of each paper. More information on the seminar is available on the course www-page at http://www.tct.hut.fi/opetus/s38130/k01/index.shtml.

Contents

Overall picture of IP telephony

Ilkka Peräläinen
The Emergency response center authority
ilkka.peralainen @112.fi

Abstract

The main trend during the last five years in telecommunications, the convergence, has lead to the development of multimedia services in packet switched networks e.g. the Internet. After a short experimental phase the H.323 standard has been laid as a basis for multi-media services and applications including IP telephony. The present installed IP telephony systems use the H.323 protocol stack. Due to the complexity albeit flexibility of the H.323 the IETF is now finishing new rivalling stan-dards the SIP and MGCP, which are aclaimed to offer better functionality and simpler implementation. Inter-operability problems have sofar hindered a breakthrough of IP telephony. The driwing force in the telephony net-work convergence, cheaper calls, has not yet compen-sated for the technical defiences.

1 Introduction

The term IP telephony is older than Voice over IP. IP telephony has earlier meant the use of telephones or hybrid equipment and PBXs over IP using gateways to overcome the barriers of various networks. Voice over IP points to a world of carrying voice over IP networks not necessarily needing any separate telephone like equipment nor PBXs. Software phones in PCs are an example of these new implementations. Today the two terms are more or less synonyms or IP telephony is a subset of VoIP.

1.1 A short history

Only ten years ago the Internet was something totally different it is today. Its use was restricted mainly to universities and research instititutes. Its interface was text based and FTP was the main tool for exchanging information alongside with email and chat.
The first revolution occurred in 1993 with the World Wide Web. The colourfull new user interface appealed to thousands and thousands of new users and emerging search engines helped the users to find interesting new sites.
Year 1996 the first attemps were made to build an Internet telephony gateway. It consisted of a modem with speakerphone capabilities. The modem could only dial the destination number. At that time some sound board drivers were capable of simultanious play and record (full-duplex), but they lacked a telephone interface. The soundboard line-in jack had to be wired to the modem microphone and the modem speaker to the sound board line-out jack.
Some software was needed and the telephony freeware of those days the VAT came to help. By adding some code to interface the modem a crude one-line gateway prototype was invented. The potential of this primitive invention was huge. The development in the voice realm of Internet has since been immensely rapid and it has made a real contribution to the much advertised conver-gence of telecommunication.
However, the new possibilities also created new problems: Internet at that time was not ready for real time applications.
Anyhow IP telephony is growing very fast and it is estimated that by year 2002 nearly 20 % of the U.S. phone traffic will be carried over data networks.
By the World Wide Web the Internet had got its face, now it was getting a voice.

1.2 The overall situation now

The beginning of IP telephony has been lucky in that videly accepted standards have emerged in an early stage. Allmost all present implementations support the H.323 protocol family.

Standards should make it easy for the equipment of various vendors to interoperate. Unfortunately this has not been the case sofar in IP telephony. On the contrary the equipment and service implementations have mostly been proprietary in that the vendors have chosen a subset of the large and complex H.323 protocol stack that has met their immediate requirements. If you have bought an IP telephony system from one vendor you have been stuck to bying all future equipment from the same vendor. This lack of interoperability has been the major impediment for the wider deployment of H.323. For this reason fastest growth in VoIP will probably occur in enterprise networks, where a uniform system and equip-ment base is easier to achieve. [3]

The capabilites negotiation phase could at least to some extent solve this problem, but unfortunately even it is often not implemented completely.

This interoperability drawback is now luckily fading. The IP telephony manufacturers are more and more acclaiming that their hardware will interoperate with other vendors systems. The International multimedia teleconferencing consortium IMTC has been set up

with the primary goal of ensuring that various vendors products and services will interoperate.

Today the standardazation situation is however not at all clear. To overcome the drawbacks of the cumbersome and difficult to implement yet flexible H.323 protocol family the IETF has created new protocols like the Session initiation protocol SIP and the Media gateway control protocol MGCP which offer much more func-tionality than H.323 to VoIP.

SIP is simpler, it scales better and it leverages the existing DNS system instead of having created its own separate hierarchy of name services. By including a clients communication features within the invite request, SIP negotiates these features and capabilities of the call within a single transaction. The call setup delay can be as low as 100 ms depending on the network.

Thus the biggest question in VoIP today is which one of the standards will prevail. H.323 is now videly accepted and deployed, but many vendors have also announced support to the newcomer protocols. At this transitional stage we will probably see systems which support both protocol families.

This paper restricts to presenting an overview of the present prevailing technology, which anyway has laid the foundation of IP telephony and leaves the deeper pre-sentation and comparison of the new standards to other presentations. The functionalities presented here in context with the H.323 are all not H.323 dependent, but general to VoIP and have thus to developped in the newer protocols also.

1.3 Characteristics of IP telephony

The characteristics of IP telephony are quite complex, especially compared to streaming video, where large buffers can be used to compensate for the imperfectness of the Internet reagarding real time applications.
The main issues of IP telephony to be dealt with include:

- The human ears perception of echo and delay
- The voice compression and packetization technics
- Silent suppresion and comfort noice generation
- The Internet shortcomings for packetized voice: delay, jitter and packet loss
- The according remedies: buffering, redundancy, time stamps and differentiated services
- Telephone signalling protocols and various call types

2 H.323

H.323 is an ITU-T standard that was first developed for multimedia (voice, video and data) conferencing over LANs and later extended to cover Voice over IP. This multimedia origin is partly the reason for its claimed complexity for mere VoIP. Its first version H.323v1 was accomplished in 1996 and the second version v2 was ready by 1998. It includes both point-to-point and multipoint connections.

H.323 is one of ITU-T's mutually compliant videocon-ferencing standards. The others are:

- H.310 for broadband ISDN (B-ISDN)
- H.320 for for narrowband ISDN
- H.321 for ATM
- H.322 for LANs with guaranteed QoS
- H.324 for public switched telephone networks (PSTN)

Clients of H.323 are able to communicate with clients of the other above mentioned networks.

The H.323 standard does not assume any QoS in the network.

2.1 Components of H.323

2.1.1 Terminal

Terminals are the LAN client endpoints providing real time two way communications. They have to support H.245, Q.931, Registration Admission Status RAS and Real Time Transport RTP protocols.

A H.323 terminal can communicate with an other H.323 terminal, a H.323 gateway or a MCU.

2.1.2 Gateway

A H.323 gateway endpoint is the interface between the Internet and the PSTN or some other network. It communicates in real time mode between H.323 terminals on the IP network and other ITU terminals on a switched network, or to an other H.323 gateway. The H.323 gateway is optional and thus is not needed in a homogenous network

Gateways perform the translation between differing transmission formats like from H.225 to H.221. They can also translate between audio and video codecs. In one single LAN the gateway is not needed, as the terminals in this case can communicate directly. The communication to other networks is done via gateways using the H.245 and Q.931 protocols.

2.1.3 Gatekeeper

The gatekeeper is the vital - yet optional - central managing point in its zone. When a gatekeeper is used all endpoints in its zone (terminals, gateways and MCUs) have to be registered with it. It supports the end-points of its zone by

- Address translation from an alias, such as an email address or a telephone number, to a transport address using a translation table, which it updates by registration messages
- Admission control denying or accepting access based on e.g. call authorization or source and destination addresses.

- Call signalling either by processing the signalling itself or with the endpoints. It may alternatively connect a call signalling channel between the end-points and let them do the signalling directly.
- Call authorization using the H.225 signalling. The gatekeeper can reject calls due to time period or particular terminal access restrictions
- Bandwidth management, complying the number of calls with the bandwidth available
- Call management maintaining optionally a list of ongoing H.323 calls for e.g. Bandwidth manage-ment purposes
- Routing all calls originating or terminating in its zone. This feature enables billing and security. Rerouting to an other gateway in case of bandwidth shortage is also included in this option and it helps in developing mobile addressing, call forwarding and voice mail diversion services.

2.1.4 Multipoint Control Unit

The Multipoint Control Unit network endpoint makes it possible for three or more terminals and gateways to participate in a multipoint conference. The MCU con-sists of a mandatory Multipoint Controller MC and an optional Multipoint Processors MP.

The MCU is an independent logical unit, but it can be combined into a terminal, a gateway or a gatekeeper.

The MC determines the common capabilities of the terminals by using the H.245 protocol, while the MP does the multiplexing of audio, video and data streams under the control of the MC.

In addition the MCU can determine whether to unicast or multicast the audio and video streams depending on the capability of the network and the topology of the multipoint conference.

In a centralized multimedia conference each terminal establishes a point-to-point connection with the MCU which then sends the mixed media streams to aech endpoint. In the decentralized model the MC manages the communication compatibility but the terminals multi-cast and mix the streams.

2.2 The H.323 protocol stack

The audio video and registration packets of H.323 use the unreliable UDP protocol, while the data and control packets are transported by the reliable TCP protocol.

2.2.1 H.225 Call signalling

The call signalling channel is used to carry the H.225 control messages. In networks where a gatekeeper does not exist, the calls are signalled directly between end-points using Call signalling transport addresses. In this it is assumed that the calling party knows the address of the called party.

If there is a gatekeeper in the network, the calling party and the gatekeeper change the initial admission message using the gatekeeper's RAS channel transport address.

Call signalling messages can be passed in two ways

- In Gatekeeper routed call signalling the signalling messages are routed between the endpoints via the gatekeeper
- In Direct endpoint call signalling the endpoints change the messages directly

After the call signalling is completed the H.245 Control channel is establshed. When Gatekeeper routed call signalling is used, there are two ways to route the H.245 Control channel. Either the control channel is established directly between the endpoints or via the gatekeeper.

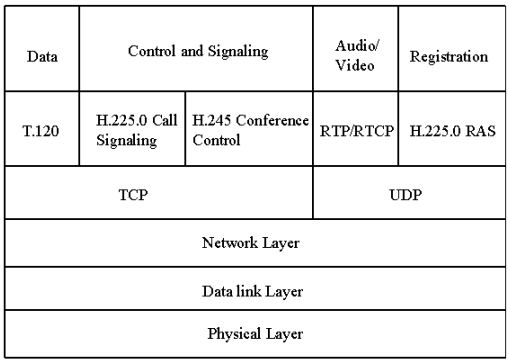


Figure 1: The H.323 protocol stack

2.2.2 H.245 Media and Conference control

After a H.323 call is established, H.245 negotiates and establishes all the media channels carried by RTP/RTCP.

The functions of H.245 are

- Determining master and slave. H.245 appoints a MC, which is in charge of central control in case a call is extended to a conference
- H.245 negotiates compatible settings between the endpoints after the call establishment. Renegotiation can take place anytime during the call
- Media channel control by which separate logical channels for audio, video and data can be opened or closed after the endpoints have agreed on capabi-lities. Audio and video channels are uni-directional while data channels are bi-directional
- Flow control messages provide feed back in case of communication problems
- Conference control keeps the endpoints mutually aware in a conference situation. A media flow model between the endpoints is also established

2.2.3 H.225 RAS Registration Admission Status

RAS defines communications between the endpoints and the gate keeper (in case one exists) by unreliable transport i.e. UDP.

RAS communications include

- Gatekeeper discovery is used by the endpoints to find their gatekeeper: endpoints multicast gate-

keeper requests to find the gatekeeper transport address
- Endpoint registration is compulsory in case where a gatekeeper exists in the network. The gatekeeper must know all the aliases and transport addresses of all the endpoints in its zone
- Endpoint location. A gatekeeper locates an endpoint with a specific transport address to update its address database for example

2.2.4 H.248 Implementors' Guide

The newcomer in the H.323 protocol family is the H.248. It is an enhancement of the centralized master slave type MGCP, Media gateway control protocol. H.248 was developed in co-operation with IETF, which calls it MEGACO.
One reason for the poor interoperability between various implementations of H.323 has been attributed to the lack of an implementation guide. This problem is now being solved by the IETF Megaco project.

2.2.5 RTP

The Real time transport protocol RTP and RTCP are both developed by the IETF. They transport the audio, video and data packets of real time media over packet switched networks. They are annexed in the H.323 protocol.

The main tasks of RTP are packet sequencing for detecting packet losses, adjusting to changing bandwidth conditions by payload identification, frame identifica-tion, source identification and intramedia syncronization to compensate for the varying delay jitter of the stream packets.

2.2.6 RTCP

The Real time transport control protocol works in conjunction with the RTP. In a RTP session participants send periodically RTCP packets to obtain information about QoS, session quitting, participant identification (email adresses, telephone numbers etc.) and intermedia synchronization.

2.2.7 Q.931

Then main purpose of Q.931 is call signalling and setting up the call.

3 Enhancements to H.323

A major drawback - especially compared to the fast SIP protocol - in the first H.323 version was the long call setup time. One message round trip is needed for
- ARQ/ACF sequence
- Setup connect sequence
- H.245 capabilities exchange
- H.245 master slave procedure
- Setup of each logical channel

In addition a TCP connection has to be setup for Q.931 and H.245 channels and each TCP connection also needs an extra round trip for the TCP window synchronization. In a WAN environment one round trip can take 100 ms, which ends up in a n unacceptably long setup delay especially when the gatekeeper routed model is used.
In a congested switched circuit network SCN, where a call cannot be setup, the network local exchange tries to send the caller a 'your call can not be connected'-message. No connect is sent because the network informs the caller and not the endpoint.
Voice messages can be sent in version v1 only after media channels have been established by sending first a connect message.

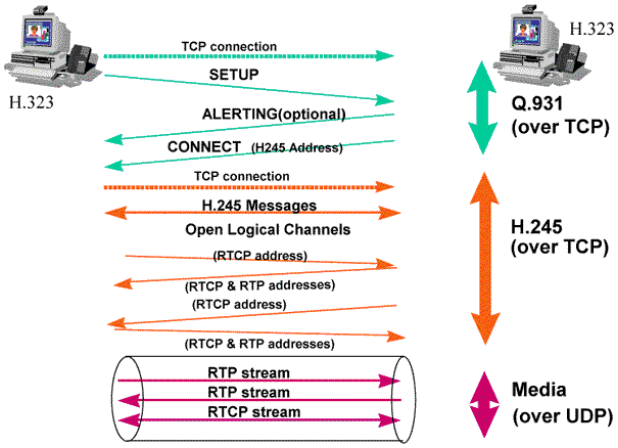


Figure 2: H.323 call sequence

There is a ITU-T Mobility Ad Hoc Group working on mobile H.323 standardization.

3.1 Faster procedures

The Fast connect procedure was invented to overcome the above mentioned deficiences. Fast connect solves the problems by
- Enabling uni- or bi-directional messages immediately after the Q.931 setup message
- Allowing a basic bi-directional audio only communication immediately after the connect message has been received
- Improving setup delays

An endpoint that uses the Fast connect procedure informs the calling party of all the media points it is prepared to receive or offers to send. This information is carried in the new fastStart parameter of the user to user Setup mesage. The description includes the codecs used and the receiving ports etc. This allows the early recei-ving of network prompts and improves also the setup delay.

The Fast connect procedure has been added as a core feature in the ETSI TIPHON project, because it resolves the interworking problem with the SCN.

Fast connect makes it possible to build simple limited capacity terminals that need only a minor part of the H.245 protocol.

H.323v2 offers an other solution with H.245 tunneling, where H.245 messages are encapsulated in Q.931 messa-ges reducing the TCP connections to one. When H.245 tunneling is used, the Q.931 channel must remain open for the duration of the call. The Tunneling method can also clear the network generated messages problem and will thus probably replace the Fast connect procedure.

The above described procedures are rather fixes to H.323v1 problems than a simpilification of the protocol.

The use of TCP causes at least one unnecessary SYN/ACK round dtrip. If the Setup message exceeds the maximum transfer unit MTU size, two or more TCP segments must be used. Most TCP implementations are network friendly mandating a slow start, where the first TCP segment has to be acknowledged before the rest can be sent.

A remedy to this problem is a special H.323v3 mode that will use UDP insted of or simultaneously with TCP signalling.

3.2 Conferencing with H.323

A multipoint control unit MCU masters a multipoint conference. It consists of one multipoint controller MC and optionally one or more multipoint processors MPs.

3.2.1 Multipoint controller MC

The MC decides

- who is allowed to participate
- how new participants are introduced to an ongoing conference
- how the participants synchronize their operation
- who is allowed to broadcast media etc.

A gatekeeper or a terminal possessing sufficient resources can include MC functionality in it and even mix media locally to a limited extent.

3.2.2 Multipoint processor MP

When several participants of a multipoint conference are simultaneously sending audio, video or data, there has to exist a network element that can mix or switch the incoming media streams. The endpoint terminals seldom have the capacity to do this. This mixer/switch element is called the MP.
When video is sent, the MP might choose the pictures of the latest speaker. When audio is the content, the MP could sum the voices of the potentially simultaneous speakers.
In a centralized conference the MP mixes and switches the media streams, where as in a decentralized conference the terminals send their streams directly to all other participating terminals.

3.2.3 H.332

The conference type where all participants retain a full H.245 control connection with the MCU is called ‘tightly coupled’. This type is resource intensive and it is obvious that it will not scale to numerous participants.
The solution to large conferences is the H.332. A large conference mostly has a panel of active speakers (5 to 10) and a large more or less passive audience of which one speaker at a time can propose a question or a comment to the panelists.
The H.332 keeps ‘tightly coupled’ conference connections with the panelists and a multicast RTP/RTCP conference with the passive listeners. The listeners have to know especially the codec and the UDP port used. H.332 uses the IETF Session description protocol SDP to encode this information.
Due to the large number of participants in a panel conference, a constraint must be set: the codec should remain stable. No new participant should have the possibility to change the codec as this would mean new negotiations for all the others.
If a listener wants to speak, he must use the regular join procedure to attain the right to speak his mind.

3.3 Directories and numbering

Most home IP telephony users are connected to Internet by a dial-up link, where the IP address is allocated on demand and is thus not static. In the early stages the users of IP telephony software contacted a server with a preconfigured IP address.

H.323 makes this kind of solutions obsolete. A terminal has to register to a gatekeeper using a RAS message, which contains all the necessary information, especially the current IP address to contact the terminal by using an alias.
At present the Internet Domain name system DNS is used to resolve the IP address when an alias name is known. The DNS servers make up an addressing network, where an address can be resolved by quering proper DNS servers top down until one is found which has detailed information of the endpoint in question. In addition to alias/IP address pairs a DNS database has much more information. It can hold information of the gatekeepers of its domain in ras://-type txt records. Once the gatekeeper is found, the caller knows to which transport address he shall send the setup message.
An important issue today for international IP calls from a PSTN network is the lack of a global IP telephony prefix. The solution has to scale to allow a large amount of users. The global prefix should tell the IP-callers network that the call that has to be setup is an IP call and should thus be routed to a home gatekeeper, which knows the location of the called party and can then resolve the phone address to a call signalling address.
It is clear that an IP call should be routed via an IP network avoiding the use of PSTN.
Several proposals have been made to define an IP telephony country code. The standardization process is not yet completed.
For example the use of DNS works well when IP address classes are used, but in the case of the ever

more popular classless interdomain routing CIDR, the reverse address resolution is supported only by few servers and is thus not applicable.

3.4 H.323 security H.235

The aim of H.235 is to provide privacy and authentication to all protocols using H.245 including H.323. Even without H.235 H.323 calls are more difficult to listen than ordinary telephone lines, which can be wiretapped. To break into a H.323 call you have to implement the codec algorithm.
With H.235 IP telephony becomes much safer than PSTN. The caller can even hide the telephone number of the endpoint it is trying to reach. Howerver, the H.235 is not yet widely deployed.
The first purpose of security was to secure the media channels so that no outsider could listen to the ongoing call. Soon it turned out that users most of all did not want to be charged for calls they did not make and that no one could monitor the called phone numbers. Providers wanted to authohrize calls when they were set up, not when media or control channels were established. So the signalling channel had also to be authenticated and secured.
The network elements that have to know the contents of the H.225 and H.245 messages need naturally to be trusted by the endpoints. This authentication can be carried out by Transport layer security TLS or a challenge response exchange using some certificate. H.323 does not specify the contents of the sertificates, but provides a way to exchange them and verify the indentities of the callers. The identity can be verified by several methods. A time stamp prevents repaly attacks.
H.323 does not ensure privacy on the RAS link between an endpoint and a gatekeeper, but it does provide authentication.
The call signalling channel H.225 can be secured by TLS or IPSEC.
The control channel H.245 security method is negotiated in the call signalling channel during the initial set up process before any other H.245 messages are sent. Various methods are accepted to initiate the secure channel.
After the H.245 channel is ensured, the terminals negotiate the media channel encryption method by capability exchange. A new capability is defined for each codec and encryption mode pair.
Many encryption algorithms can be utilized e.g. DES, Diffie-Hellm and RSA.

4 Codecs

The implementation of codecs is well developed and does not create any interoperability problems.

Audio codecs	**Title and date**
G.711	Pulse code modulation of voice frequencies at 56 or 64 kbps (11/88)
G.722	7 kHz audio coding at 64 kbps (11/88)
G.723.1	Dual rate multimedia speech coders at 5,3 and 6,3 kbps (03/96)
G.726	Speech coding at 16, 24, 32 or 40 kbps using ADPCM to encode a G.711 bit stream
G.728	Speech coding at 16 kbps using low-delay code exited linear prediction (09/92)
G.729	Speech coding at 8 kbps using conjugate-structure-algebraic-code-exited linear prediction (03/96)
Video codecs	
H.261	Audiovisual video codecs at p * 64 kbps, where p = 1 – 30 (03/96)
H.263	Low bit rate video coding (02/96)

Table 2: Audio and video codecs used with H.323

The mandatory speech codec is the G.711, which is a popular codec in telephony networks. It is not however quite suitable for Internet communication, where the subscriber loop bandwidths are much smaller. Today most H.323 terminals use G.723.1, which is much more efficient using only approximately one tenth of the G.711 bandwidth. The G723.1 uses 6,3 kbps bandwidth for continuous speech. When the call is wrapped in IP packets the additional packet headers increase the bandwidth needed to 17 kbps. When silence suppression is used the net bandwidth reduces back to ca. 6,7 kbps, which is ca. 10^{th} of the bandwidth of G.711. If IP header compression is used the relation is even greater. The G.728 and G.729 codecs are used for high quality audio with also very low bandwidth requirements.

Due to the burstiness and bandwidth hungriness of video communication efficient compression and decomp-ression technics are of utmost importance. H.323 specifies two video codecs namely the H.261 and the H.263. Other codecs can also bee used in case both endpoint support them.

Both the above mentioned video codecs use the discrete cosine transform DCT, H.261 with quantization and motion compression and H.263 with motion estimation and prediction

5 Applications and services

The vision of H.323 is interoperability between packet and circuit switched networks. H.323 also promises new value added services to the customers using circuit switched networks. These goals have not yet been achieved. Lower operational costs alone are not a reason good enough to switch to a new technology. Several Internet telephony service providers ITSPs have met the expectations of good services in North America and Europe, but the global interoperability is still a big problem. Furthermore the features and

quality of service are often inferior to plain old telephone services POTS.
The main reasons for not meeting the quality goals are the poor interoperability of the endpoints, especially gateways, of various vendors and the limited scalability of H.323 communications. [3]

5.1 The architecture of H.323

The architecture of a protocol lays the foundation for the services and applications that can be built on it. The architectural model of H.323 differs essiantially from that of the switched PSTN in that while PSTN is centralized the H.323 is decentralized.
The architectural model of H.323 is peer-to-peer, the procol design is based on the ISO QSIG standard and the services can be built using a multi-tier approach. Use of the QSIG reduces the complexity to interact with the circuit swithced PSTN networks that also use QSIG. The multitier model allows complex services to be built of building blocks of simple services.

5.2 H.450 Suplementary services

The supplementary services of H.323 rely on the H.450 series of recommendations. The key elements of it are protocol based on the QSIG, peer-to-peer signalling and a multi-tier approach of building services. [4]

H.323 architecture uses hig level Application programming interfaces APIs, so that software vendors do not have to work with low level implementation details, which would decrease interoperability risks.

5.2.1 H.450 based on ISO QSIG

The installed base of private telecommunications networks that use QSIG is wide and thus the use of QSIG in H.450 greatly helps the inter-working with that base. The migration from PBX networks to H.323 multimedia networks is simplified as well. Simpler gateways are one more advantage of using a common standard the QSIG.

5.2.2 Based on peer-to-peer signalling

In this respect the H.323 network differs essiantially from a circuit switched network. Like in the Internet, in H.450 the intelligence resides in the end and edge devices and the network simply routes the packets. The end device can be a PC or any IP phone and the edge device is a PBX or a consumer gateway at the home location. The state of the calls is also distributed in the end and edge devices.
In the traditional circuit switch model the intelligence and the state of the call reside in the network. The ends and edges are simple phones that run a stimulus-response protocol.
In H.450 new services can be installed in the ends and edges like software packages in a PC. Any software house can develop services to this standard and sell them directly to the end-user. This simplicity and straight forwardness in deployment will certainly stimulate the growth of a service building software industry. It should be remembered that the potential incompability of services in end and edge devices will be catered for by the capabilities negotiation process.
In the switch model new services are installed in the switch and may result in upgrades in other parts of the network before they are available for the customer. The switch is more over not at all so open to packages of 'outside' vendors. Yet it has to be admitted that in the central model the deployment of new services can be simpler. On the other hand the switch is a single point of failure while a software PBX can be embedded in each desktop phone. In this respect the distributed model is more fault tolerant than the switch model.

5.2.3 The multi-tier approach

The modular nature of the multi-tier approach enables the creation of basic services out of building blocks of primitives. Compound services can then be created by utilizing two or more basic services. Finally applications can be built by using compound services.
Simple services are for example:

- Multiple call handling
- Call transfer
- Call forwarding
- Call park and pickup
- Call waiting
- Message waiting indication
- N-way conference

Examples of compound services include:

- Consultation transfer
- Conference out of consultation

Consultation transfer uses call hold, multiple calls and call transfer. Conference out of consultation consists of call hold, multiple calls and n-way conference.

In Consultation transfer the user can perform three operations:

1. Put a multimedia call on hold and retrieve it later
2. Call an other person and optionally alternate between the two calls, or
3. Transfer the call

In Conference out of consultation the user has also three options:

1. Put a multimedia call on hold and retrieve it later
2. Call another person and optionally alternate between the two calls
3. Merge the calls in one conference call

6 Application examples

6.1 Call center integration

A call center gateway lets Web surfers with properly equipped multimedia PCs (typically with the right browser plug-in) connect to an existing Automatic Call Distributor (ACD) with Internet phone technology. This illustrates one of the major advantages of IP

telephony — its ability to combine voice and data on a single line.
The main advantage the IP telephony brings to Call centers is skill based routing. An incoming call can be directed to a call taker that for example can speak the same language as the caller or is a specialist in a field the caller wants help of. The call can also be directed to a personal adviser.
Emergency services provide another example of an architectural conflict since, for example IP addresses have no correlation with geographic location.

6.2 IEPS

As an other example of an application of IP telephony in the broad sense of the term application, this paper presents the basic requirements that IP telephony should take into account to support the International emergency preparedness scheme IEPS.
The ITU-T recommendation E.106 for emergency communications was first defined for PSTN and ISDN networks, but it was soon realised that this scheme had to be extended to cope with the next generation networks i.e. the Internet and especially IP telephony. In this regard the ITU-T Study group 16 is developing a new recommendation for International emergency multimedia service IEMS as an extension to E.106, to provide for enhanced emergency services over Internet based net-works in the future.
The IEPS is needed when there is a crisis situation which causes abnormal telecommunication requirements for governmental, military, civil authorities and other essen-tial users of PSTN. It allows authorized users to be able to access the International telephone service while the service is restiricted due to damage, congestion and/or other faults. [6]

6.2.1 Overall functional requirements

The primary goal of IEPS is to support crisis manage-ment arrangements by increasing the ability of the essen-tial users to communicate via the PSTN, ISDN, Public land mobile networks PLMN or IP telephony. [6]
The basic requirements include:

- International and national preference schemes are independent yet compatible: one could be activated when the other does not need to be activated
- National preference scheme users may not get access to the international scheme, but authorized users of the international scheme have to be able to use the national preference scheme
- In some national schemes IEPS features may be enabled permanently
- Calls originated by IEPS users should be given priority in the networks envolved when IEPS is enabled
- There must not be any conflict between preference for a call from an essential user and a call priority for a non-essential user to an emergency service
- If call restrictions to certain specific destinations (countries or areas) have been set when IEPS is activated, these restrictions should not apply to IEPS users
- IEPS calls should be marked from end to end

6.2.2 Established Telecommunication services

The essential features of the E.106 for the IEPS in the well established circuit switched PSTN and ISDN networks include

- Priority dial tone
- Priority call setup, including priority queuing schemes
- Exemption from restrictive management controls

In the United States the Goverment emergency telecom-munications service GETS uses the High probability of completion HPC in SS7 signalling for marking emer-gency calls. It should be noted that HPC does not include pre-emption of existing calls. In the U.S. alternate carrier routing ACR is used in the GETS in case some inter exchange carrier is not available. GETS uses a non-geographic toll free universal access number.

Some countries use IEPS access lines where all calls have a priority, while in some other countries priority is applied on a per call basis only.

6.2.3 Next generation networks

The IEPS requirements of E.106 should also be fullfilled by newly emerging next generation networks especially the Internet. The packet switching technology provides a clearly different operational evironment compared to the traditional circuit switched networks. Thus new aspects have to be considered but there also emerges the possi-bility of new innovative services based on the new featu-res of packet switched networks.
Examples of the new features are

- Quality, grade and class of service

The flexibility of the emerging object oriented and distributed technologies
For IP telephony an IEPS indicator similar to that of the HPC has to be defined, but the IP indicator has to be applied throughout the call.
There is extensive work going on in the international, national and regional standardization bodies to define the next generation networks. It is of utmost importance that they shall now start the work on the adaptation of IEPS. [7]

6.2.4 Quality

The quality of video in the Internet is poor and the audio quality is not high either especially compared to the PSTN. Because H.323 is a higher layer protocol, it can utilize the quality mechanisms of lower level protocols like the IntServ/RSVP and the DiffServ. In fact the development of QoS in the Internet is a result of the introduction of multimedia services to the Internet.
OoS features have been built in all modern LAN equipment although some critics say that enough of ever cheaper bandwidth will cater for the new multimedia services and QoS will not be necessary.

The codecs in use today squeeze an IP call to only about one 10th of the bandwidth of a traditional PSTN call and better codecs are on their way.

7 Market trends

7.1 The operator market changes

The new IP telephony technology has a markable influence on the telecommunications market. First the liberation of the telephone market legislation in the European Union. has given birth to new companies both operators and service providers. The ease to build new telephony services relying on the new IP telephony technology contributes to this trend.
In the second phase we will very probably see a conso-lidation period where new comers with unsustainable cash flows merge with profitable players or leave the hardening business campaign by going under.
Next a restructuring of the market will likely see mergers of smaller companies who cannot make the necessary investments that a new technology – how flexible and promising it ever is - inevitably requires. Figure 3. [9]

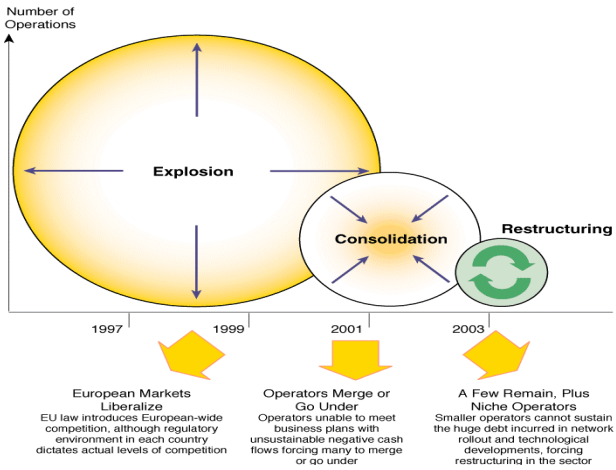


Figure 3: The operator market development

7.2 The European landscape

Unified Europe does not – at least yet - mean unified services. The main reasons are

- Current infrastructure is dispersed, difficult to deliver seamless service throughout the continent
- Back-office language and currency challenges
- Relatively new competitive environment forcing traditional players to evolve
- Consolidation and restructuring

New opportunities and new players
The EU is anyway focusing intensively in the telecom-munications field and unification will increase in the coming years.

7.3 The revenue share

As carriers move to packet networks voice and new applications pay the bills. As figure 4 shows the relative amount of pure datacommunication in the telecommu-nication networks compared to telephony will grow very fast in the next few years. Yet the overwhelming majo-rity of the revenue will continue to come from telephone calls and services. [10]

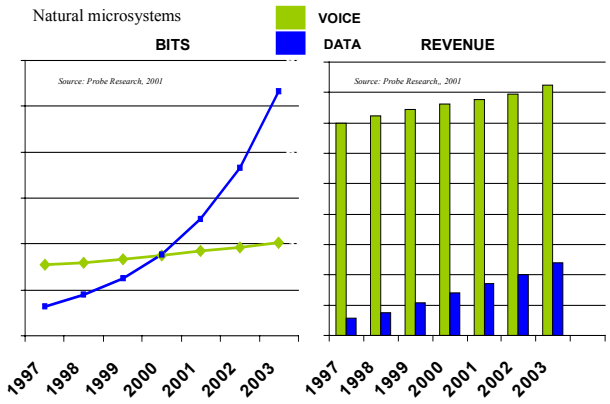


Figure 4: Voice and data

7.4 VoIP growth obstacles

According to one of the biggest European VoIP opera-tors Telia, the technological equipment compatibility process has been much too slow. Vendors have been lacking drivers for making equipment compatible. Luckily there is progress to be seen as more and more vendors are claiming to make their equipment co-operate with the equipment of the bigger vendors like Cisco systems.
An other obstacle is the price of the equipment. Carrier class VoIP equipment is still too expensive compared to traditional switching equipment. As an example to build a switched E1 connection costs $1500, while a VoIP E1 costs more than $10000. VoIP is today thus 6 - 8 times more expensive than POTS.
For these reasons the international voice carrier margins are very low which may to a certain extent slow down the development of the IP telephony business. [11]

List of acronyms

ACR: Alternate carrier routing
API: Application programming interface
CIF: Common intermediate format
Codec: Compression/decompression
DCT: Discrete cosine transform
DiffServ: Differentiated services
DTMF Dial Tone Multi-Frequency
GETS: Goverment emergency telecommunications service
HPC: High probability of completion
IEMS: International emergency multimedia service
IEPS: International emergency preparedness scheme

IETF: Internet Engineering Task Force
IMTC: International multimedia teleconferencing consortium
IntServ: Integrated services
IP: Internet protocol
ISDN: Integrated services digital network
ISO: International Organization for Standardization
ITSP: Internet telephony service provider
ITU-T: International Telecommunication Union – Telecommunications Sector
MC: Multipoint controller
MCU: Multipoint control unit
MP: Multipoint processor
MCS: Multipoint communication service
MGCP: Media gateway control protocol
OSI: Open systems interconnection
PBX: Private branch exchange
PLMN: Public land mobile network
POTS: Plain old telephone services
PSTN: Public switched telephone network
QSIG: D-channel signalling protocol at Q reference point for PBX nerworking
RAS: Registration, admission, status
RFC: Request for comments
RSVP: Resource reservation protocol
RTCP: Real-time transport control protocol
RTP: Real-time transport protocol
SCN: Switched circuit network
SDP: Session description protocol
SIP: Session initiation protocol
TCP: Transmission control protocol
TIPHON: Telecommunications and Internet protocol harmonization over networks
TLS: Transport layer security
UDP: User datagram protocol

References

[1] Arora, Rakesh: Voice over IP: Protocols and standards, Network Magazine, , 23rd of November 1999, http://www.cis.ohio-state.edu/~jain/cis788-99/voip_protocols/index.html

[2] Hersent Olivier, Gurle David, Petit Jean-Pierre: IP telephony, Addison Wesley Britain 2000, ISBN 0-201-61910-5

[3] Karim, Asim: H.323 and associated protocols, 26th of November 1999, http://www.cis.ohio-state.edu/~jain/cis788-99/h323/index.html

[4] Kumar, Vincent: Supplementary services in the H.323 IP multimedia telephony network, IEEE Communications magazine, July 1999

[5] Carlberg Ken et al: Framework for supporting IEPS in IP telephony, <draft-carlberg-ieps-framework-00.txt>, Network working group, November 2000

[6] Network working group, IETF: Description of an International emergency preparedness scheme (IEPS), <draft-itu-t-ieps-description-00.txt>, 20th February 2001

[7] Folts Hal: Functional requirements for priority services to support critical communications, TIPHON 17, Temporary document 116, Document for discussion

[8] White paper on IP telephony, A road map to supporting GETS in IP networks, Prepared under contract no. DCA 100-99-F-4413 Data item no. C002, Science applications international corporation, 27th of April 2000

[9] Indovino, Lisa, deltathree: Show me the VoIP deployment - European service providers, Spring 2001 Voice on the net conference presentatation

[10] Chase, Jack, Natural MicroSystems: Convergence of GSM, IP and VoIP, Spring 2001 Voice on the net conference presentatation

[11] Dahlgren, Paul, Telia international: Show me the VoIP deployments, Spring 2001 Voice on the net conference presentatation

H.323 Protocol Suite

Guoyou He
Helsinki University of Technology
ghe@cc.hut.fi

Abstract

Multimedia communication has affected various areas of people's life. Correspondingly numerous standards, communication technology and networks of multimedia from different vendors are evolving rapidly. The H.323 protocol suite specified by ITU is a main technology for real-time communication of audio, video, and data over packet switched networks. It also specifies the interoperability between the packet switched networks and circuit switched networks. In this paper, the history H.323 protocol suite and the architecture of H.323 system are reviewed first. Then the signaling and connection procedures of H.323 systems are presented. Finally we discuss the new features in H.323 version 4 and the features that are under development or to be specified for the future releases of H.323.

1 Introduction

At the present time, numerous multimedia applications and services are available. These applications and services include video and audio, synthesized video, audio and text, as well as interactivity. This multimedia information can be used for videoconferencing, telephony, video games, home shopping, video on demand, audio on demand and the like. However, this rapidly advancing multimedia technology is continuously spawning new products and applications, and their emergence has significant impact on a large number of people from all walks of life. This important and constantly evolving area comprises a number of technologies, which include multimedia computers, compression and multimedia networks as well as the transport mechanisms for these networks. The standards and technology for multimedia and multimedia communication are evolving at a prodigious pace. Videoconferencing provides for audiovisual communication as well as document sharing, including text, tables and images. The video and audio information must be compressed prior to entering a communication network and decompressed when leaving it. Hardware or software codec can be used for compression and decompression of video and audio information. The multimedia communication can be established with different equipment in the way of point-to-point communication or multipoint communication.

To provide interoperability for equipment from multiple vendors, standards have been established for POTS, ISDN, PSTN, and computer networks. For example, H.320 specifies the standards for ISDN videoconferencing; H.310, H.321, and H.322 specify the visual terminal for networks that guarantee quality of service (QoS); H.324 specifies videoconferencing over POTS modem connections; and T.120 standards provide the specifications for real-time data and audiographics conferencing [2]. Following the H.323 protocol suite for audiovisual communication will be discussed in detail.

2 What is H.323 suite

H.323 is a standard developed by the ITU. It specifies packet-based multimedia communications systems across networks, which might not provide any Qos guarantees. H.323 suite is a family of standards that includes many other ITU standards as shown in Table 1 [11].

Table 1: H.323 standards

Network	Non-guaranteed bandwidth packet-switched networks (e.g. IP)
Video	H.261, H.263
Audio	G.711, G.722, G.728, G.723, G. 729
Call Signalling and media packetisation	H.225.0
Call Control	H.245
Multipoint	H.323
Data	T.120

The H.323 standard is a principal technology for the transmission of real-time audio, video, and data communication over packet-based networks. It provides both multipoint and point-to-point sessions. H.323 defines the components, protocols, and procedures providing multimedia communication over packet-based networks, which include Inter-Networks (including the Internet), Local Area Networks, Enterprise Area Networks, Metropolitan Area Networks, and Intra-Networks [12]. Packet based networks also include point-to-point connections or dial up connections over the GSTN or ISDN which can use an underlying packet based transport. H.323 can be

used in a variety of mechanisms, which include audio and video (video telephony); audio only (IP telephony); audio, video and data; video and data; multipoint-multimedia communications.
The H.323 standard is part of the H.32X family of recommendations specified by ITU-T. The other recommendations of the family define multimedia communication service over different networks are shown in Table 2 [11].

Table 2: H.32X recommendations

ITU standard	Network
H.320	ISDNs
H.321, H.310	B-ISDNs
H.324	SCNs
H.323	Non-guaranteed bandwidth packet switched networks
H.322	LANs that provide guaranteed Qos

Interoperability with other multimedia networks is one of the primary goals in the development of the H.323 standard.

3 H.323 Version Suites

Since the first version of H.323 was approved in 1996, it has had 4 versions till the approval of H.323 Version 4 on November 17, 2000 [7].

3.1 H.323 Version 1

"Visual telephone systems and equipment for local area networks which provide a non-guaranteed quality of service", was published in 1996 [4] and was designed for local area networks. The first thing what companies tried to do was use H.323 in WAN, large private VoIP networks, and the Internet. It worked very well.
Recognizing the fact that H.323 was much more than a LAN protocol, the name was changed to H.323 Version2 in 1998 [10].

3.2 H.323 Version 2

H.323 "Packet-based multimedia communications systems" was approved in January of 1998. It brought in H.235 Security (Authentication of participant, Integrity of data, Encryption, and digital signature), Fast Connect, Supplementary Services (H.450.1 Signaling protocol, H.450.2 Call Transfer, and H.450.3 Call Diversion), Integration of data conferencing with T.120, and Scalability features (Alternate Gatekeepers, Time to Live, and Pre-granted ARQs) [9][10].

3.3 H.323 Version 3

H.323 version 3 was approved on September 30, 1999. It introduced a few modest improvements, mostly geared for better PSTN integration and scalability. However, H.323 has progressed substantially, mostly in the form of new Annexes to H.323 and H.225.0 that add considerable value to the overall H.323 system architecture [8].

3.4 H.323 Version 4

Many new enhancements have been introduced into the protocol H.323 Version 4, which was approved November 17, 2000. It contains enhancements in a number of important areas, including reliability, scalability, and flexibility. New features help facilitate more scalable Gateway and MCU solutions to meet the growing market requirements [7][10].

4 H.323 Architecture

The H.323 standard specifies the elements, protocols, and procedures providing multimedia communication over packet-based networks (see Figure 1 [10])

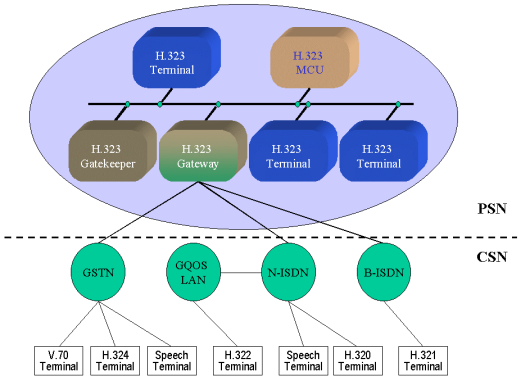


Figure 1: H.323 architecture

An H.323 system provides point-to-point or multipoint multimedia communication services. It has four main elements including terminals, gateways, multipoint control units (MCUs), and gatekeeper [12]. Terminals, gateways, and MCUs are also called endpoints.

4.1 Terminals

Terminals include Video I/O equipment, Audio I/O equipment, User Data Applications, and System Control User Interfaces. Terminals can be used for real-time bidirectional multimedia communications. An H.323 terminal can either be a personal PC or a stand-alone device, running an H.323 and multimedia applications. It supports audio, video and data communications. An H.323 terminal plays a key role in IP-telephony due to is its basic service of audio communications. Interworking with other multimedia network is the primary goal of H.323. The H.323 terminals are also compatible with the terminals on the networks given in Table 2 [4][13].

4.2 Gateways

Gateways connect H.323 networks to other networks, including the PSTN, ISDN, H.320 systems, etc. The connectivity of dissimilar networks is achieved by translating protocols for call setup and release, converting media format between different networks

[4][12]. An example of Gateway, which connects H.323 system to PSTN, is given in Figure 2 [3].

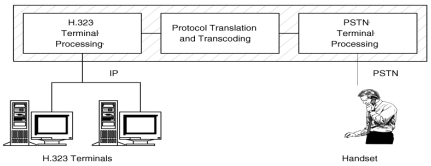


Figure 2: H.323/PSTN Gateway

4.3 MCUs

MCUs are responsible for managing multipoint conferences of three or more H.323 terminals. A two-terminal point-to-point conference can be expanded to a multipoint conference. The MCU consists of a mandatory multipoint controller (MC) and optional multipoint processor (MP). The MC supports the negotiation of capabilities with all terminals in order to insure a common level of communications. It can also control the resources in the multicast operation. The MP is the central processor of the voice, video, and data streams for a multipoint conference [13].
The MCU may (or may not) control three types of multipoint conference (Figure 3 [4]).

Centralized multipoint conference
All participating terminals communicate with the MCU point-to-point. The MC manages the conference, and the MP receives, processes, and sends the voice, video, or data streams to and from the participating terminals.

Decentralized multipoint conference
The MCU is not involved in this operation. Rather the terminals communicate directly with each other through their own MCs. If necessary, the terminals assume the terminals take the responsibility for summing the received audio streams and selecting the received video signals for display.

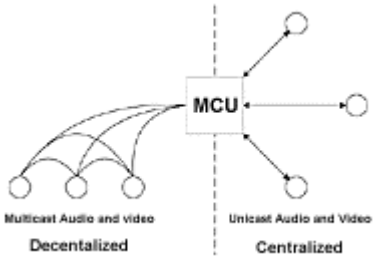


Figure 3: Multipoint conference

Hybrid multipoint conference
This conference is a mix of the centralized and decentralized modes. The MCU keeps the operations transparent to the terminals (see Figure 4 [4]).

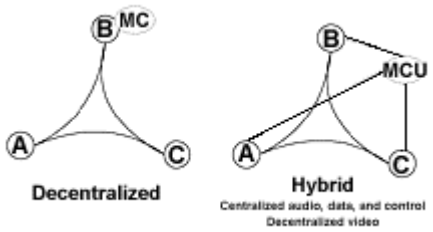


Figure 4: Hybrid multipoint conference

4.4 Gatekeepers

Gatekeepers are used for admission control and address resolution. A gatekeeper may allow calls to be placed directly between endpoints or it may route the call signaling through itself. A gatekeeper is also responsible for the services of band control, accounting, and billing. A single gatekeeper manages a collection of Terminals, Gateways, and MCUs forming a zone. A zone is logical association of these components and may span multiple LANs [4] (Figure 5 [3]).

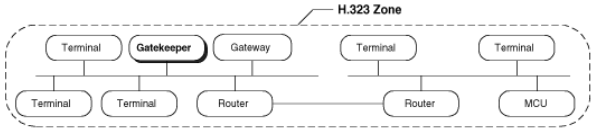


Figure 5: H.323 zone

5 The H.323 Protocol Stack

The H.323 suite consists of a set of standards. H.323 cites the use of the others shown in Figure 6 [13].
For audio applications, the minimum requirement is the support of recommendation G.711 (64 kbps channel). Other voice codec standards cited by H.323 are G.722 (48, 56, and 64 kbps channels), G.723 (5.3 and 6.3 kbps channels), G.728 (16 kbps channel), G.729 (8 kbps channel) [4].
The H.245, control protocol for multimedia communication, is used during an initial handshake between the machines to determine the audio encoding algorithm, terminal capabilities, and media channels. The terminals should be capable of sending and receiving different audio streams. After H.245 has completed the agreements on the terminals' capabilities and media channels, the H.225, call signaling and setup protocol, is used to format the audio stream.
H.261 is video coding standard. It was designed for data-rates which are multiples of 64kpbs. H.261 supports two resolutions, QCIF (Quarter CIF) and CIF (Common Intermediate format). If video is supported, the H.323 terminals must code and decode the video

streams in accordance with H.261 QCIF. Options are available, but they must use the H.261 or H.263 specifications. The coding algorithm of H.263 is similar to that used by H.261, however with some improvements and changes to improve performance and error recovery. H.263 supports five resolutions, QCIF, CIF, SQCIF (Sub-QCIF), 4CIF, and 16CIF.
Data support is through T.120, and the various control, signaling, and maintenance operations which are provided by H.245, Q.931, and the Gatekeeper specification.
The audio and video packets must be encapsulated into the Real-time Transport Protocol (RTP) and carried on a UDP socket pair between the sender and the receiver. The Real-Time Control Protocol (RTCP) is used to assess the quality of the sessions and connections as well as to provide feedback information among the communication parties. The data and support packets can operate over TCP or UDP [4][13].

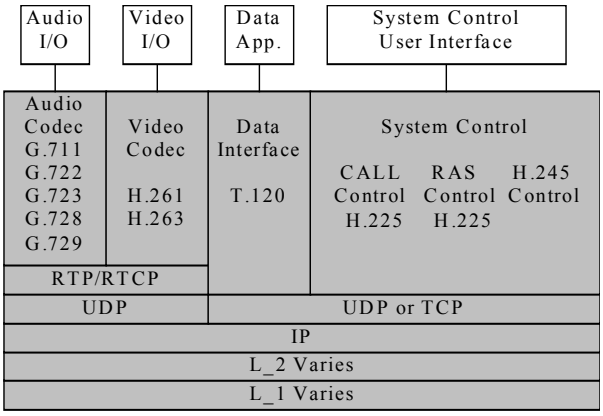


Figure 6: H.323 protocol stack

6 Call Signaling

Call signaling is the messages and procedures used to establish a call, request changes in bandwidth of the call, get status of the endpoints in the call, and disconnect the call [4].

6.1 Addresses

In H.323 system, each entity has at least one Network Address (e.g. IP address). This address uniquely identifies the H.323 entity on the network. Some entities may share a Network address (i.e. a terminal and a co-located MC). For each Network address, each H.323 entity may have several Transport layer Service Access Point (TSAP) identifiers. These TSAP Identifiers allow multiplexing of several channels sharing the same Network address. An endpoint may also have one or more alias addresses associated with it. An alias address may represent the endpoint or it may represent conferences that the endpoint is hosting. The alias addresses provide an alternate method of addressing the endpoint [4].

6.2 Registration, Admission and Status (RAS)

The RAS channel is used between H.323 endpoints and gatekeepers for gatekeeper discovery, endpoint registration, endpoint location, and admission control. The RAS messages are carried on a RAS channel that is unreliable. Hence, RAS message exchange may be associated with timeouts and retry counts.

Gatekeeper discovery

Gatekeeper discovery is the process an endpoint uses to determine which Gatekeeper to register with. The gatekeeper discovery can be done statically or dynamically. In static discovery, the endpoint knows the transport address of its gatekeeper a priori. In the dynamic method of gatekeeper discovery, the endpoint multicast GRQ message on the gatekeeper's discovery multicast address. One or more gatekeepers may respond with GCF message [4].

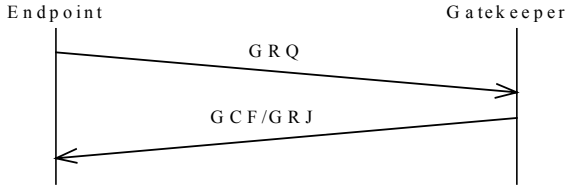


Figure 7: H.323 - Gatekeeper discovery

Endpoint registration

Endpoint registration is the process by which an endpoint joins a Zone, and informs the Gatekeeper of its Transport Address and alias address. All endpoints register with a gatekeeper as part of their configuration process. Registration occurs before any calls are attempted and occurs periodically as necessary [4] (see Figure 8).

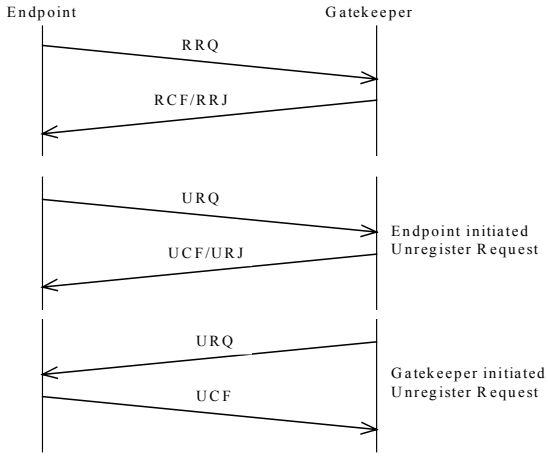


Figure 8: H.323 Endpoint registration

Endpoint location
Endpoint location is a process by which the transport address of an endpoint is determined and given its alias name or E.164 address [4].

Other Controls
The RAS channel is also used for other controls, such as admission control, to restrict the entry of an endpoint into a zone; bandwidth change, to modify the call bandwidth during a call; and disengagement control, to disassociate an endpoint from a gatekeeper and its zone [4].

6.3 H.225 Call Signaling and H.245 Control Signaling

H.225 Call signaling
The H.225 call signaling is used to set up connections between H.323 endpoints, over which the real-time data can be transported. The call signaling channel is a reliable channel, which is used to carry H.225 (adopted a subset of Q.931 messages and elements) call control messages. For example, H.225 protocol messages are carried over TCP in an IP based H.323 network [4].
In networks that do not contain a Gatekeeper, call signaling messages are passed directly between the calling and called endpoints. It is called direct call signaling. In networks that do contain a Gatekeeper, the H.225 messages are exchanged either directly between the endpoints or between the endpoints after being routed through the gatekeeper. It is called gatekeeper-routed signaling. The method chosen is decided by the gatekeeper during RAS-admission message exchange.

Gatekeeper-Routed Call Signaling
The admission messages are exchanged between endpoints and the gatekeeper on RAS channels. The gatekeeper receives the call-signaling messages on the call-signaling channel from one endpoint and routes them to the other endpoint on the call-signaling channel of the other endpoint (see Figure 9)[4].

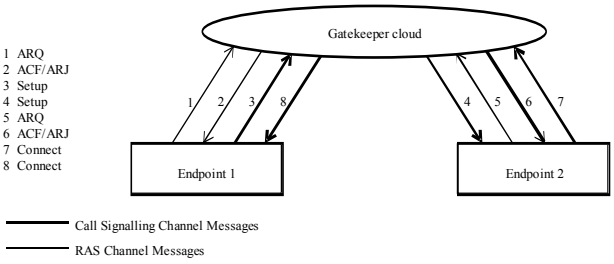


Figure 9: H.323-Gatekeeper routed call signaling

Direct Call Signaling
During the admission confirmation, the gatekeeper indicates that the endpoints can exchange call-signaling messages directly. The endpoints exchange the call signaling on the call-signaling channel (see Figure 10) [4].

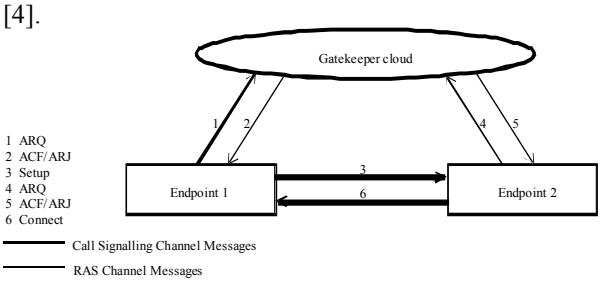


Figure 10: H.323-Direct endpoint call signaling

H.245 Control Signaling
When Gatekeeper routed call signaling is used, there are two methods to route the H.245 channel. In the first method, the H.245 control channel is established directly between the endpoints (see figure 11). In the second method, the H.245 control channel is routed between the endpoints through the Gatekeeper (see Figure 12). This method allows the Gatekeeper to redirect the H.245 Control channel to an MC when an ad hoc multipoint conference switches from a point-to-point conference to a multipoint conference. This choice is made by the Gatekeeper.
When direct endpoint call signaling is used, the H.245 control channel can only be connected directly between the endpoints [4].

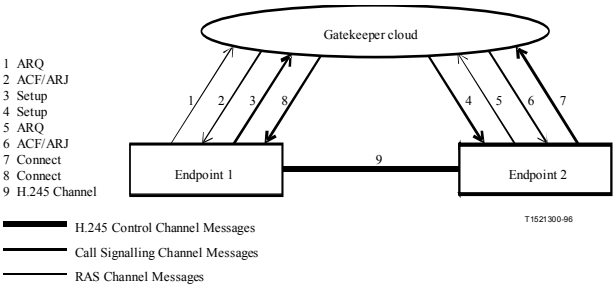


Figure 11: H.323 – H.245 control channel connection between endpoints

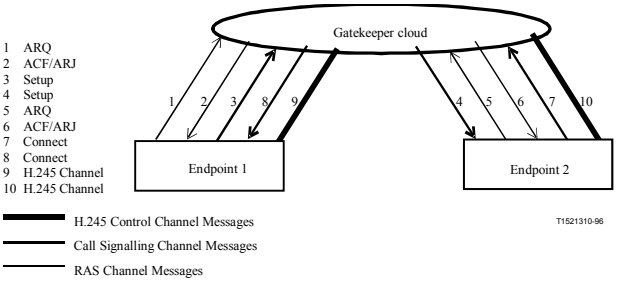


Figure 12: H.323 – Gatekeeper routed H.245 control

7 Connection Procedures

The connection procedures of the H.323 systems communication are made in the steps of Call setup, Initial communication and capability exchange, Establishment of audiovisual communication, Call services, and Call termination. This section uses an example network, which contains two endpoints connecting to a gatekeeper to illustrate the whole connection steps.

7.1 Step A: Call setup

Call setup can be in all following cases:
all combinations of Direct Routed Call signaling (DRC)/Gatekeeper Routed Call signaling (GRC), same or different Gatekeepers;
Fast connect procedures;
call forwarding using facility (restarts the procedure);
and setting up conferences [6].
Figure 13 illustrates the call setup process with the example of both endpoints registered to the same Gatekeeper. It assumes direct call signaling [12].

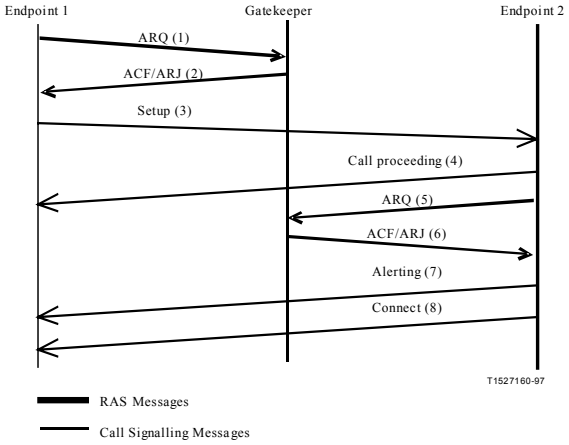


Figure 13: Call Setup

7.2 Step B: Initial communication and capability exchange

This step includes the procedures of Capability exchange, Master/Slave determination, and H.245 tunneling [4].
Once both sides have exchanged call setup messages from step A, the endpoints shall establish the H.245 Control Channel. The procedures of H.245 are used over the H.245 Control Channel for the capability exchange and to open the media channels.
The H.245 Master-slave determination procedures are used to resolve conflicts between two endpoints which can both be the MC for a conference, or between two endpoints which are attempting to open a bidirectional channel. Figure 14 is an example of H.323 control signaling flows.

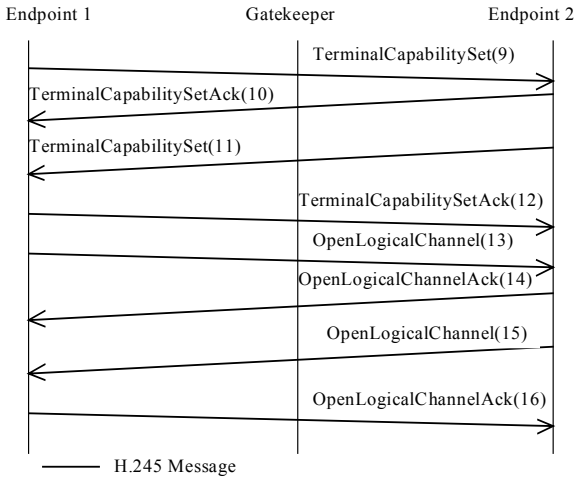


Figure 14: H.323 Control Signaling Flows

7.3 Step C: Establishment of audiovisual communication

Following the exchange of capabilities, master-slave determination, and opening of the logical channels for the various information streams, the audio and video streams, which are transmitted in the logical channels setup in H.245, are transported over dynamic Transport layer Service Access Point (TSAP) Identifiers using an unreliable protocol. Data communications, which are transmitted in the logical channels setup in H.245, are transported using a reliable protocol. Figure 15 is an example of illustrating the H.323 media stream and media control flows [4][11].

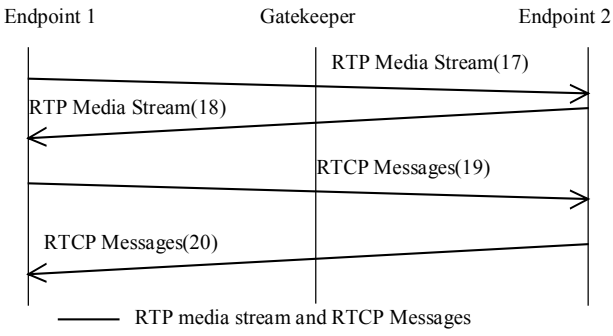


Figure 15: Media Stream and Media Control Flows

7.4 Step D: Call services

Call services include Bandwidth change, Status Information request for management, Conference expansion, multicast cascading, and H.450 Supplementary Services [4].

Bandwidth changes
Call bandwidth is initially established and approved by the Gatekeeper during the admission exchange. At any time during a conference, the endpoints or Gatekeeper may request an increase or decrease in the call bandwidth. An example of Bandwidth changes is given in Figure 16.

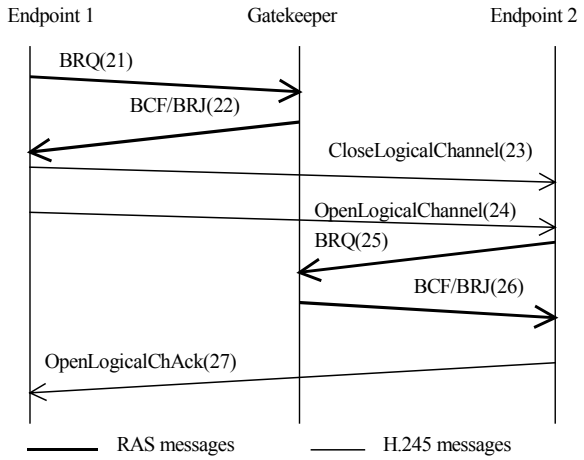


Figure 16: H.323 – Bandwidth Change

Status
Status is procedures of gatekeeper determining the work status, on/off or failure, of the endpoints. The Gatekeeper may use the H.225 Information Request (IRQ) /Information Request Response (IRR) messages to poll the endpoints periodically.
Conference expansion
Conference expansion is the procedure for expanding a point-to-point conference involving an MC to a multipoint conference. First, a point-to-point conference is created between two endpoints. At least one endpoint or the gatekeeper must contain an MC. Once the conference has been created, the conference may be expanded to multipoint conference by any endpoint in the conference inviting another endpoint into the conference through the MC, or an endpoint joins an existing conference by calling an endpoint in the conference. Figure 17, 18 [4] illustrate the H.245 Control Channel topology for the Direct Call Signaling model, and the Gatekeeper routed Call Signaling model.

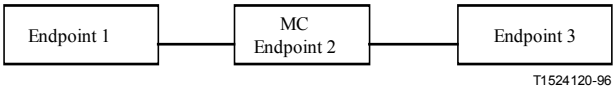


Figure 17: Direct Call Signaling model

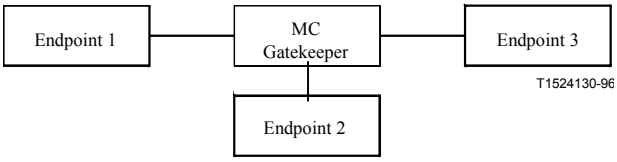


Figure 18: Gatekeeper routed Call Signaling model

Multicast cascading
Multicast cascading is the case when a call is established between the entities containing the MCs, and the H.245 Control Channel is opened, the active MC (Master/Slave procedure) may active the MC in a connected entity. Once the cascade conference is established, either the master or slave MCs may invite other endpoints into the conference. There is only one master MC in a conference. A slave MC can only be cascaded to a master MC.
H.450 Supplementary services
The H.450 supplementary services are optional to H.323 systems. These services include call forward, call hold, call waiting, message waiting indication, and name identification etc.

7.5 Step E: Call termination

Call termination can be made by any endpoint when video, audio, or data transmissions are at end. Correspondingly all logical channels for video, audio, or data are closed. Terminating a call may not terminate a conference. It can be done by MC that the terminating of a conference. Figure 19 [12] illustrates the call release procedure.

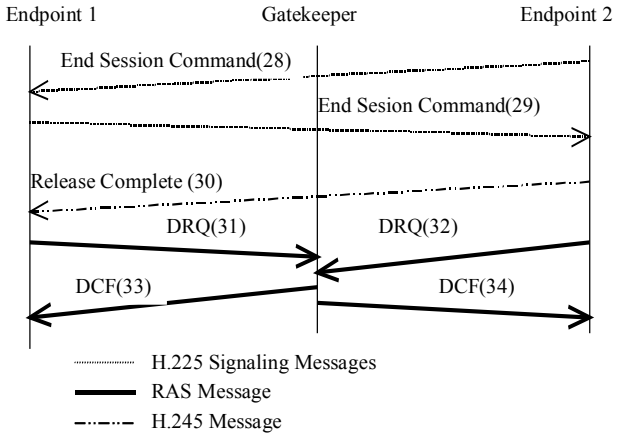


Figure 19: H.323 Call Release

8 New Feature of H.323 Version 4

H.323. Version 4 was approved on November 17, 2000. It contains enhancements in a number of important areas including scalability, reliability, flexibility, services, must have features, and generic extensibility framework [7][10][1].

8.1 Scalability, Reliability, and Flexibility

The H.323 Version 4 enhances the scalability of H.323 systems in the areas including Gateway Decomposition with H.248, Additive Registrations, Alternate Gatekeepers, and Endpoint Capacity Reporting.

Gateway Decomposition

Traditional Gateways were designed so that both media and call control were handled in the same box. Recognizing the need to build larger, more scalable gateway solutions for carrier solutions, the ITU-T worked jointly with the IETF produced the Recommendation H.248, which describes the protocol between the Media Gateway Controller (MGC) and the Media Gateway (MG). H.323 version 4 supports the decomposition of Gateway into Media Gateway Controller (MGC) and Media Gateway (MG).

The decomposed Gateway separates the MGC function and the MG function. Multiple MGs may exist to allow the decomposed Gateway to scale to support much more capacity than a composite Gateway. The communication between the MGC and MGs is done through H.248 (see Figure 20 [11]).

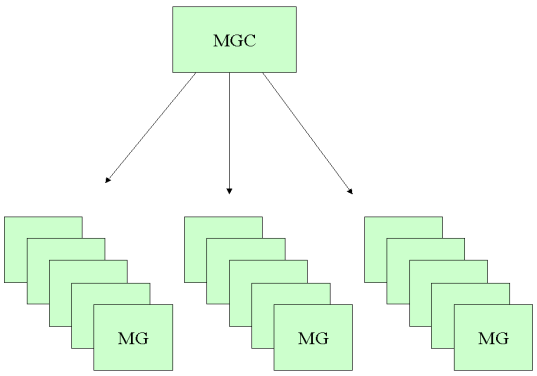


Figure 20: Decomposition Gateway

Alternate Gatekeepers

The architecture of alternate Gatekeepers is shown in Figure 21 [11]. By using Alternate Gatekeepers, endpoints can continue functioning when the communication between the endpoints and one or more Gatekeepers. It increases the reliability and never loses calls.

Endpoint Capacity reporting

H.323 endpoints report capacity to Gatekeepers. By utilize endpoint capacity reporting, Gatekeepers may select an endpoint that is best capable of handling the call. It is very useful for large scale deployments of Gateways, and extremely increases the availability (see Figure 22 [11]).

* GK selects the GW with the most capacity.
* *H.323 terminals report capacity in absolute terms, not in percentages.*

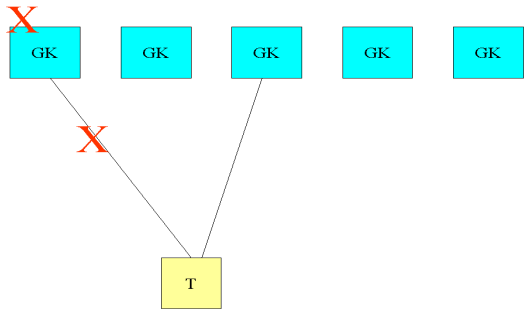


Figure 21: Alternate Gatekeepers

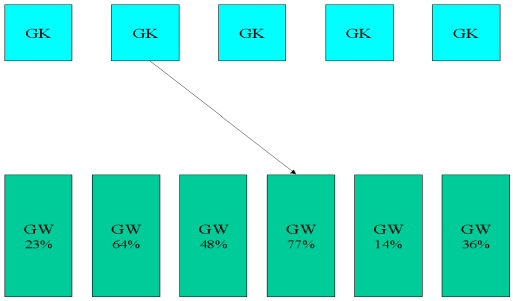


Figure 22: Endpoint Dispatcher

8.2 Services

One of the most important features of a VoIP protocol is its ability to provide services to the service provider and end users. H.323 has a rich set of mechanisms to provide supplementary services. Version 4 introduces a few more supplementary services to strengthen the protocol in this regard. These services mainly include HTTP-based Service Control, Stimulus-based Control, and Call completion [1][10][7].

HTTP-based Service Control

H.323 version 4 specifies a means of providing HTTP-based control for H.323 devices. With HTTP-based control, service providers have the ability to display web pages to the user with meaningful content that ties into the H.323 systems. In essence, it is a third party call control mechanism that utilizes a separate HTTP connection for control.

Stimulus-based Control

H.323 version4 provides a new "stimulus-based" control mechanism. With this mechanism, an H.323 device may communicate with a feature server to provide the user with various services. The H.323 endpoint may possess some intelligence, but some intelligence may reside only in the feature server or multiple feature servers. The features may be numerous. Any new features may be added to the feature servers without the delay by standard procedure.

Call completion
This is a new H.450 supplementary service, which provides a standard means of allowing calls to complete when the user is either busy or there is no answer.

8.3 "Must Have" Features

The features included are listed below [7][10][5]:

Usage Information Reporting
To help providing accurate billing information, the Gatekeeper can request the endpoint to provide usage information reporting to the Gatekeeper at various times during the call, including at the beginning of the call, during the call, and at the end of the call.

Caller Identification
H.323 Version 4 contains complete information for providing caller identification services with H.323.

Tones and Announcements
H.323 version 4 details the procedure for indicating the presence of in-band tones and announcements. Such tones and announcements are often heard when the destination number is incorrect or unreachable.
In addition to in-band tones and announcements, the Gatekeeper may signal an endpoint to play specific announcements at various times: pre-call, mid-call, or end-call.

Alias Mapping
When routing calls, a telephone number in the IP-world may not be sufficient for proper routing into the SCN. In addition, it might be that a service provider would like to use the same Gateways to provide Virtual Voice Private Networks, but need some intelligence in a device to perform proper mapping. With Version 4, a Gateway, for example, can indicate that it can perform alias mapping at either the ingress or egress side of a call. This will reduce the number of malformed numbers, as well as provide a means for providing Voice Virtual Private Network (VVPN) services.

Better Bandwidth Management (multicast)
Prior to H.323 Version 4, and endpoint could request much more bandwidth than it actually needs, and thus, cause wasting network resources. With Version 4, it is mandatory that an endpoint made bandwidth requests with a lower value if, indeed, the endpoint is using less bandwidth than it had initially indicated in the ARQ.
In addition, managing bandwidth for multicast sessions has been nearly impossible since, unless the Gatekeeper routed the H.245 signaling and carefully monitored the media channels that were opened, it could not determine whether two endpoints that request bandwidth are actually requesting bandwidth for a multicast session or unicast session. This becomes a much bigger issue when many people are participating in a multipoint multicast conference. With Version 4, specific details about the media channels are conveyed to the Gatekeeper in (Information Request Response) IRR messages (if the Gatekeeper requests them), so that the Gatekeeper can better control bandwidth utilization.

Fax Enhancements
Version 4 of H.323 allows an endpoint to be able to initiate a voice call and then switch to fax at some point. It allows an IP-based fax device to operate in a similar manner as today's PSTN fax devices. Version 4 also enhanced to utilize TCP for carrying fax data. Previously, UDP was the only real option for carrying fax data.

Tunneling other protocols
H.323 is often used to inter-work between two circuit networks. To provide better inter-working, Version 4 provides a mechanism whereby QSIG (Signaling between the Q reference points) and ISUP may be tunneled without translation essentially. H.323 may act as a transparent tunnel for those non-H.323 signaling protocols (see Figure 23 [5]).

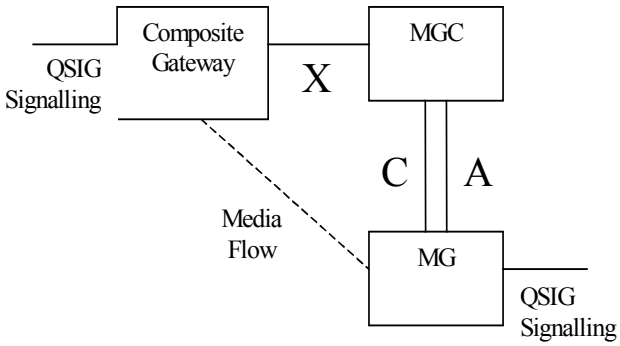


Figure 23: H.323 – QSIG tunneling example

H.323 specific URL
Version 4 introduced URL scheme "h323". The H.323 URL allows entities to access users and services in a consistent manner. The form of the H.323 URL is "h323:user@host", where "user" is a user or service and "host" might be the Gatekeeper that can translate the URL into a call signaling address.

Call Credit-related capabilities
H.323 v4 provides the means of communicating available funds or for the Gateway to control early call termination based on available funds for the prepaid IP telephony. H.323 v4 adds these features to the RAS protocol.

Multiplexing audio and video
One weakness with the current usage of RTP is difficulty in synchronizing the separate audio and video streams. Version 4 now includes an optional procedure, which allows both video and audio to be multiplexed in a single stream. This will assist endpoints in synchronizing video and audio.

DTMF Relay via RTP
H.323 version 4 allows an endpoint to utilize RFC 2833 "RTP Payload for DTMF Digits, Telephony Tones and Telephony Signals" to send and receive DTMF digits.

8.4 Further Features are under developing on H.323

ITU-T is working or is going to work on some of further enhanced features of H.323, which include Generic Extensibility Framework, Protocol Inter-working, Mobility, and Robustness [1][10].

Generic Extensibility Framework

The Generic Extensibility Framework (GEF) introduces new means by which H.323 may be further enhanced or extended with optional features, which does not require changes to the current ASN.1 syntax

Inter-working or integrating with other protocols.

The inter-working or integrating with newly developed protocols may need to be developed. These protocols include SIP, H.248/Megco, and Bearer Independent Call Control (BICC).

SIP is gaining in popularity as a VoIP protocol. H.248/Megaco may find its way into many "media gateway" devices, ranging from residential gateways to large-scale service provider gateways. The Bearer Independent Call Control (BICC) protocol will compete with both H.323 and SIP for a place in the service provider network. Making H.323 work with is also important.

Mobility

Mobility includes terminal mobility, user mobility, and service mobility. To implement the mobility of H.323, the functions of mobility management need to be defined, which include Home Location Function (HLF), Visitor Location Function (VLF), Authentication Function (AuF), and Inter-working Function (IWF) (see Figure 24 [1])

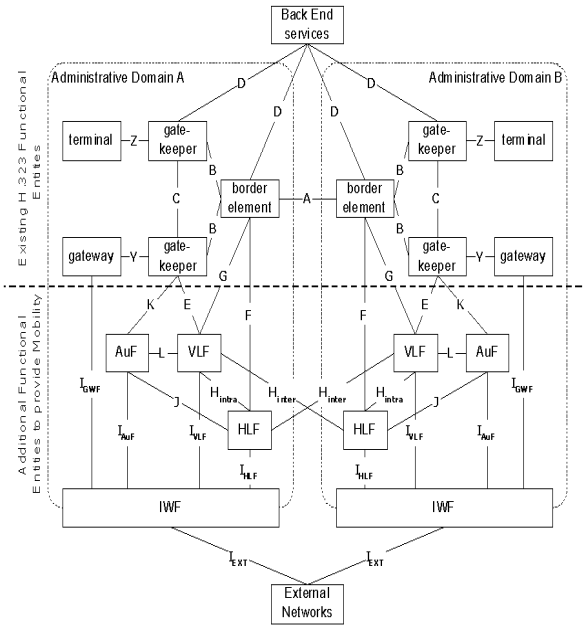


Figure 24: H.323 - Mobility

Robustness

Robustness is developing, it requires refining the architecture for recovery from crashes. Currently two architectures are proposed: small scale systems and large scale systems.

In smallscale systems, the architecture makes each element responsible for detecting failures of the others. If one element fails in the system, the others can go to the backup element. Some state information of the failure element is then need to be provided. For large scale system, the architecture is very complex and need to be specified.

9 Comments on H.323

H.323 is a very complex system with all kinds of features for multimedia communications, but not every part of H.323 has to be implemented when building a powerful and useful system. Multimedia over IP, in itself, has a certain amount of complexity associated with it. It results in that the system for implementing the inter-working between different multimedia systems with various features and services is inevitably complex. The complexity does exist in a H.323 system for a reason, the reason may become even more evident as video, audio, and data conferencing become more prevalent [10].

H.323 allows the use of multiple codecs. In the systems, there is a good reason for using each of the codecs.

Gatekeepers are optional in H.323 system. They provide consistent means for H.323 endpoints to perform address resolution, and may perform inter-working between simple H.323 (set devices) and more protocol-complete H.323 entities. Gatekeepers can act as a platform from which powerful new IP-based services can be built and provided.

H.323 is scalable. Service providers can deploy H.323 networks in small scale or large scale depending on the expected features and services.

H.323 is a proven technology used in large networks. It has excellent integration with PSTN.

Multimedia conferencing shows the real potential for H.323 used in multimedia communication

Many equipment manufacturers, software vendors, and service providers have built products and services supporting H.323. It greatly supports the success of H.323.

With the constantly coming of new technologies, for example BICC, H.323 has big pressure on keeping its place in the service provider network.

10 Conclusions

As just presented, H.323 is organized around four major facilities: (a) terminals, (b) Gateways (which can perform protocol conversion), (c) Gatekeeper (bandwidth manager), and (d) multipoint control units

(MCUs), responsible for multicasting. The H.323 standard is a principal technology for the transmission of real-time audio, video, and data communication over packet-based networks. It provides both multipoint and point-to-point sessions. One of the primary goals of developing H.323 standards is to provide the interoperability between packet switched networks and other multimedia networks. H.323 is a rich and complex specification. Especially the version 4 is a powerful system for multimedia communication. It contains enhancements in a number of important areas, including, scalability, reliability, flexibility, supplementary services, and new features. Future releases will be even more powerful. Especially the inter-working or integrating with other newly developed protocols will strengthen its position in the multimedia communication area. Mobility will greatly increase flexibility for using H.323 system in the fields of terminal mobility, user mobility, and service mobility. Of course, mobility will also greatly increase the complexity of the H.323 system.

Even though H.323 is a powerful system for multimedia communication, if has faced great competition from some newly developed protocols, such as SIP, H.248/Megco, and BICC. Reducing the complexity of H.323, and simplifying its usage will hopefully improve its leading position in fast changing multimedia communication world.

Acronyms

ACF/ARJ – Admission Confirm/Reject
ARQ – Admission Request
AuF – Authentication Function
BCF/BRJ – Bandwidth Confirm/Reject
BICC – Bearer Independent Call Control
B-ISDN – Broadband ISDN
BRQ – Bandwidth Request
CIF – Common Intermediate Format
DCF/DRJ – Disengage Confirm/Reject
DRC – Direct Routed Call signaling
DRQ – Disengage Request
DTMF – Dual-Tone Multi-Frequency
GCF/GRJ – Gatekeeper Confirm/Reject
GEF – Generic Extensibility Framework
GK – Gatekeeper
GQOS – Guaranteed Quality of Service
GRQ – Gatekeeper Request
GSTN – General Switched Telephone Network
GRC – Gatekeeper Routed Call signaling
GRQ – Gatekeeper Request
HLF – Home Location Function
IRR – Information Request Response
IRQ – Information Request
ISDN – Integrated Services Digital Network
ISUP – ISDN User Part
ITU – International Telecommunication Union
IWF – Inter-working Function
MC – Multi-point Controller
MCU – Multi-point Control Unit
MG – Media Gateway
MGC – Media Gateway Controller
MP – Multi-point Processor
N-ISDN – Narrow-band ISDN
PISN – Private Integrated Services Network
POTS – Plain Old Telephone Service
PSN – Packet Switched Network
PSTN – Public Switching Telephone Network
QCIF – Quarter Common Intermediate Format
QoS – Quality of Service
QSIG - Signaling between the Q reference points
RAS – Registration/Admission Status
RCF/RRJ – Registration Confirm/Reject
RRQ – Registration Request
RTCP – Real Time Control Protocol
RTP – Real-time Transport Protocol
SCN – Switched Circuit Network
SIP – Session Initiation Protocol
SQCIF – Sub Quarter Common Intermediate Format
TCP – Transmission Control Protocol
TSAP – Transport Service Access Point
UCF/URJ – Unregistration Confirm/Reject
UDP – User Datagram Protocol
URQ – Unregistration Request
VLF – Visitor Location Function
VoIP – Voice over Internet Protocol
VVPN – Voice Virtual Private Network

References

Boaz Michaely: H.323 Overview, November 2000. http://www.packetizer.com/iptel/h323/papers/

Chan-Hwa Wu ja J. David Irvin: Emerging Multimedia Computer Communication Technologies, Prentice Hall, 1998, ISBN 0-13-079967-X.

Databeam Corporation: A Primer on the H.323 Series Standard, 1999. http://www.packetizer.com/iptel/h323/primer/

ITU-T: Recommendation H.323, 1998.

ITU-T: Recommendation H.323, 2000.

Olivier Hersent, David Gurle & Jean-Pierre Petid: IP Telephony Packet-based multimedia communications systems, Pearson Education Limited 2000, ISBN 0-201-61910-5.

Packetizer: H.323 Version 4 – Overview, 2001. http://www.packetizer.com/iptel/h323/whatsnew_v4.html

Packetizer: H.323 Version 3 – Overview, 2001. http://www.packetizer.com/iptel/h323/whatsnew_v3.html

Packetizer: H.323 Version 2 – Overview, 2001. http://www.packetizer.com/iptel/h323/whatsnew_v2.html

Paul E. Jones: H.323 Past, Present and Future, January 2001. http://www.packetizer.com/iptel/h323/papers/

Phillips Omnicom Training: Voice Over IP Training Material, 2000.

Trillium: H.323, 2000. http://www.iec.org/tutorials/h323/

Uyless D. Black: Voice Over IP, Prentice Hall PTR 2000, ISBN 0-13-022463-4.

Voice Quality in IP Telephony

Vesa Kosonen
Networking Laboratory
Helsinki University of Technology
P.O.Box 3000, FIN-02015 HUT, FINLAND
vesa.kosonen@hut.fi

Abstract

This paper has been presented at the licentiate course in the Networking Laboratory of Helsinki University of Technology in April 2001. The topic of the course was 'IP Telephony'.

In this paper we will study voice quality in IP telephony. We will look at the causes of impairments along the end-to-end path and how to recover from them. We will also introduce common criterions to measure voice quality. Some results based on measurements in our laboratory with different commercial VoIP phones will also be included.

1 Introduction

The improvement of voice quality has been one of the main interests in the telephony industry since the invention of telephone in 1876. Especially delay and echo have caused most problems. Nowadays the quality of the SCN phones is very good. The codec using G.711 -standard with 8 kHz sampling frequency gives MOS value 4.2, while the theoretical maximum value is 5.0. On the contrary voice quality of IP telephony is far away from quality of SCN phones. But for the surprise of many IP telephony draws the attention of ever increasing number of people. Also telephone companies have realized the potential of IP telephony, especially the threats that lie ahead of them.

The first applications of IP telephony were programs that made it possible to call anywhere in the world that had Internet access. One could use a personal computer to call to another personal computer with the same program without ever entering SCN network. Because calling with this new invention was not so handy as with normal phone only a few people liked to use it even it meant free calls. Voice quality varied from bad to moderate. Later on new telephone operators appeared to the market who gave low price overseas telephone services based on the utilization of IP telephony. Now you could make a call from your own SCN telephone. Even though voice quality was only moderate many people got interested. The same thing happened earlier with mobile telephones: mobility was favored despite of lower voice quality. Ever since new IP telephony solutions have came to market. Especially companies benefit from the new technology: it allows cheap internal calls and the same infrastructure can be used for data and voice. This is especially tempting since the amount of voice traffic is decreasing compared to data traffic.

2 End-to-End Route of a Voice Call

2.1 Scenarios from ETSI/TIPHON

ETSI/TIPHON has defined four different scenarios of making an IP call (See Appendix A). The scenario 0 defines a call from an IP network to another IP network. The scenario 1 on the other hadn defines a call from IP network to SCN. The scenario 2 is the opposite of the scenario 1: a call from SCN to IP network. Finally the scenario 3 defines a call between two SCNs using Internet between them.

2.2 Path of a Voice Call

We will use scenario 0 to show the route of an end-to-end voice call (Fig. 1). The analog speech of a caller is first transformed to digital bits. It is called A/D transformation and it is done by taking samples from the speech and quatising them. The bitstream is then encoded. Encoding or speech coding is the process of transforming digitized speech into a form that can be efficiently transported over the network. The reverse function of encoding is decoding which is performed at the receiving end [2]. After encoding bits are framed. The size of the frame depends on the used codec. E.g. G.723.1 codec uses 30 ms frames. Several frames are grouped together and packetized by adding RTP+UDP+IP header (12+8+20=40 bytes). Now the packets are ready to be sent to Internet.

At the other end the header information is removed. While travelling through Internet some delay is always introduced. Delay is not the same for all packets which causes variation in delay or interarrival jitter as it is also called. Packets might also be lost. Jitter buffer is used to correct those impairments. After a playout time the packets are deframed, decoded and transformed back to analog voice.

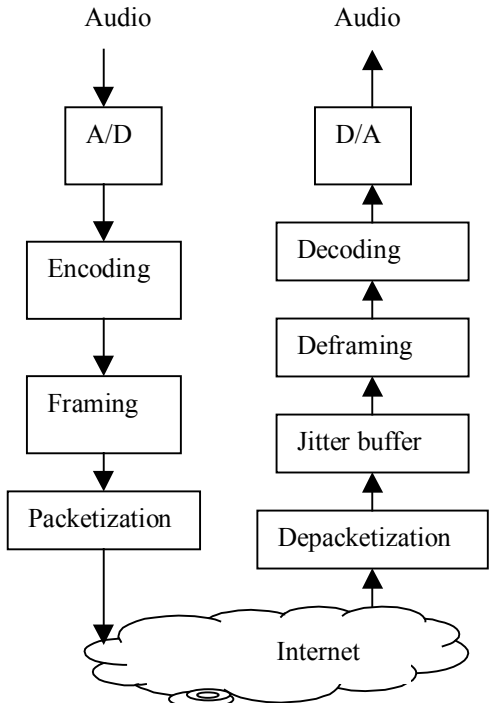


Figure 1. End-to-End route of voice in an IP-to-IP call [2].

3 Causes of Impairments and how to Improve Voice Quality

There are several factors that cause impairments to the end-to-end voice quality of IP telephony. We will consider five of them, namely delay, jitter, packet loss, echo and bandwidth usage. They occur either at terminal or along the transmission path or in both of them.

3.1 The Operating System

As we speak the sound card samples the microphone signals and accumulates them into a memory buffer. When the buffer is full the sound card tells the operating system by the help of an interrupt signal that it can retrieve the buffer. There is a limit how many interrupts an operating system allows. For instance Windows allows an interrupt not more often than every 60 ms. This means that the buffer collects speech samples in chunks of at least 60 ms which is the introduced minimum delay [3].

To avoid this problem some vendors use real-time operating systems, which allow as many interrupts as needed. Another way is to do all real-time functions using dedicated hardware and perform only the control functions from the non real-time operating system [3].

3.2 Encoder/Decoder

Typically in a telephone conversation there are periods of silence. These silence periods don't contain intelligent information and are cut off. This is done with the help of voice activity detector (VAD) which cuts the silence periods and sends only silence information descriptor (SID) frames. The other end adds the silence into the speech. This is a very efficient way of saving bandwidth since the estimated time of silence is close to 50% [2]. In a bitstream there is also always redundant information that can be removed before sending e.g. information that can be forecasted by extrapolating or that is repeated by certain intervals. To save bandwidth further bitstream is also compressed, both payload and headers (40 bytes → 2 or 4 bytes).

The bit rate of a typical G.711 codec is 64 kbit/s. The quality is good but the problem is the big bandwidth usage. Codecs with smaller bit rates have been developed. E.g. G.723.1 codec has bitrate only 5.3/6.3 kbit/s. Yet the voice quality is almost the same (MOS value 3.7/3.9).

3.3 Audio equipment

There are two kinds of echos namely talker echo and listener echo. Talker echo means that the speaker hears his/her own voice but delayed and attenuated. It can be caused by electric (hybrid) echo or by acoustic echo picked up at the listener side. If the talker's echo is reflected twice the listener will hear the talker's voice twice - a loud signal first, then attenuated and much delayed. This is called listener echo [3].

An IP phone can be either a separate microphone and loudspeakers, personal computer together with headset or it may look like an ordinary SCN phone. When used a separate microphone and loudspeakers there will exist acoustic echo which can be eliminated with the help of an acoustic echo canceller (AEC). Typical values for acoustic echo attenuation in loudspeakers phones are 10-15 dB, in hands-free phones are 35-40 dB and in phone with good quality handset are 35-40 dB [3]. If the two other types of phones are used then acoustic echo doesn't exist.

3.4 An IP/ISDN Call

When a call is made between an IP phone and an ISDN phone there is a need to use a gateway. The gateway makes the mapping from IP network to ISDN network and vice versa. This introduces some delay.

3.5 An IP/PSTN Call

If a call is made from an IP phone to a PSTN phone we also need to use a gateway. In addition to that PSTN

network will also introduce electric echo which is caused by the 2wire/4wire transformation. The phones at user side use only two wires but the network uses four wires. Thus the 2wire/4wire transformation has to be performed. The electric echo can be eliminated with an electrical echo canceller (EEC) which should be positioned as close to the user as possible [3].

3.6 Jitter Buffer

Jitter buffer is used to eliminate the impairments caused by the transmission path. Real-time Transmission Protocol (RTP) was developed to handle the situations that may occur to the packets as they travel through Internet. If packets are lost codecs try to hide that. This procedure is called 'error concealment'. Lost packets are decomposed by interpolating the previous packet [2]. This prevents gaps in the speech. The packets that arrive in wrong order or are delayed are desequenced with the help of RTP time stamp and sequence number. The size of the jitter buffer can be adjusted and it is a trade-off between delay and voice quality. If the size of jiffer buffer is long it has time to wait for delayed packets but in that case it will introduce more delay. To minimize delay all packets will not arrive in due time and that causes gaps into the speech.

4 Methods to Assess Voice Quality

4.1 Mean Opinion Score

In order to be able to compare voice quality of different telephony systems we need some common criterions. One possibility is to assess voice quality subjectively with the help of MOS (Mean Opinion Score) scale. Voice quality is given values between 0 - 5. Table 1 below shows the MOS values of the most common ITU-T standardized codecs.

Table 1. MOS values of the most common ITU-T standardized speech codecs [3].

Standard	Bitrate (in kbit/s)	MOS value
G.711	64	4.2
G.726	32	4.0
G.728	16	4.0
G.729	8	4.0
G.723.1	6.3/5.3	3.9/3.7

4.2 E-model

E-model (ITU-T standard G.107) was originally developed by ETSI. It is a computational tool to assess end-to-end voice quality. It was developed for the use of network planners to help to ensure that users will be satisfied with end-to-end transmission performance while avoiding over-engineering of networks [4]. The model estimates the conversational quality from mouth to ear as perceived by the user at the receive side, both as listener and talker. The primary output from the model is the "Rating Factor" R. The model combines the effect of several impairment factors instead of considering them separately [4]. The Rating Factor R is defined as follows:

$$R = R_0 - I_s - I_d - I_e + A \qquad [4]$$

Where

- R_0 represents the basic signal-to-noise ratio, including noise sources such as circuit noise and room noise
- I_s is the combination of all impairments that occur simultaneously with voice signal, such as the quantization distortion or too load side tone
- I_d represents impairments caused by delay including impairments caused by talker and listener echo or by loss of interactivity
- I_e represents the impairments caused by use of special equipment, such as low bit rate codecs or by e.g. packet loss [4]
- A is the advantage factor, which expresses the decrease in the rating R that a user is willing to tolerate lower voice quality, e.g. the A factor for mobile telephony is 10 [5] and for multi-hop satellite connections A is 20 [4].

The values of the rating factor R can lie between 0 and 100, where R=0 means an extremely bad quality and R=100 means a very high quality. The values of R can also be compared with MOS values and user satisfaction as shown in the Table 2. The lower limit of R is included but not the upper limit.

Table 2. Comparing R, MOS and user satisfaction according to [4], [6]

R-value	R-value (attribute)	MOS-value (lower limit)	User Satisfaction
90 - 100	Best	4.34	Very satisfied
80 - 90	High	4.03	Satisfied
70 - 80	Medium	3.60	Some dissat.
60 - 70	Low	3.10	Many dissat.
50 - 60	Poor	2.58	Nearly all dissat.

PSTN quality is an example of desirable level of voice quality. It is described as “good intelligibility, good speaker identification, naturalness, only minor disturbing impairments” [7] If a call is considered to be PSTN quality then rating factor values R=>70 should be reached. G.107 standard lists the default values, which

are recommended to be used for all parameters that don't vary during the calculation. If only default values are used the calculation results in a very high quality with rating factor of R = 93.2. [4]. In that case (and if echo is perfectly controlled, that is echo loss = ∞) a call retains its quality up to a mouth-to-ear delay of 150 ms. Also delay values even up to 400 ms are still within the limits of PSTN quality [5].

5 Measurements

5.1 A Practical Test on Delay

To get an idea about the end-to-end delay in different telephony systems you can perform the following trendsetting test. Make a call with an ISDN phone to another ISDN phone. Start counting. When the other person hears you say 'one' he should say 'two'. When you hear the other person say 'two' you should say 'three' and so on. Count until fifty and take the time elapsed. You should repeat the test several times. Then make another call with your mobile phone to another mobile phone and repeat the same procedure. Compare the times [3]. Table 3 lists the results from the previous measurements including the same measurements done in our laboratory environment. It shows that the end-to-end delay in ISDN network is lower than in GSM network. On the other hand Selsius/Cisco phone seems to have lower delay than NetMeeting.

Table 3. A Practical Test on Delay

	Time is takes to count to fifty	
ISDN	32 s	
GSM	42 s	
Selsius/Cisco phone		40 s
NetMeeting		55 s

5.2 The Measuring Environment

We have measured some commercial VoIP phones (Selsius/Cisco IP phone and Microsoft's NetMeeting). The phones were connected to 10 Mbit/s Ethernet local LAN (Fig 2).

We used Dummynet software to simulate different real life situations by altering its parameters, such as delay, bandwidth and packet loss. The packets were captured and analyzed by DNA-323 analyzer software. Before introducing some of the results we will explain the key concepts namely packet spacing difference (D(i)) and jitter (J).

Packet spacing difference is defined as the difference between the consecutive received packets subtracted with the difference between the consecutive sent packets (Fig. 3).

$$D(i) = (R_i - R_{i-1}) - (S_i - S_{i-1}) \qquad [8]$$

When delay is constant the value of D(i) is zero. But when delay varies the spacing of the packets at the

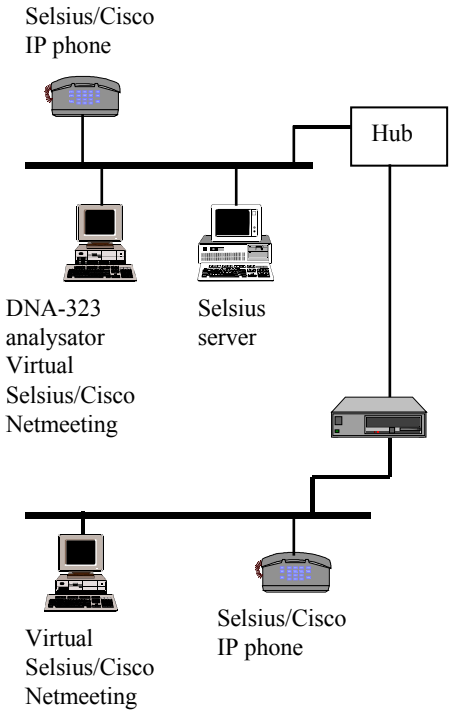
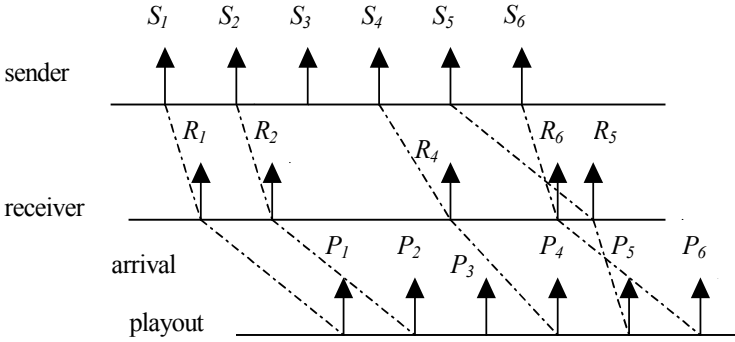


Figure 2. Measurement arrangements.

receiving end varies, too. The parameter that describes this difference is called delay variance or jitter J. There is a connection between D(i) and J as follows:



$$J_n = J_{n-1} + 1/16 * (|D(i)| - J_{n-1}) \qquad [8]$$

Figure 3. Synchronization in Jitter Buffer.

Jitter gives the size of the jitter buffer that is needed to synchronize the packets before they are played out. The playout time depends on the position of the first packet as follows:

$$P_n = P_{n-1} + (S_I - S_{I-1}), \text{ for } n>1 \qquad [8]$$

Where

- R_I is the arrival time of the received packet
- S_I is the generation time of the packet
- P_n is the playout time of the packet

5.3 Some Measuring Results

In Appendix B. you will find some measuring results of Selsius/Cisco phone and NetMeeting. We measured the packet spacing difference and jitter without any load (Dummynet parameters: bandwidth=10 Mbit/s, packet loss = 0 and delay = 0 ms) and then with load by altering the parameters separately.

5.3.1 Measurements without load

Without load voice quality of Selsius/Cisco phone (Fig. 4 in Appendix B.) was good and there was no noticeable delay either. The clipping of voice when not talking caused the only inconvenience. But it was faded out if there was some background noise. But voice quality was lower than SCN quality. The summary of statistics is show in Table 4.

Voice quality of NetMeeting (Fig.5 in Appendix B) on the other hand was considerably worse compared with Selsius/Cisco phone. There was clearly noticeable delay and the tone of the voice was softer. The graphs of packet spacing difference and jitter are quite different. The graph of packet spacing difference has two peaks. It is due to variance in delay and packet loss.

Table 4. Statistic of D and J without load

	Selsius	**Selsius**	**NetM.**	**NetM.**
	D [ms]	J [ms]	D [ms]	J [ms]
Average	0,0156	0,3986	0,021	20,026
St.Dev.	0,6694	0,1775	21,041	1,9179
Var.	0,4480	0,0315	442,703	3,6785

5.3.2 Measurements with load

Table 5 shows the effects of packet loss on voice quality. See also figures 6 and 7 in Appendix C.

Table 5. Effects of introducing packet loss.

Packet loss	**Selsius/Cisco**	**NetMeeting**
20%	Small crackings	Small crackings
25 %	Gaps in speech	Gaps in speech
30 %	It took a few seconds to connect, more gaps	Big gaps in speech
35 %	Severe gaps in speech, the connection was cut	Speech difficult to understand

As can be seen from the figures Selsius phone seems to be more robust than NetMeeting. The shape of the packet spacing difference curve of Selsius phone remains the same even when load was applied. Where as the same curve of NetMeeting has changed considerably.

Table 6 shows how decreasing bandwidth effects on voice quality. See also figures 8 and 9 in Appendix D.

Table 6. Effects of changing bandwidth.

Bandwidth	**Selsius/Cisco**	**NetMeeting**
80 bit/s	Noticeable decrease in quality	Noticeable decrease in quality
60 kbit/s	Difficult to understand	Difficult to understand
50 kbit/s	Nearly impossible to understand	More difficult to understand
20 kbit/s	----	Nearly impossible to understand

References

[1] ETSI, Telecommunications and Internet Protocol Harmonization Over Network (TIPHON); End to end Quality of Service in TIPHON Systems; Part 1: General Aspects of Quality of Service (QoS), France, 2000 (TR 101 329-1, V 3.1.1 (2000-07)).

[2] Selin, J.: Media Management in IP Telephony Systems, Master Thesis of the Networking Laboratory, Helsinki University of Technology, Espoo, Feb. 2001.

[3] Hersent O et al, IP Telephony: Packet-based multimedia communications systems, Great Britain, 2000, ISBN 0-201-61910-5.

[4] ITU-T Recommendation G.107, The E model, a computational model for use in transmission planning, 2000.

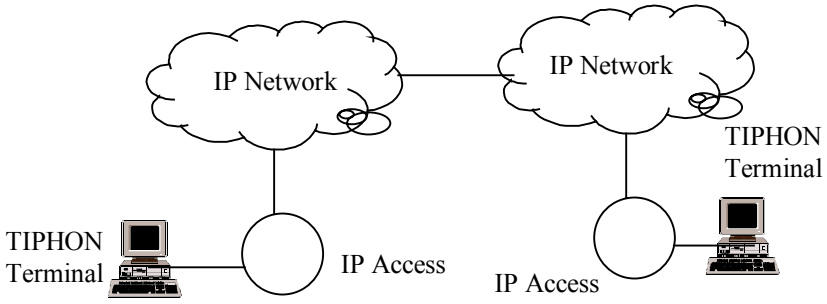
[5] Janssen J et al, Delay and Distortion Bounds for Packetized Voice Calls of Traditional PSTN Quality, Proceedings of the 1st IP-Telephony Workshop (IPTel 2000), Berlin, 2000.

[6] ETSI Telecommunications and Internet Protocol Harmonization Over Network (TIPHON); TIPHON; End to end Quality of Service in TIPHON Systems; Part 2: Definition of Quality of Service (QoS) Classes, France, 2000 (TR 101329-2, V 1.1.1 (2000-07)).

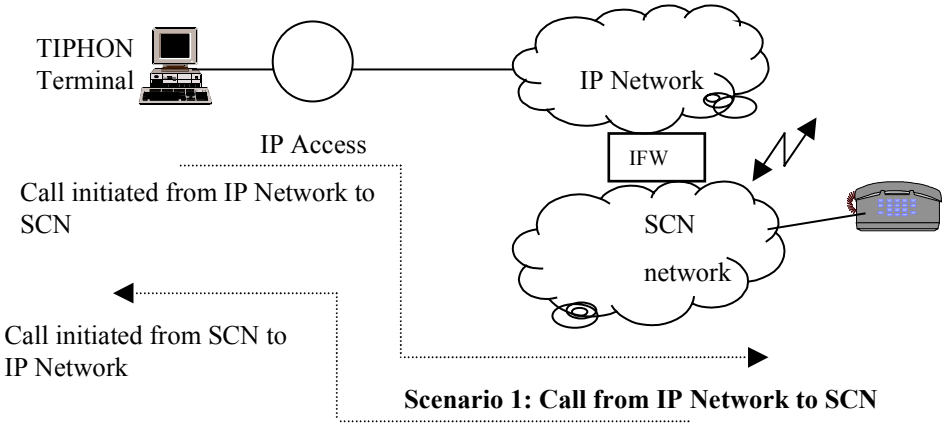
[7] ITU-T Recommendation G.113, Transmission Impairments, 1996.

[8] Yletyinen, T.: The Quality of Voice over IP, Master Thesis of the Laboratory of Telecommunication Technology, Helsinki University of Technology, March 1998.

Appendix A. Scenarios of End-to-End Voice Call by ETSI/TIPHON [1]

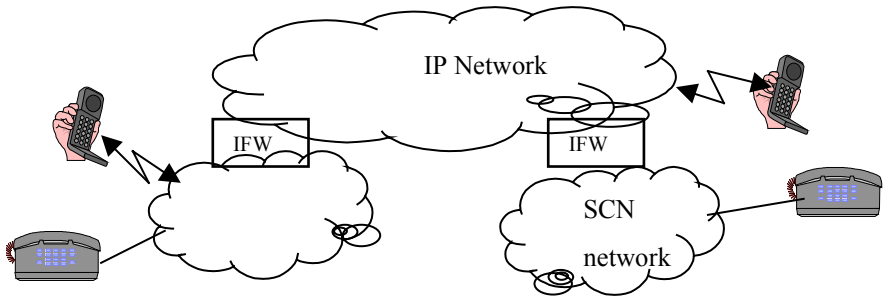


Scenario 0: Call from IP network to IP network



Scenario 1: Call from IP Network to SCN

Scenario 2: Call from SCN to IP Network



Scenario 3: Call from SCN to SCN over IP Network

Appendix B. Measuring Packet Spacing Difference and Jitter on Selsius/Cisco IP phone and NetMeeting program with no load

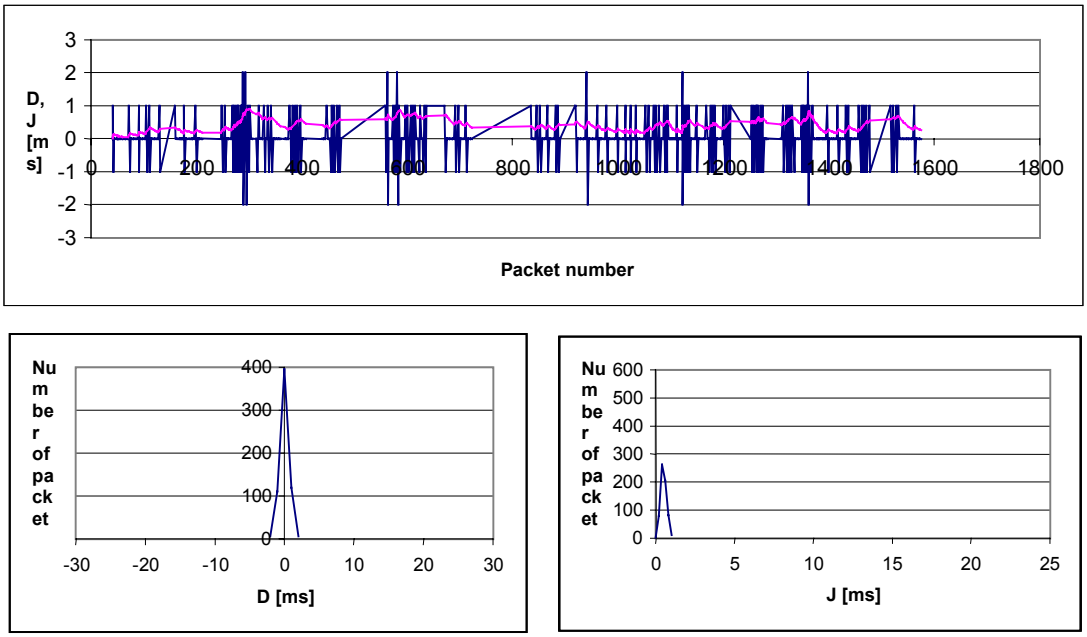


Figure 4. Selsius/Cisco IP phone with no restrictions (Bandwidth = 10 Mbit/s, Delay = 0 ms, Packet loss = 0%)

Upper picture shows the measured D and J as the packets were captured, down left is the histogram of D [ms] and down right is the histogram of J [ms].

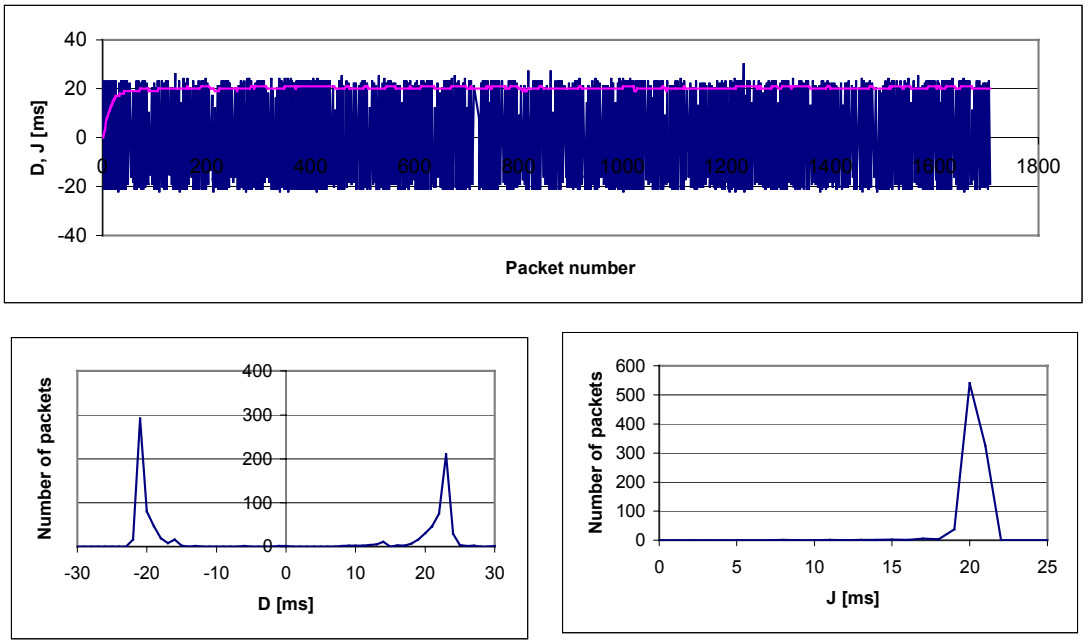


Figure 5. NetMeeting program with no restrictions (Bandwidth = 10 Mbit/s, Delay = 0 ms, Packet loss = 0%)

Upper picture shows the measured D and J as the packets were captured, down left is the histogram of D [ms], and down right is the histogram of J [ms].

Appendix C. Measuring Packet Spacing Difference and Jitter on Selsius/Cisco IP phone and NetMeeting program with packet loss 25 %

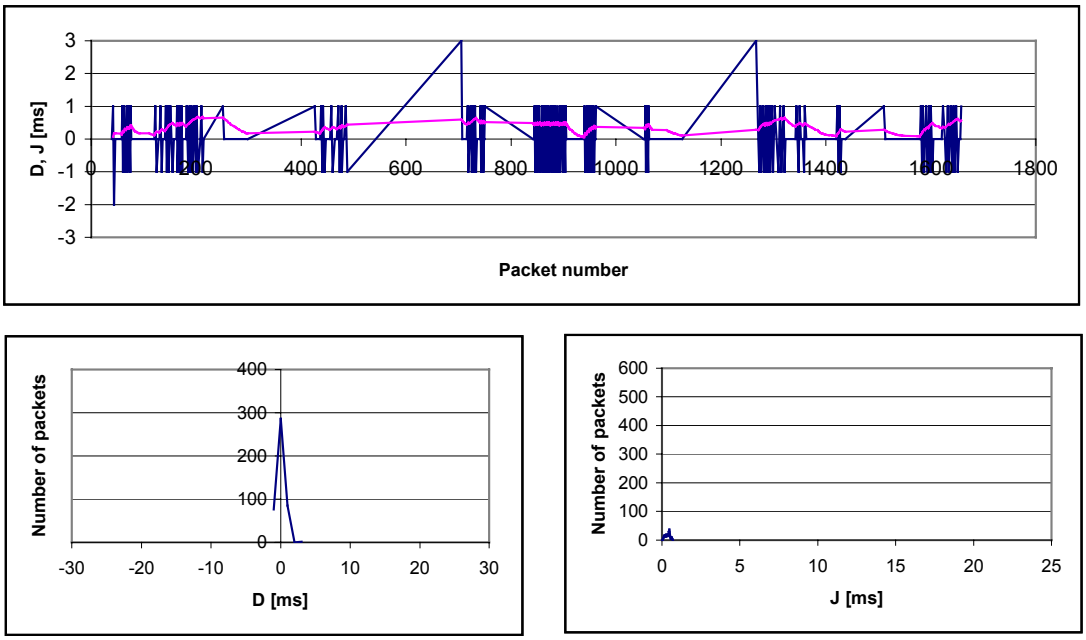


Figure 6. Selsius/Cisco IP phone with Packet loss = 25 % (Bandwidth = 10 Mbit/s, Delay = 0 ms)

Upper picture shows the measured D and J as the packets were captured, down left is the histogram of D [ms], and down right is the histogram of J [ms].

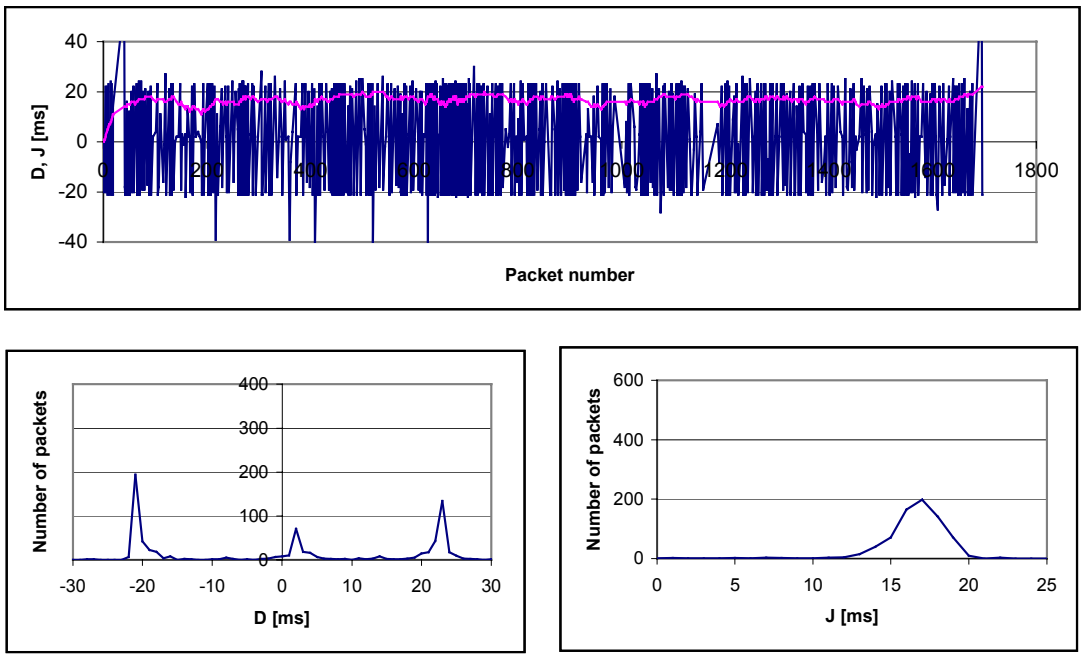


Figure 7. NetMeeting program with Packet loss = 25 % (Bandwidth = 10 Mbit/s, Delay = 0 ms)

Upper picture shows the measured D and J as the packets were captured, down left is the histogram of D [ms], and down right is the histogram of J [ms].

Appendix D. Measuring Packet Spacing Difference and Jitter on Selsius/Cisco IP phone and NetMeeting program with bandwidth 80 kbit/s

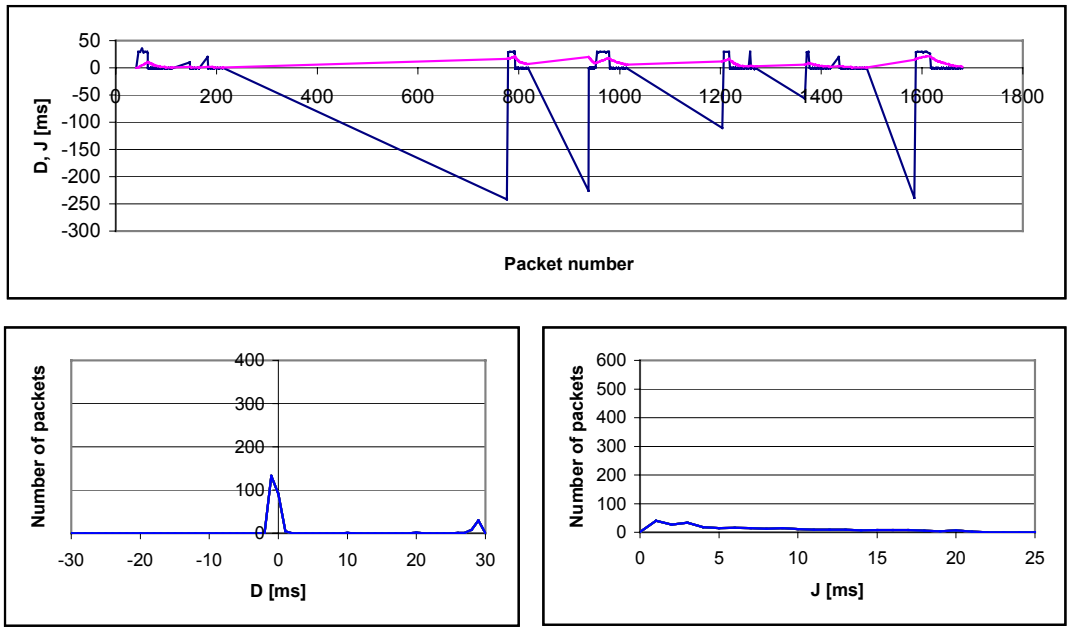


Figure 8. Selsius/Cisco IP phone with Bandwidth = 80 kbit/s (Delay = 0 ms, Packet loss = 0 %)

Upper picture shows the measured D and J as the packets were captured, down left is the histogram of D [ms], and down right is the histogram of J [ms].

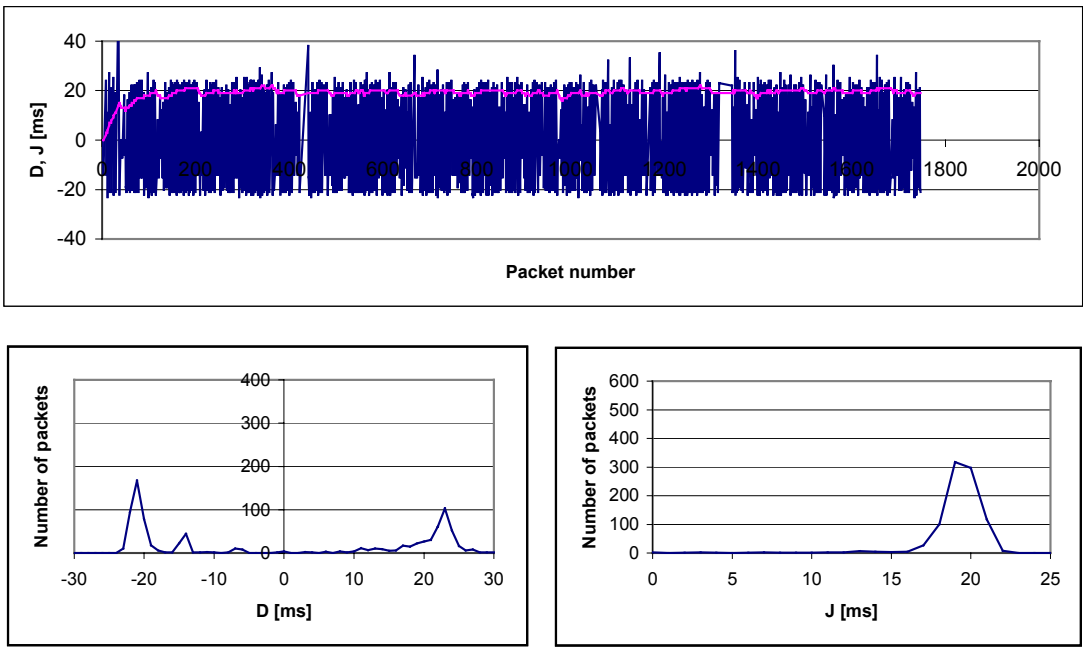


Figure 9. NetMeeting program with Bandwidth = 80 kbit/s (Delay=0 ms, Packet loss=0%)

Upper picture shows the measured D and J as the packets were captured, bottom left is the histogram of D [ms], and bottom right is the histogram of J [ms].

Voice in Packets: RTP, RTCP, Header Compression, Playout Algorithms, Terminal Requirements and Implementations

Jani Lakkakorpi
Nokia Research Center
P.O. Box 407
FIN-00045 NOKIA GROUP
Finland
jani.lakkakorpi@nokia.com

Abstract

RTP/RTCP protocol suite provides the means for sending packetized voice by introducing time stamps and sequence numbers in packet headers. Playout buffering is needed to re-synchronize the received voice stream. In this paper, a new adaptive playout delay adjustment algorithm is introduced.

A major problem, especially on low-bandwidth links, with Voice over IP (VoIP) packets is that they include a lot of overhead. The solution is header compression, which is done on link-by-link basis.

All terminals that support real time interactive voice should have enough processing power. The computational requirements of voice codecs usually increase with the voice compression ratio.

1 Introduction

During an average telephone conversation, each party usually talks only 35 per cent of the time. Most of the techniques that are used today to transform voice into data have the ability to detect silent periods. This allows asynchronous voice transmission and statistical multiplexing [Her00].

With statistical multiplexing, bandwidth can be used more efficiently. However, statistical multiplexing also introduces some uncertainty in the network. This uncertainty is variation in delay – also known as jitter. It needs to be corrected by the receiving side by adding some playout delay in order to restore the original packet spacing. Otherwise the original speech would sound incomprehensible.

Real-Time Transport Protocol (RTP) gives us the means to re-synchronize a voice stream. Each RTP packet is equipped with a timestamp and a sequence number. Chapter 2 describes RTP and its companion protocol, Real-Time Control Protocol (RTCP). Chapter 3 addresses the problem of RTP header compression – the overhead can be really large, as we will see. Chapter 4 introduces some playout algorithms developed for packetized voice, and chapter 5 gives a short overview about the terminal requirements and implementations for Voice over IP.

2 RTP and RTCP

Real-Time Transport Protocol (RTP) provides end-to-end transport functions suitable for applications that transmit real time data, such as audio or video over multicast or unicast networks. RTP does not provide any Quality of Service (QoS) guarantees but it is only responsible of synchronizing the received packets – within a single stream or across two streams. To achieve this, RTP packets are equipped with timestamps and sequence numbers.

RTP is designed to work together with Real-Time Control Protocol (RTCP) to get feedback on the quality of data transmission and information about participants in the on-going session. Request For Comments (RFC) 1889 (and its later revisions) includes complete descriptions of these protocols and their uses [Sch96a].

2.1 RTP

The first twelve octets in an RTP header (the first three rows in Figure 1) are included in all RTP packets, while the list of Contributing Source identifiers (CSRC) is present only when inserted by a mixer (see Section 2.2).

- **Version (V, 2 bits)**

  This field identifies the RTP version. The current version, defined in RFC 1889, is two.

- **Padding (P, 1 bit)**

  If padding bit is set, packet contains one or more padding octets at the end of the payload. The last octet of the payload contains the number of padding octets.

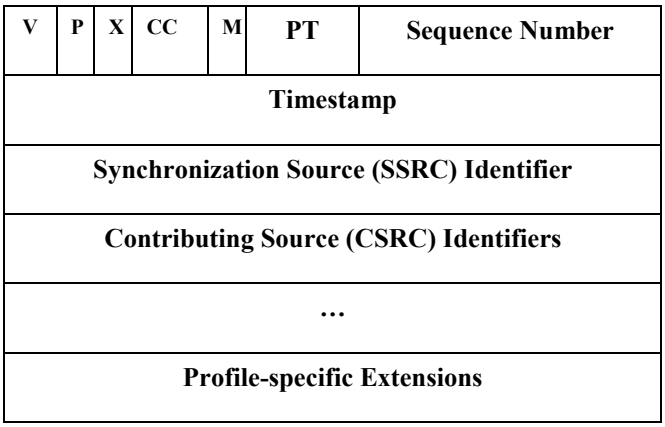


Figure 1. RTP Header Format

- **Extension (X, 1 bit)**

  If extension bit is set, the fixed RTP header and possible CSRCs are followed by extensions that use the format defined in RFC 1889.

- **CSRC count (CC, 4 bits)**

  CSRC count contains the number of Contributing Source identifiers that follow the fixed header. This number is usually zero [Her00].

- **Marker (M, 1 bit)**

  The interpretation of marker bit is defined by RTP profile. The marker bit is intended for marking significant events, such as frame boundaries, in the packet stream. H.225.0, for example, says that for audio codings supporting silence suppression, the marker bit must be set to one in the first packet of each talkspurt after a silence period [Her00].

- **Payload Type (PT, 7 bits)**

  This field identifies the format of the RTP payload, and determines its interpretation by the application. A profile specifies default static mapping of payload type codes to payload formats. An initial set of default mappings for audio and video is specified in RFC 1889 [Sch96b].

- **Sequence Number (16 bits)**

  Sequence number starts from a random value and it is incremented by one for each RTP packet sent. The sequence number is used by the receiver to detect packet losses and to restore packet sequence.

- **Timestamp (32 bits)**

  Timestamp reflects the sampling instant of the first payload octet. The clock frequency is defined for each payload type, and the clock is initialized with a random value [Her00].

- **SSRC (32 bits)**

  SSRC field identifies the synchronization source. This identifier is chosen randomly, with the intent that no two synchronization sources within the same RTP session have the same SSRC identifier.

- **CSRC list (0 to 15 items, 32 bits each)**

  CSRC list identifies the contributing sources for the payload contained in this packet. The number of identifiers is given by the CC field. Only 15 sources can be identified. CSRC identifiers are inserted by mixers, using the SSRC identifiers of the contributing sources.

2.2 Mixers and Translators

RTP-level relays called mixers re-synchronize incoming packets in order to reconstruct the packet spacing generated by the sender, mix these reconstructed streams into a single stream, possibly translate the encoding, and then forward the packet stream. These packets might be unicast to a single recipient or multicast to multiple recipients. RTP header includes the means for mixers to identify the sources that contributed to a mixed packet so that correct talker indication, for example, can be provided at the receiver.

Another type of RTP-level relay, translator, just takes a stream and passes it through. Translator can be used, for example, in a situation where the receiver is beyond a firewall.

2.3 RTCP

Real-Time Control Protocol (RTCP) is based on the periodic transmission of control packets to all participants in the session. The underlying protocol must provide multiplexing of data and control packets, for example, by using separate port numbers with UDP (User Datagram Protocol). RTP is usually assigned an even UDP port, and RTCP the next odd UDP port. RTCP performs three mandatory functions:

1. The primary function is to provide feedback on the quality of the data distribution. This function is performed through sender and receiver reports.

2. RTCP carries a persistent transport-level identifier for an RTP source, called Canonical Name (CNAME). Since the SSRC identifier may change, all receivers require the CNAME to keep track of each participant.

3. Since the first two functions require that all participants in a session send RTCP packets, the RTCP packet rate must be controlled in order to scale up to a large number of participants. Each participant can independently observe the number of other participants and thus control its RTCP packet rate. The maximum rate at which a participant can send RTCP reports is one per five seconds.

It is recommended that translators and mixers combine individual RTCP packets from multiple sources into compound packets whenever feasible.

RFC 1889 defines several RTCP packet types to carry control information:

- **SR (Sender Report)** contains transmission and reception statistics for active senders

- **RR (Receiver Report)** contains reception statistics for participants that are not active senders

- **SDES (Source Description Items)** describes various parameters about the source, including the CNAME

- **BYE** packet is sent by a participant when leaving the session

- **APP**: application specific functions

Each RTCP packet begins with a fixed part, followed by structured elements that may be of variable length according to the packet type, but always end on a 32-bit boundary.

2.3.1 Sender and Receiver Reports

RTP receivers provide reception quality feedback using RTCP report packets of two types. The only difference between sender reports (SR) and receiver reports (RR), besides the packet type code, is that the sender report includes a 20-byte sender information section (highlighted in Figure 2). Sender report is issued if the participant has sent at least one RTP packet during the last report period – otherwise a receiver report is issued.

Both the sender report and the receiver report include reception report blocks, one for each of the synchronization sources from which this participant has received RTP data packets since the last report. Reports are not issued for contributing sources listed in the CSRC list. Each reception report block provides statistics about the data received from the particular source indicated in that block.

<table>
<tr><td>V</td><td>P</td><td>RC</td><td>PT=SR=200</td><td>Length</td></tr>
<tr><td colspan="5">SSRC of Sender</td></tr>
<tr><td colspan="5">NTP Timestamp, Most Significant Word</td></tr>
<tr><td colspan="5">NTP Timestamp, Least Significant Word</td></tr>
<tr><td colspan="5">RTP Timestamp</td></tr>
<tr><td colspan="5">Sender's Packet Count</td></tr>
<tr><td colspan="5">Sender's Octet Count</td></tr>
<tr><td colspan="5">SSRC_n (SSRC of nth Source)</td></tr>
<tr><td colspan="2">Fraction Lost</td><td colspan="3">Cumulative Number of Packets Lost</td></tr>
<tr><td colspan="5">Extended Highest Sequence Number Received</td></tr>
<tr><td colspan="5">Interarrival Jitter</td></tr>
<tr><td colspan="5">Last SR (LSR)</td></tr>
<tr><td colspan="5">Delay Since Last SR (DLSR)</td></tr>
<tr><td colspan="5">…</td></tr>
<tr><td colspan="5">Profile-specific Extensions</td></tr>
</table>

Figure 2. RTCP Sender Report

The first section of the sender report, called header, is 8 octets long.

- **Version (V, 2 bits)**

  Identifies the current version (which is the same for RTCP and RTP packets). The version defined in RFC 1889 is two.

- **Padding (P, 1 bit)**

  If the padding bit is set, this RTCP packet contains some additional padding octets at the end which are not part of the control information. The last

octet of the packet contains the number of these padding octets.

- **Reception Report Count (RC, 5 bits)**

  The number of reception report blocks contained in this packet. A value of zero is valid.

- **Packet Type (PT, 8 bits)**

  Contains the constant 200 to identify this packet as an RTCP sender report.

- **Length (16 bits)**

  Length of this RTCP packet in 32-bit words subtracted by one. (Includes the header and any padding.)

- **SSRC (32 bits)**

  Synchronization source identifier for the originator of this sender report.

The second section, sender information, is 20 octets (five rows in Figure 2) long and it is present in all sender reports. It summarizes the data transmissions from this sender.

- **NTP Timestamp (64 bits)**

  Indicates the wallclock time when this report was sent.

- **RTP Timestamp (32 bits)**

  Corresponds to the same time as the NTP timestamp, but in the same units, and with the same random offset as the RTP timestamps in data packets. This correspondence may be used for intra- and inter-media synchronization for sources whose NTP timestamps are synchronized.

- **Sender's Packet Count (32 bits)**

  The total number of RTP data packets transmitted by the sender since starting transmission up until the time this sender report was generated. The count is reset if the sender changes its SSRC identifier.

- **Sender's Octet Count (32 bits)**

  The total number of payload octets (not including header or padding) transmitted in RTP data packets by the sender since the start of transmission up until the time this sender report was generated. The count is reset if the sender changes its SSRC identifier. This field can be used to estimate the average payload data rate.

The third section contains reception report blocks. The amount of these blocks depends on the number of other sources that this sender has been listening to since last report.

- **SSRC_n (Source Identifier, 32 bits)**

  The SSRC identifier of the source that we are reporting about.

- **Fraction Lost (8 bits)**

  The fraction of RTP data packets from source SSRC_n that were lost since the previous sender or receiver report was sent. If the loss is negative due to duplicates, the fraction lost is set to zero.

- **Cumulative Number of Packets Lost (24 bits)**

  The total number of lost packets from source SSRC_n since the beginning of reception. This figure is defined to be the number of packets expected subtracted by the number of packets actually received. The number of packets received also includes late and duplicate packets. Thus packets that arrive late are not counted as lost, and the loss may be negative if there are duplicates. The number of packets expected is defined to be the last extended highest sequence number received subtracted by the initial sequence number received. This may be calculated as shown in RFC 1889 [Sch96a].

- **Extended Highest Sequence Number Received (32 bits)**

  The low 16 bits contain the highest sequence number received in an RTP packet from source SSRC_n, and the most significant 16 bits extend that sequence number with the corresponding count of sequence number cycles.

- **Interarrival Jitter (32 bits)**

  An estimation of the variance of the RTP packet interarrival time measured in timestamp units and expressed as an unsigned integer.

  Interarrival jitter can be calculated as a difference in the relative transit time for two packets. The relative transit time is the difference between the packet's RTP timestamp and the receiver's clock at the time of arrival, measured in same units. If S_i is the RTP timestamp of packet i and R_i is the time of arrival of packet i (in RTP timestamp units), the difference in packet spacing for the two packets, i and j, can be expressed as:

$$D_{(i,j)} = (R_j - R_i) - (S_j - S_i).$$

Interarrival jitter is updated each time when a packet is received from source SSRC_n (using the difference in packet spacing for that packet and the previous packet) according to the following formula:

$$J = J + (|D_{(i-1,i)}| - J)/16.$$

When the reception report is issued, the current value of J is sampled.

- **Last SR Timestamp (LSR, 32 bits)**

  The middle 32 bits of the NTP timestamp of the most recent RTCP sender report from source SSRC_n. If no sender report has been received yet, the field is set to zero.

- **Delay Since Last SR (DLSR, 32 bits)**

  The delay (expressed in NTP form: units of $1/65536$ seconds) since the last sender report arrived. Together with the last SR timestamp, the sender of this last SR can use it to compute the round trip time. If no SR packet has been received yet from SSRC_n, the DLSR field is set to zero.

2.3.2 Receiver Report RTCP Packet

Receiver report (RR) shares the same format with the sender report except that the packet type field contains the constant 201, and the five words of sender information are omitted (these are the NTP & RTP timestamps and sender's packet & octet counts).

3 Header Compression

The high overhead of RTP/UDP/IP packets is a challenging issue, especially on slow links [Her00]. For example, a popular video-conferencing application, Microsoft NetMeeting [Net01], uses voice codec G.723.1, where a frame of 24 bytes is sent every 30 ms. This will produce a data rate of 6.3 kbit/s. Since RTP/UDP/IP headers add at least 40 bytes of overhead, and the link layer some bytes as well (PPP+HDLC add four bytes), the resulting bit rate will be over 18 kbit/s.

A common trade-off for reducing the bit rate is to put several frames in a single packet. However, this can set the conversational delay to a level that is far too high for most users. The overhead issue can be solved also by using header compression [Cas99].

Header compression is based on the simple idea that since most of the data packet overhead is constant for a given stream, it is possible to negotiate a shorter index for those constants (e.g. source and destination IP addresses and ports) when the stream is set up [Her00]. Other (variable) values can be reconstructed at the receiving end. To put it short: the sending host replaces the large RTP/UDP/IP header to a small index, and the receiving host reverses this operation. An RTP/UDP/IP header compression mechanism for low-speed links is described in RFC 2508. In many cases, all three headers can be compressed to 2-4 bytes. The compression is done on a link-by-link basis [Cas99].

3.1 RFC 2508

The compression algorithm proposed in RFC 2508 draws heavily upon the design of TCP/IP header compression, which is described in RFC 1144 [Jac90].

3.1.1 The Basic Idea

In TCP/IP header compression, it has been observed that about half of the header bytes remain constant over the duration of the connection. After the header has been sent uncompressed once, the constant fields can be excluded from the following compressed headers. Headers can be further compressed with the help of differential coding on the changing fields.

In RTP header compression, some of the aforementioned techniques can be applied, but the major gain comes from the fact that although several fields change in every packet, the difference from packet to packet is often constant. If the compressor and decompressor maintain both the uncompressed header and the first-order differences, the only information that must be conveyed is an indication that the second-order difference was zero. If that is the case, the decompressor can reconstruct the original header without any loss of information by adding the first-order differences to the uncompressed header as each compressed packet is received.

3.1.2 Header Compression for RTP/UDP/IP Packets

In IPv4 header, only the total length, packet ID, and header check-sum fields typically change. The total length can be excluded, because it is provided by the link layer. Since the RFC 2508 compression scheme depends upon the link layer to provide good error detection, the header checksum may also be excluded.

In order to maintain lossless compression, changes in the packet ID are transmitted. The packet ID is usually incremented by one for each packet. In IPv6 base header, neither packet ID nor header checksum exist, and only the payload length field changes.

In UDP header, the length field is redundant with the IP total length field and the length indicated by the link layer. UDP checksum field will be zero in the case, where source does not generate any UDP checksums. Otherwise, the checksum must be sent intact in order to preserve lossless compression.

In most RTP headers, only the sequence number and timestamp change from packet to packet. If packets are not lost or misordered, the sequence number is incremented by one for each packet. For audio packets of constant duration, the timestamp is incremented by the number of sample periods conveyed in each packet.

If the second-order differences of the sequence number and timestamp fields are zero, the next packet header can be constructed from the previous header by adding the first-order differences (that are stored in the session context along with the uncompressed header) for these fields.

The marker bit is set on the first packet of an audio talkspurt. If it were treated as a constant field, such that each change would require sending the full RTP header, the compression would become quite inefficient. Because of this, one bit in the compressed header is reserved for the marker bit.

3.2 Other Proposals

There exist a number of other RTP/UDP/IP header compression mechanisms that have emerged after RFC 2508. They all should perform slightly better than the mechanism described in the RFC 2508. Some of the most recent proposals are Ericsson's ROCCO [Lar00] and Nokia's ACE [Khi00].

4 Playout Algorithms

4.1 Playout Delay

In most packet audio applications, packets are buffered at the receiving host in order to compensate for variable network delay. The receiver buffer sizes can be constant or adaptively adjusted. Keeping the delay as small as possible, and avoiding excessive packet losses at the same time is not an easy task. The results of [Ram94] indicate that an adaptive algorithm, which explicitly adjusts to the sharp, spike-like increases in packet delay, can achieve a lower rate of lost packets.

Adaptive playout delay can be either per-talkspurt or per-packet based. In the former approach, playout delay remains constant throughout the talkspurt and the adjustments are done between talkspurts. The latter approach introduces gaps in speech, and thus it is not recommended for VoIP [Yle97].

Three different playout delay adjustment algorithms for packetized voice are presented in [Moo98]. The paper is focused on the tradeoff between packet playout delay and packet playout loss. The authors present an adaptive delay adjustment algorithm that tracks the network delay of recently received packets and maintains delay percentile information.

Some playout delay adjustment algorithms assume that the sender and receiver clocks are synchronized, but in [Moo98] this is not the case. The propagation delay is removed from end-to-end delay by subtracting out the minimum of measured end-to-end delays. Thus it is possible to concentrate on the variable delay component.

In the following section, we present a similar, although somewhat simpler, algorithm for adaptive playout delay adjustment. This algorithm does not assume synchronization of the sender and receiver clocks.

Waiting time in playout buffer is calculated with the following algorithm:

All packets are played out at:

$PlayAt_i = ReceivedAt_i + Twait_{,i}$

For the first packet of the connection, playout delay is constant (given by the user):

$T_{wait,\,0} = t_{wait}$

For other packets, waiting time is calculated as follows:

$T_{wait,\,i} = (TStamp_i - TStamp_{i-1}) - (ReceivedAt_i - PlayAt_{i-1})$

If the result is negative, packet is discarded.

Whenever playout delay is adjusted, it will be the maximum of the initial playout delay and the current playout delay subtracted by the minimum T_{wait} of the latest measurement period.

The following events will trigger the playout delay adjustment:

- If N or more packets among the last M packets (measurement period) arrive late, playout delay is adjusted upwards when the next talkspurt arrives.
- Similarly, if M successive packets have been received all in time, playout delay is adjusted downwards before the next talkspurt.

Table 2 shows some simulation results for constant and adaptive playout delay. Network delay was simply modeled with exponential distribution with a mean of 30 milliseconds. Parameters used were: N = 2, M = 100, t_{wait} = 100 ms. Simulation duration was 200 seconds.

Table 2: Constant vs. adaptive playout delay

Playout delay	Packet loss ratio	Mean	Min.	Max.
Constant	3.7%	100 ms	100 ms	100 ms
Adaptive	1.2%	150 ms	100 ms	240 ms

Simulation results show that there is a clear tradeoff between playout delay and playout loss. If we had selected a larger playout delay in the constant playout delay case, packet loss ratio would have been smaller. If we the variation of network delay is unknown, it can be very hard to set the constant playout delay. Simulation results also show that upper bound for adaptive playout delay is probably needed, because end-to-end delays longer than 400 milliseconds are not acceptable for voice [ITU00].

Figure 3 illustrates the simulated sequence of sent, received and synchronized VoIP packets, while Figure 4 illustrates the changes in playout delay.

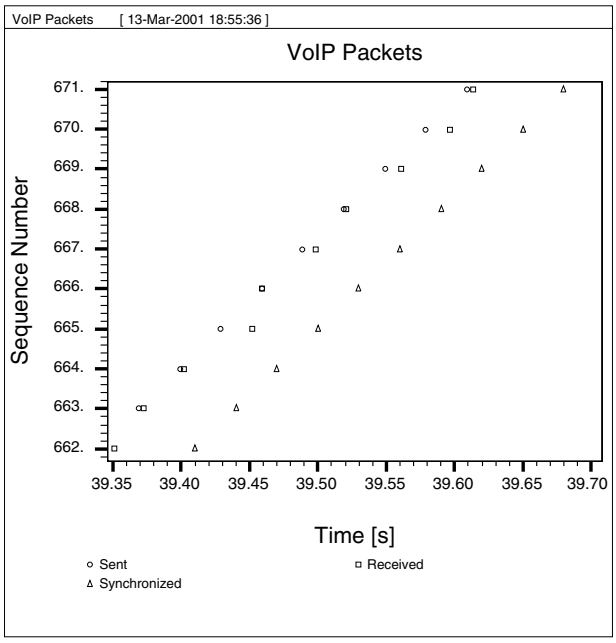


Figure 3. Sent, Received and Synchronized Packets

4.2 Synchronization Delay and Clock Drift

Audio device driver is the link between the operating system and the hardware. Implementation of the device driver is a crucial factor in audio device performance. The driver can namely introduce unnecessary delay in both directions. If the audio device requests data at fixed intervals, that are not synchronized with the reception of incoming packets, an additional delay of half of the audio block duration is introduced [Sel01].

Clock drift means that the sampling and playout rates of the audio devices do not match. If the clocks at each end drift in different directions, buffer underruns at one end, and increased delay is experienced at the other end. Buffer underrun occurs when the receiver does not have anything to play. Modifying the playout rate by adding or dropping samples before the frame is transferred to the audio device can compensate for clock drift [Sel01].

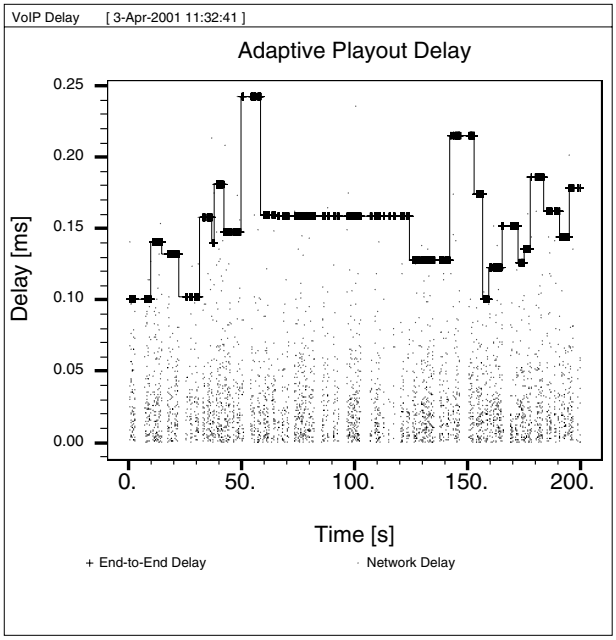


Figure 4. Adaptive Playout Delay

5 VoIP Terminal Requirements and Commercial Implementations

According to [Kos98], PCs (or other terminals) that support real time interactive voice must have considerable processing power. The computational requirements of voice codecs increase with the voice compression ratio.

Microsoft NetMeeting [Net01] is a popular remote conferencing tool. The hardware requirements that have to be met in order to use the data, audio, and video features of NetMeeting are as follows:

- For Windows 95 or Windows 98: Pentium 90 processor with 16 MB of RAM
- For Windows NT: a Pentium 90 processor with 24 MB of RAM
- 56,000 bps or faster Internet connection
- Sound card with microphone and speakers
- Video capture card or camera that provides a Video for Windows capture driver

VocalTec Internet Phone Lite is mainly targeted for voice only connections. It has the following system requirements [Voc01]:

- Windows 95 or Windows NT 4.0 or higher
- Pentium 75 processor or higher
- 14,400 bps or faster Internet connection
- Sound card with microphone and speakers

6 Conclusions

RTP/RTCP protocol suite provides the means for real-time communication over IP by introducing time stamps and sequence numbers.

The amount of overhead in VoIP packets is a serious issue, especially on low-bandwidth links. This problem can be alleviated by header compression, which is done on link-by-link basis.

In order to make transmitted voice stream human understandable, playout buffering is needed to re-synchronize the stream. A new adaptive playout delay adjustment algorithm has been introduced. The operation of the algorithm is basically that if certain criteria are met, the delay between the sender and the receiver can be adaptively adjusted upwards or downwards.

Terminals that support real time interactive voice must have considerable processing power. The computational requirements of voice codecs increase with the voice compression ratio.

References

[Cas99] S. Casner, V. Jacobson: Compressing IP/UDP/RTP Headers for Low-Speed Serial Links (Request for Comments: 2508), February 1999.

[Her00] Olivier Hersent, David Gurle, Jean-Pierre Petit: IP Telephony, Packet based multimedia communications systems, Addison Wesley, 2000, ISBN 0-201-61910-5.

[ITU00] ITU-T Recommendation G.114 (05/00): One-way transmission time.

[Jac90] V. Jacobson: TCP/IP Compression for Low-Speed Serial Links (Request for Comments: 1144), February 1990.

[Khi00] Khiem Le, Christopher Clanton, Zhigang Liu, Haihong Zheng: Adaptive Header ComprEssion (ACE) for Real-Time Multimedia (Internet Draft, Expired 24.11.2000), 24.5.2000.

[Kos98] T. J. Kostas, M. S. Borella, I. Sidhu, G. M. Schuster, J. Grabiec, J. Mahler: Real-time Voice over Packet-switched Networks, IEEE Network, Volume: 12, Issue: 1, 1998.

[Lar00] Lars-Åke Larzon, Hans Hannu, Lars-Erik Jonsson, Krister Svanbro: Efficient Transport of Voice over 1P over Cellular links, Proceedings of IEEE Globecom 2000.

[Moo98] Sue B. Moon, Jim Kurose, and Don Towsley: Packet Audio Playout Delay Adjustment: Performance Bounds and Algorithms, ACM/Springer Multimedia Systems, Vol. 6, pp. 17-28, January 1998.

[Net01] Microsoft Netmeeting: http://www.microsoft.com/windows/netmeeting/, 12.2.2001.

[Ram94] Ramachandran Ramjee, Jim Kurose, Don Towsley Henning Schulzrinne: Adaptive Playout Mechanisms for Packetized Audio Applications in Wide-Area Networks, Proceedings of IEEE Infocom '94, Montreal, Canada, April 1994.

[Sch96a] H. Schulzrinne, S. Casner, R. Frederick, V. Jacobson: RTP: A Transport Protocol for Real-Time Applications (Request for Comments: 1889), January 1996.

[Sch96b] H. Schulzrinne: RTP Profile for Audio and Video Conferences with Minimal Control (Request for Comments: 1890), January 1996.

[Sel01] Jari Selin: Media Management in IP Telephony Systems, Master's Thesis, Helsinki University of Technology, Networking Laboratory, February 2001.

[Voc01] VocalTec Internet Phone Lite: http://www.vocaltec.com/iptelephony/products/iplite_vea_sp/iplite_overview.htm, 12.2.2001.

[Yle97] Tomi Yletyinen: An Introduction to Protocols for Real-Time Communications in Packet Switched Networks, Laboratory of Telecommunications Technology, Helsinki University of Technology, 1997.

Voice Coding in 3G Networks

Tommi Koistinen
Signal Processing Systems
Nokia Networks
Tommi.Koistinen@nokia.com

Abstract

The 3G networks will introduce several new additions to the basic speech service. The adaptive wideband speech codec will enhance the naturalness of speech and the transcoder free operation will remove unnecessary encodings that otherwise would degrade the speech quality. The speech processing on network side in 3GPP reference architecture model is focused around two network elements, namely the Media Gateway (MG), and the Media Resource Functions (MRF) unit. However, as the speech applications utilize the network more or less in transparent end-to-end mode the characteristics and speech enhancement capabilities of mobile terminals will finally determine the perceived overall speech quality.

1 Introduction

Voice compression techniques have been utilized in digital telecommunication networks for decades (G.711 standard [1] dates back to 1972). The G.711 standard presents a coding technique that operates at rate of 64 kbit/s and is widely used in all digital switched telephone networks. But where does the exact rate of 64 kbit/s come ?

The most essential frequency range for the human speech production system (that is the glottis and the vocal tract) and for the auditory system happens to be between 300-3400 Hz. As the sampling theorem says; to reproduce the original signal after sampling we must use a sampling rate that is double the desired frequency band. If sampling rate is less the reproduced signal will be distorted by image frequencies of the original signal. Speech in telecommunication networks is commonly sampled at 8 kHz to obey this law.

The number of bits per sample that is used to quantize the analog signal is a compromise between the quantisation noise that is introduced and the quality of the original signal. If the input signal is already band limited to 300-3400 Hz there is no point of using 24-bit converter. Commonly 13 bits per sample is seen to be a practical value for restricted voice band quantisation. The uniform quantisation however is not the most efficient quantisation method.

The main idea behind the G.711 standard is to use a logarithmic quantizer which results the same signal-to-noise ratio (SNR) with only 8 bits per sample compared to original 13 bits per sample.

This is achieved by allocating more quantisation steps to lower amplitude levels that in fact are the most important to perceived overall speech quality. The drawback is that the logarithmic scale will result a reduced SNR in the area of high-powered input signals but happily the effect of this is insignificant with speech signals.

As a result we can multiply 8000 samples per second (that came from the sampling theorem) with 8 bits per sample (that resulted from the logarithmic quantisation) to get the final bit stream of 64 kbit/s.

The compression ratio of G.711 standard can be seen to be 1.625:1 (13:8). And all compression is usually good. To transfer more telephone calls with less transmission equipment means money for the operator and this has resulted that several more advanced compression techniques have been developed.

Speech coding techniques in general can be separated to waveform coders (e.g. G.711, G.726, G.722) and to analysis-by-synthesis type of coders (e.g. G.723, G.729, GSM FR). The waveform coders operate in time domain and they are based on sample-by-sample approach that utilizes the correlation between speech samples. Analysis-by-synthesis types of coders try to imitate the human speech production system by a simplified model of a source (glottis) and a filter (vocal tract) that shapes the output speech spectrum on frame basis (typically frame size of 10-30 ms is used). A short introduction to details of both basic techniques (and their intermediate

versions; hybrids) is presented in [2] on pages 270-287.

The waveform coders are mainly used to compress speech on transmission links, for example, on PCM trunks between two switching centers. The compression ratios range from 2:1 to 4:1 and quite high speech quality can be maintained.

The analysis-by-synthesis types of coders were mainly introduced together with digital mobile networks (GSM Full Rate codec [3] dates back to 1988). As frequency band in the radio interface between a mobile terminal and a base station is restricted (and regulated) compression techniques are a meaningful way to save money in that interface. A typical full rate channel (16 kbps) utilizes a compression rate of 4:1. A half rate channel (8 kbps) is half of that and it operates at compression rate of 8:1. Lossy compression has always some effects on speech quality and more compression means usually less quality. The G.711 standard is common reference point for "real" speech codecs and e.g. GSM Enhanced Full Rate codec [4] almost reaches the quality of G.711.

The frame based handling that is natural to analysis-by-synthesis coders is also in line with the characteristics of packet based transmission techniques (IP, ATM) that are becoming quite common not only in core networks (or backbone) but also as building blocks of radio access networks.

This article will discuss the voice coding and user plane issues particularly in 3G networks. The first chapter presented the basic reasons and means for speech coding in general. The second chapter will review the basic 3G network architecture models. The most important 3G network elements that provide speech related processing are discussed in chapters four and five. The sixth chapter will discuss the issues related to tandeming of speech codecs and finally the seventh chapter will conclude the presentation.

2 Network Architectures

This chapter will present the basic 3G network evolution according to 3GPP (Third Generation Partnership Project [5]) reference architectures. 3GPP has scheduled its work to releases of R99, R4 and R5 and so on. In the following the basic reference architecture model of each release is shortly described emphasizing the voice coding and user plane issues.

Release 99

The basic architecture of R99 compatible network is shown in Figure 1. The IP packet data from UTRAN (Universal Terrestrial Radio Access Network, that is basically base stations and Radio Network Controllers (RNC)) goes through Iu-PS interface to 3G SGSN. Voice data goes through Iu-CS interface to 3G Mobile Switching Center (MSC) that converts the Adaptive Multirate (AMR) coded speech to G.711 format and vice versa for the PSTN network. The circuit switched speech is transferred in packet mode (ATM/AAL2) from UTRAN (from Radio Network Controller) to 3G MSC but the codec level packet mode speech is not yet originated from the terminal.

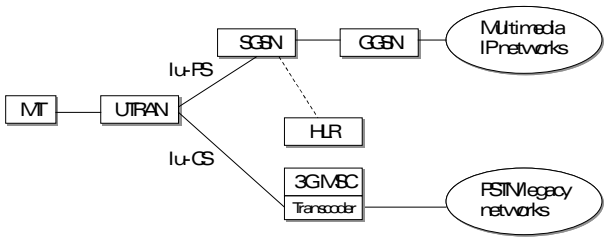


Figure 1. 3GPP Reference Architecture of Release 99.

Release R4

The next step that is taken with release 4 (formerly known as Release 2000) is to separate the signaling and the user data in Iu-CS interface. The signaling goes now to MSC Server and the transcoder is separated as a standalone media gateway. Figure 2 presents the R4 architecture with clear separation to packet side and to circuit switched side. Media gateway in the PSTN interface converts the AMR coded speech to G.711. Speech goes in packet mode from UTRAN to PSTN interface.

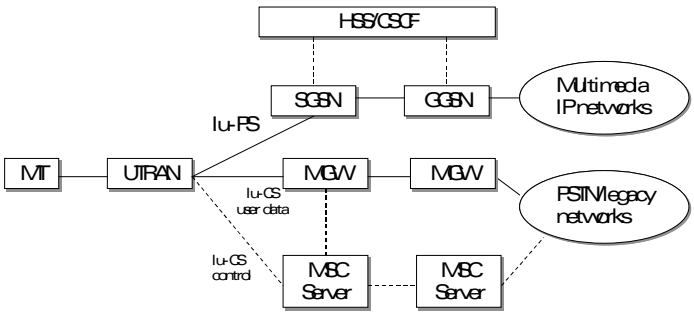


Figure 2. 3GPP Reference Architecture of Release 4.

The final architecture model, also called as All-IP network [6], moves also speech to full end-to-end

packet mode. The IP packets that are generated in a mobile terminal go as such either to another IP terminal or to MGW from GGSN. The architecture is presented in Figure 3. A new network entity is also introduced, namely the Multimedia Resource Functions (MRF) unit that implements mainly conferencing services for the IP based calls.

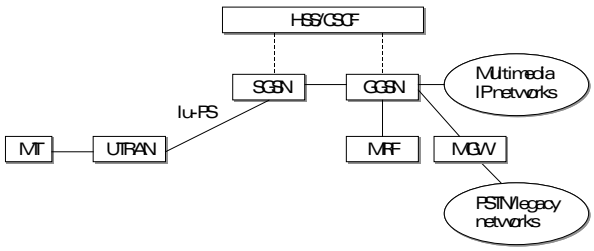


Figure 3. All-IP reference architecture.

Of course, the different phases of 3GPP releases may coexist at the same time depending on operators' needs.

3 Media Gateway

In 2G networks (like GSM) the speech related functionalities have been implemented around the transcoder unit (TRAU). The basic task of transcoder has been speech encoding and decoding of narrowband codecs like GSM Full Rate (FR), Enhanced Full Rate (EFR) or Half Rate (HR) codecs. Some extra features like noise cancellation or acoustic echo cancellation are also offered by 2G transcoders. The Mobile Switching Center has then additionally offered tone and DTMF generators, echo cancellers, fax and modem pools and announcement and conferencing services. Control mechanisms for these functionalities have usually been proprietary. In 3G networks, all of these functions must be offered by the Media Gateway that is controlled by the Media Gateway Controller (MGC) with the standard H.248 control protocol [7].

An example (and quite full) set of functions that Media Gateway could implement is:

- support for several interfaces (A-interface for 2G and Iu-interface for 3G) and for several transmission protocols (ATM, IP, TDM)
- support for several codecs including the Adaptive Multirate (AMR) codec and future coming wideband codecs
- electric and acoustic echo cancellation
- announcement services
- DTMF and call progress tone generation and detection
- support for fax/modem/data protocols
- support for Tandem Free Operation (TFO) and Transcoder Free Operation (TrFO)
- bad frame handling
- IP protocol handling (RTP/RTCP, encryption, QoS support)

Some functions, especially the conferencing service and possible speech enhancement services, are basically thought to be provided by the Multimedia Resource Functions (MRF) unit, but they may optionally be added to Media Gateway responsibilities.

A lot of signal processing (DSP) power is required to provide the Media Gateway's functions. Typically, one DSP chip may process 4-16 channels, and on one processor card there might be 8-32 DSPs which totals 32-512 channels per processor card.

4 Media Resource Functions

The Multimedia Resource Functions (MRF) unit according to 3GPP standard shall provide the audio/video conferencing services for the All-IP network. The basic requirement is to support several speech codecs to be able to sum up the conference for each party. As it is impossible for today's technology to sum up signals in parameter domain, all signals must be first decoded for linear domain processing. The summed signals are then encoded again for each party.

The 3GPP work on MRF entity has not progressed further than the conferencing requirement. However, the MRF entity is a natural place also for other speech enhancement services. It should be remembered that most of the calls in an All-IP network are staying inside the core network and they are not going to Media Gateway at all (see figure 4).

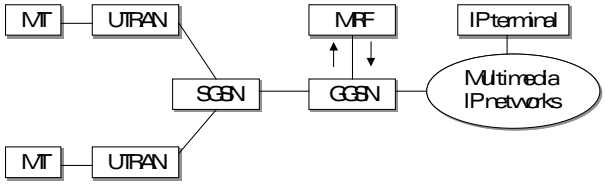


Figure 4. MRF unit as a network side speech enhancement server.

Calls between mobile IP terminals are transferred in coded format end-to-end and if any speech enhancement services are desired to be provided on the network side, the MRF entity could do the

necessary operations (as it already has to support all coding formats for the conferencing service). The other option is that all speech enhancement services shall be provided by mobile terminals.

A set of speech enhancements that the MRF entity could provide is:

- Noise suppression
- Gain (volume) control
- Acoustic echo cancellation

It should also be mentioned that the Media Gateway and the Multimedia Resource Functions unit are logical entities only and physically they may co-locate in the same device.

5 Tandem Avoidance

5.1 Tandem Free Operation (TFO)

Every time voice is encoded or decoded the speech quality will degrade a little bit. Thus, as few conversion as possible are desired. The basic 2G mobile-to-mobile call suffers from tandem coding that means that separate speech coding happens in both radio interfaces and between the transcoders voice goes in 64 kbps G.711 format. In general two encodings in clear speech conditions is no problem but more than two encodings especially in bad line conditions cause severe degradations.

To overcome this kind of quality problem ETSI has specified so called Tandem Free Operation (TFO) [8] that establishes a sub channel (of 16 or 8 kbps) inside the 64 kbps G.711 stream for the encoded speech. Also the transcoders must support TFO feature as they must omit the decoding and pass encoded parameters as such forward.
An end-to-end connection (of 16 or 8 kbps) can now be formed with only one encoding (in originating mobile) and only one decoding (in receiving mobile). The figures 5 and 6 present the cases without TFO and with TFO in operation.

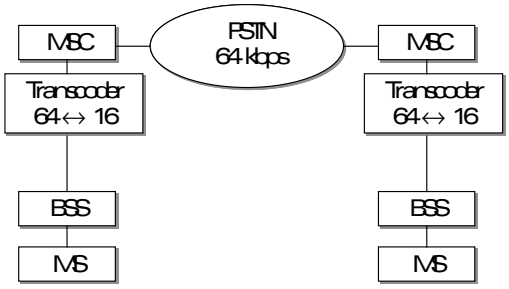


Figure 5. No Tandem Free Operation.

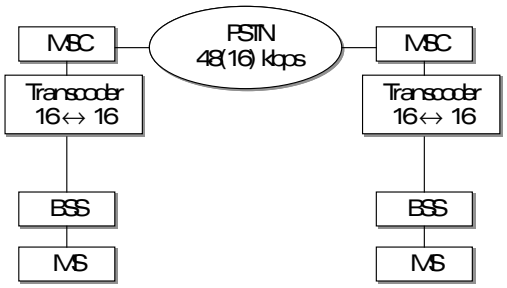


Figure 6. Tandem Free Operation is utilised.

TFO is based on inband procedures that means that no outband signaling is used to form a TFO connection. In practice, the TFO connection establishment starts with a negotiation phase where certain TFO protocol messages are exchanged between transcoders to agree on the used codecs. If the other end doesn't support TFO it will not acknowledge the negotiation and also the TFO capable transcoder will start to encode and decode the 64 kbps as in figure 5.

5.2 Transcoder Free Operation (TrFO)

For the 3G networks a slightly different approach is taken considering tandem avoidance. Firstly, outband signaling is used for codec negotiation and if codecs match there is no need for the transcoders at all. Operation is called as Transcoder Free Operation (TrFO) [9].

TrFO is relevant mainly for the MSC Server concept and for intersystem compatibility as in the final All-IP network calls are by nature of TrFO type. In figure 7 is presented a basic call where outband signaling travels from MSC Server to another until the whole link is negotiated. If a common codec can be agreed no transcoding resources are reserved from the intermediate media gateways.

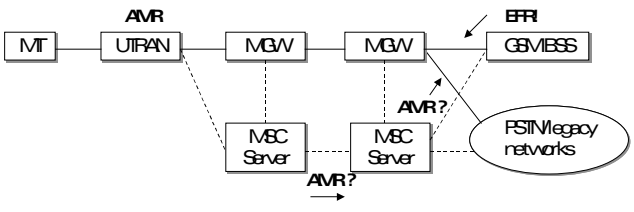


Figure 7. A basic TrFO call.

6 Adaptive Speech Coding

The traditional GSM speech codecs operate in the radio interface at a fixed source rate with a fixed level of error protection (e.g. Full Rate codec with framing overhead consumes 16 kbps and error protection adds 6.8 kbps resulting a 22.8 kbps gross bit rate over the air). The codec itself do not have means (except bad frame handling mechanism) to adapt to changing radio conditions.
For this reason, ETSI (and later 3GPP) has asked for new adaptive coding schemes that could select the optimum channel mode (full rate or half rate) and the optimum codec mode (speech rates) based on the radio conditions. As a result, the Adaptive Multirate (AMR) codec [10,11] has now been standardized as an additional codec for the GSM system and as the only mandatory codec (thus far) for the 3G system. Two most important design targets for the AMR codec were:

- improved speech quality in both half-rate and full-rate modes by means of codec mode adaptation i.e. varying the balance between speech and channel coding for the same gross bit-rate.

- ability to trade speech quality and capacity smoothly and flexibly by a combination of channel and codec mode adaptation; this can be controlled by the network operator on a cell by cell basis.

The AMR codec consist of 2 channel modes (full rate (FR) and half rate (HR)) and 8 codec modes that are presented in table 1. The ninth mode is for discontinuous transmission (DTX) meaning that during silence only silence description (SID) frames are periodically sent to other end. All modes operate on 20 ms frame basis.

Codec mode	Source codec bit-rate
AMR_12.20	12.20 kbit/s FR
AMR_10.20	10.20 kbit/s FR
AMR_7.95	7.95 kbit/s FR / HR
AMR_7.40	7.40 kbit/s FR / HR
AMR_6.70	6.70 kbit/s FR / HR
AMR_5.90	5.90 kbit/s FR / HR
AMR_5.15	5.15 kbit/s FR / HR
AMR_4.75	4.75 kbit/s FR / HR
AMR_SID	1.80 kbit/s FR / HR

Table 1. 8+1 different AMR modes.

The choice between the full rate and the half rate channel mode can be made off-line based on the capacity requirements of the operator. The selection of the codec mode happens continuously by the radio resource management. Basically, as a lower AMR mode is selected, more bits from the gross bit rate are freed for the channel coding and error protection. Even that we use a very low codec bit rate the high error protection keeps the overall speech quality sufficiently high. The figure 8 shows reasoning for the mode selection. To follow the optimum quality curve (MOS=Mean Opinion Score of speech quality) against decreasing signal-to-noise ratio (C/I) the AMR mode that is used must be changed accordingly.

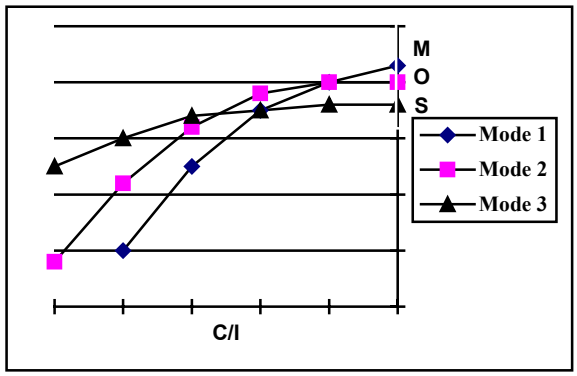


Figure 8. Different AMR modes have different quality curves.

It should be however noted that in the 3G radio interface the power control mechanism (fast power control and outer loop power control) is used to keep the optimum speech quality by adjusting the transmit power of a mobile terminal and the base station. The adaptiveness of AMR in fact doesn't bring such benefits for 3G as it does for 2G radio interface.

Principles of the AMR encoder

The AMR codec is based on the Code-Excited Linear Predictive (CELP) coding model that imitates the glottis and the vocal tract by an excitation signal and a linear prediction synthesis filter (Figure 9).

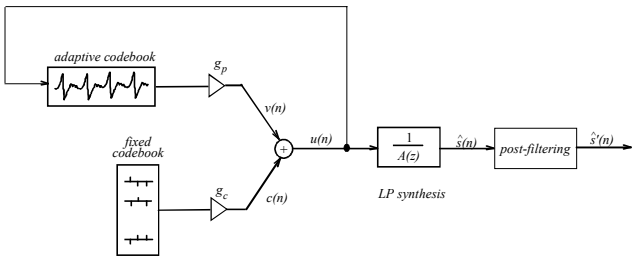


Figure 9. The CELP model.

The excitation signal at the input of the short-term LP synthesis filter is constructed by adding two excitation vectors from adaptive and fixed codebooks. The speech is synthesized by feeding the two properly chosen vectors from these codebooks through the short-term synthesis filter. The optimum excitation sequence in a codebook is chosen using an analysis-by-synthesis search procedure.

The AMR coder operates on speech frames of 20 ms corresponding to 160 samples at the sampling frequency of 8 000 sample/s. At each 160 speech samples, the speech signal is analysed to extract the parameters of the CELP model (LP filter coefficients, adaptive and fixed codebooks' indices and gains). These parameters are encoded and transmitted. At the decoder, these parameters are decoded and speech is synthesized by filtering the reconstructed excitation signal through the LP synthesis filter.

Table 2 shows the resulting parameters from the encoder operating in 12.2 kbps mode. The LP analysis is performed twice per 20ms frame resulting 2 sets of line spectrum pairs (LSP). The adaptive codebook (pitch delay and gain) and the fixed codebook are found for 4 subframes of 5 ms each. Total number of bits is 244 per frame.

Parameter	**1st**	**2nd**	**3rd**	**4th**	**Total**
2 LSP sets					38
Pitch delay	9	6	9	6	30
Pitch gain	4	4	4	4	16
Fixed code	35	35	35	35	140
Fixed gain	5	5	5	5	20
Total					244

Table 2. Encoder output.

RTP payload specification for AMR codec

In the 3GPP Release 99 architecture the AMR codec payload is packed in the Radio Network Controller in IuUP protocol frames [12] that are carried as such to transcoder in 3G MSC. The specified frame format for AMR codec is restricted to Iu interface.

In the All-IP model (figure 3) the AMR payload data travels all the way from the mobile terminal through UTRAN and the core network either to media gateway or another IP terminal. The GGSN will output the application level protocols, that in this case, are the RTP (Real-time Transport Protocol) frames carrying the AMR payloads. So, concerning IP Telephony the RTP payload specification for AMR codec [13] has grown in importance as AMR is the codec that should converge the traditional IP Telephony with the mobile IP Telephony. The RTP for AMR specification includes the following extra features:

- codec mode request procedure
- robust sorting of payload bits
- bad frame indication
- compound payloads
- CRC calculation

The specification is still under finalisation in IETF.

7 Wideband Speech Coding

The 300-3400Hz speech band frequency range has been used for decades in all telephony applications. As the range is heavily restricted all non-speech signals, like music, are degraded badly when forced to go through this narrow frequency pipe. Even speech contains plenty of information above 3400 Hz that affects the naturalness of speech.

Basically, the existing terminals that conform to this traditional frequency band have been one barrier in front of wideband speech. Second reason has been that more bandwidth is needed to transfer the highest quality wideband signals.

However, as the difference in quality between narrowband and wideband speech is so clear it is evitable that more wideband applications will be introduced in the near future. Wideband speech coding can easily be seen as the next fundamental improvement in speech quality for mobile telecommunication systems. 3GPP has understood this and wideband AMR specifications are already getting ready.

The principles of wideband AMR [14] are copied from the narrowband AMR. The frequency band, as a difference, is extended in both directions, and it is now from 50 Hz to 7000 Hz. The resulting speech quality exceeds the wireline quality of narrowband G.711. In figure 10 is shown an illustrive graph on speech quality comparison of EFR, AMR-NB and AMR-WB in 16kbps full-rate channel [15].

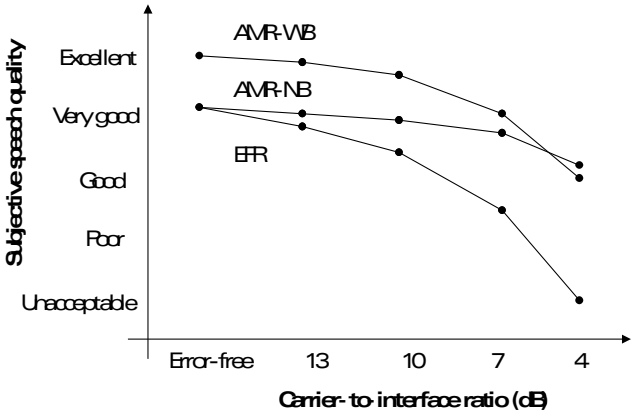


Figure 10. AMR-WB vs. AMR-NB and EFR.

As can be seen, in error-free conditions the AMR-WB is superior over AMR-NB or EFR (that is the highest quality GSM codec at the moment). Even in very bad conditions AMR-WB can maintain high quality far above fixed rate GSM codecs. The nine modes of AMR-WB (plus one mode for DTX) are presented in in table 3.

AMR-WB is specified for GSM full rate radio traffic channel, for future GSM EDGE (GERAN) and for the 3G (UTRAN) radio channel. The 3GPP specifications for a wideband AMR codec (AMR-WB) are expected to be finalized in March 2001.

Codec mode	Source codec bit-rate
AMR-WB_23.85	23.80 kbit/s
AMR-WB_23.05	23.05 kbit/s
AMR-WB_19.85	19.85 kbit/s
AMR-WB_18.25	18.25 kbit/s
AMR-WB_15.85	15.85 kbit/s
AMR-WB_14.25	14.25kbit/s
AMR-WB_12.65	12.65 kbit/s
AMR-WB_8.85	8.85 kbit/s
AMR-WB_6.6	6.6 kbit/s
AMR-WB_SID	1.75 kbit/s

Table 3. 9 Different AMR-WB modes.

8 Conclusion

Packet data services have been advertised to be the major application of future 3G networks. However, also the voice services are strongly enhanced with new wideband codecs that can adapt to network conditions. Also the transcoder free operation, and the new speech enhancement services will make speech quality better, even to level never experienced before.

This article has mainly focused on the application level. Good network conditions (low delay, no lost packets due congestion) are a starting point also for superior application level speech quality. Media gateways shall support the network level QoS mechanisms (like DiffServ) that are used to optimize and prioritise the real-time and the non-real-time traffic (see for example [16]).

In the past, speech service has been closely tied on technical level to providing network. Within All-IP networks also speech service will be lifted more and more up to user-level. End-to-end user applications will not even see the underlying transport network and the overall speech quality that is perceived will heavily depend on the characteristics and features of the All-IP terminals.

As also the speech service will include more choices of used codecs, used bandwidth and used speech enhancements there shall be opportunity to differentiate the pricing of these features. The user may in the future have means to select the speech quality that he or she is willing to pay.

References

[1] ITU-T G.711; Pulse Code Modulation (PCM) of Voice Frequencies. 1972.

[2] Hersent O, Gurle D, Petit J-P. IP Telephony. Packet-based multimedia communications system. Addison Wesley, 2000.

[3] GSM 06.10; Full Rate Speech; Transcoding.

[4] GSM 06.60; Enhanced Full Rate Speech; Transcoding.

[5] Third Generation Partnership Project (3GPP) www.3gpp.org

[6] 3GPP/TR 23.922; Architecture for an All-IP network, v1.0.0, October 1999.

[7] ITU-T H.248; Gateway Control Protocol, June 2000.

[8] GSM 08.62; Inband Tandem Free Operation (TFO) of Speech Codecs; Service Description; Stage, v8.0.1, August 2000.

[9] 3GPP/TS 23.153; Out of Band Transcoder Control – Stage 2, v2.0.3, October 2000.

[10] 3GPP/TS 26.071; AMR Speech Codec; General Description, v3.0.1, August 1999.

[11] 3GPP/TR 26.975; Performance Characterization of AMR Speech Codec, v1.1.0, January 2000.

[12] 3GPP/TS 25.415; UTRAN Iu Interface User Plane Protocols, v3.5.0, December 2000.

[13] IETF Internet Draft: RTP Payload Format and File Storage Format for AMR Audio, v0.5, February 2001.

[14] 3GPP/TR 26.901; AMR Wideband Speech Codec; Feasibility Study Report, v4.0.1, April 2000.

[15] Advance – Information from Nokia Research Center. Number 1, 2001.

[16] Ferguson P, Huston G. Quality of Service; Delivering QoS on the Internet and in Corporate Networks. Wiley 1998.

Session Initiation Protocol (SIP)

Jouni Soitinaho
Jouni.Soitinaho@nokia.com

Abstract

This paper describes the basic characteristics of the SIP protocol and especially its extension mechanism. Several Internet draft specifications are studied in order to get an overall picture of the maturity of the protocol. Some interesting application areas are examined for demonstrating how the SIP protocol suite can be used in a wider context.

1 Introduction

SIP is a simple but extendable signaling protocol for setting up, modifying and shutting down communication sessions between two or more participants. One or more media or even no media at all, can be transmitted in the session context. SIP is independent of the actual media and the route of the media can be different to the route of signaling messages. SIP can also invite participants to IP multicast session.

SIP is part of the IETF multimedia architecture and it's designed to cooperate with several other protocols, which is a fundamental principle of the SIP design. Other protocols include, for example, RTP and RTCP for media transport, RTSP for controlling streaming and SDP for describing the capabilities of the participants. Limiting the SIP protocol to the controlling of the session state is also more likely to keep it simple and easy to implement.

Another fundamental aspect of SIP design is the easy way it can be extended with additional capabilities. Actually, the basic protocol specification defines rather limited signaling protocol. It is missing several capabilities needed by real life applications. Several general extensions are being defined currently and some of these are expected to be included in the basic standard after reaching the required stability.

SIP was first developed within the Multiparty Multimedia Session Control (MMUSIC) working group and then continued in the SIP working group. Active communications with MMUSIC is important since the Session Description Protocol (SDP) is developed by MMUSIC. The working group has also close relationship with the IP telephony (iptel) working group, whose Call Processing Language (CPL) relates to many features of SIP, and the PSTN and Internet Internetworking (pint) working group, whose specification is based on SIP. Distributed Call Signaling Group (DCS) is giving input to SIP for distributed telephony services. Recently it was decided to split the SIP working group to two: SIP WG will concentrate on the basic protocol and general extensions and SIPPING WG will concentrate on applications and generate input to the SIP WG.

Besides all the activities taken by the IETF task forces 3GPP technical specification groups currently investigate SIP. Since SIP was chosen as the signaling protocol for the IP multimedia subsystem of 3G network 3GPP will set new requirements for the protocol.

The basic SIP protocol is defined in RFC2543 that is currently in "proposed" state. The corresponding Internet draft document [1] contains many updates and is the reference document for describing the basic protocol in the next section. Some of the current development activities are discussed in section three. Finally, a few application areas of SIP are studied in section four before conclusions in the last section.

2 Basic Protocol

2.1 Characteristics

The basic features of SIP:

- Locating user: determination of the end system to be used for communication;
- Determining user capabilities: determination of the media and media parameters to be used;
- Determining user availability: determination of the willingness of the called party to engage in communications;
- Setting up the call: "ringing", establishment of call parameters at both called and calling party;

- Controlling the call: including transfer and termination of calls.

Main technical properties and some implications of SIP:

- Text-based (ISO 10646 in UTF-8 encoding), similar to HTTP: Easy to learn, implement, debug and extend. Causes extra overhead, which is not a serious drawback for a signaling protocol. Header names can be abbreviated.
- Recommended transport protocol is UDP: It is not meant to send large amounts of data.
- Application level routing based on Request-URI: The signaling path through SIP proxies is controlled by the protocol itself not by the underlying network. Requires routing implementation in SIP proxies.
- Independence on the session it initiates and terminates (capability descriptions, transport protocol, etc.): Cooperates with different protocols, which can be developed independently. It is not a conference control protocol (floor control, voting, etc.) but it can be used to introduce one.
- Supports multicasting for signaling and media but no multicast address or any other network resource allocation.
- Support for stateless, efficient and "forward" compatible proxies (re-INVITE carries state, ignore the body, ignore extension methods).

2.2 Operations

Protocol operations of SIP:

- `INVITE` initiates session establishment
- `ACK` confirms successful session establishment
- `OPTIONS` requests capabilities
- `BYE` terminates the session
- `CANCEL` cancels a pending session establishment
- `REGISTER` binds a permanent SIP URL to a temporary SIP URL for the current location.

The following diagram demonstrates SIP protocol operations for user registration and session handling.

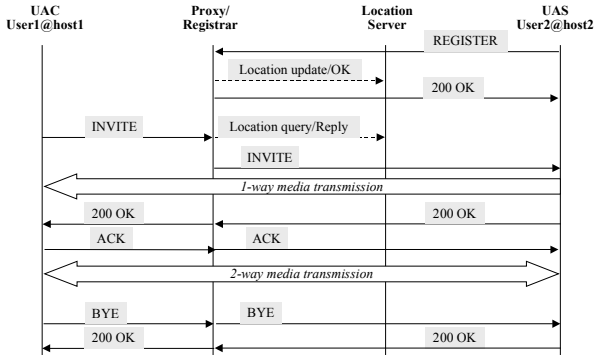


Figure 5. An example of SIP protocol operations.

2.3 Network elements

SIP has been designed for IP networking. The protocol makes use of standard elements like DNS and DHCP servers, firewalls, NATs and proxies. Special support in DNS and DHCP servers is not needed but it makes the protocol operations more efficient. The SIP protocol is implemented by the user agent client (UAC) and server (UAS), redirect servers, proxies and registrars. Registrars and location servers maintain the mapping between user's permanent address and current physical addresses.

The SIP specification does not actually define the network architecture. However, the logical elements and their relationships can be determined based on the protocol specification. The following figure demonstrates an example of inter-domain session setup. Both UAC and UAS are located in their home domains. Thin lines represent SIP signaling messages and thick lines represent media transmission and dotted line represent non-SIP protocol.

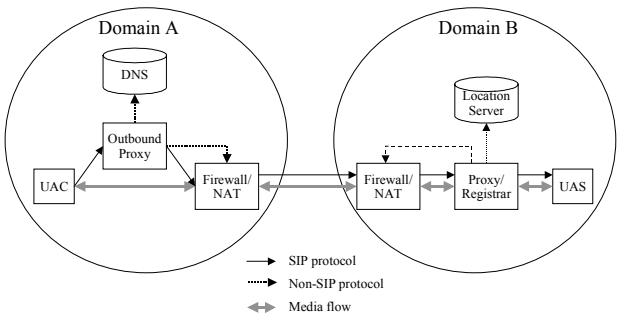


Figure 6. Logical network elements involved in an inter-domain session setup.

In this scenario UAC composes an INVITE message in order to set up a call with UAS. The message contains the session data in its headers and media descriptions in the body in SDP format [2]. INVITE is sent to

Outbound Proxy whose address may have been configured in UAC using DHCP. Outbound Proxy uses DNS to resolve the recipient's address. It also controls Firewall/NAT to open the ports for media transmission. Domain B has configured all the incoming requests to go to Proxy/Registrar that controls Firewall/NAT of Domain B. Proxy/Registrar queries the current location of UAS from Location Server and forwards the message to UAS. In an intra-domain call a redirect server could be used instead of a proxy in Domain B to return the current location of UAS who could then be contacted directly by UAC without having any proxy involved in the communications.

Since the request carried the media descriptions of UAC and since the corresponding ports were opened in firewalls media can immediately flow back from UAS to UAC. The signaling response is routed along the same path as the request and it carries the media descriptions of UAS. UAC can now send media to UAS. Finally UAC has to send ACK message to UAS for acknowledging the successful session establishment.

2.4 Addressing and routing

SIP uses e-mail like addresses for users but it also includes the protocol keyword in the SIP URL. SIP URLs are used to identify the originator (From), current destination (Request-URI), final destination (To) and redirection address (Contact).

Two formats exist:

- `sip:user@host`

  when UA exists, e.g. From and To fields in INVITE

- `sip:host`

  when no UA exists, e.g. Request-URI in REGISTER

Including the protocol keyword in the URL allows SIP server use the Contact-header to redirect a call to a web page or to a mail server, for example. This facilitates integration of audio and video applications with other multimedia applications.

Routing of SIP messages is included in the protocol itself since finding the user is one of the primary functions of SIP. The host part of the SIP URL indicates the next hop for a request. Even if clients could send the request directly to this address in practice they are typically forced to go through a proxy for security or address translation reasons.

Furthermore two headers are in central position for routing SIP messages:

- `Via` header indicates the request path taken so far. It prevents looping and is used for routing the response back the same path as request has traveled. Proxies must add "received" parameter in the top-most Via header if the field contains different address than the sender's source address. This feature supports NAT servers. Proxies can also forward the request as multicast by adding "maddr" parameter in the Via field.

- `Route` header is used for routing all requests of a call leg along the same path, which was recorded in the `Record-Route` header during the first request. This is to guarantee that stateful proxies will receive all the subsequent messages that affect the call state.

SIP proxies can also fork the incoming request to several outgoing requests in order to accelerate the processing of INVITE method. The forking can create several simultaneous unicast INVITEs to the potential locations or one multicast INVITE to a restricted subnetwork. Even if forking is an efficient mechanism it is a potential source of difficult problems and needs to be paid special attention during implementation.

2.5 Registering

A client uses REGISTER method to bind its permanent address to one or more physical addresses where the client can be reached. The request is sent to the registrar, which is typically co-located with a proxy server. Alternatively the request can be sent to the well-known SIP multicast address "sip.mcast.net".

REGISTER method is also ideally suited for configuration and exchange of application layer data between a user agent and its proxy. This may produce modest amounts of data exchanges. However, because of the infrequency of such exchanges and their typical limitation to one-hop this is acceptable if TCP is used.

The most important fields for the REGISTER method:

- `Request-URI` names the domain of the registrar. `user` part must be empty.

- `To` indicates the user to be registered

- `From` indicates the user responsible for the registration (typically equal to To header value)

- `Contact` (optional) indicates the address(es) of the user's current location. List of current locations can be queried by leaving the Contact header empty in the REGISTER request. An optional `expires` parameter indicates the expiration time of the particular registration. By giving the wildcard address "*" in a single contact header a client can remove all the registrations. By giving zero as the value for the expires parameter a client can remove the corresponding registration.
- `Expires` tell the default value for expiration unless the corresponding parameter is present in the `Contact` header. If neither one is present default value of one hour is used.

It is particularly important that REGISTER requestor is authenticated.

2.6 SIP Security

Security must be addressed at several levels. At the network level the security is based on regular firewalls and NATs since SIP is designed for IP networking. Controlling the firewall with a SIP proxy is an essential enhancement for the standard IP security mechanisms.

At the protocol level both the media security and signaling security must be addressed. Media encryption is specified in the message body with SDP [2].

Signaling security includes user authentication and encryption of the signaling messages. User authentication is based on HTTP authentication mechanism [3] with minor modifications as specified in [1]. Besides "Basic" and "Digest" authentication schemes SIP supports also stronger authentication with "PGP" scheme [4]. It is based on public key cryptography, which requires the client to sign the request with the private key and the server to verify the signature with the public key. It is recommended to authenticate the REGISTER requestor with the PGP scheme instead of the other schemes.

SIP also supports PGP encryption of the signaling messages. By setting the "Encryption" header to "PGP" scheme all following headers can be encrypted as well as the message body. Note that sending the media encryption key in the body requires the message body to be encrypted. Note also that there are special considerations for the encryption of the Via header since it is used by the proxies.

Obviously, standard IPSec protocol can be used for IP level encryption.

2.7 Expandability

In order to keep the basic protocol compact SIP provides the protocol designers with means for extending its capabilities. Protocol elements that can be extended without change in the protocol version include:

- Methods
- Entity headers
- Response codes
- Option tags

In addition to the SIP extensions the session description (SDP) can be extended to contain new attributes and values for the session.

Several definitions in the protocol set the limits for the extensions. First of all, proxy and redirect servers treat all methods other than INVITE, CANCEL and ACK in the same way by forwarding them. User agent server and registrar respond with the "501 Not Implemented" response code for request methods they do not support.

SIP servers and proxies ignore header fields not defined in the specification [1] and they do not understand, i.e. treating them as entity headers. General headers, request headers and response headers are extended only in combination with a change in the protocol version. Furthermore, stateless proxies are required to recognize only the values defined in the basic protocol. They will forward new values without actions. Session stateful proxies need to support the extension if it can change the call state in a way, which is meaningful for the proxy.

SIP applications are not required to understand all registered response codes. They must treat any unrecognized response code as being equivalent to the x00 response code of that class, with the exception that an unrecognized response must not be cached.

Option tags are unique identifiers used to designate new extensions for SIP. These tags are set in Require, Proxy-Require, Supported and Unsupported header fields to communicate the signaling capabilities between UACs, UASs and proxies. The extension creator must either prefix the option with the reverse

domain name or register the new option with the Internet Assigned Numbers Authority (IANA).

Clients can always call the OPTIONS method for explicitly querying the capabilities of the server and proxies lying on the path.

Since there are multiple ways to define a SIP extension special attention needs to be paid on the semantic compliance with the basic protocol. An informational Internet draft sets the guidelines for writing a SIP extension [5].

3 Protocol Extensions

About 30 extension drafts can be found on http://www.cs.columbia.edu/~hgs/sip/drafts_base.html. Some of these add reliability or functionality missing in the basic protocol for supporting real time services like VoIP. Examples of these are "reliable provisional responses", "resource management" and "INFO method". Some extensions add functionality for implementing existing PBX services, like call transfer. Examples are "call control-transfer" and "caller identity and privacy". Some extensions add new functionality for enabling new type of services, like presence based instant messaging. Examples are "event notification" and "caller preferences". Finally some extensions add resilience to the basic protocol for implementing reliable and scalable networks. Examples are "session timer" and "distributed call state".

3.1 Reliable provisional responses

When run over UDP, SIP does not guarantee that provisional responses (1xx) are delivered reliably, or in order. However, many applications like gateways wireless phones and call queuing systems make use of the provisional responses to drive state machinery. This is especially true for the 180 Ringing provisional response, which maps to the Q.931 ALERTING message.

The Internet draft document [6] specifies an extension to SIP for providing reliable provisional response messages ("100rel"). When a server generates a provisional response which is to be delivered reliably, it places a random initial value for the sequence number (RSeq). The response is then retransmitted with an exponential backoff like a final response to INVITE.

The client uses a new method (PRACK) for acknowledging the provisional response. Unlike ACK, which is end-to-end, PRACK is a normal SIP message, like BYE. Its reliability is ensured hop-by-hop through each stateful proxy. PRACK has its own response and therefore existing proxy servers need no modifications. A new header (RAck) in the PRACK message indicates the sequence number of the provisional response, which is being acknowledged.

The following diagram demonstrates how the support and need for reliable provisional response is negotiated and implemented.

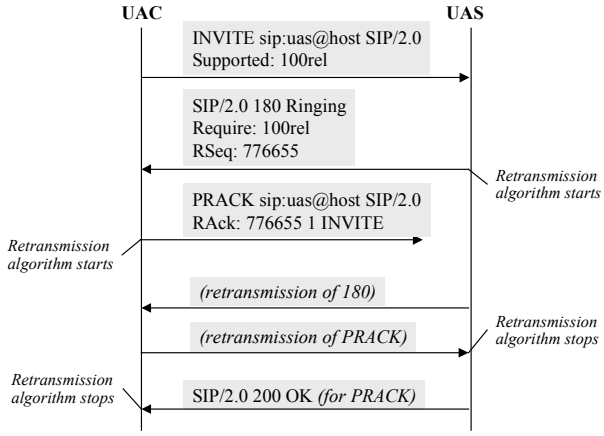


Figure 7. Reliable provisional response.

3.2 Resource Management

In order to become a successful service Internet telephony must meet the quality expectations based on the existing telephony services. This implies that the resources must be reserved beforehand for each call. Cooperation is therefore needed between call signaling, which controls access to telephony specific services, and resource management, which controls access to network-layer resources

The Internet draft document [10] discusses how network QoS and security establishment can be made a precondition to sessions initiated by SIP, and described by SDP. These preconditions require that the participant reserve network resources or establish a secure media channel before continuing with the session. In practical terms the "phone won't ring" until the preconditions are met. The draft proposes new attributes for SDP:

- `"a=qos:" strength-tag SP direction-tag`

- `"a=secure:" SP strength-tag SP direction-tag`

where the strength can have values "mandatory", "optional", "success" and "failure" and the direction can have values "send", "recv" and "sendrecv".

The document also proposes a new method to SIP. The COMET method is used to confirm the completion of all preconditions by the session originator. The following diagram presents the message flow for a single-media session setup with a "mandatory" quality-of-service "sendrecv" precondition, where both the UAC and UAS can only perform a single-direction ("send") resource reservation.

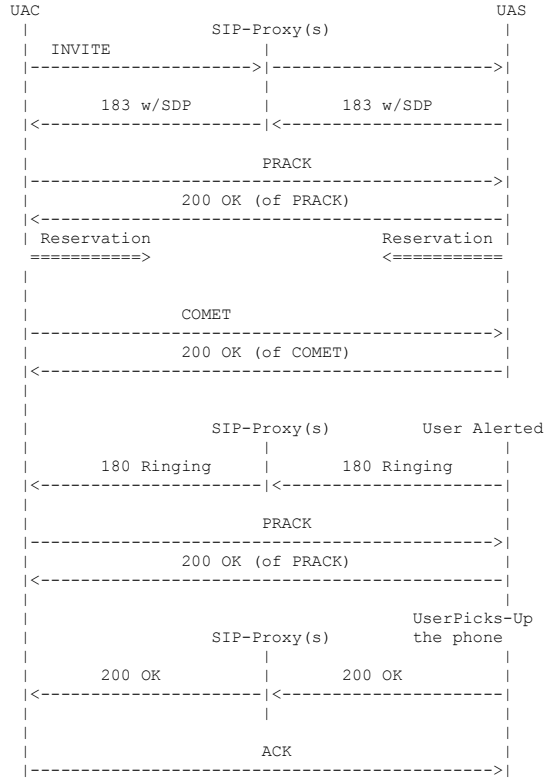


Figure 8. Resource management signaling.

The session originator (UAC) prepares an SDP message body for the INVITE describing the desired QoS and security preconditions for each media flow, and the desired direction "sendrecv." This SDP is included in the INVITE message sent through the proxies, and includes an entry "a=qos:mandatory sendrecv." The recipient of the INVITE (UAS), returns a 183-Session-Progress provisional response containing SDP, along with the qos/secure attribute for each stream having a precondition. The UAS now attempts to reserve the qos resources and establish the security associations. The 183-Session-Progress is received by the UAC, and the UAC requests the resources needed in its "send" direction, and establishes the security associations.

The diagram also demonstrates the usage of PRACK and COMET methods for confirming the responses and resource allocations respectively.

3.3 INFO method

The SIP INVITE method can be called one or more times during the established session (re-INVITE) to change the properties of media flows or to update the SIP session timer. However, there is no general-purpose mechanism to carry session control information along the SIP signaling path during the session.

RFC2976 [14] defines the INFO method for communicating mid-session information during the call. It is not used to change the state of the session but it provides means for exchanging additional information between the peers. One example of such session control information is ISUP and ISDN signaling messages used to control telephony call services.

The information can be conveyed either in the header of the INFO message or as part of the message body. The definition of the message body and/or message headers used to carry the mid-session information is outside the scope of this document. However, consideration should be taken on the size of message bodies since it can be fragmented while carried over UDP bearer.

3.4 Call Control - Transfer

The basic SIP protocol does not support any of the multiple ways a call can be transferred to a third party. In an "unattended transfer" the transferor is not participating the call simultaneously with the transferee and transfer target whereas in an "attended transfer" the three actors participate the call simultaneously (ad-hoc conference). In an "consultation hold transfer" the transferor establishes and terminates a second call with the transfer target before performing the actual transfer.

The Internet draft document [11] proposes a SIP extension, which can be used, for example, to implement traditional unattended and consultation hold transfers. The attended transfer is not drafted yet since the call control framework has not addressed

conferencing. The following figure presents the message sequence of unattended transfer with consultation hold.

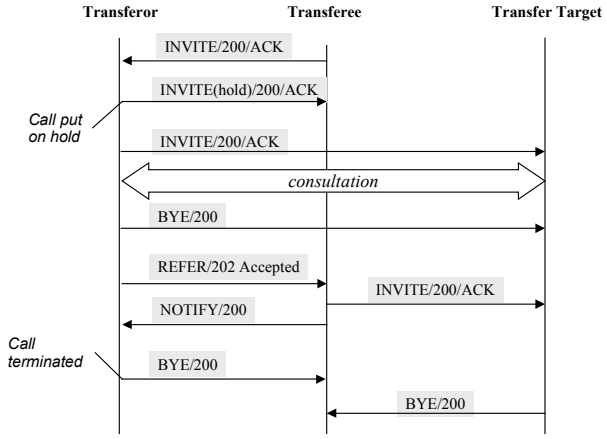


Figure 9. Unattended call transfer with consultation on hold.

The new REFER method indicates that the recipient (Request-URI) should contact a third party identified by the contact information (Refer-To). Once the transferee knows whether the transfer succeeded or failed it notifies the transferor by sending "refer" event using the NOTIFY mechanism as if the REFER message had established a subscription.

3.5 Caller Identity and Privacy

In order for SIP to be a viable alternative to the current PSTN, it must support certain telephony services including Calling Identity Delivery, Calling Identity Delivery Blocking, as well as the ability to trace the originator of a call. While SIP can support each of these services independently, certain combinations cannot be supported. The issue of IP address privacy for both the caller and callee needs to be addressed as well.

The Internet draft document [12] specifies two extensions to SIP that allow the parties to be identified by a trusted intermediary while still being able to maintain their privacy. A new general header, Remote-Party-ID, identifies each party. Different types of party information can be provided, e.g. calling, or called party, and for each type of party, different types of identity information, e.g. subscriber, or terminal, can be provided. Another new general header, Anonymity, is also defined for hiding the IP addresses from the other parties.

3.6 Caller preferences

When a SIP server receives a request, there are at least three parties who have an interest and each of which should have the means for expressing its policy:

- The administrator of the server, whose directives can be programmed in the server.
- The callee, whose directives can be expressed most easily through a script written in the call processing language (CPL)
- The caller, who doesn't have obvious ways to express the preferences within the SIP server.

The Internet draft document [9] specifies an extension mechanisms by which the caller can provide its preferences for processing a request. These preferences include the ability to select which URIs a request gets proxied or redirected to, and to specify certain request handling directives in proxies and redirect servers. It does so by defining three new request headers, Accept-Contact, Reject-Contact and Request-Disposition. The extension also defines new parameters for the Contact header that describe attributes of a UA at a specified URI.

3.7 Event Notification

The ability to request asynchronous notification of events is useful in many types of services. Examples include automatic callback services (based on terminal state events), buddy lists (based on user presence events), message waiting indications (based on mailbox state change events), and PINT status (based on call state events).

The Internet draft document [13] proposes a framework by which notification of events can be ordered. The draft can't be used directly, i.e. it doesn't specify any event types and it must be extended by other specifications (event packages). In object-oriented terminology, this is an abstract base class which must be derived into an instantiatable class by further extensions.

The extension is based on two new methods: SUBSCRIBE and NOTIFY and a new header "Event" together with the "Expires" header. Neither SUBSCRIBE nor NOTIFY necessitates the use of "Require" or "Proxy-Require" header and no extension token is defined for "Supported" header. Clients may probe for the support of SUBSCRIBE and NOTIFY using the OPTIONS method.

There is no separate media transmission between the subscriber and notifier as in normal SIP session. The message body of the NOTIFY method is to carry the actual notification.

Removing and refreshing subscriptions are performed in the same way as for REGISTER method. Usage of the message body in SUBSCRIBE request is left up to the concrete extensions. It may be used to filter and set thresholds for the events.

The basic scenario of a notification session is presented in the following figure. Note that according to the SIP principle proxies need no additional behavior to support SUBSCRIBE and NOTIFY methods but they can act as subscribers and notifiers.

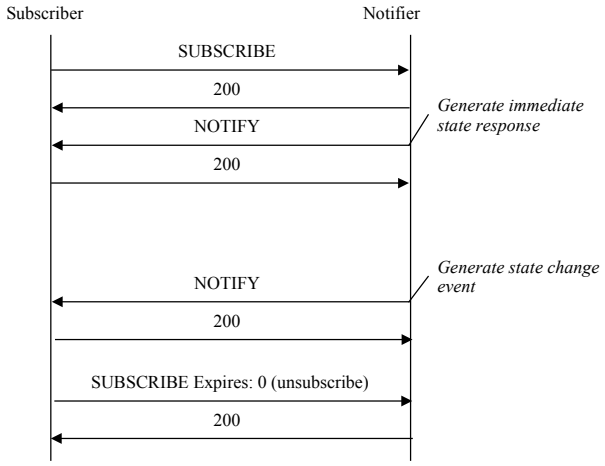


Figure 10. Event notification messages.

This extension is not targeted to very frequent notifications. The interval must be minutes instead of seconds. For better performance and for simplifying the subscriber implementation the new state after the event must be notified in addition to the event itself. The extension is not either for transferring large amounts of data since the preferred transport protocol is UDP. Therefore this extension is not fully in line with the SIP extension guidelines.

3.8 Session timer

SIP does not currently define a keepalive mechanism. The result is that call stateful proxies are not always able to determine whether a call is still active or not. For instance, when a user agent fails to send a BYE message at the end of a session, or the BYE message gets lost due to network problems, a call stateful proxy will not know when the session has ended.

This is especially important feature for proxies controlling firewalls or NATs or performing billing tasks. Holes and address bindings are dynamically created in firewall and NATs to allow the media for the session to flow. These settings represent state which must be eventually removed.

The Internet draft document [7] specifies the session timer extension ("timer") for solving the problem and improving the reliability of the basic SIP protocol. UAC, UAS and proxies communicate the support for the extension and assign the responsible party (UAC or UAS) for sending the re-INVITEs in the original INVITE message. If UAC supports the extension it sets "timer" in the Supported header and if it wants to turn the extension on it sets the refresh interval in Session-Expires header. UAC will then be responsible for sending the re-INVITEs. A proxy may adjust the refresh interval to a smaller value and also require (Proxy-Require) UAS to send the re-INVITEs in case UAC does not support the extension. If a re-INVITE is not received before the refresh interval passes, the session is considered terminated, and call stateful proxies can release the session.

Note that using INVITE as the refresh method, as opposed to a new method, allows sessions to be recovered after a crash and restart of one of the UAs.

3.9 Distributed Call State

Many types of services require proxies to retain call state. Unfortunately, maintaining call state presents problems. It introduces scalability problems and makes fallback and load balancing more complex.

The extension proposed in the Internet draft document [8] allows proxies to encapsulate any state information they desire into a header, called State header. The header is sent to the user agents and reflected back in subsequent messages.

The idea is similar to the use of cookies with HTTP user agent clients and proxies. In essence, it allows proxies to behave as stateful proxies while still being stateless.

4 Applications

4.1 Call centers

There are multiple ways to implement a SIP based call center where more than one operators can provide the same service for incoming requests. In a very simple model a redirect server is used together with the registrar to redirect the calls to a free operator according to a round robin algorithm, for example. The server can use the Contact header with the maddr parameter to instruct the caller to send the next

INVITE with the same Request-URI but connect to the host indicated by the maddr parameter.

This is a very limited solution since the redirect server has no automatic means to record the state of the operators. Of course, they could send re-REGISTER message whenever they are free for a new call but this is not according to the semantics of the REGISTER message. In fact, SIP provides a better way for implementing the application.

Using a SIP proxy instead of a redirect server the state of each call can be maintained by listening to the SIP messages. The address of the proxy is published externally and no direct connections to the operator addresses are allowed through the firewall. The proxy includes itself in the message path using Record-Route and Via headers in order to get the CANCEL and BYE requests as well as all the responses. When a new call arrives the proxy decides the operator based on its own call state information and information in the registrar.

Sending the INVITE message using IP multicast can accelerate the seeking of operator. Free operators generate a response within a random time interval. Since all operators will hear the first response they can drop the request without responding. If no operator is free proxy retries until one is free or the client terminates the call by sending CANCEL request which is responded by the proxy. The proxy generates all call statistics.

If the network does not support IP multicast yet another option is to fork the request in the proxy into simultaneous requests to the current locations of the free operators. In this case the cancellation of the other INVITE messages need to be performed by the proxy whenever the first operator responds.

4.2 Presence and Instant Messaging

Presence is considered as a promising application area in all-IP networks. When combined with instant messaging it creates a lot of opportunities for application developers. A new working group, called SIMPLE (SIP for Instant Messaging and Presence Leveraging), has been established in IETF for developing specifications in this area. 3GPP is also considering presence as one service for the IM subsystem.

Presence is defined as user's reachability, capabilities and willingness to communicate with other users. Presence application obviously has to provide the means to deliver this information to other users. A lot of room exists for differentiating applications from each other's. For example, intelligent filters for exposing the presence and accepting calls can be built based, for example, on user's location and caller's identity.

Instant messaging (IM) is defined as the exchange of content between a set of participants in real time, like in IRC. The content is mainly small textual messages but they can also contain pictures or audio or video clips. The main difference to emails is the real time nature requiring all the parties to be online.

It is very important to keep presence and IM separate from each other even if these are mixed in the existing, proprietary solutions. The separation enables independent development of the two protocols. This is important also because of the existing IM applications (multiplayer online games).

SIMPLE bases its work on the existing SIP and extension drafts. The foundation of using SIP for the presence and IM protocols derives from two factors: the SIP registrars already hold some information about the user's presence and SIP networks already route messages from user to the proxy that can access this information [15,16]. Extending SIP for this area is rather small step in terms of protocol operations but semantically it is a bigger step, however.

The presence extension is an instantiation of the abstract notification extension. A new event package, named as "presence", is defined for this purpose. The body of the NOTIFY message contains a presence document. An XML data format and a MIME type will be defined for the document. The following figure shows the logical elements for SIP presence.

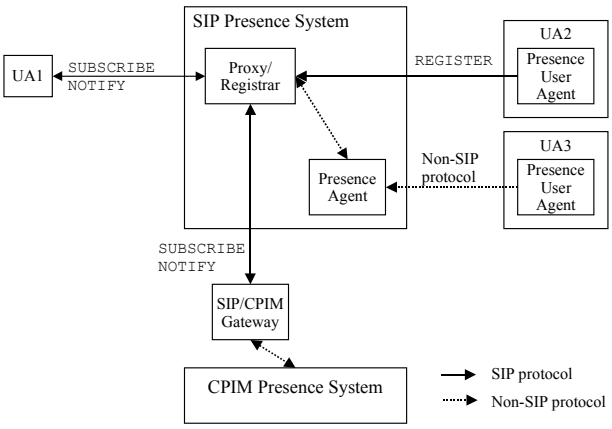


Figure 11. Logical network elements for SIP presence.

The presence agent (PA) is capable of storing the subscriptions and generating notifications based on the events. Present user agent (PUA) updates presence information.

Authorization is a critical component of a presence protocol. Authorization can be pushed to the server ahead of time or, more typically, determined at the time of subscription. Since this is not covered by the basic SIP protocol an Internet draft [17] proposes a new method (QUATH) for querying the authorization from the subscription authorizer (e.g. PUA). This draft seems to be arguable, however.

The IM protocol extensions are defined in the Internet draft [18]. When a user wishes to send an instant message to another, the sender issues a SIP request using the new MESSAGE method. The request URI can be in the format of "im: URL" or normal SIP URL. The body of the request contains the message to be delivered. Provisional and final responses will be returned to the sender as with any other SIP request. The following diagram shows two message exchanges between two users.

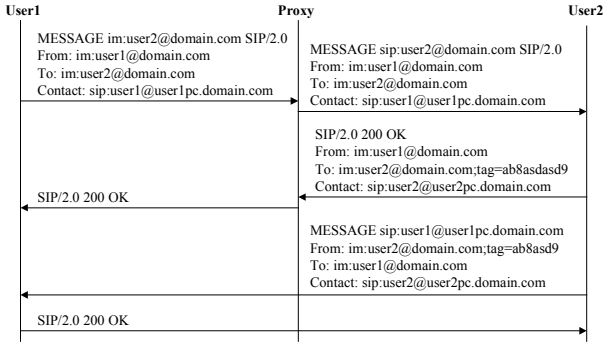


Figure 12. Instant messaging between users in the same domain.

Proxy looks up the registration database for the binding from im address to sip address of User2 and forwards the message to the current location. The response traverses the same path. Based on the Contact header of the message User2 can send the second message directly to User1's current location because Proxy added no Record-Route header in the first message. The From and To headers are reversed, however.

The specifications for presence and instant messaging are still rather insufficient. This is indicated by the long list of open issues listed in the drafts.

The semantic difference between presence and IM protocols and basic SIP protocol is in the type of session they create. Presence protocol creates a passive session which is used asynchronously for notifying the subscriber using the signaling channel without any media channel. Establishment and termination of the session is done differently to the basic protocol. IM does not create a session at all which is currently discussed in the working group. Surrounding the related MESSAGE requests with INVITE and BYE requests would be consistent with the basic protocol..

5 Conclusion

Simplicity is a key characteristic of SIP. It facilitates interoperable clients, servers and proxies coming from independent vendors. Sharing a lot of similarities with HTTP makes the understanding of SIP rather easy for a large developer community.

Expandability is another key characteristic. Being inbuilt in the basic protocol it provides the means for extending the protocol capabilities. Network elements can dynamically negotiate their capabilities. The basic protocol specification can concentrate on its primary function.

Supporting different protocols for different purposes is yet another key characteristic of SIP. This facilitates protocol development independence between SIP and other protocols and makes the overall adoption of SIP more likely.

A lot of SIP related development activities are going on in IETF (over 70 drafts). This is an evidence of its potential on one hand but an evidence of its immaturity on the other hand. The potential is demonstrated by the application examples presented in this paper. The immaturity for IP telephony is demonstrated by the large number of suggested extensions described in this paper that are fundamental for this area.

Many extensions seem to be very useful and easy to specify at first sight. However, they may not share the semantics of the basic protocol and should not be defined as a SIP extension. The ability of IETF to respond to the needs and at the same time control the specification work will be tested in near future.

The slowness of the IETF process is indicated also by its inability to promote the basic SIP specification to

"draft" state after being in "proposed" state over two years. At the same time 3GPP is stating its requirements for SIP in the IP multimedia subsystem of 3G. If these requirements are not included in the IETF specifications the risk of SIP fragmentation may come true.

References

[1] Internet Draft, SIP WG, Handley/Schulzrinne/Schooler/Rosenberg: SIP: Session Initiation Protocol, November 24, 2000, http://www.ietf.org/internet-drafts/draft-ietf-sip-rfc2543bis-02.txt

[2] RFC2327, Network WG, M. Handley, V. Jacobson: SDP: Session Description Protocol, April 1998, http://www.ietf.org/rfc/rfc2327.txt

[3] RFC2617, Network WG, J. Franks, et al: HTTP Authentication: Basic and Digest Access Authentication, June 1999, http://www.ietf.org/rfc/rfc2617.txt

[4] RFC2440, Network WG, J. Callas, et al: OpenPGP Message Format, November 1998, http://www.ietf.org/rfc/rfc2440.txt

[5] Internet Draft, SIP WG, J.Rosenberg, H.Schulzrinne: Guidelines for Authors of SIP Extensions, March 5, 2001, http://www.ietf.org/internet-drafts/draft-ietf-sip-guidelines-02.txt

[6] Internet Draft, SIP WG, J.Rosenberg,H.Schulzrinne: Reliability of Provisional Responses in SIP, March 2, 2001, http://www.ietf.org/internet-drafts/draft-ietf-sip-100rel-03.txt

[7] Internet Draft, SIP WG, S.Donovan, J.Rosenberg: The SIP Session Timer, November 22, 2000, http://www.ietf.org/internet-drafts/draft-ietf-sip-session-timer-04.txt

[8] Internet Draft, SIP WG, W. Marshall, et all: SIP Extensions for supporting Distributed Call State, February, 2001, http://www.ietf.org/internet-drafts/draft-ietf-sip-state-01.txt

[9] Internet Draft, SIP WG, Schulzrinne/Rosenberg: SIP Caller Preferences and Callee Capabilities, November 24, 2000, http://www.ietf.org/internet-drafts/draft-ietf-sip-callerprefs-03.txt

[10] Internet Draft, SIP WG, W. Marshall, et al: Integration of Resource Management and SIP, February, 2001, http://www.ietf.org/internet-drafts/draft-ietf-sip-manyfolks-resource-01

[11] Internet Draft, R. Sparks: SIP Call Control – Transfer, February 26, 2001, http://www.ietf.org/internet-drafts/draft-ietf-sip-cc-transfer-04.txt

[12] Internet Draft, SIP WG, W. Marshall, et al: SIP Extensions for Caller Identity and Privacy February, 2001, http://www.ietf.org/internet-drafts/draft-ietf-sip-privacy-01.txt

[13] Adam Roach: Event Notification in SIP, Internet Draft, February 2001, http://www.ietf.org/internet-drafts/draft-roach-sip-subscribe-notify-03.txt

[14] RFC2976, Network WG, S. Donovan: The SIP INFO Method, October 2000, http://www.ietf.org/rfc/rfc2976.txt

[15] RFC2778, Network WG, M. Day, J. Rosenberg, H. Sugano: A Model for Presence and Instant Messaging, February 2000, http://www.faqs.org/rfcs/rfc2778.html

[16] Internet Draft, SIMPLE WG, Rosenberg et al: SIP Extensions for Presence, March 2, 2001, http://www.cs.columbia.edu/sip/drafts/draft-rosenberg-impp-presence-01.txt

[17] Internet Draft, IMPP WG, Jonathan Rosenberg et.al: SIP Extensions for Presence Authorization, June 15, 2000, http://www.cs.columbia.edu/sip/drafts/draft-rosenberg-impp-qauth-00.txt

[18] Internet-Draft, J. Rosenberg, et al: SIP Extensions for Instant Messaging, February 28, 2001, http://www.cs.columbia.edu/sip/drafts/draft-rosenberg-impp-im-01.txt

[19] 14 2000, http://www.softarmor.com/sipwg/teams/sipt/index.html

[20] Ericsson: Best Current Practice for ISUP to SIP mapping , IETF, September 2000, http://www.softarmor.com/sipwg/teams/sipt/index.html

[21] Phillips Omnicom: Voice over IP, Phillips Omnicom, July 2000.HERTS SG1 1EL – UK

[22] Srinivas sreemanthula etc: 'RT Hard Handoff Concept for All-IP System, version V1.0.2, and IPMN project.

A transport protocol for SIP

Gonzalo Camarillo
Advanced Signalling Research Lab.
Ericsson
Finland
Gonzalo.Camarillo@ericsson.com

Henning Schulzrinne
Department of Computer Science
Columbia University
USA
hgs@cs.columbia.edu

Raimo Kantola
Networking Laboratory
Helsinki University of Technology
Finland
Raimo.Kantola@hut.fi

Abstract

Current SIP implementation typically use TCP or UDP as a transport protocol. The differences between SIP over UDP and SIP over TCP have already been analyzed and are relatively well-known. However, there have not been so far SIP implementations that use SCTP as a transport. This paper analyzes the advantages that can be derived from the use of SCTP as a transport for SIP. It shows how while SCTP is an excellent transport protocol for high levels of traffic its performance decreases when the number of SIP transactions transmitted in parallel decreases.

1 Introduction

The Session Initiation Protocol (SIP) is an application-layer protocol for creating, modifying and terminating sessions. SIP [1] is designed in a modular way so that it is independent of the type of session established and of the lower-layer transport protocol used. Its modularity is one of the most important strengths of SIP. It makes SIP flexible and easy to extend with new features.

The SIP specification describes how the protocol operates over TCP [2] and over UDP [3]. Both transport protocols have different characteristics and provide a particular SIP application with different services. TCP provides reliable in-order transfer of bytes while UDP does not ensure neither reliability nor in-order delivery. Both UDP and TCP present certain advantages and disadvantages, and also both of them present certain limitations regarding signalling transport.

The limitations present in TCP and UDP for transporting signalling traffic led to the design of a new transport protocol within the IETF. The SIGTRAN working group developed the Stream Control Transmission Protocol (SCTP). SCTP [4] was first intended to transport telephony signalling over an unreliable network such as an IP network. However, the protocol has been designed so that SCTP can be used as a general-purpose transport protocol.

There have been already attempts to define SIP operation on top of SCTP [5]. However, although there are already implementations of telephony signalling protocols such as ISUP on top of SCTP, so far there has not been any implementation of SIP over SCTP that could show the gains that SCTP might achieve. This document discusses advantages and disadvantages derived from the use of SCTP as a transport protocol for SIP.

The remainder of this document is organized as follows. Section 2 and 3 describes SIP operation on top of TCP and UDP respectively. Pros and cons of each protocol are analyzed. Section 4 provides an introduction to SCTP. Section 5 analyzes advantages and disadvantages of using SCTP as a transport for SIP and finally section 6 outlines some conclusions.

2 SIP over TCP

The natural choice to transport a signalling protocol whose messages have to be reliably delivered to the destination seems to be a reliable transport protocol. Since the most widespread reliable transport protocol is TCP, it would not have been surprising if SIP had been designed to run only over TCP. Besides, SIP is based on HTTP [6], which uses TCP as a transport.

However, TCP presents some limitations regarding signalling transport. Therefore, SIP was designed to be independent of the transport protocol. This way, SIP can also run over UDP overcoming some of TCP's limitations. At present, UDP is the most widespread transport for SIP.

2.1 TCP limitations

TCP was designed to transport large amounts of data between two end-points. Once a connection is established, TCP implements flow control and error correction based on the dynamic behavior of the end-

to-end traffic. However, signalling traffic does not consist of large amounts of data. Signalling traffic usually consists of small bursts of information. TCP's flow control mechanisms are not designed for such as traffic pattern, and therefore do not perform as well as it might be expected.

Fast retransmit algorithm

When a large bulk of data is being transmitted by TCP, ack messages from the receiver are continuously received indicating which segments have been successfully received. The receiver sends duplicate acks when out-of-order segments arrive. Thus, arrival of duplicate acks indicates that a segment was lost. Therefore, the sender retransmits it without waiting for a timeout. This mechanism is referred to as fast retransmit and it is used together with the fast recovery algorithm.

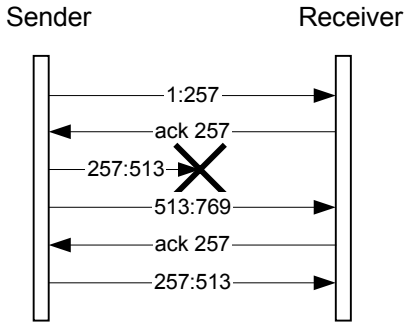


Figure 13 : Fast retransmit

Note that the sender in figure 1 retransmit the missing segment upon reception of a duplicate ack. This flow has
been simplified. A typical implementation waits until three duplicate acks are received before retransmitting a segment.

Figure 1 shows how TCP behaves when large bulks of data are transmitted. Retransmissions are usually triggered by duplicate acks rather than timeouts. This is the reason why TCP timeouts are relatively high, in the order of 1,5 seconds. This allows using the fast retransmit algorithm before a timeout occurs.

However, SIP messages are relatively small, in the order of 500 bytes. A SIP message usually fits into a TCP segment. So, if a TCP segment that contains a SIP message gets lost, TCP will not be able to receive duplicate acks, since it is not sending any more data. Therefore, TCP will have to wait for a timeout in order to retransmit the missing segment. This results in a too conservative retransmission policy when TCP transports SIP signalling.

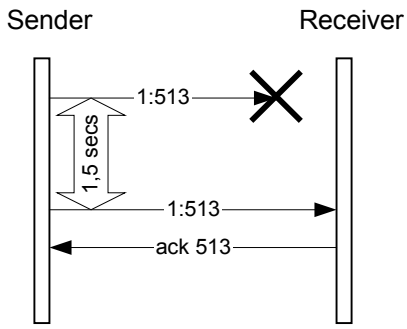


Figure 2 : TCP timeout

TCP connection establishment

TCP performs a three-way handshake before any user data can be transmitted between both ends. In a long-lived connection, the connection establishment time is negligible compared to the whole connection duration. However, signalling traffic is delay sensitive. If a SIP UAC wants to send an INVITE over TCP it will have to wait until the TCP connection is established before sending the INVITE.

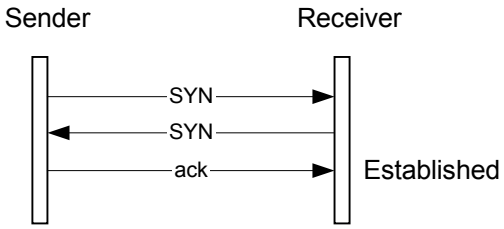


Figure 3 : TCP three-way handshake

The receiver of figure 3 will not pass any data to the application until it does not reach the "established" state. This overhead is not acceptable when the user is expecting an answer for his INVITE.

TCP implements a special timer for connection establishment. When a SYN gets lost, a typical implementation retransmits it after 6 seconds. Therefore, a single packet loss increases enormously the connection establishment delay introduced by TCP.

2.2 Multiple SIP sessions

One straightforward attempt to resolve both issues previously described consists of bundling several SIP sessions into a single TCP connection. With a high number of SIP sessions the TCP connection transports data continuously so that packet losses are detected by receiving duplicated acks rather than by timeouts. This increases the performance of TCP and reduces the delay introduced to SIP messages.

Another advantage of bundling SIP sessions is that the first SIP message of a new session does not have to wait for a new TCP connection to be established before being transmitted. Since the TCP connection is already

established SIP messages belonging to a new SIP session are not affected by any additional delay. They can be sent immediately.

A SIP UAC usually handles a single SIP session, but proxies in the network have several ongoing SIP sessions between them at the same time. Therefore, proxies handling a high number of SIP sessions can typically take advantage of bundling SIP sessions. Another example were bundling can be performed is between a large gateway towards the PSTN and its outbound proxy.

Byte stream service

However, TCP presents an important limitation regarding bundling of sessions. TCP provides ordered delivery of a stream of bytes. When TCP is used to transmit messages it preserves the order in which the messages were sent by the sender. This property causes interaction problems between different SIP sessions carried on a single TCP connection.

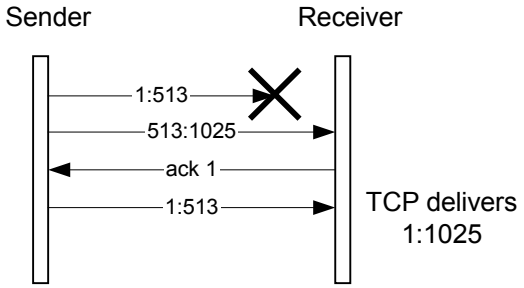


Figure 4 : TCP provides ordered delivery

The sender of figure 4 sends two INVITEs that belong to different sessions using the same TCP connection. The segment carrying the first INVITE gets lost (1:513), but the segment carrying the second INVITE arrives properly to the receiver (513:1025). However, since TCP provides ordered delivery, it will not delivered the second INVITE to the application until it has delivered the first INVITE. Therefore, the second INVITE is delayed until the first INVITE is retransmitted. The consequence is that a particular SIP session might suffer delay without having experimented any packet loss, as it is shown in figure 4.

3 SIP over UDP

Transporting SIP over UDP overcomes some of the problems associated with TCP. UDP is a connectionless protocol. Thus, it does not perform any kind of connection establishment before sending data. Therefore, a particular INVITE will be sent encapsulated in a UDP packet without any establishment delay introduced by the transport protocol.

Since UDP does not provide reliable transport, reliable delivery is achieved through application level retransmissions. The SIP application retransmit a particular SIP messages when the retransmission timer expires. This retransmission timer is lower than in TCP. Its default value is 0,5 seconds. Therefore, the retransmission policy of SIP when it runs over UDP is more aggressive than when it runs over TCP.

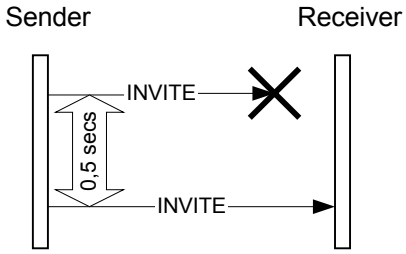


Figure 5 : SIP retranmission policy using UDP

SIP can afford to have a more aggressive retransmission policy over UDP than TCP because it transmits a small number of small messages. Therefore, SIP assumes that it is not going to congest the network because they are retransmitted more often than TCP.

Therefore, when a single or a small number of SIP sessions are handled, UDP is a better choice than TCP. However, UDP, as opposed to TCP, does not hide retransmissions from the application layer. Thus, although a SIP application using UDP has to store more state information than when TCP is used this does not represent an important issue for most of the applications.

3.1 Multiple SIP sessions

When there are multiple SIP sessions between two proxies they can be bundled in a single TCP session to take advantage of the congestion control mechanisms built in TCP. Losses are detected before and thus, performance improves.

However, when UDP is used, the same retransmission timers apply to every session. This can lead to a poorer performance and even to network congestion, since UDP does not provide congestion information to the application and by default SIP uses a more aggressive retransmission policy than TCP.

Therefore, for proxies handling a large amount of connections, the choice between UDP and TCP is not clear. TCP presents the previously described head of the line blocking issue and UDP does not implement any congestion control mechanism. The choice between TCP and UDP depends on how the network is loaded at a certain moment and the RTT between sender and receiver.

4 SCTP

The Stream Control Transmission Protocol (SCTP) is intended to resolve the issues derived from the use of TCP and UDP when there are multiple SIP sessions between sender and receiver. SCTP [4] also provides a certain level of fault tolerance through multihoming.

4.1 SCTP connection establishment

SCTP is a connection oriented transport protocol. In SCTP terminology, a connection is referred to as an association. An association is established through a four-way handshake in which the last two messages can already carry user data.

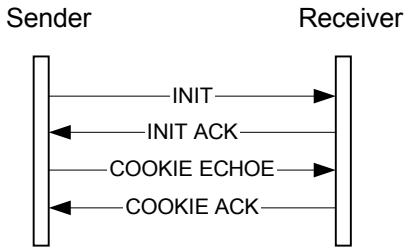


Figure 6 : SCTP four-way handshake

In this handshake end users exchange one or multiple IP addresses or host names. One destination address will be marked as the primary. The rest of them will be used in case the primary destination becomes unavailable. This feature, known as multihoming, allows a SCTP connection to survive network failures. The data is just sent to another destination address in case of failure.

The four-way handshake provides also a certain level of protection against resource attacks. The receiver, upon reception of an INIT message sends back a cookie in the INIT ACK. The receiver does not allocate any resources for this SCTP association until it receives the same cookie in the COOKIE ECHOE message. This way, resources are allocated when it is ensured that the party sending the INIT message is really willing to establish an SCTP association.

4.2 Multiple streams within an association

SCTP provides multiplexing/demultiplexing capabilities within an association. A single association can contain several streams. Each stream is identified by its stream id. During the four-way handshake the number of streams in both directions is negotiated.

An association can contain several types of streams. The base SCTP specification [4] defines two services: reliable ordered delivery and reliable unordered delivery. However, there are extensions [7] that provide an unreliable delivery service.

It is important to note that a particular service is provided on stream basis. Therefore, one stream within an association might be an ordered stream while another is unordered.

4.3 Flow and congestion control per association

Even if an association contains several streams, SCTP performs flow and congestion control per association. This allows to use the behavior of all the traffic within the association as input for the flow control mechanisms, which are effectively very similar to the ones used by TCP.

For instance, the fast retransmit algorithm can be used effectively without waiting for timeouts in order to retransmit data. Figure 7 shows how stream demultiplexing and flow control work together in an example.

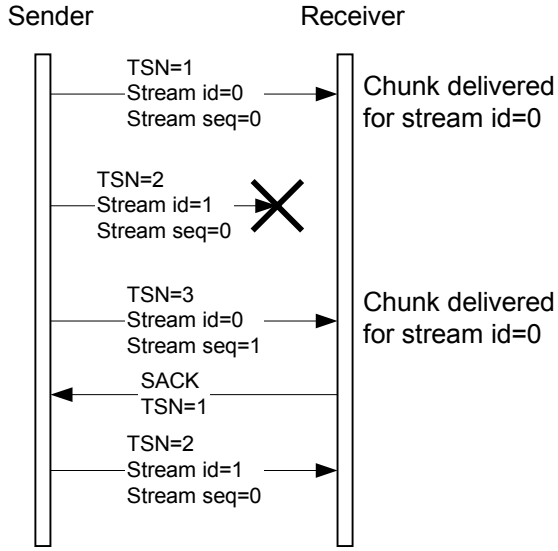


Figure 7 : Multiple streams within an association

The association of figure 7 consists of two ordered streams (stream id=0 and stream id=1). SCTP implements a general sequence number space (Transmission Sequence Number) and a sequence number space per stream. The general TSN is used to perform flow control and packet loss recovery and the stream sequence numbers are used to deliver individual streams.

When the message with TSN=3 arrives to the receiver, this knows that TSN=2 is lost. However, it also knows that TSN=3 is the next packet of stream id=0 (Stream seq=1). Therefore, it delivers the packet to the application without waiting to receive TSN=2. In the SACK (Selective ACK) the receiver reports that TSN=2 was missing.

Therefore, losses in one stream do not introduce delay on other streams. Besides, since the whole association is used to perform flow control, the sender detects that TSN=2 got lost thanks to the SACK sent upon

reception of TSN=3, that belongs to a different stream. This way, SCTP does not have to wait for a timeout to retransmit TSN=2.

So, SCTP combines good features of both TCP and UDP. It bundles streams to take advantage of flow control mechanisms and delivers separately packets belonging to different streams.

5 SIP over SCTP

It seems clear that proxies that handle multiple SIP sessions between them can obtain a better performance using an SCTP association than using TCP or UDP. If each SIP session is sent over an ordered stream, SIP messages can take advantage of flow control without being delayed by lost messages from other sessions.

However, even when multiple ordered streams are used, it is still possible that messages are delayed by other messages belonging to the same SIP session. The example of figure 8 shows how the loss of a provisional response can delay the delivery of the final response which was successfully received.

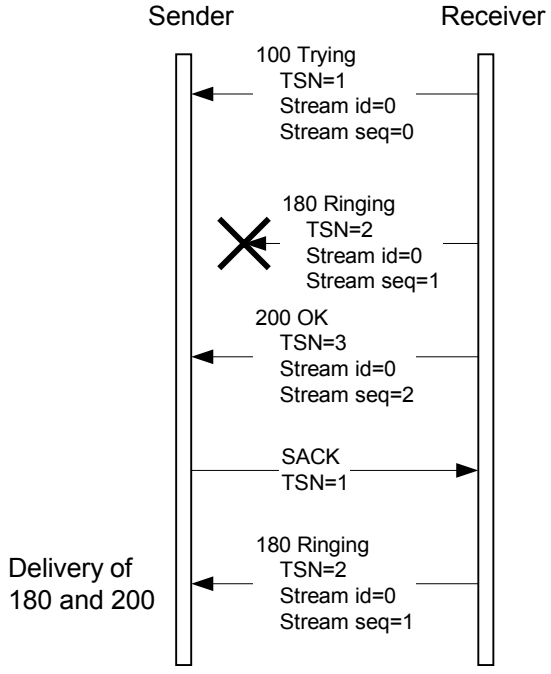


Figure 8 : SIP over ordered SCTP streams

In figure 8 all SIP responses are sent over an ordered SCTP stream (stream id=0). Therefore, SCTP delivers messages to the application in order within the stream. Since the provisional response "180 Ringing" got lost, SCTP cannot deliver the final response "200 OK" to the application. SCTP waits until TSN=2 arrives before delivering both responses.

Unordered service for final responses

In order to overcome this problem SIP final responses can be sent using the SCTP unordered service. SCTP allows to send unordered messages within an ordered stream. Therefore, all SIP messages within a SIP session are still sent using the same stream, but messages carrying final responses are sent with the SCTP unordered flag set.

Note that a receiver performs demultiplexing of incoming SIP messages based on the Call-ID of the SIP message rather than on the SCTP stream id. Stream ids are used here to solve the head of the line blocking problem. They are not intended to provide further demupltiplexing.

General unordered service

The method just described would be the most efficient way of transporting SIP over SCTP. However, there is a simpler mechanism that behaves nearly as well and simplifies implementations. It consists of sending all SIP traffic using the SCTP unordered service. When all SIP messages are sent with the unordered flag set SCTP delivers any message received immediately, independently of which stream the message belongs to. Thus, SIP entities can perfectly use the same stream id for all SIP sessions.

This mechanism is simpler because an implementation does not have to ensure that SIP messages belonging to a particular SIP session are always sent using the same stream id. Implementation that are not willing to perform stream id management should use this mechanism.

An example of such an implementation is a proxy that does not store state information about SIP transactions (stateless) but has SCTP associations continuously open to send SIP messages to certain common destinations.

Note that the use of a hash of the Call-ID of a SIP message module the number of SCTP streams available in order to choose the outgoing stream id for the message has some limitations. Although with a high number of available streams it is not likely no happen, a system using this method might end up sending requests with different Call-IDs using the same stream id. This would result in the head of the line blocking problem previously mentioned.

These two methods have the advantage of interworking together. Any receiver is able to receive traffic from senders using any of both mechanisms.

5.1 Differences between both methods

The only difference between both methods is that sending just final responses with the SCTP unordered flag set avoids re-ordering of requests and provisional responses in the parts of the path where SCTP is used.

However, there are just a few scenarios where this can happen.

Provisional responses

Provisional responses are sent unreliably by SIP. SIP systems do not rely on provisional responses to drive any protocol state machine. Therefore, receiving out of order provisional responses does not represent a problem for a SIP UAs.

When a SIP UA is interested in provisional responses it uses the extension defined in [9]. Then, provisional responses are transmitted reliably. [9] recommends SIP servers sending provisional responses not to send subsequent responses until the previous one has been acknowledged with a PRACK. Thus, using ordered or unordered SCTP to transport provisional responses does not make a difference, since the SIP layer ensures that they are received in order.

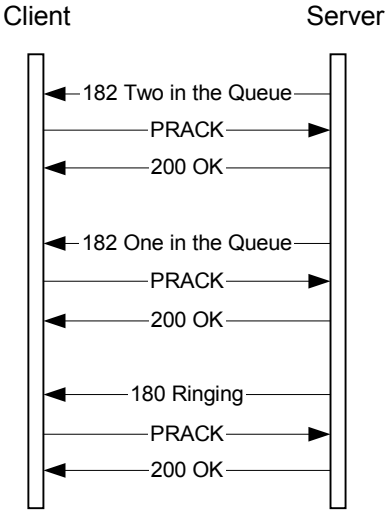


Figure 9 : SIP ensures in order delivery

Requests

The behavior of a SIP entity sending requests is similar to the one described for reliable provisional responses. A SIP client does not send a request until the previous transaction has completed. There are two exceptions to this rule, but in general it does not make a difference the transport used (ordered or unordered) for requests either.

The only two exceptions when a SIP client sends overlapping requests are: an INVITE followed by a CANCEL and an INVITE followed by a BYE. Note that other methods such as COMET or PRACK are just sent after a response for the INVITE has been received. Note also that CANCEL can terminate any request other than CANCEL and ACK. However, since non-INVITE requests are responded immediately by the server, CANCEL is typically used only for INVITE requests.

These two situations are the only ones where both uses of SCTP described previously differ. If the requests are sent unordered, a CANCEL or a BYE might overtake the INVITE sent before. Ordered SCTP ensures that they arrive in the same order as they were sent. However, this is only ensured in the part of the path where ordered SCTP is used. If other transport protocol such as UDP is used in another part of the path, reordering can still happen. Therefore, even systems using ordered SCTP have to be prepared to handle out of order CANCELs and BYEs. Figure 10 shows how a system using ordered SCTP might still receive out of order requests.

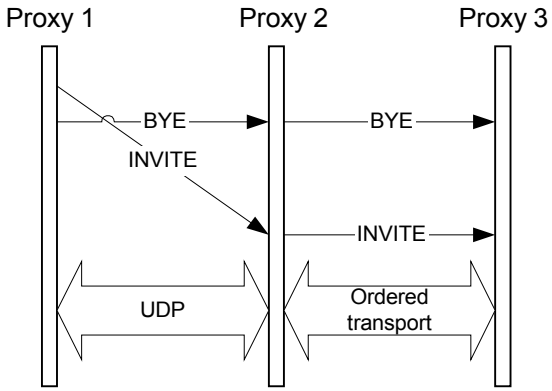


Figure 10 : INVITE followed by a BYE

Even if ordered SCTP streams are used, a SIP entity has to be prepared to received a BYE before an INVITE. A “481 Transaction Does Not exist” will be sent as response to the BYE.

Therefore, the only difference between sending all the SIP traffic with the SCTP unordered flag set and sending just final responses with this flag is that the likelihood of receiving a BYE or a CANCEL before an INVITE decreases using the latter method, although it might still happen.

5.2 Other strengths of SCTP

The previous section described SCTP behaves like a TCP connection without the head of the line blocking problem. Besides resolving this problem, SCTP has some other strengths that SIP can take advantage of.

Message based

SCTP is a message-oriented protocol, as opposed to TCP that is stream oriented. SCTP delivers messages while TCP delivers a stream of bytes. This makes it possible for SCTP to provide unordered delivery of SIP messages. In TCP this concept would not make any sense, since delivering unordered bytes would be useless for an application.

Message-oriented protocols such as SCTP or UDP allow implementing simpler parsers. When these

transport protocols deliver a message to the application it contains a single SIP message. In order to parse a SIP message received over TCP it is necessary to implement application level boundaries such as the SIP Content-Length header.

Transport-layer fragmentation

However, although both SCTP and UDP are message-oriented transport protocols, SCTP has an advantage over UDP. SCTP implements transport-level fragmentation while UDP does not. If a SIP message inside a UDP packet is larger than the path MTU the packet will be fragmented at the IP layer.

IP-layer fragmentation presents several problems. The likelihood of having packet losses increases and firewall and NAT traversal becomes impossible. The fragments of the UDP packet do not carry the UDP header, which contains the source and the destination port number of the UDP packet. Therefore, network devices that need to examine port numbers will simply discard the packets.

SCTP implements transport-layer fragmentation. Messages larger than the path MTU are transported in different SCTP chunks. Every chunk carries complete transport information, and thus, problems derived from IP fragmentation are avoided. Different chunks are reassemble at the destination and delivered to the application as a single message.

Currently fragmentation does not represent a serious problem for SIP, since SIP messages are usually smaller than the path MTU. However, new session description protocols or new SIP extensions might increase the size of SIP messages. SCTP fragmentation would then represent an important advantage.

Bundling of chunks

Figure 11 shows the format of a SCTP packet. It contains a common header and several chunks. Unless fragmentation is performed, a chunk contains an application-level message.

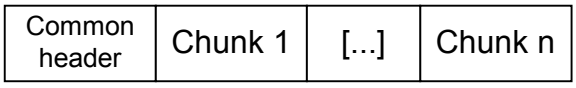


Figure 11 : SCTP message format

Therefore, a single SCTP packet can carry several SIP messages that belong to different sessions. Bundling SCTP chunks decreases the number of packets sent through the network. This avoids certain congestion problems in IP routers and typically achieves a better performance than sending various individual packets.

Multihoming

SCTP provides several source and destination addresses within an association. They are intended to provide alternative paths to be used in case of network failures. This feature increases the reliability of an association.

Multiple destination addresses are not intended to provide a load balancing mechanism. SCTP marks one address as the primary, and all the traffic is routed to that address until it fails. Other mechanisms such as DNS SRV [8] records might be used to provide load balancing. SCTP multihoming just provides a fail over mechanism.

5.3 A single SIP session over SCTP

It is clear that SIP entities that handle a high amount of SIP traffic between them can take advantage of SCTP and all its features. However, SCTP advantages are not so evident when a single SIP session (or a small number of them) is transported. In this scenario SCTP shares some problems that TCP has. SCTP association establishment delays the delivery of the first INVITE, and once the association is established, SCTP timeouts are more conservative than the ones used by SIP over UDP. The initial value for the SCTP retransmission timer is 3 seconds and even when RTT measurements are performed its minimum value is 1 second.

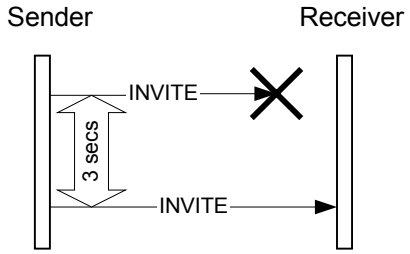


Figure 12 : SCTP's initial retransmission timer

The only advantage of SCTP over UDP in a scenario with low level of SIP traffic is the transport-layer fragmentation provided by SCTP, since multihoming can be achieved using UDP in conjunction with DNS SRV records.

6 Conclusions

The best transport protocol for SIP depends on the amount of SIP traffic that a particular SIP entity handles. SIP entities that handle a large amount of SIP traffic between them such as proxies and large SIP gateways have in SCTP their best choice. SCTP bundles together several SIP sessions into a single SCTP association and then performs flow and congestion control per association. This way, packet losses are detected before retransmission timers expire leading to an increase in the overall performance. Among all the possible services provided by SCTP, unordered delivery and ordered delivery with unordered final responses are the ones that suit SIP better.

However, SIP entities that handle a small number of SIP sessions such as the SIP UA of a individual user cannot take advantage of the flow control provided by SCTP. When a small number of SIP messages are transported over SCTP packet losses are detected by timeouts. This leads to a too conservative retransmission policy, since timers in SCTP are not designed for situations where the traffic load is very low. Therefore, small SIP entities have in UDP their best choice. UDP does not introduce any connection establishment time and retransmit lost packets in a more aggressive way than SCTP. However, since SIP applications using UDP do not perform any congestion control other than implementing a back-off retransmission timer, the use of UDP is not recommended for high volumes of SIP traffic.

While TCP is an excellent protocol for transferring large amounts of data such as files or the contents of a particular web page, it presents important limitation regarding signalling transport. Therefore, depending on the SIP entity, UDP or SCTP are better choices to transport SIP signalling.

Acronyms

ACK: Acknowledgement
DNS: Domain Name System
HTTP: HyperText Transfer Protocol
IP: Internet Protocol
ISDN: Integrated Services Digital Network
ISUP: ISDN User Part Protocol
MTU: Maximum Transmission Unit
NAT: Network Address Translator
PRACK: Provisional ACK
PSTN: Public Switched Telephone Network
RTT: Round Trip Time
SACK: Selective ACK
SCTP: Stream Control Transmission Protocol
SIGTRAN: Signalling Transport
SIP: Session Initiation Protocol
SYN: Synchronize sequence numbers flag
TCP: Transmission Control Protocol
TSN: Transmission Sequence Number
UA: User Agent
UAC: User Agent Client
UDP: User Datagram Protocol

References

[1] Handley M., Schulzrinne H., Schooler E., Rosenberg J., “SIP: Session Initiation Protocol”, RFC 2543. IETF. March 1999.

[2] Postel J., “Transmission Control Protocol”, RFC 793. IETF. September 1981.

[3] Postel J, “User Datagram Protocol”, RFC 768. IETF. August 1980.

[4] Stewart R., Xie Q., Morneault K., Sharp C., Schwarzbauer H., Taylor T., Rytina I., Kalla M., Zhang L., Paxson V., “Stream Control Transmission Protocol”, RFC 2960. IETF. October 2000

[5] Rosenberg J, Schulzrinne H., “SCTP as a Transport for SIP”, draft-rosenberg-sip-sctp-00.txt. IETF. Jone 2000. Work in progress.

[6] Fielding R., Gettys J., Mogul J., Frystyk H., Berners-Lee T., “Hypertext Transfer Protocol -- HTTP/1.1”, RFC 2068. IETF. January 1997.

[7] Xie Q., Stewart R., Sharp C., Rytina I., “SCTP Unreliable Data Mode Extension”, draft-ietf-sigtran-usctp-01.txt. IETF. February 2001. Work in progress.

[8] Gulbrandsen A., Vixie P., Esibov L., “A DNS RR for specifying the location of services (DNS SRV)”, RFC 2782. IETF. February 2000.

[9] Rosenberg J., Schulzrinne H., “Reliability of Provisional Responses in SIP”, draft-ietf-sip-100rel-03.txt. IETF. March 2001. Work in progress.

Session Initiation Protocol in 3G

Tuomo Sipilä
Nokia Research Center, Helsinki, Finland
tuomo.sipila@nokia.com

Abstract

This article gives an overview of the 3GPP IP Multimedia subsystem and how SIP protocol is used with it. Also the 3G required changes to SIP are identified. In addition some key problems with SIP protocol are identified with regards to 3G mobile networks.

1 Introduction

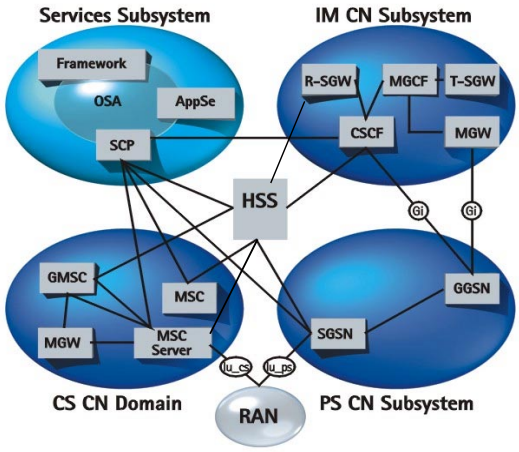
The 3rd Generation Partnership Project started the development of IP based multimedia services to the 3rd generation (3G) mobile networks, known as UMTS in Europe and IMT-2000 in Japan, during autumn 1999. The initiation and pressure for the work came through an organisation called 3G.IP which is a group formed to push IP based ideas to 3G networks. The target was to standardise the required enhancements for the 3G network so that IP telephony and multimedia can be provided with equal user perceived quality as with the current mobile network services. Another requirement was that the future 3G network could function fully based on packet and IP connections without the traditional circuit switched domain. Essentially also the IP multimedia would in the future provide via IP a wider and more flexible service set than the current networks. In Spring 2000 IETF defined Session Initiation Protocol (SIP) was selected as the base protocol that shall provide the IP Multimedia sessions to the mobile terminals (UE). This decision tied the 3GPP solution and work into co-operation with IETF.

Already during year 2000 it became evident that the specification work would take longer that expected since it requires specification of a completely new network subsystem with all required mobile functions. Therefore the specification target was set to the end of 2001 when the 3GPP Release 5 specifications should be finalised.

2 3GPP Rel5 system architecture

The 3GPP Release 5 architecture is illustrated in the figure 1. The 3GPP mobile system consists of the following network domains:

- **Radio Access Network Domain (RAN)** consists of the physical entities, which manage the resources of the radio access network, and provides the user with a mechanism to access the core network. It can be either WCDMA based UTRAN or GSM/EDGE based GERAN
- **Circuit Switched Core Network Domain (CS CN)** comprises all core network elements for provision of Circuit switched services
- **Packet Switched Core Network Subsystem (PS CN)** comprises all core network elements for provision of PS connectivity services i.e. guaranteeing the IP flow to and from the mobile terminal (UE)
- **IP Multimedia Core Network Subsystem (IMSS)** contains all the network elements that are used to provide the IP base multimedia services
- **Service Subsystem** Comprises all elements providing capabilities to support operator specific services (e.g. IN and OSA)



- **Figure 1: 3GPP Release 5 network architecture and domains (not all elements are shown)**

One of the fundamental principles of the 3GPP network architecture is to maintain to large extent an independence of the network domains and subsystem so that independent evolution is possible. This means that the IP multimedia is seen to large extent as an transparent service through the Radio Access and Packet Core Network domains. Other names for this is layered approach and access indepdendency.

IP Multimedia Core Network Subsystem (IMSS) and its functionality is the main focus of this article. For

making the system functionality more understandable the PS CN subsystem functionality is briefly illustrated. Also the IMSS linkage with the service subsystem that provides the open service generation is briefly mentioned.

2.1 PS CN Subsystem

The PS CN subsystem main functions are to establish and maintain the connection between the terminal and the GGSN, route the IP packets in both directions and do charging. The Packet Switched Core Network subsystem consists of the following GPRS based network elements and functions [4]:

- **Serving GPRS Support Node** which maintains the subscription data (identities and addresses) and follows the location of the terminal within the network
- **Gateway GPRS Support Node** which maintains the subscription information, allocated IP addresses and follows the SGSN under which the terminal is.

The PS CN subsystem in connected to the IMSS via Go and Gi interfaces that are located in the GGSN. The Gi interface is the one that is also used for standard Internet access and it is relatively transparent. The Go interface is used for policy control between IM Subsystem and GGSN and packet core. The reasons for policy control is to allow the operators to limit the utilisation of the best 3G packet QoS classes to their own IP Multimedia services.

The IP connections between terminal (UE) and GGSN are provided by PDP contexts . At PDP context establishment the used QoS profile and terminal IP address are allocated.

3 IP Multimedia Subsystem

3.1 Requirements

The 3GPP TS 22.228 [6] specifies the following main service requirements for the IP Multimedia Subsystem:

- **at least equal end-to-end QoS for voice as in circuit switched** (AMR Codec based) wireless systems
- **equal privacy, security or authentication as in GPRS and circuit switched services**
- **QoS negotiation possibility** for IP sessions and media components by both ends
- **access independence** i.e. the IP Multimedia network and protocols evolve independently of radio access (WCDMA, EDGE/GSM/GPRS, WLAN etc)
- **applications shall not be standardised**
- **IP policy control possible** i.e the operators shall have the means to control which IP flows use the real-time QoS bearers
- **automated roaming** with the services in home and visited network
- **hide the operator network topology** from users and home/visited network. The network topology is regarded as a key competitive factor between operators
- **the resources shall be made available before the destination alerts**
- identification of the entities with **either SIP URL or E.164 number**
- procedures for incoming and outgoing calls, emergency calls, presentation of originator identity, negotiation, accepting or rejecting incoming sessions., suspending, resuming or modifying the sessions
- user shall have the choice to select which session components reject or accept

3.2 Architecture

The IP Multimedia subsystem current architecture showing the functions (March 2001) is in figure 2. Note that several of the illustrated functions can be merged into real network elements. The functions and their purposes are clarified in the following subsections. It should be noted that the standardisation for the system is still ongoing so changes can be expected.

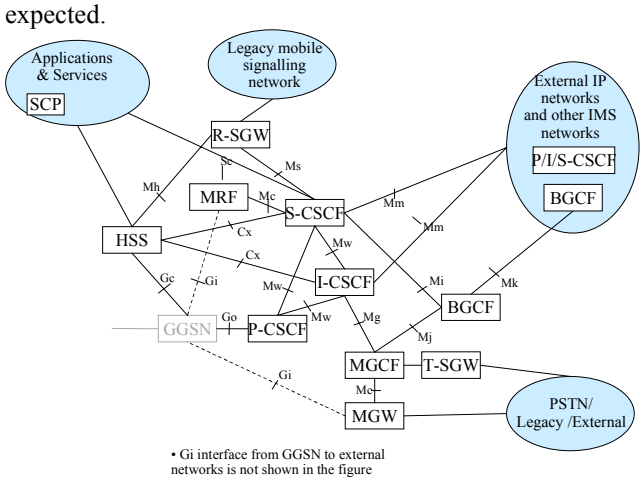


Figure 2: IP Multimedia Subsystem

3.3 HSS

Home Subscriber Server (HSS) is a combination of the currently existing UMTS/GSM HLR and the needed register functions for IP Multimedia Subsystem. HSS will provide the following functions:

- user identification, numbering and addressing information.
- user security information: Network access control information for authentication and authorisation
- user location information at inter-system level; HSS handles the user registration, and stores inter-system location information, etc.
- the user profile (services, service specific information…) [3]

3.4 P-CSCF

Proxy Call State Control Function (P-CSCF) performs the following functions:

- Is the first contact point for UE within IM CN subsystem, forwards the registration to the I-CSCF to find the S-CSCF and after that forwards the SIP messages between UE and I-CSCF/S-CSCF
- Behaves as like a proxy in RFC 2543 [9]i.e. accepts requests and services the internally or forwards them possibly after translation
- may behave also like a RFC 2543 [9] User agent i.e. in abnormal conditions it may terminate and independently generate SIP transactions
- is discovered using DHCP during registration or the address is sent with PDP context activation
- may modify the URI of outgoing requests according to the local operator rules (e.g. perform number analysis, detect local service numbers)
- detect and forward emergency calls to local S-CSCF
- generation of charging information
- maintains security association between itself and UE, also provides security towards S-CSCF
- provides the policy control function (PCF)
- authorisation of bearer resources, QoS management and Security issues are currently open in standardisation [3].

3.5 I-CSCF

Interrogating Call State Control Function (I-CSCF) performs the following functions:

- is the contact point within an operator's network for all connections destined to a subscriber of that network operator, or a roaming subscriber currently located within that network operator's service area. It can be regarded as a kind of firewall between the external IMSS and the operators internal IMSS network. There may be multiple I-CSCFs within an operator's network
- Assigns a S-CSCF to a user performing SIP registration
- Route a SIP request received from another network towards the S-CSCF
- Obtains from HSS the Address of the S-CSCF
- charging and resource utilisation
- in performing the above functions the operator may use I-CSCF to hide the configuration, capacity, and topology of the its network from the outside
- additional functions related to inter-operator security are for further study

3.6 S-CSCF

Serving Call State Control Function) (S-CSCF) performs the following functions:

- performs the session control services for the terminal. Within an operator's network, different S-CSCFs may have different functionality
- It maintains session state and has the session control for the registered endpoint's sessions
- Acts like a Registrar defined in the RFC2543[9], i.e. it accepts Register requests and makes its information available through the location server (e.g. HSS)
- may also behave as a proxy or as a user agent as defined by RFC 2543 [9]
- Interacts with Services Platforms for the support of Services
- obtain the address of the destination I-CSCF based on the dialled number or SIP URL
- on behalf of a UE forward the SIP requests or responses to a P-CSCF or an I-CSCF if an I-CSCF is used in the path in the roaming case
- generates charging information
- Security issues are currently open in standardisation [3]

3.7 MGCF

Media Gateway Control Function (MGCF) Provides the following functions:

- protocol conversion between ISUP and SIP
- routes incoming calls to appropriate CSCF
- controls MGW resources [3]

3.8 MGW

Media Gateway (MGW) provides the following functions:

- Transcoding between PSTN and 3G voice codecs
- Termination of SCN bearer channels
- Termination of RTP streams [3]

3.9 T-SGW

Transport Signalling Gateway provides the following functions

- Maps call related signalling from/to PSTN/PLMN on an IP bearer
- Provides PSTN/PLMN <-> IP transport level address mapping [3]

3.10 MRF

Multimedia Resource Function provides the following functions:

- Performs multiparty call and multimedia conferencing functions [3]

3.11 BGCF

The S-CSCF, possibly in conjunction with an application server, shall determine that the session should be forwarded to the PSTN. The S-CSCF will forward the Invite information flow to the Breakout Gateway control function (BGCF) in the same network.

The BGCF selects the network in which the interworking should occur based on local policy. If the BGCF determines that the interworking should occur in the same network, then the BGCF selects the MGCF which will perform the interworking, otherwise the

BGCF forward the invite information flow to the BGCF in the selected network. The MGCF will perform the interworking to the PSTN and control the MGW for the media conversions

3.12 IMSS functionality

Figure 3 shows the call model in mobile to mobile call case when both callee and caller are roaming. In the roaming case i.e. when a user roams to network that is outside his home network the IP multimedia services are provided by the S-CSCF in the home network. The P-CSCF in the visited network forwards the service request to the home network. However in some cases some services can be provided directly via the visited network i.e. by the P-CSCF. The P-CSCF is needed in the home network to allow for the network flexiblity because S-CSCFs may contain different services and also in the roaming case allow the visited operator handle the call and provide local services. The local services can be an emergency call or other localised services such as services related to geographical location of the user or local numbering plans. I-CSCF is acting as a protective firewall between home and visited networks. Notice that the true physical elements may contain one or several of the CSCF functions [6].

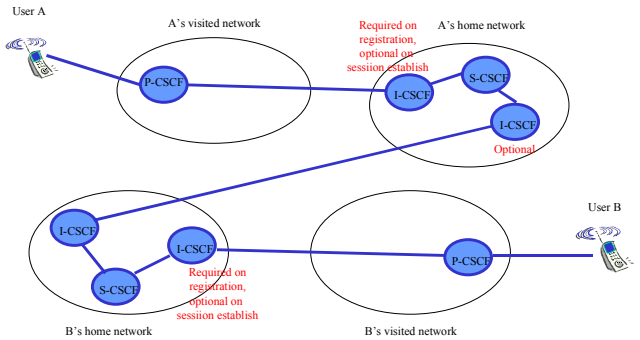


Figure 3: Call model in roaming case [5]

4 SIP protocol in 3GPP Rel5

4.1 SIP in IMSS

SIP and SDP as a protocol has been selected to some and IPv6 as the only solution to all of the IP Multimedia Subsystem interfaces.
As shown by the figure 4 the basic SIP (RFC2543[9]) has been selected as the main protocol on the following interfaces:

- Gm: P-CSCF - UE
- Mw: P-CSCF – S-CSCF and P-CSCF – I-CSCF
- Mm: S/I-CSCF - external IP networks & other IMS networks
- Mg: S-CSCF – BCGF

Mk: BCGF – external IP networks & other IMS networks

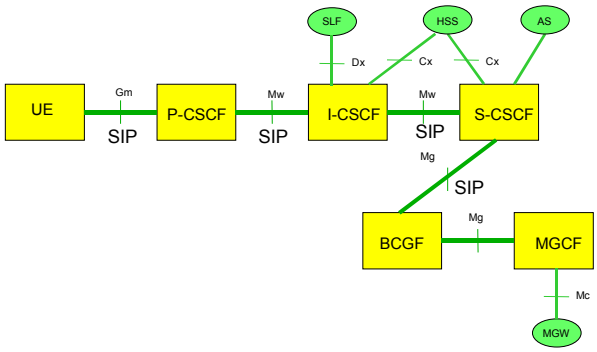


Figure 4: SIP protocol in IM SS [5]

Eventually there may be differences in the SIP procedures of Gm and Mw reference points. This implies that there is a difference in UNI and NNI interfaces [3].

The following procedures have been defined for the 3GPP IM subsystem in [3]:

- Local P-CSCF discovery: Either using DHCP or carrying address in the PDP context
- S-CSCF assignment and cancel
- S-CSCF registration
- S-CSCF re-registration
- S-CSCF de-registration (UE or network initiated)
- Call establishment procedures separated for
  - Mobile origination; roaming, home and PSTN
  - Mobile termination; roaming, home and PSTN
  - S-CSCF/MGCF – S-CSCF/MGCF; between and within operators, PSTN in the same and different network
- Routing information interrogation
- Session release
- Session hold and resume
- Anonymous session establishment
- Codec and media flow negotiation (Initial and changes)
- Called ID procedures
- Session redirect
- Session Transfer

4.2 SIP in Service SS

The service subsystem and its connections to IM subsystem is shown in the figure 5. The S-CSCF interfaces the application development servers with SIP+ protocols. The SIP application server can reside either outside or within operators network [3]. The OSA capability server and Camel refer to already standardised 3G and GSM based service generation elements.

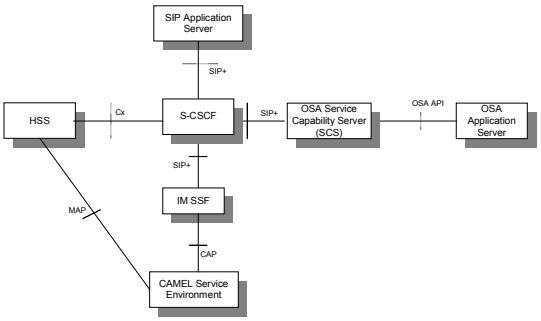


Figure 5: Service Subsystem connections with IMSS

SIP+ is used to interface the Application servers on the following interfaces:

- S-CSCF- SIP Application server
- S-CSCF- Camel Server
- S-CSCF-OSA Service Server

The plus sign implies here that the standard SIP may need to be modified. The modifications have not been identified yet [3].

4.3 The 3GPP Release 5 IMSS procedures

The PS CN Subsystem is strongly linked with the IP Multimedia since it shall provide the bearer through radio and the packet core network for the IP Multimedia Signalling (SIP) and also for the IP media streams. Thus co-operation is required on some level. Special key topics are:

- Handling of mobile terminated calls
- Bearer reservation before alerting

4.4 Mobile terminated calls

For mobile terminated calls the options are:

1) have network initiated PDP Context activation
2) provide an always on PDP context.

The network initiated PDP conext is currently discussed in 3GPP in the context of push services. The problem of the network initiated context activation is that the usage of dynamic IP addressing is not possible without enhacements to the network. The discussions are still open and the solution for the address allocation is sought [10].

For the second option i.e. using signalling PDP context there are two alternate methods how the P-CSCF address is provided to the terminal: either during the PDP context activation or after that with DHCP procedures. The latter case requires that the PDP Context is modified after the IP address of the P-CSCF has been found so that the GGSN can filter the SIP traffic to the correct PDP flow or a new PDP context is established for SIP with the correct filter information and the old is released. At the moment both options for the CSCF discovery are available in the specifications [1][3].

4.5 Bearer reservation before alerting

For the session flow (user plane traffic) a secondary PDP context with different QoS requirements is activated. A timing synchronism has to be sought between signalling PDP context establishment, secondary PDP context establishment, SIP connection negotiation and callee alerting. This is needed to avoid alerting before the resources are available and to find the fastest call establishment solution. This can be resolved with 2 phase call setup. The procedures are shown in the figure 6. For simplicity only the call originating part is shown. Also it should be noted that the PDP Context activation or the radio access bearer setups are not shown. In the receiving end the PDP context is established after the message 19: 200 OK [1][3][4].

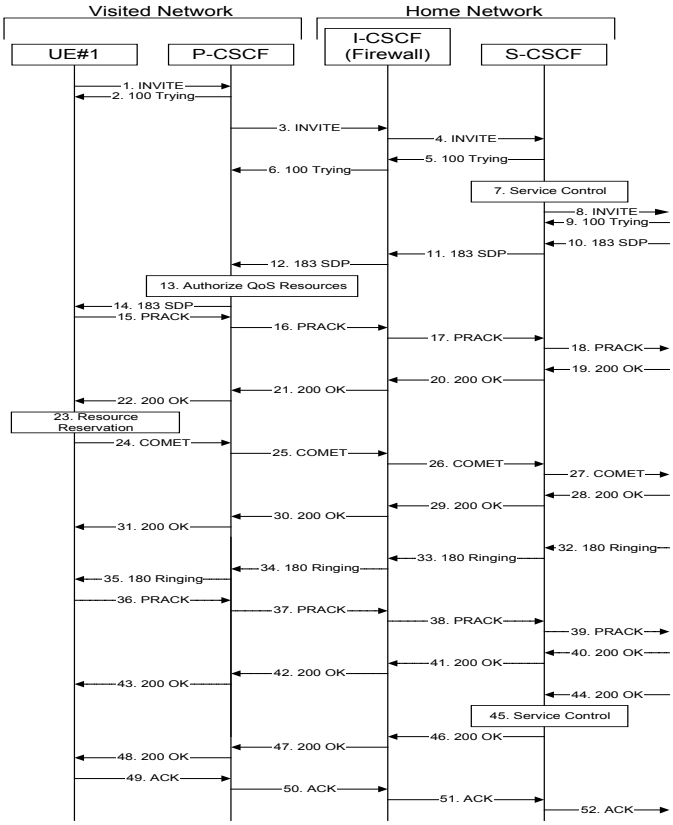


Figure 6: The 3GPP SIP 2-phase call setup [4]

5 3GPP SIP requirements

3GPP is in its specifications referring to IETF specifications and the target has been to minimise the changes. 3GPP is currently dependent on completion of the following SIP WG items [5] :

- draft-ietf-sip-rfc2543bis: SIP: Session Initiation Protocol
- draft-sip-manyfolks-resource: Integration of resource management and SIP

- draft-ietf-sip-100rel: Reliability of Provisional Responses in SIP
- draft-ietf-sip-privacy: SIP extensions for caller identity and privacy
- draft-ietf-sip-call-auth: SIP extensions for media authorization
- draft-roach-sip-subscribe-notify

3GPP has found out that a major part of the features are already provided by the SIP protocol and thus very few candidates for 3GPP originated enhancements have been identified [7]. The following SIP enhancements have been recognised so far [1][8]:

- **addition of routing PATH header** to the SIP messages to record the signalling path from P-CSCF to S-CSCF
- **location information in the INVITE message** to carry the location of the terminal (for instance Cell ID)
- **emergency call type** is needed to indicate the type of emergency call i.e. is it police, ambulance etc.
- **filtering of routing information** in the IM SS before the SIP message is sent to the terminal to hide the network topology from terminal
- **refresh mechanism inside IM SS**
- **Network-initiated de-registration**
- **183 Session Progress provisional response** for INVITE to ensure that the altering is not generated before PDP contexts for session are activated
- **Reliability of provisional responses** – PRACK method to acknowledge the 183 message
- **Usage of session timers** to keep the SIP session alive
- **Indication of resource reservation status** – COMET method
- **Security for privacy**
- **Extensions for caller preferences and callee capabilities**
- **Media authorisation token**

Discussions are currently ongoing on the changes between 3GPP and IETF.

6 Problems and open issues

The following problems can be identified in the 3GPP IP Multimedia Subsystem:

- **architecture complexity** i.e. with several functions there will be several interfaces, implementation may differ from vendor to vendor thus the multivendor cases may become challenging
- **call establishment delay problems** due to the signalling taking place on multiple levels (RAN, PS CN, IMSS). By making some calculations based on figure 6 for establishing a call there will be 6 round trip times (RTT) end to end on SIP level. In addition to that there are the PDP context reservations which take one round trip time between UE and GGSN.
- **guarantees of QoS**: Several elements and several IP based interfaces, in addition the packet radio included in the path while the requirements are at the same level as current GSM circuit voice calls
- **lengthy standardisation time:** more issues there are to standardise, more there are opinions and more time it will take
- **suitability of the SIP protocol for the radio interface** i.e. it is a character based protocol with long signalling messages and requires certain transport quality
- **IETF and 3GPP standardisation co-operation:** the operations and the behaviour are different in IETF and 3GPP
- **Terminal complexity:** the terminals become more and more complex with several protocol stacks, only to provide very similar services than today. SIP has to provide true revolution in applications and services

7 Conclusions

The major identified differences with the SIP IETF in and 3GPP are as follows:

1. the architecture of the IMSS is defined based on 3G model (home and visited), messages run always via S-CSCF
2. Registration is mandatory
3. The CSCFs interrogate the SIP and SDP flows either actively modifying the messages or reading the data, also the I-CSCF hides the names of CSCF behind it
4. Codec negotiations in 3GPP do not allow different codecs in different directions
5. in 3G networks there is a separation of UNI and NNI interface
6. due to radio and packet core functionality there are some change proposals to the SIP and SDP
7. due to the P-CSCF – S-CSCF interface and the 3G roaming mode there are some requirements to the SIP and SDP protocols
8. in 3G SIP is used also to interface the application development elements, they set requirements for SIP and SDP protocols

Despite of the above mentioned differences it seems that the SIP protocol is suitable to the needs of the UMTS network. The identified problems can be overcome and some of them have political or architectural nature thus they are more of choices than problems. The current work in 3GPP is still unfinished and the discussion with IETF has just been started. It is likely that the 3GPP Release 5 shall contain some specifications on SIP and IMSS architecture but their maturity is not probably too good by the end of 2001 to guarantee fully functioning network. One major advantage is that the SIP changes so far required by 3GPP are not extensive thus the SIP can probably be tailored for 3GPP. However, since the specification work for a new subsystem is a relatively large, it can

be expected that the specification work will continue also during 2002 and beyond. When the SIP and IMSS has been finalised for the UMTS network the real-time packet services can provide for the operators a true way to differentiate from each other and thus generate longed for revenues.

References

[1] Ahvonen, Kati: Master's Thesis: IP telephony signalling in a UMTS All IP network, Helsinki University of Technology, 24.11.2000

[2] 3G TS 23.002 version 5.1.0 Network Architecture (Release 5)

[3] 3G TS 23.228 version 2.0.0 IP Multimedia (IM) Subsystem - Stage 2

[4] 3G TS 24.228 version 2.0.0 IP Multimedia (IM) Subsystem - Stage 3

[5] Drage, Keith: 3GPP and SIP. Presentation in IETF #50 (March 18-23, 2001). http://www.softarmor.com/sipwg/meets/ietf50/slides/drage-3gpp-sip.ppt

[6] 3G TS 22.228 V5.0.0 (2001-01) Service requirements for the IP Multimedia Core Network Subsystem (Stage 1) (Release 5)

[7] Meeting minutes of 3GPP TSG-CN/SA SIP ad-hoc, February 12-14 http://www.3gpp.org/ftp/TSG_CN/WG1_mm-cc-sm/SIP_meetings/CN1_SA2_03_(New%20Jersey)/Report/NewJersey0102.zip

[8] Tdoc N1-010233; 3GPP TSG-SA WG2 / TSG-CN WG1 SIP ad-hoc meeting, 13-15 February, 2001, New Jersey, USA; Nokia: Feedback from IETF's interim SIP WG meeting held on week #6 http://www.3gpp.org/ftp/TSG_CN/WG1_mm-cc-sm/SIP_meetings/CN1_SA2_03_(New%20Jersey)/Tdocs/N1-010233%20.zip

[9] RFC 2543 SIP: Session Initiation Protocol . , March 1999. http://www.ietf.org/rfc/rfc2543.txt

[10] 3GPP TR 23.874 V1.3.0 (2000-11) Feasibility study of architecture for push service (Release 4) http://www.3gpp.org/ftp/Specs/Latest_drafts/23874-130.ZIP

SIP Service Architecture

Markus Isomäki
Senior Research Engineer
Nokia Research Center
Markus.isomaki@nokia.com

Abstract

Session Initiation Protocol (SIP) is an application layer signaling protocol for creating and modifying multimedia sessions. In this paper an overview of SIP based service architecture is presented. This includes introduction to SIP Application Servers, which implement the "service logic". Service programming methods such as Call Processing Language are briefly described, as well as service building blocks such as Third Party Call Control and call transfer. In order to draw everything together, an example of "autoconferecing" service is provided. Finally 3GPP Service Control architecture is described to point out how the principles could be applied in future mobile networks.

1 Introduction

Session Initiation Protocol (SIP) is an application layer signaling protocol for creating, modifying and terminating multimedia sessions with two or more participants [1]. These sessions include IP telephony and video calls, multimedia conferences and media distribution. SIP supports user mobility by proxying and redirecting requests to user's current registered location.

The protocol is still under heavy development and innovation in the Internet Engineering Task Force (IETF). The first version of the protocol was published as RFC 2543 already in March 1999, but since then the effort has only intensified. The current work includes refinements to the base protocol (known as rfc2543bis), as well as a large number of extensions for various purposes. Some of the extensions are merely ways of using existing SIP, such as third party call control. Others are true extensions offering new capabilities, such as session transfer method and generic event subscription, notification and messaging framework.

Besides IETF, several standardization organizations have selected SIP as a cornerstone in their architecture models. The most notable of these is perhaps the Third Generation Partnership Project (3GPP), who decided to use SIP as a session control protocol in the future releases of Third Generation cellular network specifications.

In addition to traditional call control and supplementary services, SIP can be used to build more advanced session related services by adding more intelligence to SIP servers and User Agents. The services cover those currently provided by Intelligent Networks (IN) in the PSTN, as well as completely new service types. By combining SIP with protocols such as HTTP, VoiceXML, RTSP and SMTP a full-blown multimedia infrastructure combining voice, video, web, e-mail and instant message communication can be created.

In this paper an overview of SIP based service architecture is presented. Chapter 2 presents a generic SIP service model that works as basis for further discussion. Chapter 3 introduces Application Servers and provides insight into their functionality. Chapter 4 briefly describes some useful service building blocks that have been developed for SIP. Chapter 5 lists different possible tools for programming Application Servers. Chapter 6 draws everything together by going through a complex service example. Finally, Chapter 7 intoduces 3GPP service control architecture and Chapter 8 points out the major conclusions of the paper.

2 Service Model

The service architecture depends in large degree on routing and security model of the servic providers. Figure 1 depicts a generic SIP service model with roaming support, with P denoting a Proxy server and AS denoting Application Server. Roaming and mobility are emphasized in the model, as most users and terminals are expected to be mobile in any future network, and a model excluding them would soon be obsolete.

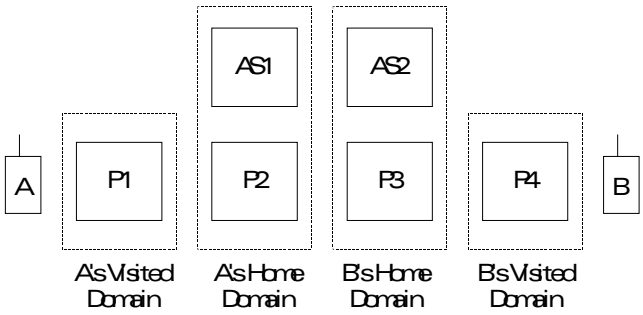


Figure 1. SIP Domains and Elements.

Each user has a home domain with whom he shares a security association, either based on shared secret or PKI. User registers to one of the registrars in his home domain, and home domain should also be aware of his services for both originating and terminating requests. Note that SIP home domain does not have to have anything to do with IP-layer home network, where e.g. the Mobile IP Home Agent resides.

All incoming requests to user are first routed to his home domain, where registration information is used to route the requests to the user himself. Outgoing requests can bypass home domain servers what comes to basic routing, but if some "originating" session services are required, they should also traverse home domain. It is the responsibility of the home domain to execute both "terminating" and "originating" request services for the user. Usually the overall control is left to a specific proxy, which is always on the signaling path of both incoming and outgoing requests. Sometimes this proxy is called a "feature proxy", in 3GPP terminology it is called Serving Call State Control Function (S-CSCF). In the Figure P2 and P3 denote user A's and user B's feature proxies, respectively.

In order to implement more complex services, the feature proxy can use specific Application Servers (AS1 and AS2 in the Figure) by routing certain requests to them. Application Servers can be either specialized for one task (number translation, conferencing) or they can do several tasks. In SIP nomenclature Application Servers can be either proxies, redirect servers or back-to-back User Agents. More details of Application Servers are provided in Chapter 3. In order to implement services Application Servers can interact with e.g. Intelligent Network, or use other means such as those introduced in Chapter 5. They can also utilize mechanisms described in Chapter 4 as building blocks for services. It is usually Application Servers who interact with other protocols such as HTTP or RTSP.

If user is roaming outside his home domain, he may need to use a local outbound proxy in the visited domain. The outbound proxy may authenticate the user using some AAA protocol such as Diameter to make a query to home domain's AAA server. Outbound proxy may also help the user with NAT or firewall traversal and authorize network layer QoS resources for user's media flows. In the Figure P1 and P4 denote local outbound proxies for users A and B, respectively. If the user is able to bypass the local outbound proxy, he is effectively in his home domain. Outbound proxies are usually not intended for service control, but they might be useful in offering local or location based services for the visiting user. These include emergency call services.

So, when user A issues a request to user B, his terminal first sends it to P1. P1 routes the request to A's home network, eventually to P2. P2 controls user A's originating services and can use various Application Servers to execute them. After that, if no specific action is taken by the service logic, the request is routed to recipient's, that is B's, home domain to P3. P3 controls B's terminating services, and utilizes Application Servers to achieve them. After that, if that is what the service logic tells, request is routed to P4 and finally to B. B's responses and the further negotiation drives the service machinery in both A's and B's domains in those proxies that decided to stay on the signaling path during the initial request.

It should be noted that User Agents or terminals themselves can do services, such as forwarding or screening or special ringing tones. However, terminals may be out of network coverage or otherwise disconnected, so everything can not be left to them. Otherwise basically every element on the signaling path can do a "service".

3 Application Server Mechanisms and Components

In SIP there are basically four different types of servers in addition to User Agent software running in terminals:

- **Registrar** is the server that handles registrations and maintains state from which user's current contact point and preferences can be determined (actually, part of this state can be stored to external elements, location server and presence server). Registrar is always in the home domain.
- **Proxy** forwards the incoming request further based on some internal logic. Proxies can be transaction stateless or stateful. In some cases they can be

even session stateful, which of course reduces their scalability. There can be proxies for different purposes:

- Core Routing proxy
- Gateway controlling proxy
- Firewall controlling proxy
- QoS controlling proxy
- "Feature proxy" or "Regional Routing proxy", whose duty is to orchestrate the service routing and execution as explained in Chapter 2.

- **Redirect Server** does not forward the request further, but returns it towards its originator with redirection information. Redirect servers are very easy to implement and scalable, and thus they are powerful tools for simple services.
- **Back-to-Back User Agent** is the term used for any "proxy-like" element in the network which does something more than a proxy is supposed to do. This includes e.g. issuing requests to ongoing sessions "in the middle" or modifying SDP parameters.

SIP Application Server (APSE) is a vague term which is not defined anywhere in the official specifications. Basically Application Server can be a Proxy, Redirect Server or Back-to-Back User Agent. If simple change in the Request-URI is all that a service requires, redirection is the way to go. However, if changes in other headers, call state monitoring or acting upon responses is desired it takes at least a proxy. APSE acting as a proxy differs from common proxy only in how much intelligence or programmability it has. Obviously there is no official distinction between the two. If proxy functionality is not enough, a Back-to-Back UA is needed.

It may be useful to let one APSE to handle only one specific task and treat them as components from which complete services can be build. In addition to routing and control APSEs also other special media handling components are needed in the network:

- Media Server **can play announcements or stream other audio or video content to the user. It can also be used to collect DTMF "digits". Media Server is needed for e.g. announcements, voicemail and interactive voice response. Such a server can be controlled by SIP (session creation and termination), RTSP (media control) and HTTP/VoiceXML (interactive voice response dialogues).**
- Conferencing Server **or bridge is able to mix medias coming from different parties together to implement a multi-party conference. SIP is used as a control protocol, also more tight control can be obtained using a special conference control protocol.**
- Presence Server **obtains information on users communication state and preferences or announcements ("I'm eating") and can convey this information to interested parties. More details in the next Chapter.**
- Text-to-Speech Server **is able to translate a text stream to speech and possibly even vice versa.**
- Messaging Server **is able to issue instant messages to users.**
- Web Server **and** E-mail Server **can be used to enhance services by alternative communication methods. For example interactive voice response can be replaced by web-based dialogues. Web-pages can contain embedded SIP or mailto URLs and SIP messages can contain HTTP URLs, so for example an initial voice session initiation can be redirected to fetch a HTML/XML document.**

Complex services can be built by combining the capabilities of different servers. It is feasible to treat the servers as resources or service components addressible by either SIP, RTSP or HTTP URL, as proposed in [2]. In that way the servers who use other servers do not need to know the internal details of what they are using. This is an opposite approach compared to device control protocols such as MGCP or Megaco, where the controller has tight control over the slave and needs to understand slave's internal structure to some extent. MGCP and Megaco can still be used to separate media part from the control part.

4 Service Building Blocks

This Chapter introduces some of the building blocks that APSEs and User Agents can use to construct services.

Record-Routing **is one of the key features in SIP protocol design and it is very useful to APSEs, as it allows them to choose whether to remain on the signaling path of certain session after the first transaction has been completed.**

Forking is another nice feature of SIP, which allows for "parallel searches" in order to save time. Unfortunately complete forking solution is only possible for INVITE requests.

Event Notification is a SIP extension to provide a generic and extensible framework by which SIP nodes can request notification from remote nodes indicating that certain events have occurred [3]. The framework is based on two new methods – SUBSCRIBE and NOTIFY.

Using SUBSCRIBE SIP User Agents can subscribe to state of any resource in the network that is addressible with a SIP URL. When the state changes, a NOTIFY with relevant information is sent to each of the subscribers.

It is not intended that SIP SUBSCRIBE and NOTIFY would be used for all types and classes of events. However, events related to session or user registration state fall well within the intended scope. In fact, by extending user's registration state to include so called Presence information to which other users can subscribe, innovative new service types are made possible.

A typical SUBSCRIBE-NOTIFY message flow is depicted in Figure 2 [3]. If SUBSCRIBEs are Record-Routed, NOTIFYs follow the same (reverse) path.

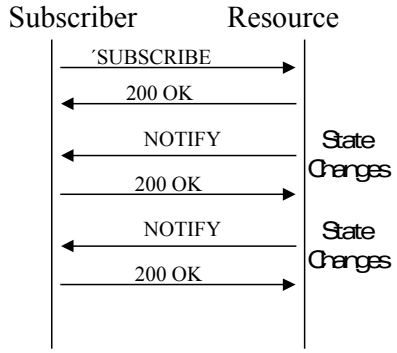


Figure 2. SUBSCRIBE and NOTIFY.

Third Party Call Control allows a third party to setup a session between two User Agents by issuing both of them a separate INVITE request and relaying the session descriptions (SDPs) from one to another. Thus, media will flow directly between the two UAs. The scenario is depicted in Figure 3.

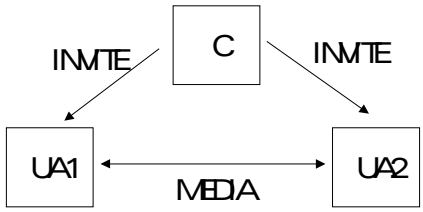


Figure 3. Third Party Call Control.

Third Party Call Control is suitable e.g. for situations where an Application Server wishes to create a session between a user and a Media Server or it wants to invite a user to a centralized conference. APSE can terminate the session when it so wishes, but it does not have to cope with the media. Third Party Call Control does not require any SIP extensions, as it should not make any difference to UA in terms of processing incoming sessions. Third Party Call Control is explained in detail in [4].

REFER **is a new SIP method for implementing various types of call or session transfer. Figure depicts a common use of REFER, namely unattended transfer. Three parties are involved: Transferor (party initiating transfer), Transferee (party being transferred) and Transfer Target (party to whom the transfer is intended). First Transferee calls Transferor, and a session between them is established. After they have e.g. talked for a while Trasferor decides that it is time to initiate the transfer. It first puts Transferee on hold by issuing a re-INVITE with special SDP parameters. After that it issues a REFER request to Transferee indicating in Refer-To header the address of the Transfer Target. Transferee accepts the request (202), and using the information in REFER issues an INVITE to Transfer Target.**

If INVITE is successful, Transferee and Transfer target now have an ongoing session. After that Transferee sends Transferor NOTIFY to indicate the success of transfer. Transferor is then able to terminate the original session. Thus, session has been transferred. If the transfer were not successful, Transferor could resume the original session by issuing re-INVITE instead of BYE.

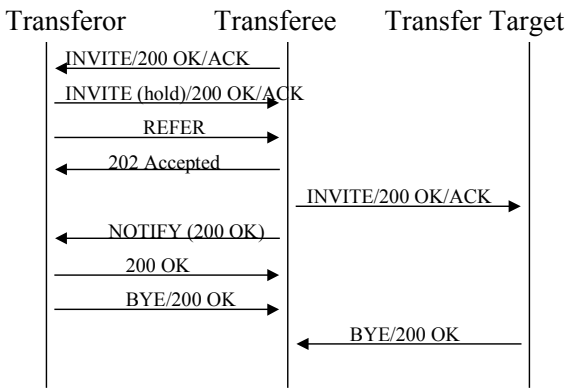


Figure 4. Unattended transfer with REFER method.

According to current specifications, REFER can occur outside of call-leg, thus it can basically initiate sessions from scratch. Refer-To header can contain other URLs than SIP URLs, thus it would be possible to initiate also other than SIP-based forms of communication. REFER is suitable to similar types of scenarios as Third Party Call Control, although both have their advantages. REFER details can be found in [5].

Messaging is a new form of communication supported by SIP and is currently under development. Messaging can be done either with MESSAGE method or by opening a messaging session with INVITE. Messaging is a useful tool between two users but as well between an APSE and a user. MESSAGE is specified in [6].

Caller Preferences and **Callee Capabilities** can be expressed easily in SIP by header parameters. Examples include indication of supported media types or spoken languages or type of the device the user currently has.

5 Service Creation Tools

After talking about Application Servers and service building blocks it is useful to look briefly how the services can actually be programmed. As an APSE is usually a proxy or redirect server, the task is to find suitable tools to program them to handle incoming SIP messages in a desired way.

There are currently a lot of tools available for the task:

- **Call Processing Language (CPL)** is an XML-based language that can be used to describe and control Internet telephony services. CPL is not tied to any signaling protocol, but the expectation is that either SIP or H.323 is used. CPL is designed so that it is powerful enough to describe a large number of services and features, but it is limited in power so that it can run safely in Internet telephony servers [7]. Thus, even end-users themselves could program CPL-scripts and download them to servers without being able to do any severe harm. The goal of the language definition has been that it would be possible to generate it with graphical tools. Nothing prevents running CPL directly in the User Agent. CPL should become IETF standard-track RFC during spring 2001.
- **SIP Common Gateway Interface (CGI)** is equivalent to popular HTTP CGI, the difference being the protocol under control. SIP CGI opens the contents of SIP message headers directly accessible by external programs, in a programming language independent way.
- **SIP Servlets** or SIPlets are equivalent to Java Servlets used in Web programming. They provide a certain class library for developers to access and control the SIP stack. Servlets usually offer better performance than CGI scripts due to their more advanced handling of processes. Being tied to Java, Servlets share the advantages and drawbacks of the language.
- **SIP JAIN** is another Java-based programming interface to control the SIP stack.

Of the four presented mechanisms, the three latter ones seem to be suitable for same type of tasks and are thus competing with each other. CPL has a bit different scope, as it has certain limitations. One of the trade-offs in the control interface is how much the developer has to understand SIP. Low-level interfaces such as CGI require certain knowledge of SIP messages and allow efficient and pinpointed control. On the other hand high-level interfaces such as CPL do not require SIP expertice, but lack efficiency and advanced features. There are already Application Server products (or at least prototypes) available supporting at least CPL, SIP CGI and Servlets.

Also Intelligent Networks can be used to provide service control for SIP proxies. This can be achieved by mapping SIP state machine to Basic Call State Machine (BCSM) and controlling it by INAP or CAP protocols. In this architecture SIP proxy plays the role of "Soft" Service Switching Function (SSF) which interacts with external Service Control Function (SCF). While this model is suitable for bringing "legacy" telephony services to SIP network, it lacks the capabilities and flexibility required for integration with other protocols and services, and the most advanced features of SIP are not utilized. Open Services Architecture (OSA) is another telephone network oriented service control platform, which can be used to control SIP services.

6 Example Service

At this point it is useful to go through an example service to illustrate how different APSEs could interoperate using the building blocks defined in Chapter 4.

"Autoconferencing" service is used as an example. The service works so that a user is able to schedule an audio or video conference to start when all required participants indicate that they are available for conferencing. Figure 5 depicts the needed components: Controller for orchestrating the service, Presence Server for obtaining users' availability, Messaging and Media Servers to send announcements and finally a Conferencing Server to bridge the conference participants together.

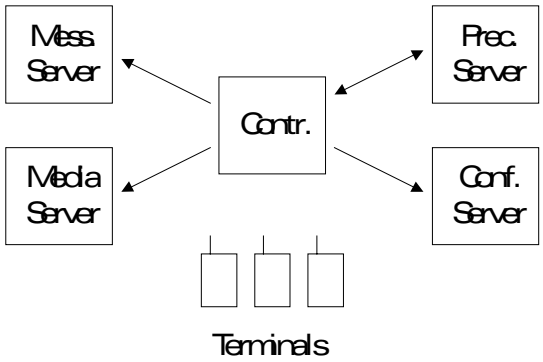


Figure 5. Autoconferencing Service.

First, a user fills in a web page order for the conference listing all the required participants and sends it to the Controller Server using standard HTTP/HTML. Controller then subscribes to presence information of all participants with SIP SUBSCRIBE and starts to receive NOTIFYs when their presence state changes. When all participansts seem to be available for conferencing, Controller Server either sends them an instant message using Messaging Server or invites them to a session with Media Server using Third Party Call Control. The purpose of these actions is to get an acknowledgement from the participants on their willingness to join the conference. Message can contain push buttons or links to achieve this, while Media Server can use IVR dialogues to get users opinion. IVR dialogue can be controlled by the Controller Server.

In the end the Controller receives a "yes" or "no" answer from each participant. If everything is proceeding well, Controller now invites each participant to a session with Conference Server using Third Party Call Control. This can be preceded with some kind of authentication scheme using a web page or Media Server IVR. Conference Server ties all participants together based on the Request-URI in INVITE it receives. Thus, all participants are in the same conference. When new participants join, Controller can invite Media Server to the conference to play short announcements like: "Bob just joined the conference".

All interaction between different servers happens by exchanging standard SIP and HTTP messages. Inside Media and Conferencing servers, MGCP or Megaco could be used to separate media and control processing from each other.

7 3GPP Service Architecture

3GPP is currently defining its service control architecture for Release 5 IP Multimedia Subsystem (IMS). The specifications are to be completed by the end of year 2001. IMS is based on SIP protocol with minor 3GPP specific modifications. In IMS each incoming and outgoing request is routed via subscribers home network through an element called Serving Call State Control Function (S-CSCF).

S-CSCF plays the roles of registrar, feature proxy and in some cases also back-to-back user agent. It has to route the incoming and outgoing requests to correct SIP Application Servers and other external service platforms according to subscribers' service profile. Besides SIP APSEs, also CAMEL and OSA service control are supported in Release 5.

3GPP Release 5 service architecture is depicted in Figure 6 [8]. S-CSCF routes incoming and outgoing requests to APSEs and other service platforms using a protocol called "SIP+". SIP+ requirements and definition work has just started in 3GPP, so the only known fact about it is that it should resemble SIP as much as possible. Mapping to CAMEL (CAP) and OSA does not happen in S-CSCF but rather in the specialized gateway elements shown in the Figure.

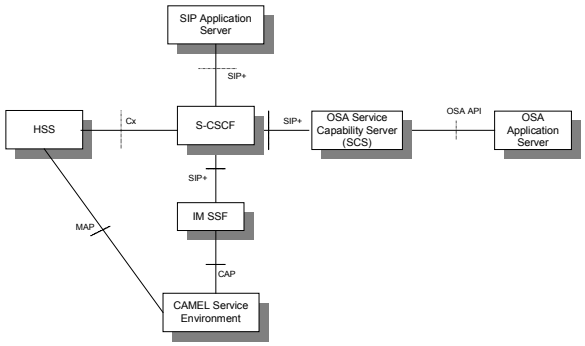


Figure 6. S-CSCF and Service Control interfaces.

3GPP IMS Application Servers should be quite similar to standard SIP APSEs described in this paper. Some differences may arise, if SIP+ turns out to be much different from SIP. The most challenging problem is to define how different APSEs and service platforms interoperate and how S-CSCF makes service routing decisions.

8 Conclusions

SIP and related protocols such as HTTP, RTSP and SMTP provide a powerful machinery to implement services that integrate different forms of communication. In SIP services are provided by specialized Application Servers, which are actually proxy or registrar servers with extended intelligence.

The intelligence can be brought to the servers with various methods including Call Processing Language (CPL), SIP CGI and SIP Servlets. Even traditional Intelligent Network can be used for some purposes.

Application Servers utilize standard SIP features such as Record-Routing and make use of extensions such as Third Party Call Control, SUBSCRIBE and NOTIFY, REFER and SIP messaging. Also HTTP is useful in carrying out simple transactions. All resources can be addressed by SIP or RTSP or HTTP or mailto: URLs, which can be carried in protocol headers or be embedded e.g. in HTML/XML documents. By making different APSE components play together, complex services such as "autoconferencing" presented in Chapter 6 can be accomplished.

3GPP is currently in the process of defining a SIP-based Service Control architecture for UMTS. The main challenges in the process are how services can be made work together and how service specific routing of requests should be orchestrated.

References

[1] M. Handley, H. Schulzrinne, E. Schooler, J. Rosenberg, "SIP: Session Initiation Protocol", Internet-Draft (work in progress), November 2000, http://search.ietf.org/internet-drafts/draft-ietf-sip-rfc2543bis-02.txt

[2] Jonathan Rosenberg, "An Application Server Component Architecture for SIP", Internet-Draft (work in progress), March 2001, http://search.ietf.org/internet-drafts/draft-rosenberg-sip-app-components-01.txt

[3] Adam Roach, "Event Notification in SIP", Internet-Draft (work in progress), February 2001, http://search.ietf.org/internet-drafts/draft-roach-sip-subscribe-notify-03.txt

[4] Jonathan Rosenberg, Jon Peterson, Henning Schulzrinne, Gonzalo Camarillo, "Third Party Call Control in SIP", Internet-Draft (work in progress), March 2001, http://search.ietf.org/internet-drafts/draft-rosenberg-sip-3pcc-02.txt

[5] Robert Sparks, " SIP Call Control – Transfer", Internet-Draft (work in progress), February 2001, http://search.ietf.org/internet-drafts/draft-ietf-sip-cc-transfer-04.txt

[6] Jonathan Rosenberg et al., " SIP Extensions for Instant Messaging", Internet-Draft (work in progress), February 2001, http://search.ietf.org/internet-drafts/draft-rosenberg-impp-im-01.txt

[7] Jonathan Lennox, Henning Schulzrinne, " CPL: A Language for User Control of Internet Telephony Services", Internet-Draft (work in progress), November 2000, http://search.ietf.org/internet-drafts/draft-ietf-iptel-cpl-04.txt

[8] 3G TS 23.228 version 2.0.0 IP Multimedia (IM) Subsystem - Stage 2

IP TELEPHONY SERVICES IMPLEMENTATION

Eero Vaarnas
eero.vaarnas@iki.fi

Abstract

There is a wide variety of tools – both traditional, PSTN-like (Public Switched Telephone Network) and web-oriented – for implementing services in IP telephony. There are so many alternatives for service creation that only some of them are described here. The scope of this document is in all-IP environment, where many of the paradigms come from the World Wide Web (WWW). Some of the techniques are more or less standardized, like Call Processing Language (CPL), SIP-CGI (SIP Common Gateway Interface) and SIP Servlet API (SIP Servlet Application Program Interface).

CPL is a simple scripting language with rapid implementation cycle but limited capabilities. It is independent of the signalling protocol. SIP-CGI is a more powerful interface for executing arbitrary programs in a SIP proxy server. The interface is language independent, but the process handling causes some overhead. SIP Servlet API is a similar technique to SIP-CGI. It is designed using Java, so it's platform independent. All services run on the same Java Virtual Machine (JVM), so the overhead of process generation is eliminated. There are also H.323-based services, but their major disadvantage is in interoperability problems.

1 Introduction

IP Telephony protocols are in a quite mature state. There are some competing and/or overlapping standards, but the overall picture is pretty clear. It seems more and more likely that SIP (Session Initiation Protocol [1]) is going to be the signalling protocol of All-IP multimedia sessions, including voice. SIP is text-based, HTTP-like (HyperText Transfer Protocol) protocol standardized by IETF. It is simple but easy to be extended. Of course also H.323 from ITU-T [2] will have its own role because its current installed base, mainly in corporate use. However, H.323 has its difficulties, such as scalability and interoperability.

Also PSTN-interoperability can be handled with a limited number of protocols. In media gateway control, there are practically two protocols, MGCP (Media Gateway Control Protocol) and Megaco/H.248. Megaco/H.248 can be seen – if not directly as an extension – as the successor of MGCP. Both can possibly be used also in dumb IP terminals directly. ISUP (Integrated Services User Part) and similar signalling over IP networks can be done quite straightforwardly, either by mapping ISUP messages to SIP, H.225/H.245 or similar, or tunneling them transparently using e.g. BICC (Bearer Independent Call Control) or SIP-T (SIP for Telephones). Media transmission is merely a matter of standard codecs and packetization.

In service creation there are more decisions to be made. In the PSTN many services have been implemented using Intelligent Networks (IN). IN is controlled by the operator and typically users activate services using DTMF (Dual Tone MultiFrequency) tones. New kind of service creation paradigms come from the World Wide Web, where users can more freely control the services and user interfaces are more intuitive.

There are some interfaces that can be used to integrate IN services to IP telephony environment. With for example JAIN (Java Advanced Intelligent Networks, Java APIs for Integrated Networks) and/or Parlay, Intelligent Networks could be utilized from the IP environment. IN connectivity is an important issue, but it isn't considered here.

The emphasis of this document is in services implemented totally in the IP environment. Most of the new techniques – especially SIP based – borrow slightly from techniques already used in WWW. The idea is that the more open the architecture is, the easier it is for the third parties and even users themselves to create new services.

Four service implementation techniques are presented here: CPL, SIP-CGI, SIP Servlet API and H.323 services. First three of them work conceptually quite similarly. The server has some default mechanism for handling requests, which is used for normal signalling operation. By some means the server decides, which messages are handled by the default processing and which are sent to the service interface. Then the service interface can perform signalling or other operations and/or pass the message back to the default processing. H.323 services introduced later on form an exception. They are more similar to traditional PSTN services.

2 Call Processing Language

CPL (Call Processing Language) [3] is an XML-based (eXtensible Markup Language) markup language that can be used to describe telephony services. It describes

the logical behavior of the signalling server, in principle it isn't tied to any specific protocol.

Like XML, CPL is based on tags that are hierarchically arranged according to the information that they contain. The tags are traversed according to the hierarchy and the rules they contain. Eventually the traversal ends and the action specified by the script is executed. In some cases the action remains unspecified, so some default policy is resumed.

2.1 Structure of CPL

CPL is specified as an XML DTD (Document Type Definition). It is going to have a public identifier in XML (`-//IETF//DTD RFCxxxx CPL 1.0//EN`) and corresponding MIME (Multipurpose Internet Mail Extensions) type. Here is only an overview of the structure, the complete DTD can be seen in [3] and XML specification in [4].

After the standard XML headers, CPL script is enclosed between tags `<cpl>` and `</cpl>`. The script itself consists of nodes and outputs, arranged hierarchically in a nested structure. Nodes and outputs can be thought of as states and transitions, respectively (for a tree representation, cf. 2.2). The structure is represented by nested start and end tag pairs, so both nodes and outputs can be simply referred as tags. Tags can have parameters that describe their exact behavior .

At the top level, there can be four kinds of tags: `ancillary`, `subaction`, `outgoing` and `incoming`. The `subaction` tag is used to describe repeated structures to achieve modularity and to avoid redundancy. The implementation is under the `subaction` tag with the `id` parameter as an identifier. One or more references to the implementation can be made using the `sub` tag with the desired subaction identifier as the `ref` parameter. The `outgoing` and `incoming` tags are top level actions, similar to sub-actions in their implementation structure. The `ancillary` tag contains information that is not part of any operation, but possibly necessary for some CPL extension.

The actual node-output structure of the script is inside the action tags, i.e. `subaction`, `outgoing` and `incoming`. There are four categories of CPL nodes: switches, which represent choices a CPL script can make; location modifiers, which add or remove locations from the set of destinations; signalling operations, which cause signalling events in the underlying protocol; and non-signalling operations, which trigger behavior which does not effect the underlying protocol.

2.1.1 Switches

Switches represent choices a CPL script can make, based on either attributes of the original call request or items independent of the call. The attributes are represented by variables, depending on the switch type. Switch has a list of output tags, that are traversed and the first matching output is selected. If the variable doesn't exist, the optional `not-present` tag can be chosen instead. If none of the outputs match (including `not-present`), the optional output `otherwise` is chosen. There are four types of switches: `address-switch`, `string-switch`, `time-switch` and `priority-switch`.

The `address-switch` makes decisions according to addresses. With the `field` parameter either `origin`, `destination`, or `original-destination` of the request can be chosen. Moreover, the optional `subfield` parameter can be used to access the `address-type`, `user`, `host`, `port`, `tel`, or `display` (display name) of the selected address. In the `address` output it can be compared if the address `is` an exact match, `contains` substring of the argument (for `display` only) or is in the `subdomain-of` the argument (for `host`, `tel` only). The `address-switch` is essentially independent of the signalling protocol. The specific meaning of the entire address depends on the protocol and additional subfield values may be defined for protocol-specific values.

The `string-switch` allows a CPL script to make decisions based on free-form strings present in a request. The `field` parameter selects either `subject`, `organization`, `user-agent` (program or device name that made the request), `language` or `display`. The `string` output checks if the selected string `is` an exact match or `contains` a substring of the argument. String switches are dependent on the signalling protocol being used.

The `time-switch` handles requests according to the time and/or date the script is being executed. It uses a subset of iCalendar standard [5], which allows CPL scripts to be generated automatically from calendar books. It also allows us to re-use the extensive existing work specifying calendar entries such as time intervals and repeated events. Parameters `tzid` (time zone identifier) or `tzurl` (time zone url) select the current time zone and the output `time` match calendar entries such as starting or ending times (`dtstart`, `dtend`), days of the week (`byday`) and frequencies (`freq`). Time switches are independent of the underlying signalling protocol.

With the `priority-switch` it is possible to consider priorities specified for the requests. Priority switches take no parameters. The `priority` output can be used to match against `less` than, `greater` than or `equal` to the argument. The priorities are `emergency`, `urgent`, `normal`, and `non-urgent`. The priority switches are dependent on the underlying signalling protocol.

2.1.2 Location modifiers

The set of locations to which a call is to be directed is not given as node parameters. Instead, it is stored as an implicit global variable throughout the execution of a processing action (and its subactions). Location modifiers add, retrieve or filter the set of locations.

There are three types of location nodes defined. Explicit locations add literally-specified locations to the current location set; location lookups obtain locations from some outside source; and location filters remove locations from the set, based on some specified criteria.

The explicit `location` node has three node parameters. The mandatory `url` parameter's value is the URL of the address to add to the location set. The optional `clear` parameter specifies whether the location set should be cleared before adding the new location to it. The optional `priority` parameter specifies a priority for the location. There are no outputs, next node follows directly. Explicit location nodes are dependent on the underlying signalling protocol.

Locations can also be specified up through external means, through the use of location lookups. The `lookup` node initiates lookups according to the `source` parameter. With the optional parameters, one can `use` or `ignore` caller preferences fields or `clear` the location set before adding. The outputs are `success`, `notfound`, and `failure`, one of them is selected depending on the result of the lookup.

The `remove-location` is used to filter the location set. Filtering is done based on the `location` parameter and caller preferences `param` - `value` pairs. There are no outputs, next node follows directly. The meaning of the parameters is signalling-protocol dependent.

2.1.3 Signalling operations

Signalling operation nodes cause signalling events in the underlying signalling protocol. Three signalling operations are defined: `proxy`, `redirect`, and `reject`.

The `proxy` node causes the request to be forwarded on to the currently specified set of locations. With the corresponding parameters, a `timeout` can be set, the server can be forced to `recurse` to subsequent redirection responses, and the `ordering` of the location set traversal can be set to `parallel`, `sequential`, or `first-only`.

The `redirect` node causes the server to direct the calling party to attempt to place its call to the currently specified set of locations. The redirection can be set `permanent`, otherwise considered temporary. Redirect immediately terminates execution of the CPL script, so this node has no outputs and no next node. The specific behavior the redirect node invokes is dependent on the underlying signalling protocol involved, though its semantics are generally applicable.

The `reject` nodes cause the server to reject the request, with a `status` code and possibly a `reason`. Similarly to `redirect`, rejection terminates the execution, and specific behavior depends on the signalling protocol.

2.1.4 Non-signalling operations

With non-signalling operations, it is possible to invoke operations independently of the telephony signalling. If supported, `mail` can be sent, `log` files can be generated, and also other operations can be added as so called extensions.

2.2 Tree representation of CPL

For illustrative purposes, CPL scripts can be represented as trees. Also graphical editors might utilize the tree representation. Node tags represent nodes of the tree, output tags are edges between them. In Figure 1 is an example CPL script from [3]. It is converted into a tree in Figure 2.

```
 1: <?xml version="1.0" ?>
 2: <!DOCTYPE cpl
 3:  PUBLIC "-//IETF//DTD RFCxxxx CPL 1.0//EN"
 4:  "cpl.dtd">

 5: <cpl>
 6:  <subaction id="voicemail">
 7:   <location
 8:    url="sip:jones@voicemail.example.com">
 9:    <redirect />
10:   </location>
11:  </subaction>

12:  <incoming>
13:   <address-switch field="origin"
14:    subfield="host">
15:    <address subdomain-of="example.com">
16:     <location url="sip:jones@example.com">
17:      <proxy timeout="10">
18:       <busy> <sub ref="voicemail" />
19:       </busy>
20:       <noanswer> <sub ref="voicemail" />
21:       </noanswer>
22:       <failure> <sub ref="voicemail" />
23:       </failure>
24:      </proxy>
25:     </location>
26:    </address>
27:    <otherwise>
28:     <sub ref="voicemail" />
29:    </otherwise>
30:   </address-switch>
31:  </incoming>
32: </cpl>
```

Figure 1 Example CPL script

Let us have a brief look at the example script (also the graphical representation can be followed and compared to the script structure). In lines 6-11 there is an example of a subaction. It defines a redirection to the user's voicemail. This is accomplished by adding the address of the voicemail to the location set (lines 7-8) and then activating the redirection (line 9). Lines 12-31 describe how incoming calls are handled. The address switch in lines 13-30 selects the host part of the callers address. If the caller is from the same domain as the owner of the script (line 15), the call is considered urgent and it is let through. Again, this is done in two stages: first the address is added to the location set (line 16), then the actual proxy behavior is activated (line 17). All the unsuccessful cases are directed to the

voicemail (lines 18-23). The voicemail is implemented as a reference to the previously defined subaction. Also unimportant calls go to the voicemail (lines 27-29).

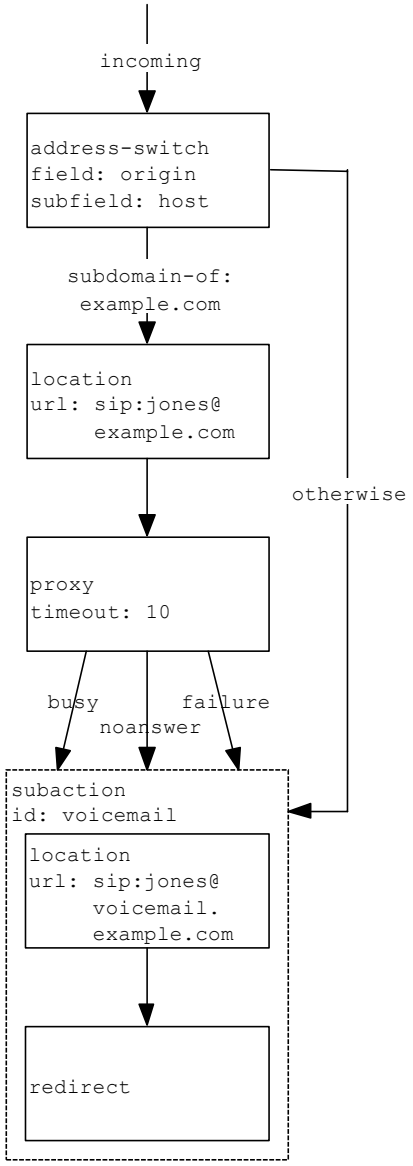


Figure 2 Tree representation of the example script

2.3 General feasibility of CPL

CPL is a simple but powerful tool for IP telephony service implementation. It is concentrated in basic call control functions, but it is possible to create extensions – some of them already available – for different kinds of advanced services. Of course CPL isn't a programming language, so constructions like loops aren't possible and all the features must be actually implemented outside the scripts.

CPL is based on XML, which is a widely accepted industry standard. This, along with its general simplicity, provides a good starting point for its utilization. First of all, people already familiar with XML can easily adopt CPL. Even with minimal knowledge of XML it is possible to start writing CPL scripts. It is also possible to generate scripts automatically. Generation could be based on simple, standard text-processing languages. From other types of XML documents, XSLT (eXensible Style Language Translation) transformations could apparently be used. Because of its tree representation CPL (and XML) can be expressed and edited also graphically. With GUI (Graphical User Interface) based editors also people not so familiar with the syntax can create and edit services. Users could upload their own CPL scripts using SIP registration messages, HTML forms, FTP, or whatever method seems proper.

Things like scalability, stability and security depend much on the implementation of the CPL server. However, because of the limited expression power of the language, these problems are more easily treated. Scripts can be exhaustively validated upon their uploading, so in principle malicious or erroneous code can be eliminated. Also the lack of loops and other more complex programming structures makes CPL scripts potentially more compact.

CPL execution is already implemented at least in a few SIP proxy servers [6]. There are also plenty of XML editors available and recently even some specialized CPL editors. Some service creation environments are based on automatic CPL generation.

3 Common Gateway Interface for SIP

SIP-CGI (Common Gateway Interface for SIP) [7] is an interface for running arbitrary programs from a SIP proxy server or similar software. Since SIP borrows a lot from HTTP, also the CGI interface is adopted. Of course, the technical specification is different, but the basic idea is similar to HTTP-CGI.

When the server decides to invoke a SIP-CGI script, it executes it as a normal process in the underlying operating system. It then uses standard input and output (stdin, stdout) and environment variables to exchange information with the process. Script status throughout invocations is maintained with special tokens.

3.1 Input and metadata

The header fields (with some exceptions, such as potentially sensitive authorization information) of the received SIP message are passed to the script as metavariables. In practice, metavariables are represented by the operating system environment variables. Each SIP header field name is converted to upper case, has all occurrences of "–" replaced by "_", and has `SIP_` prepended to form the metavariable name. For example `Contact` header would be represented by `SIP_CONTACT` metavariable. The values of the header fields are converted to fit the requirements of the environment variables. Similar transformations are applied for other protocols.

There are some additional metavariables that are passed to the script. Some of them are derived from the

header fields or even match the values of the fields. This redundancy is for the script to distinguish between information from the original header fields and information synthesized by the server.

The type of the message is seen from metavariables REQUEST_METHOD and RESPONSE_STATUS. If REQUEST_METHOD is defined, the message was a request and the method (INVITE, BYE, OPTIONS, CANCEL, REGISTER or ACK) is stored in the metavariable. REQUEST_URI is the intended recipient of the request. REGISTRATIONS contains a list of the current locations the server has registered for the recipient (REQUEST_URI).

For responses, RESPONSE_STATUS is the numeric code of the response and RESPONSE_REASON is the string describing the status. For example SIP/2.0 404 Not Found response contains the protocol version, status code and reason phrase, respectively. REQUEST_TOKEN and RESPONSE_TOKEN are used to match requests and responses. SCRIPT_COOKIE can be used to store state information across invocations within the same transaction.

REMOTE_ADDR and REMOTE_HOST determine the IP address and DNS name of the client that sent the message to the server, respectively. REMOTE_IDENT can be used to supply identity information with Identification Protocol, but it isn't too widely used.

The AUTH_TYPE metavariable determines the authorization method, if any. Authentication methods comply to SIP/2.0 specification. Currently the options are Basic, Digest or PGP. REMOTE_USER identifies the user to be authenticated.

CONTENT_LENGTH and CONTENT_TYPE describe the message body. Content type can be any registered MIME type, as stated in [1]. Actual message body can be read from stdin.

Some additional information of the server and the outside world is provided in some special metavariables. The SERVER_NAME metavariable is set to the name of the server. The SERVER_PROTOCOL metavariable is set to the name and revision of the protocol with which the message arrived, e.g. SIP/2.0. The SERVER_SOFTWARE metavariable is set to the product name and version of the server software handling the message. GATEWAY_INTERFACE is the version of SIP-CGI used, e.g. SIP-CGI/1.1. Servers and CGI implementations can check their compatibility based on the information provided. SERVER_PORT is the port on which the message was received.

3.2 Output

The output (stdout) consists of any number of messages determining the desired actions of the server. The messages are like arbitrary SIP messages possibly containing some additional information as special CGI header fields. The status line can be replaced by CGI actions, thus referred as the action line. The messages are separated by double line feeds – in the same way that in a UDP packet in which multiple requests or responses are sent. It is intended that all the actions are performed, but the server can choose which actions it will perform. An example of a SIP-CGI output can be seen in Figure 3. It is explained in the following chapters.

```
1: SIP/2.0 100 Trying
2:
3: CGI-PROXY-REQUEST sip:user@host SIP/2.0
4: Contact: sip:server@domain
5: CGI-Remove: Subject
6:
7: CGI-AGAIN yes SIP/2.0
8:
9: CGI-SET-COOKIE abcd1234 SIP/2.0
```

Figure 3 Example SIP-CGI output

3.2.1 Action lines

If the action line is a normal status line, a normal SIP response is generated according to the status code. CGI header fields (and possibly some others) are discarded and missing fields are filled according to the original message, if needed. For example line 1 in Figure 3 would generate a provisional response to the request being processed.

The action line CGI-PROXY-REQUEST causes the server to forward a request to the specified SIP URI. Message to be sent depends on the triggering point: if the script is triggered by a request, the triggering request is forwarded; if it is triggered by a response, the initial request of the transaction is sent. The initial request can only be known by a stateful server. The request can be supplemented with the header fields possibly contained in the CGI output. Message body can be inserted, substituted or deleted. However, message integrity must be maintained. An example use of CGI-PROXY-REQUEST can be seen in Figure 3, lines 3-5. It forwards the request to sip:user@host, adds a Contact header and removes the Subject (cf. 3.2.2 for details).

CGI-FORWARD-RESPONSE causes the server to forward a response on to its appropriate final destination. The same rules apply for accompanying SIP headers and message bodies as for CGI-PROXY-REQUEST. RESPONSE_TOKEN metavariable can be set.

CGI-SET-COOKIE sets the SCRIPT_COOKIE metavariable to store information across invocations (Figure 3, line 9).

CGI-AGAIN determines whether the script will be invoked for subsequent requests and responses for this transaction. If it won't, the default action is performed for all later invocations. Default action results also if the script doesn't generate any new messages. Line 7 in Figure 3 instructs the script to be invoked again.

3.2.2 CGI Header Fields

CGI header fields pass additional instructions or information to the server. They resemble syntactically

SIP header fields, but their names all begin with `CGI-`. The SIP server strips all CGI header fields from any message before sending it.

To assist in matching responses to proxied requests, the script can place a `CGI-Request-Token` CGI header in a `CGI-PROXY-REQUEST` or a new request. This header contains a token, opaque to the server. When a response to this request arrives, the token is passed back to the script as a meta-header. This allows scripts to fork (send to multiple locations in parallel) a proxy request, and correlate which response corresponds to which branch of the request.

The `CGI-Remove` header allows the script to remove SIP headers from the outgoing request or response. The value of this header is a comma-separated list of SIP headers. If the headers exist in the message, they are removed before sending, for example line 5 in Figure 3 removes the subject, if it exists. It is illegal to try to remove a header that is inserted elsewhere in the script.

3.3 General feasibility of SIP-CGI

SIP-CGI is an interface that provides practically endless possibilities in service creation within the SIP architecture. Since CGI scripts can be whatever programs, it is possible to perform any kind of operations or access external services. This can be considered as a weakness also: If the programs are extensively complex, they can cause severe overloading of the system. Also access to local file systems or similar resources can be misused. This is why care should be taken, when considering third party implementations in CGI. Even though the uploading of scripts can be done straightforwardly, it is impossible to verify the functionality of the code. Therefore it is not advisable to let third party developers or service users freely create new CGI programs. Of course with proper supervision and access restrictions it is possible to expose CGI programmability to a limited number of people/organizations.

CGI scripts can be written in any programming language available for the platform in use. There are many powerful scripting languages such as Perl and various shell scripts that can be used for simple specialized tasks. When more complex operations are needed, actual programming languages can be used. There can be portability problems concerning the variety of languages: in order to implement the service on a different platform, the compiler or interpreter for the implementation language must be available. Even if the language is implemented in the new platform, there can be some dialect variations that can mess up the functionality.

One more disadvantage of SIP-CGI is that every invocation of a script generates a new process. This is quite resource consuming in most of the operating systems. Thus, large number of simultaneous service users can cause overloading.

There are some proxy/application servers with SIP-CGI support available [6]. Programming tools can be used depending on the platform, but their usage is invisible to the CGI interface. Because of its similarity to HTTP-CGI, SIP-CGI will be easy to adopt for experienced web programmers. However, CGI programming is getting a bit “old-fashioned”.

4 SIP Servlet API

SIP Servlet API is an interface for Java programs which control the processing of SIP messages. Similarly to SIP-CGI and HTTP-CGI, the basic idea of SIP Servlet API is from HTTP Servlet API. Currently there is no single standard for SIP Servlet API. Here, we describe the first one of the proposals [8]. The rest of the proposals [9] are either extensions to the first one or competing drafts.

The API is based on Java `interface` definitions. Any server/servlet that `implements` the appropriate interfaces can be used together. The server and the servlets communicate through the API and the state of the servlets is maintained by the JVM (Java Virtual Machine).

The interface for all SIP servlets to be implemented is `SipServlet`. After instantiation (creation of a new object in Java), servlets are initialized and eventually “cleaned” with `init` and `destroy` methods, respectively. Their main function is to pass configuration information and handle the allocation and deallocation of needed resources.

The `SipServlet` interface has methods for different types of messages: `gotRequest` for requests and `gotResponse` for responses. In its `abstract` implementation class, `SipServletAdapter`, `gotRequest` divides requests to their subtypes. Their implementation lies in methods `doInvite`, `doAck`, `doOptions`, `doBye`, `doCancel` and `doRegister`. When the server decides that some servlet is responsible for handling a message, it calls the appropriate method. The methods return boolean values depending on the success. If `false` is returned, the server should apply its default processing to the message.

The work distribution between servlets is based on transactions. When a servlet is registered as a listener to a transaction, it receives all messages related to that transaction. Initially, the server is responsible for this registration. Servlets can register to further transactions and remove registrations via the `SipTransaction` interface.

`SipMessage` and its sub-interfaces `SipRequest` and `SipResponse` represent messages. A new request in a `SipTransaction` can be initiated with its method `createRequest`. A response to a `SipRequest` can be created with its method `createResponse`. The method `send` is used to send messages. A Request needs a next hop address, whereas responses are routed according to their `Via`

fields. Servlets can have different authorizations to generate messages.

Servlets can inspect and modify the messages with certain restrictions. The body of the message can be accessed through the methods `getContent` and `setContent`. Header fields can be inspected with methods `getHeaderNames`, `getHeaders` and `getHeader`. Method `setHeader` is used to modify the headers, excluding so-called system headers that are managed by the SIP stack.

Similarly to SIP-CGI, requests and responses can be tied together with tokens. Sending a request returns a request token that can be used by servlets to match against similar tokens contained in responses. This can be used, for example, in forking requests to different destinations in parallel.

Current registrations of the users can be accessed through the interface `ContactDatabase`. Servlets can inspect (`getContacts`), substitute (`setContacts`), add (`addContact`) or remove (`removeContact`) registrations. Despite of its name, `ContactDatabase` doesn't have to be a database: its internal implementation is hidden and it provides only generic contact information.

`SipURL` represents SIP URL's in the destination of the messages, user addresses etc. With additional information such as display name, URL's can be stored id `SipAddress` interface. `SipAddress` represents the values of `From` and `To` headers. `Contact` is an extended version of `SipAddress`, including expiration information and similar information. `Contact` represents values of `Contact` header and individual entries in the `ContactDatabase`.

Besides the message manipulations and database access, the server can set other restrictions for sensitive operations such as file system or network access. For untrusted code, so-called servlet sandbox or similar models can be used. The idea of the sandbox model is to restrict the set of operations that can be performed. If feasible, even the bytecode of newly installed servlets can be analyzed to ensure that they don't contain buggy or malicious code such as endless loops.

Figure 4 is an example SIP Servlet from [8]. To understand it completely the reader should be familiar with Java API specification [10], but the following brief explanation can be understood cursorily even without prior knowledge about Java. The servlet implements an unconditional call reject. As a service it isn't interesting, but it serves as an example about servlet programming.

The example servlet extends `SipServletAdapter` (line 4), which means that by default it doesn't react to any messages. Only `INVITE` requests are processed (lines 20-25). They are responded with a generic response (lines 21-23), with status code and reason phrase (line 22) stored in the servlet instance (lines 5-6). Customized codes and reasons can have been determined (lines 11-14) during the initialization (lines 8-18), otherwise the default one (line 16) is used. The servlet returns true, which means that no default message processing is needed.

```
 1: import org.ietf.sip.*;
 2:
 3: public class RejectServlet
 4:     extends SipServletAdapter {
 5:  protected int statusCode;
 6:  protected String reasonPhrase;
 7:
 8:  public void init(ServletConfig config) {
 9:   super.init(config);
10:   try {
11:    statusCode = Integer.parseInt(
12:        getInitParameter("status-code"));
13:    reasonPhrase =
14:        getInitParameter("reason-phrase");
15:   } catch (Exception _) {
16:    statusCode = SC_INTERNAL_SERVER_ERROR;
17:   }
18:  }
19:
20:  public boolean doInvite(SipRequest req) {
21:   SipResponse res = req.createResponse();
22:   res.setStatus(statusCode, reasonPhrase);
23:   res.send();
24:   return true;
25:  }
26: }
```

Figure 4 Example SIP Servlet

4.1 General feasibility of SIP Servlet API

In its expression power, SIP Servlet API is quite similar to SIP-CGI. As independent programs, servlets can carry out any kind of tasks needed for the service. However, there are some key differences in these two techniques. Mainly they are the same that those between HTTP-CGI and HTTP Servlet API.

The Java Virtual Machine is running as long as the servlet engine is up. This saves resources, since it is not necessary to generate a new process for every servlet invocation. Once the servlet is instantiated, its methods can be called over and over again. Also the state information is conserved in the servlets themselves, no external mechanism is needed for distributing it.

The tight connection to the server has also other advantages. As stated above, messages and even the database are represented through the API. This makes access to them "handier". It is more convenient and safer to handle headers, database fields etc. when they are readily parsed by the server. It is also easier to control the access when it is done explicitly through the interface. In addition, different kinds of sandbox-like environments can be used.

SIP Servlet API (like practically anything written in Java) is platform independent. Unfortunately it is tied to Java language, so obviously some flexibility is lost. Some operations are more suitable to be performed with a scripting language like Perl, than with a general-purpose language like Java. If it is necessary or more efficient to use scripting languages, some of them can be run natively in Java. There are packages for Perl, regular expressions and many other tools. External scripts can also be invoked as system processes from Java (even CGI can be run from a servlet), but that

should be avoided because it effectively destroys the original idea of tight integration.

There are some proxy/application servers with SIP Servlet API support available [6]. Java itself is widely adopted, with many development environments to choose from. Because of their similarity to HTTP Servlets, SIP Servlets will be easy to adopt for experienced web programmers.

5 H.323 services

5.1 H.450-based services

Originally H.323 intended to handle only basic call control signalling [11]. The first solution to enable advanced services in on top of H.323 was ITU-T specification series H.450. Its idea was to specify individual supplementary services similar to current PSTN services.

The protocol for all H.450-based services is defined in H.450.1. It is derived from QSIQ protocol used between private branch exchanges (PBX), so it can be seen as a protocol for IP PBX services. One large difference to PSTN model is that most of the service logic is in terminal equipment (TE). Since the services are visible in the protocol and the TE's execute the services, it is necessary to both endpoints to understand the logic of the service to be used. This is a major disadvantage, because services will work completely correctly only if all the TE's have the same release of H.323.

The actual services are defined in H.450.2 and up. C Current version (H.323 v. 4) includes H.450.2 to H.450.12: H.450.2 for call transfer, H.450.3 for call diversion (forwarding, deflection) H.450.4 for call hold, H.450.5 for call park and pickup, H.450.6 for message waiting indication, H.450.7 for call waiting, H.450.8 for name identification, H.450.9 for call completion, H.450.10 for call offer, H.450.11 for call intrusion, and H.450.12 for additional common information network services.

5.2 Non-H.450-based services

H.450-based services are a bit cumbersome to deploy. All the services are specified by ITU-T and often all the TE's must support the same version of H.323. Another solution is to separate the service logic from the TE's and implement the services in the gatekeeper. Particularly routing related services could be offered by the gatekeeper.

So far gatekeeper services have been proprietary implementations. There's been some discussion, whether IN should be integrated with gatekeepers. Also other alternatives – maybe similar to CGI or Servlets – could be developed. Since CPL is independent of the signalling protocol, also CPL servers could be implemented in an H.323 environment.

5.3 General feasibility of H.323 services

It can be seen that H.323 is largely based on PSTN-like models. The most significant service implementation proposals are based on PBX and possibly IN technologies.

It is worth thinking over, whether conventional models should be used in IP telephony service implementation. It is clear that for example IN based services must be accessible from IP environment, but it is a completely different issue to reproduce the implementation mechanisms. There are already standards like JAIN for integrating IP telephony systems to IN. Services that are purely developed for the new environment should provide some real added value utilizing the new possibilities.

Many vendors and carriers have already made significant investments in H.323. Equipment and software have been at commercial stage for quite a period. However, at the services side the progress has been a lot slower. Apart from H.450 services and the proprietary implementations, there hasn't been very much service implementation capabilities.

6 Example service architecture

The interfaces presented in chapters 2-4 are typically implemented within a SIP proxy server. Also other SIP signalling server types can host services and the system can also be referred as an application server. More precise description about the overall architecture can be found in [12]. Figure 5 depicts an example of the internal architecture of the application server.

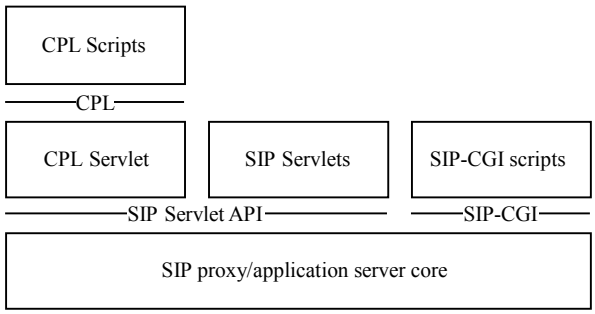


Figure 5 Example service architecture

In the example architecture, both servlets and CGI scripts communicate directly with the signalling server through respective interfaces. CPL scripts are handled by a servlet specialized in that task. CPL support could be also implemented directly in the signalling server or through CGI scripts. In general, this is only a reference architecture, application servers or similar components can be realized in various ways.

7 Conclusions

What comes to signalling and media transmission, IP telephony isn't going to change much. In the long term,

of course operation costs will reduce, because it won't be necessary to maintain two separate networks. Issues like signalling delays and voice quality are going to stay pretty much the same (if they will degrade, users will complain). Of course more advanced codecs and other improvements are being developed but generally there isn't much to do.

The part that is going to change most radically is the services. The existing services in the PSTN and the WWW can be combined. Some examples of the combination are click-to-dial, Unified Messaging (UM) and different kinds of information services. Also completely new kind of services will emerge. The tools used to implement these services are going to be numerous, which can be seen already from the variety of service implementation techniques used in WWW. Some of them have already been adopted in IP telephony. CGI and servlets are being standardized for SIP, and components like Java Beans are widely used in service creation environments. Just wait for the IP telephony equivalents of ASP, JSP, JavaScript, VBScript, VRML, FutureSplash, Shockwave and others to appear.

Like now everyone can run a web server, in the future communications services could be distributed among individuals. There is a project similar to Apache starting to implement an open source SIP proxy server with CGI and servlets. It could be downloaded and installed by anyone, and services could be developed as in a kind of "home-made telephone exchange". Of course carrier grade communications services will have their own role regardless of the new, more open solutions. How exactly the transition is going to happen, is still to be seen.

References

[1] Schulzrinne, Henning et al: SIP: Session Initiation Protocol, IEFT, March 1999 - April 2001, http://www.ietf.org/rfc/rfc2543.txt, http://search.ietf.org/internet-drafts/draft-ietf-sip-rfc2543bis-02.txt

[2] ITU-T Recommendation H.323, Packet-Based Multimedia Communications Systems, since 1996

[3] Lennox, Jonathan; Schulzrinne, Henning: CPL: A Language for User Control of Internet Telephony Services, IETF, November 14 2000, http://search.ietf.org/internet-drafts/draft-ietf-iptel-cpl-04.txt

[4] Bray, T. et al: Extensible markup language (XML) 1.0 (second edition), W3C, October 2000

[5] Dawson, F; Stenerson, D.: Internet Calendaring and Scheduling Core Object Specification (iCalendar), IETF, November 1998, http://www.ietf.org/rfc/rfc2445.txt

[6] Schulzrinne, Henning: SIP Implementations, Columbia University, ongoing work, http://www.cs.columbia.edu/~hgs/sip/implementations.html

[7] Lennox, Jonathan et al: Common Gateway Interface for SIP, IETF, January 2001 http://www.ietf.org/rfc/rfc3050.txt

[8] Kristensen, Anders; Byttner, Anders: The SIP Servlet API, IETF, September 1999, http://www.cs.columbia.edu/~hgs/sip/drafts/draft-kristensen-sip-servlet-00.txt

[9] Schulzrinne, Henning: SIP Drafts: APIs and Programming Environments, Columbia University, ongoing work, http://www.cs.columbia.edu/~hgs/sip/drafts_api.html

[10] Java 2 Platform, Standard Edition, v 1.3 API Specification, Sun Microsystems, 1993-2000, http://java.sun.com/j2se/1.3/docs/api/index.html

[11] Liu, Hong; Mouchtaris, Petros: Voice over IP Signalling: H.323 and Beyond, IEEE Communications Magazine, October 2000

[12] Isomäki, Markus: SIP Service Architecture, Helsinki University of Technology, May 2001

MASTER SLAVE PROTOCOL

Sunesh Kumra
Nokia Networks
Takimo 1, Pitajanmaki
Helsinki
Sunesh.Kumra@nokia.com

Abstract

This article explains how the MGCP and MEGACO protocols work. A brief introduction about how MGCP was born is given and then the various messages in the MGCP protocols are explained. Couple of scenarios are then presented where we see how the protocol actually works. This is followed by brief look at the other variant of this Master Slave protocol called Megaco. Conclusion of the paper is then presented. Appendix A contains the glossary of terms used in this article, while Appendix B contains the notations used to explain MGCP Messages. Appendix C contains some interesting comments made at the VON conference.

1 Introduction

In 1998 some R&D departments started to realize that H.323v1 was not satisfying some very important requirements from the carriers. Lack of mature products, lack of some features in H.323v1, lack of marketing efforts in favor of H.323v2 and time to market issues pushed the incumbent vendors to react against H.323 and propose alternative protocols to address the needs of large-scale phone-to-phone deployments. In mid 1998, the important RFI (Request for Information) and RFP (Request for Proposal) for building large VoIP networks were sent to vendors.
The first proposal came from Bellcore (now Telcordia) and Cisco by the name of SGCP (Simple Gateway control protocol). The second proposal came from ITU-T SG16, ETSI TIPHON and IETF by the name of IPDC (Internet Protocol Device Control).
It was not long before the forces behind these two protocols realized that by unifying their efforts they could get bigger consensus and foster the adoption of their position. Bellcore and Level3 played a key role in merging these protocols into one, the MGCP (Media Gateway Control Protocol).
Some time later another protocol by the name of MEGACO was introduced. Megaco is now a coordinated standard between IETF (MEGACO) and the ITU (H.248).
In both MGCP and MEGACO/H.248 the main two components are Media Gateway and Media Gateway Controller.

Media Gateways are low intelligence distributed devices, which terminates lines/trunks and provide translation of POTS voice/fax signals for IP transport.
Media Gateway Controller provides centralized intelligence for

a) total control over Media Gateways
b) Call admission and billing
c) Signaling interface to PSTN
d) Translation for other protocols. E.g. SIP and H.323.

2 MGCP

MGCP is designed to interface a media gateway controller and media gateway. MGCP is a text-based protocol and supports centralized call model. MGCP is a master slave protocol. MGCP assumes a call control architecture where the call control "intelligence" is outside the gateways and handled by external call control elements.
In its principle MGCP is very close to the proprietary protocols of the switch manufacturers that convey information back and forth between call control points and service switching points. This principle in the context of MGCP clearly places the intelligence on the physically separate entity, the media gateway controller and not on the hardware endpoint, the media converter. But unlike the switch architecture as specified in IN documents where the call control remains close to the actual hardware endpoints, in the MGCP architecture the call control functionality is no longer attached to the media part.
The MGCP assumes that these call control elements, or Call Agents, will synchronize with each other to send coherent commands to the gateways under their control. MGCP does not define a mechanism for synchronizing Call Agents. MGCP is, in essence, a master/slave protocol, where the gateways are expected to execute commands sent by the Call Agents.
MGCP allows combination of commands to be sent in one PDU, this combination reduces the number of messages necessary to establish a call. However, MGCP still requires at least 11 round trips to establish a phone to phone call.
MGCP has seamless PSTN Integration. Many existing Internet Telephony solutions require two stage dialing where a gateway number must be dialed prior to dialing the actual destination number. This is cumbersome for the end-user. However, if gateways are made dumb then they will be inexpensive enough

for the end-users to buy and place in their home. This avoids the need for two-stage dialing since the users telephone will already be connected to the gateway! MGCP assumes a connection model where the basic constructs are endpoints and connections. Endpoints are sources or sinks of data and could be physical or virtual. Example of physical endpoints is an interface on a gateway that terminates a trunk connected to a PSTN switch. Example of a virtual endpoint is an audio source in an audio- content server.
Connections may be either point to point or multipoint. A point to point connection is an association between two endpoints with the purpose of transmitting data between these endpoints. Once this association is established for both endpoints, data transfer between these endpoints can take place. A multipoint connection is established by connecting the endpoint to a multipoint session.
Connections can be established over several types of bearer networks:

- Transmission of audio packets using RTP and UDP over a TCP/IP network.
- Transmission of audio packets using AAL2, or another adaptation layer, over an ATM networks.
- Transmission of packets over an internal connection, for example the TDM backplane or the interconnection bus of a gateway.

2.1 Telephony Gateway

A telephony gateway is a network element that provides conversion between the audio signals carried on telephone circuits and data packets carried over the Internet or over other packet networks. Examples of gateways are:

- Trunking gateways, that interface between the telephone network and a Voice over IP network. Such gateways typically manage a large number of digital circuits
- Voice over ATM gateways, which operate much the same way as voice over IP trunking gateways, except that they interface to an ATM network.
- Residential gateways, that provide a traditional analog (RJ11) interface to a Voice over IP network. Examples of residential gateways include cable modem/cable set-top boxes, xDSL devices, broad-band wireless devices
- Access gateways, that provide a traditional analog (RJ11) or digital PBX interface to a Voice over IP network. Examples of access gateways include small-scale voice over IP gateways.
- Business gateways, that provide a traditional digital PBX interface or an integrated "soft PBX" interface to a Voice over IP network.
- Network Access Servers that can attach a "modem" to a telephone circuit and provide data access to the Internet. It is expected, in the future, the same gateways will combine Voice over IP services and Network Access services.
- Circuit switches, or packet switches, which can offer a control interface to an external call control element.

Note: The examples of gateways give above are just functional classification of gateway. It is possible that two or more gateways explained above are present in the same physical gateway.

2.2 Calls and Connections

Connections are created on the call agent on each endpoint that will be involved in the "call." Each connection will be designated locally by a connection identifier, and will be characterized by connection attributes.
When the two endpoints are located on gateways that are managed by the same call agent, the creation is done via the three following steps:

1. The call agent asks the first gateway to "create a connection" on the first endpoint. Denoted by Step 1 in Figure 1. The gateway allocates resources to that connection, and respond to the command by providing a "session description." (step 2) The session description contains the information necessary for a third party to send packets towards the newly created connection, such as for example IP address, UDP port, and packetization parameters.
2. The call agent then asks the second gateway to "create a connection" on the second endpoint. (Step 3) The command carries the "session description" provided by the first gateway. The gateway allocates resources to that connection, and respond to the command by providing its own "session description."(Step 4).
3. The call agent uses a "modify connection" command to provide this second "session description" to the first endpoint.(Step 5) Once this is done, communication can proceed in both directions.

.

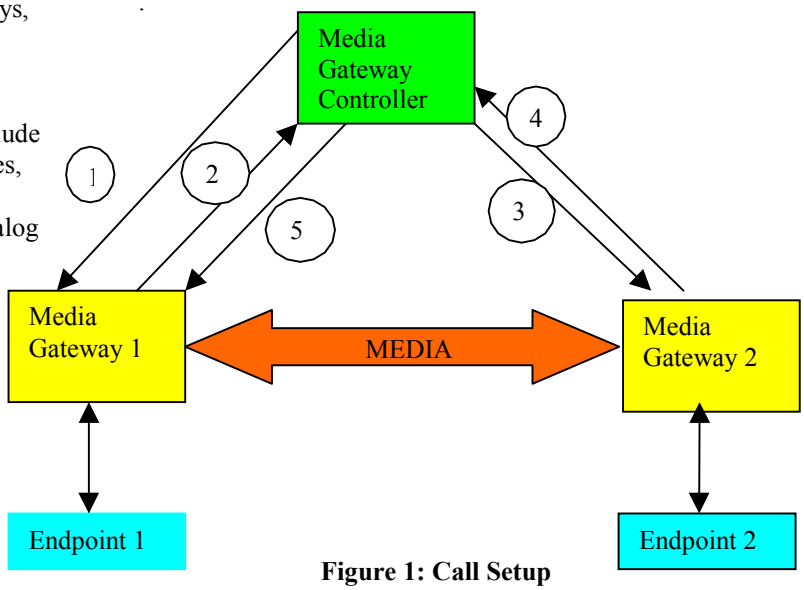


Figure 1: Call Setup

When the two endpoints are located on gateways that are managed by the different call agents, these two call agents shall exchange information through a call-agent to call-agent signaling protocol, in order to synchronize the creation of the connection on the two endpoints. Once established, the connection parameters can be modified at any time by a "modify connection" Command. The call agent may for example instruct the gateway to change the compression algorithm used on a connection, or to modify the IP address and UDP port to which data should be sent, if a connection is "redirected."

The call agent removes a connection by sending to the gateway a "delete connection" command. The gateway may also, under some circumstances, inform a gateway that a connection could not be sustained

2.3 Usage of SDP

The Call Agent uses the MGCP to provision the gateways with the description of connection parameters such as IP addresses, UDP port and RTP profiles. These descriptions will follow the conventions delineated in the Session Description Protocol which is now an IETF proposed standard, documented in RFC 2327.
SDP allows for description of multimedia conferences. This version limits SDP usage to the setting of audio circuits and data access circuits. The initial session descriptions contain the description of exactly one media, of type "audio" for audio connections, "nas" for data access.

2.4 High Availability and Load Balancing in MGCP

Call Agents are identified by their domain name, not their network addresses, and several addresses can be associated with a domain name. In a typical configuration, the MG sends Notifications to the CA. After trying to contact the CA for some configurable number of times and not getting any response back, it starts contacting the other (back-up) MGC within the same domain name.
If a CA is overloaded, it can inform the MG about the same, by changing the Notified Entity with the MG to a new CA. Therefore, when the MG has to deliver the next Notification, it does so to the new CA.

2.5 MGCP Commands

The table below lists the various MGCP Commands. CA denotes the Call Agent and GW denotes the Gateway.
CA --> GW would mean that the command is send from CA to GW.

Table 3: MGCP Commands

Sr no.	Commands	Command flow
1	CreateConnection	CA --> GW
2	ModifyConnection	CA --> GW
3	DeleteConnection	CA -> GW
4	NotificationRequest	CA --> GW
5	Notify	CA <-- GW
6	AuditEndpoint	CA --> GW
7	AuditConnection	CA --> GW
8	RestartInProgress	CA <-- GW
9	Endpoint Configuration	CA --> GW

We shall now look into the individual MGCP Commands. Every command is represented by a few parameters, details on what those parameters can be found in Appendix B. For more information on how command is represented, check the RFC 2705.

2.5.1 Endpoint Configuration

The EndpointConfiguration commands are used to specify the encoding of the signals that will be received by the endpoint. For example, in certain international telephony configurations, some calls will carry mu-law encoded audio signals, while other will use A-law. The Call Agent will use the EndpointConfiguration command to pass this information to the gateway.

Command is represented by:
ReturnCode
EndpointConfiguration(EndpointId,
BearerInformation)

2.5.2 Notification Request

The Notification Request commands are used to request the gateway to send notifications upon the occurrence of specified events in an endpoint. For example, a notification may be requested for when a gateway detects that an endpoint is receiving tones associated with fax communication.
One of the nice features of this command is the association of actions with each of the events. Using this facility, the communication and processing of information between the two entities can be optimized. To each event is associated an action, which can be:

- Notify the event immediately, together with the accumulated list of observed events,
- Accumulate the event in an event buffer, but don't notify yet.
- Accumulate according to Digit Map.

Command is represented by:
ReturnCode
NotificationRequest(EndpointId,
[NotifiedEntity,]
[RequestedEvents,]
RequestIdentifier,
[DigitMap,]
[SignalRequests,]
[QuarantineHandling,]
[DetectEvents,]
[encapsulated EndpointConfiguration])

2.5.3 Create Connection

This command is used to create a connection between two endpoints. In addition to the necessary parameters that enable a media gateway to create a connection, the

localConnectionOptions parameter provides features for quality of service, security, and network related QOS.

Command is represented by:
ReturnCode,
ConnectionId,
[SpecificEndPointId,]
[LocalConnectionDescriptor,]
[SecondEndPointId,]
[SecondConnectionId]
CreateConnection(CallId,
EndpointId,
[NotifiedEntity,]
[LocalConnectionOptions,]
Mode,
[{RemoteConnectionDescriptor |
SecondEndpointId},]
[Encapsulated NotificationRequest,]
[Encapsulated
EndpointConfiguration])

2.5.4 Modify Connection

This command is used to modify the characteristics of a gateway's "view" of a connection. This "view" of the call includes both the local connection descriptors as well as the remote connection descriptor.

Command is represented by:
ReturnCode,
[LocalConnectionDescriptor]
ModifyConnection(CallId,
EndpointId,
ConnectionId,
[NotifiedEntity,]
[LocalConnectionOptions,]
[Mode,]

[RemoteConnectionDescriptor,]
[Encapsulated NotificationRequest,]
[Encapsulated
EndpointConfiguration])

2.5.5 Delete Connection

This command is used to terminate a connection. As a side effect, it collects statistics on the execution of the connection. If there are more than one gateway involved, the call agent will send the Delete Connection command to each of the media gateways. It is also possible for the Call Agent to delete multiple connections at the same time, using for example wild card options.

Command is represented by:
ReturnCode,
Connection-parameters
DeleteConnection(CallId,
EndpointId,
ConnectionId,
[Encapsulated NotificationRequest,]
[Encapsulated
EndpointConfiguration])

In some circumstances, a gateway may have to clear a connection, for example because it has lost the resource associated with the connection, or because it has detected that the endpoint no longer is capable or willing to send or receive voice. The gateway terminates the connection by using a variant of the DeleteConnection command.

2.5.6 Audit Endpoint

The Audit EndPoint command can be used by the Call Agent to find out the status of a given endpoint. This feature has been inherited from the switch environment.

Command is represented by:
ReturnCode,
EndPointIdList|{
[RequestedEvents,]
[DigitMap,]
[SignalRequests,]
[RequestIdentifier,]
[NotifiedEntity,]
[ConnectionIdentifiers,]
[DetectEvents,]
[ObservedEvents,]
[EventStates,]
[BearerInformation,]
[RestartReason,]
[RestartDelay,]
[ReasonCode,]
[Capabilities]}
AuditEndPoint(EndpointId,
[RequestedInfo])

2.5.7 Audit Connection

The Audit Connection command can be used by the Call Agent to retrieve the parameters attached to a connection.

Command is represented by:
ReturnCode,
[CallId,]
[NotifiedEntity,]
[LocalConnectionOptions,]
[Mode,]
[RemoteConnectionDescriptor,]
[LocalConnectionDescriptor,]
[ConnectionParameters]
AuditConnection(EndpointId,
ConnectionId,
RequestedInfo)

2.5.8 Restart in Progress

The RestartInProgress command is used by the gateway to signal that An endpoint, or a group of endpoint, is taken in or out of service.

Command is represented by:
ReturnCode,
[NotifiedEntity]

RestartInProgress (EndPointId,
RestartMethod,
[RestartDelay,]
[Reason-code])

3 Protocol At Work

We shall now see how MGCP works with the help two examples.

3.1 MGCP in all IP Network

Let us now see how MGCP works in the case of all IP network. In the figure RGW = Residential Gateway, CA = Call Agent and EP = Endpoint.
For the sake of discussion, it is assumed that the two EPs, which want to talk with each other, are under the control of the same CA.
In the figure below the solid lines denote the signalling path and the dashed line denotes the media flow. The RGW, CA and database are all part of the IP Network.

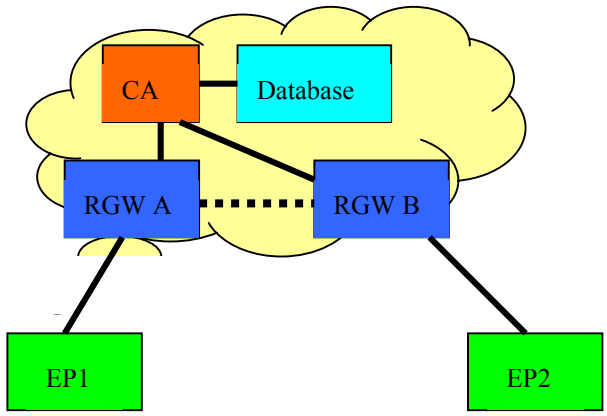


Figure 2: MGCP in all IP Network

1 CA directs the RGW A to look for an off-hook event and report it. Sends a Notification request to RGW A.

2 RGW A goes off-hook and the same is detected by the RGW A and Notification is sent to the CA.
3 CA looks for the service associated with the off-hook event and asks the RGW A to collect the digits and play dial tone to EP1
4 RGW A accumulates the digits and send Notification to CA.
5 CA send a Notification Request to RGW A to stop collecting digits and look for an on-hook event.
6 CA seizes the incoming circuit (asks the RGW to create a call context) and then send the Create Connection command to RGW A.
7 RGW A sends back the SDP (Session Description Parameter) to the CA.
8 CA finds the IP address that serves the dialed number for EP2 from the database.
9 After CA knows the IP address of RGW B, it sends Create Connection command to it.
10 RGW B responds sending back its SDP.
11 CA now sends the SDP from RGW B to RGW A in the Modify Connection command. AT this point two legs of the call are established in half duplex mode.
12 CA instructs RGW B to start ringing by sending Notification Request.
13 CA notifies EP1 that EP2 is ringing
14 EP2 answers the call and the RGW B sends the CA Notification that EP2 is answering the call.
15 CA sends a Notification Request to RGW A to stop ringing
16 CA sends Modify Connection to RGW A to change the communication mode from half duplex to full duplex.
17 The EP1 and EP2 are now talking!

3.2 MGCP in PSTN - IP Network

This is the case where the User A is in the PSTN network and he wants to call to an IP phone.
As before the solid lines denote the signaling path and the dotted lines denote the media path.
Point to be noted is that SG and TGW are on the edge of the IP cloud. They interface with both the IP world and SS7 and PSTN world respectively.

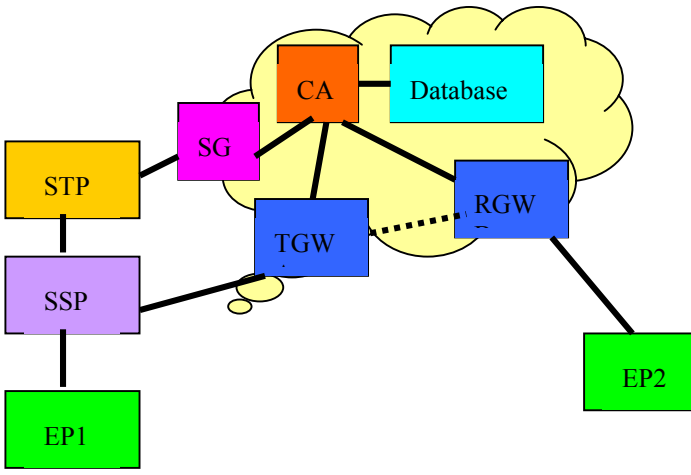


Figure 3: MGCP in PSTN-IP Network

1 EP1, which is in the PSTN world, dials the number of EP2.
2 This number reaches SSP through EP1s local exchange.
3 SSP issues IAM (IAM is the ISUP Initial Address Message) to the CA, which is in the IP world. This IAM reaches SG via STP. SG is connected to IP world on one side and the SS7 world on the other.

SG converts the ISUP on SS7 to ISUP on IP and sends the message to CA.

4 CA finds the IP address that serves the dialed number for EP2 from the database.

5 CA now sends the Create Connection command to the TGW to connect to the incoming trunk using CIC. TGW returns the SDP of the connection.

6 CA seizes the incoming trunk (asks the RGW to create call context) and reserves the outgoing trunk by sending the Create Connection to the RGW passing the SDP of TGW.

7 CA now sends Modify Connection to the TGW.

8 CA requests the RGW to ring the called line by sending Notification Request to the RGW.

9 When the CA receives the Ack from the RGW, it issues ACM to the SG.

10 The SG forwards the ACM (ACM is the ISUP Address Complete Message) to the SSP.

11 EP2 goes off-hook, the RGW notifies the CA by sending the Notification Request.

12 Now the voice channel has to be turned into the full duplex mode, the CA does this be sending the Modify Connection command to the TGW.

13 CA then sends the answer message to the SG, the STP forwards this message to the SSP.

14 The EP1 and EP2 are now talking!

4 MEGACO

MEGACO is used between elements of a physically decomposed multimedia gateway, i.e. a Media Gateway and a Media Gateway Controller. Megaco meets the requirements for a MGCP as defined in RFC 2705.

4.1 Connection Model

The connection model for the protocol describes the logical entities,or objects, within the Media Gateway that can be controlled by the Media Gateway Controller.

The main abstractions used in the connection model are Terminations and Contexts.

A Termination sources and/or sinks one or more streams. In a multimedia conference, a Termination can be multimedia and sources or sinks multiple media streams. The media stream parameters, as well as modem, and bearer parameters are encapsulated within the Termination.

A Context is an association between a collection of Terminations. There is a special type of Context, the null Context, which contains all Terminations that are not associated to any other Termination.For instance, in a decomposed access gateway, all idle lines are represented by Terminations in the null Context.

The following figure is a graphical depiction of these concepts.

The empty dashed box in each context represents the logical association of Terminations implied by the Context.

The protocol provides commands for manipulating the logical entities of the protocol connection model, Contexts and Terminations.

Commands provide control at the finest level of granularity supported by the protocol. For example, Commands exist to add Terminations to a Context, modify Terminations, subtract Terminations from a Context, and audit properties of Contexts or Terminations. Commands provide for complete control of the properties of Contexts and Terminations. This includes specifying which events a Termination is to report, which signals/actions are to be applied to a Termination and specifying the topology of a Context (who hears/sees whom). Most commands are for the specific use of the Media Gateway Controller as command initiator in controlling Media Gateways as command responders. The exceptions are the Notify and ServiceChange commands.

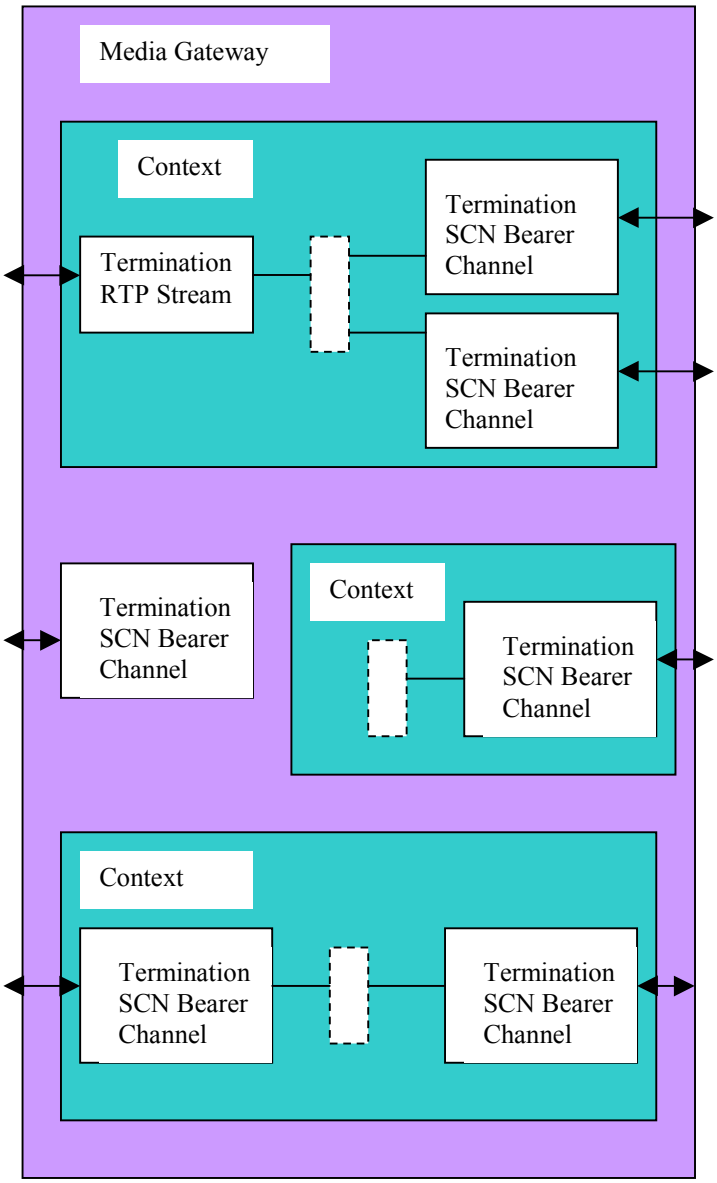


Figure 4: Connection Model

4.2 MEGACO Commands

Following are the various Megaco Commands.

- Add. The Add command adds a termination to a context. The Add command on the first Termination in a Context is used to create a Context
- Modify. The Modify command modifies the properties, events and signals of a termination.
- Subtract. The Subtract command disconnects a Termination from its Context and returns statistics on the Termination's participation in the Context. The Subtract command on the last Termination in a Context deletes the Context.
- Move. The Move command atomically moves a Termination to another context.
- AuditValue. The AuditValue command returns the current state of properties, events, signals and statistics of Terminations.
- AuditCapabilities. The AuditCapabilities command returns all the possible values for Termination properties, events and signals allowed by the Media Gateway.
- Notify. The Notify command allows the Media Gateway to inform the Media Gateway Controller of the occurrence of events in the Media Gateway.
- ServiceChange. The ServiceChange Command allows the Media Gateway to notify the Media Gateway Controller that a Termination or group of Terminations is about to be taken out of service or has just been returned to service. ServiceChange is also used by the MG to announce its availability to an MGC (registration), and to notify the MGC of impending or completed restart of the MG. The MGC may announce a handover to the MG by sending it a ServiceChange command. The MGC may also use ServiceChange to instruct the MG to take a Termination or group of Terminations in or out of service.

5 Comparison between MGCP and MEGACO

Now that we have had a brief look at the two protocols, let us make a comparison between them.

Table 2: Comparison

Features	**MEGACO**	**MGCP**
Server	Media Gateway Controller	Call Agent
Call Representative	Terminations within a call context	Endpoints with Connections
Call Types	Any combination of multimedia and conferencing	Point to point and multipoint audio.
Coding	Text or binary	Text
Internet Protocol Evolution	TCP or UDP Formal extension process defined within the IETF and the ITU	UDP Less structured process, managed by industry consortia

6 Conclusion

With Megaco you can do everything that you could have done with MGCP and more. Megaco would be primarily used for the Media Gateway Control in the future. MGCP is being tested in many networks today and should soon be operational commercially, but the popularity is Megaco is fast rising. Since MGCP would be soon deployed, so it is likely to stay for some time. However the networks that will appear maybe a year from now will likely use Megaco for Media Gateway Control. So I see that MGCP and Megaco will co-exist for some years, before we mainly have Megaco for Media Gateway Control.

References

[1] Request for Comments: 2705: Media Gateway Control Protocol

[2] JAIN MGCP API, version 0.9.

[3] IP Telephony Packet-based multimedia communications systems

[4] www.pulver.com

Appendix A

Glossary

Terms	Meaning
STP	Signaling Transfer Points
SP	Signaling Point
ISUP	ISDN User Part
SSP	Service Switching Points
SCP	Service Control Points
TGW	Trunk Gateway
RGW	Residential Gateway
EP	Endpoint
MGCP	Media Gateway Control Protocol
CA	Call Agent
MG	Media Gateway
SG	Signaling Gateway
JAIN	Java APIs for Integrated Networks
SGCP	Simple Gateway control protocol
RFI	Request for Information
RFP	Request for Proposal
SS7	Signaling System No. 7
PSTN	Public Switched Telephone Network
IN	Intelligent Network
UDP	User Datagram Protocol
ITU	International Telecommunication Union
IETF	Internet Engineering Task Force
IP	Internet Protocol
IAM	Initial Address Message
CIC	Circuit identification code
ACM	Address Complete Message
VON	Voice on Net

Appendix B

- ReturnCode: ReturnCode is a parameter returned by the gateway. It indicates the outcome of the command and consists of an integer number optionally followed by commentary.
- EndpointId: EndpointId is the name for the endpoint in the gateway where command executes.
- BearerInformation: BearerInformation is a parameter defining the coding of the data received from the line side.
- NotifiedEntity: NotifiedEntity is specifies where the notifications should be sent. When this parameter is absent, the notifications should be sent to the originator of the NotificationRequest.
- RequestedEvents: RequestedEvents is a list of events that the gateway is requested to detect and report. Such events include, for example, fax tones, continuity tones, or on-hook transition. To each event is associated an action
- RequestIdentifier: RequestIdentifier is used to correlate the request with the notifications that it triggers.
- DigitMap: DigitMap allows the Call Agent to provision the gateways with a digit map according to which digits will be accumulated. If this parameter is absent, the previously defined value is retained.
- SignalRequests: SignalRequests is a parameter that contains the set of signals that the gateway is asked to apply to the endpoint, such as, for example ringing, or continuity tones. Signals are identified by their name, which is an event name, and may be qualified by parameters.
- QuarantineHandling: The QuarantineHandling parameter specifies the handling of "quarantine" events, i.e. events that have been detected by the gateway before the arrival of the NotificationRequest command, but have not yet been notified to the Call Agent.
- DetectEvents: DetectEvents specifies a list of events that the gateway is requested to detect during the quarantine period.
- ConnectionId: ConnectionId uniquely identifies the connection within one endpoint.
- SpecificEndpointId: SpecificEndPointId parameter identifies the responding endpoint when returned from a CreateConnection command.
- LocalConnectionDescriptor: LocalConnectionDescriptor is a session description that contains information about addresses and RTP ports, as definedin SDP.
- SecondEndpointId: When a SecondEndpointId is returned from a CreateConnection command, the command really creates two connections that can be manipulated separately through ModifyConnection and DeleteConnection commands.
- SecondConnectionId: When this is returned from a CreateConnection, it identifies the second connection.
- LocalConnectionsOptions: LocalConnectionOptions is a parameter used by the Call Agent to direct the handling of the connection by the gateway. Some of the fields contained in LocalConnectionOptions are: Encoding Method, Packetization period, Bandwidth, Type of Service,Usage of echo cancellation and so on.
- Mode: Mode indicates the mode of operation for this side of the connection. The mode are "send", "receive", "send/receive", "conference", "data", "inactive", "loopback", "continuity test", "network loop back" or "network continuity test."
- DetectEvents: DetectEvents, the list of events that are currently detected inquarantine mode.
- RestartMethod: The RestartMethod parameter specified the type of restart of the endpoint. The methods include "graceful" and "forced".
- RestartDelay: The parameter is expressed as a number of seconds. If the number is absent, the delay value should be considered null.

- Capabilities: The capabilities for the endpoint are similar to the LocalConnectionOptions parameter and including event packages and connection modes.

Appendix C

Mentioned below are some interesting comments from Speakers during the VON conference.

- In 1998 there were more than 1 trillion minutes of POTS usage.
- The US market for Telephony services is about $250 billions and the global telecom service market is about $800 billion.
- The cross-over for the wide-area data traffic exceeding voice traffic is happening about now, but voice revenues are much greater than the data revenues.
- By 2004, 5% to 20% of long distance calls will be VoIP.
- Circuit switching will be dead by 2005.
- Voice will be only 1% of the total global network traffic by 2008.
- The worldwide market for IP Telephony will grow from $480 million in 1999 to $19 billion in 2004.

Network dimensioning for voice over IP

Tuomo Hakala
Oy Datatie Ab
tuomo.hakala@datatie.fi

Abstract

This article concentrates in the issues of network dimensioning for voice over IP (VoIP). The network under dimensioning is an IP network between VoIP user devices. First, a short introduction to VoIP in general is given. Second, the issues in network dimensioning for VoIP are identified. Third, bandwidth requirements of VoIP are calculated. Fourth, basic approaches to Quality of Service are discussed and finally conclusions are drawn.

1 Introduction

VoIP represents the best opportunity so far for voice and data convergence and it is now one of the fastest-growing industries [12]. An IP network with mixed voice and data makes the network management easier than managing separate voice and data networks. A VoIP call uses less bandwidth than a circuit-switched call. VoIP makes new services possible.

IP networks, like the current Internet, offering only best-effort service, cannot satisfy the Quality of Service (QoS) requirements of VoIP. This is primarily because of the variable queuing delays and packet loss during network congestion [12].

The end-to-end Quality of Service of VoIP is composed of factors related to the network and factors related to the applications. Factors related to the network are [13] [1]:

- Network delay
- Network jitter
- Network packet loss and desequencing

Factors related to the applications are:

- Overall packet loss
- Jitter buffers
- Codec performance
- Overall delay

Currently there are several approaches to improve the audio quality of VoIP [7]:

- **Integrated Services (IntServ)** is a stateful approach where resources are reserved in the network before data starts to flow along the reserved path. [8]
- **Differentiated Services (DiffServ)** is a stateless approach where real-time traffic is marked to get preferred treatment in the network. [9]
- **Forward Error Correction (FEC)** algorithms reduce the impact of data loss by sending redundant data along with the audio data. The redundant data helps to reconstruct lost data. [10] [14]
- **Loss Concealment** algorithms try to reduce the impact of data loss by replacing the lost audio with an approximation. [11]

Forward Error Correction and Loss Concealment algorithms are methods used in the VoIP user devices. IntServ and DiffServ are methods used in the IP network.

2 The issues

During an average conversation, each party usually talks only about 35 percent of the time. Most of the techniques used to transform voice into data, the codecs, have the ability to detect silences. With this voice activity detection, data is transmitted only when needed. When several conversations are multiplexed on a single transmission line, statistical multiplexing can be used which leads to more efficient use of bandwidth.

When a VoIP packet is transferred through an IP network, it will experience delay that is caused by:

- Transmission delay between the nodes, depends on the frame size and the transmission speed
- Queuing delay in the nodes because of buffering

- Switching and processing delay in the nodes, the time to switch a frame from an input port to an output port

- Propagation delay, depends on the characteristics of the transmission media and the distance between the nodes

The use of statistical multiplexing means that the delay of sent packets within a conversation will vary. This varying delay is called jitter. The jitter must be minimized in the network and the remaining jitter needs to be corrected by the receiving side using jitter buffers to make the original speech intelligible. Jitter buffers increase the overall delay.

Several technologies enable the use of statistical multiplexing and mixing of voice and data on the same transmission lines. Such technologies are voice over Frame Relay, voice over ATM (Asynchronous Transfer Mode) and VoIP. VoIP is the most flexible technology because it does not require virtual channels to be set up between the parties having a conversation. Also, VoIP scales in terms of connectivity much better than frame relay or ATM.

In IP networks, routers are the devices that execute the statistical multiplexing functionality. IP packets belonging to the same conversation may use different routes having different delays and therefore they may be received in different order than in which they were sent. This is called desequencing.

When an overflow of the buffers in the network nodes occurs because of network congestion, there will be some packet loss, which must be handled by the receiving side. It makes no sense to resend part of speech because of the overall delay. [1]

The bandwidth required by VoIP must be calculated considering the bandwidth requirements of a single conversation and the number of conversations on each link in the network. Acceptable packet loss and the buffering capacity of the nodes in the network must be considered as well. Delay and jitter must be minimized in the network.

The receiving side must take care of the remaining network jitter and the desequencing of packets. The Real-time Transport Protocol (RTP) was designed to allow the receiver to do the correction [4]. RTP includes:

- Information on the type of data transported
- Timestamps
- Sequence numbers

Real-Time Control Protocol (RTCP) allows the conveyance of feedback on the quality of the transmission and it can also carry information on the identity of the participants [4]. RTP and RTCP are mostly used on top of User Datagram Protocol (UDP) [3], which provides the use of a port number and a checksum. The use of UDP enables also the use of IP multicast i.e. sending packets to IP multicast addresses. This means that a RTP stream generated by a single source can be received by several destinations. [1]

3 Bandwidth requirements

3.1 The number of calls per link

When bandwidth needs to be reserved for voice in an IP network designed for both voice and data, information needs to be gathered in order to know who phones where, how often and how long. When an existing circuit switched telephone network is planned to be realized by using VoIP, this information can be derived from existing statistics. When VoIP network is planned for a new implementation and no statistics are available, calculations of the number of calls can be done using the Erlang model [1].

An optimal route on the network is chosen for each of the calls considering the cost of each link per unit of bandwidth. After this, it is possible to calculate the number of simultaneous calls, or conversations, on each link in the network at any given time. The maximum number of simultaneous calls is used as the value of N in the following discussion.

Their is some signaling traffic in VoIP before and after a call but the amount of signaling traffic per voice channel is negligible compared to the voice traffic itself and therefore it is not considered in the following discussion.

3.2 The number of active voice channels in one direction of a link

During a voice conversation the proportion of active one-way speech intervals of the whole time of the conversation is called the activity rate a. An average value is usually 0,35 because the parties of a conversation normally think a while before their speak. To be on the safe side a=0,5 should be used in the calculations. This allows each of the two parties of a conversation to use a half of the time of the conversation to speak.

The probability P(I) of having exactly I active voice channels, out of N conversations, in one direction of a link can be expressed with the binomial distribution:

$$P(I) = \frac{N!}{I!(N-I)!} * a^{I} * (1-a)^{N-I} \qquad (1)$$

Figures 1 to 5 show this probability P(I) with activity rate a=0,5 and different values of N.

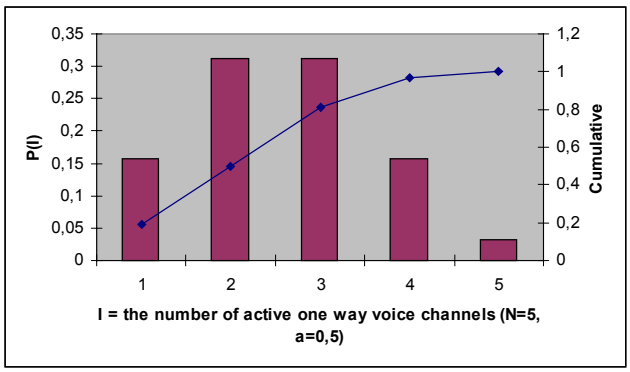


Figure 1: The probability of having I active voice channels in one direction of a link when the number of conversations is N=5 and the activity rate is a=0,5.

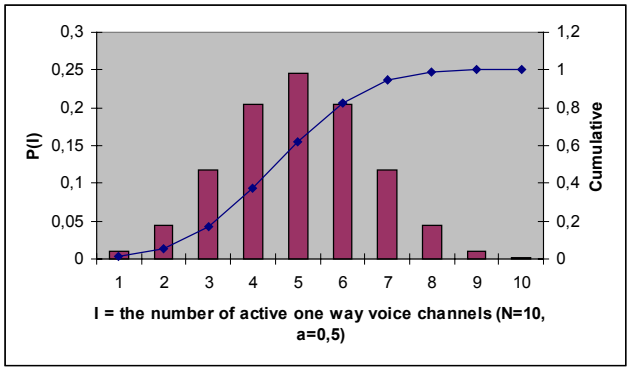


Figure 2: The probability of having I active voice channels in one direction of a link when the number of conversations is N=10 and the activity rate is a=0,5.

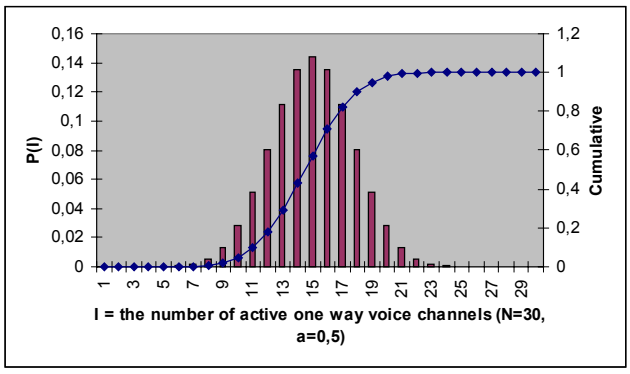


Figure 3: The probability of having I active voice channels in one direction of a link when the number of conversations is N=30 and the activity rate is a=0,5.

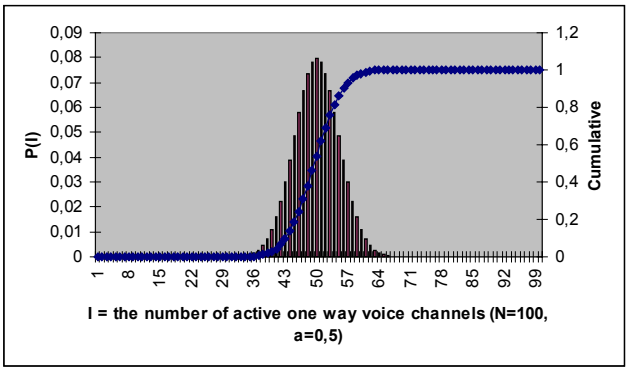


Figure 4: The probability of having I active voice channels in one direction of a link when the number of conversations is N=100 and the activity rate is a=0,5.

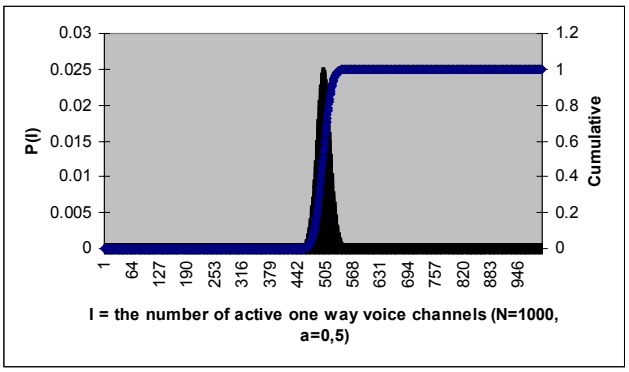


Figure 5: The probability of having I active voice channels in one direction of a link when the number of conversations is N=1000 and the activity rate is a=0,5.

The cumulative graphs in the figures 1 to 5 show the probability of having maximum I voice channels active out of N at the same time in one direction of a link. This can be used in link sizing. As an example, if we know that the maximum number of conversations on a link is 1000 and we want to be 99% sure that all simultaneously active voice channels get all of their packets through the link, we size the link for the bandwidth of 536 times the bandwidth required by a single voice channel.

Table 1 shows I, the maximum number of active voice channels in one direction of a link with various values of N and with the probability of 99%. For N=5, the 97% value is shown. It can be seen, that when N increases the 99% value of I maximum gets closer and closer to a*N. This means that, as a rule of thumb, with a small number of conversations the link must be sized as all conversations were active at the same time and with a large number of conversations the link can be sized as only a bit more than a half of the conversations were active at the same time.

Table 1: I, the maximum number of active voice channels out of N in one direction of a link, with the probability of 99% and a=0,5. For N=5, the 97% value is shown.

N	I	Probability
5	4	97%
10	8	99%
30	21	99%
100	61	99%
1000	536	99%

3.3 Buffering

Buffering in the network increases jitter and therefore reduces interactivity. It is good practice to dimension VoIP links assuming no buffering in the network. This leads to some overprovision for slow links, but this overhead can be used by non real-time traffic in an IP network designed for both voice and data [1]. In an IP network with mixed voice and data the bandwidth requirements of VoIP are small compared to the bandwidth used for data in today's IP networks.

3.4 Link sizing

Table 2 shows the transport header overhead of IPv4, UDP and RTP.

Table 2: Transport header overhead

Protocol	Overhead (octets)
IPv4 (Internet Protocol version 4) [2]	20
UDP (User Datagram Protocol) [3]	8
RTP (Real-time Transport Protocol) [4]	12

Table 3 shows the header overhead with the following level 2 technologies: Frame Relay, PPP (Point-to-Point protocol), POS (Packet over SONET/SDH), ATM/AAL5 with LLC/SNAP (Asynchronous Transfer Mode, ATM Adaptation Layer 5, Logical Link Control/Subnetwork Access Protocol).

Table 3: Total header overhead

Level 2 framing	Frame Relay	PPP	POS	ATM, AAL5, LLC/SNAP (Note)
Level 2 header (octets)	2	8	16	8+8
IPv4+UDP+RTP headers (octets)	40	40	40	40
Header overhead (octets)	42	48	56	56

Note: With ATM, add AAL5 padding octets to get multiples of 48 and add 5 octets for every 48 octets to get the 53 octet cell size.

Table 4 shows the frame frequencies of three codecs. The frame frequency can be used to calculate the total bandwidth of a single active voice stream when the total header overhead is known.

Table 4: Frame frequencies of three codecs

Codec	G.723.1 (5,3 kbit/s)	G.723.1 (6,3 kbit/s)	G.729 (8 kbit/s)
Payload size (octets)	20	24	10
Sample (ms)	30	30	10
Frame frequency (1/s)	33,125	32,8125	100

Table 5 shows the total level 2 frame size when one frame contains a single voice sample and considering the notes from table 3. Including more than one voice sample in one packet would cause additional packetization delay and therefore it is not recommended.

Table 5: Total level 2 frame size (octects)

Codec	G.723.1 (5,3 kbit/s)	G.723.1 (6,3 kbit/s)	G.729 (8 kbit/s)
Frame Relay	62	66	52
PPP	68	72	58
POS	76	80	66
ATM	106	106	106

Table 6 shows the total bandwidth of a single active voice stream considering the frame frequencies calculated in table 5. It is shown that the G.729 coded voice stream that is run over an ATM link requires more bandwidth than 64 kbit/s which is the bandwidth required by a G.711 codec run over a TDM link.

Table 6: Total required bandwidth of a single active voice stream (kbit/s)

Codec	G.723.1 (5,3 kbit/s)	G.723.1 (6,3 kbit/s)	G.729 (8 kbit/s)
Frame Relay	16,430	17,325	41,600
PPP	18,020	18,900	46,400
POS	20,140	21,000	52,800
ATM	28,090	27,825	84,800

Table 7 shows the maximum number of simultaneously active voice streams in one direction with zero packet loss on a SDH/STM-1 link with POS and ATM/AAL5. The available bandwidth on a SDH/STM-1 link is 149,76 Mbit/s from the 155 Mbit/s link speed.

Table 7: Maximum number of simultaneously active voice streams in one direction with zero packet loss on a SDH/STM-1 link with POS and ATM/AAL5 (149,76 Mbit/s available from the 155 Mbit/s link speed)

Codec	G.723.1 (5,3 kbit/s)	G.723.1 (6,3 kbit/s)	G.729 (8 kbit/s)
POS	7 436	7 131	2 836
ATM, AAL5	5 331	5 382	1 766

Table 8 shows the maximum number of simultaneously active voice streams in one direction with zero packet loss on a 64k TDM link with Frame Relay and PPP.

Table 8: Maximum number of simultaneously active voice streams in one direction with zero packet loss on a 64k TDM link with Frame Relay and PPP.

Codec	G.723.1 (5,3 kbit/s)	G.723.1 (6,3 kbit/s)	G.729 (8 kbit/s)
Frame Relay	4	4	2
PPP	4	3	1

4 Delay, jitter and packet loss

When the bandwidth required by VoIP is calculated for a low packet loss assuming no buffering in the network, the network delay and jitter are minimized. In an IP network with mixed voice and data traffic, some mechanism must be used to ensure that the bandwidth calculated for VoIP is not used by other real-time traffic or non real-time traffic. When calculations for VoIP are done for a low packet loss in the network, somehow it must be taken care of that the buffers in the network nodes are not filled with packets of other traffic types which would cause VoIP packets to get dropped causing packet loss. Also, when the calculations for VoIP are done assuming that there is no buffering in the network nodes, because buffering would lead to increased delay and jitter, it must be somehow taken care of, that VoIP packets get sent first to the outgoing link even though there are packets of other traffic type in the buffers.

First of all, VoIP traffic must be somehow differentiated from other traffic types in the network so that it can be treated better. The nodes in the IP network, the routers, can differentiate traffic according to source and destination IP addresses, protocol type, port numbers and by the Differentiated Services (DS) field. The DS field means the type of service (TOS) byte in IPv4 and the traffic class byte in IPv6.

There are two basic approaches in an IP network with mixed voice and data traffic that can be used to improve the quality of VoIP [7]:

- Integrated Services (IntServ) is a stateful approach where resources are reserved in the network before data starts to flow along the reserved path. [8]
- Differentiated Services (DiffServ) is a stateless approach where real-time traffic is marked to get preferred treatment in the network. [9] [5]

4.1 Integrated Services (IntServ)

IntServ model proposes two service classes in addition to best-effort service: guaranteed service and controlled-load service. Guaranteed service is for applications requiring a fixed delay bound. Controlled-load service is for applications requiring reliable and enhanced best-effort service. [12]

IntServ requires that resources are explicitly managed for each real-time application. Routers must reserve resources (e.g. bandwidth and buffer space) in order to provide specific QoS for each packet flow. This requires flow-specific states in the routers. [12]

The four components of IntServ are:

- **Flow specification** - *Flowspec* describes the characteristics of the flow and it has two separate parts, *Tspec* (describes flow's traffic characteristics) and *Rspec* (specifies the service requested from the network)
- **Signaling protocol** - e.g. Resource ReSerVation Protocol (RSVP) [6]
- **Admission control routine** - determines whether a request for resources can be granted.
- **Packet classifier and scheduler** - packets entering a router are classified and put in the appropriate queue and then scheduled accordingly.

4.2 Differentiated Services (DiffServ)

In DiffServ model traffic entering an IP network is classified, marked, policed and shaped at the edge of the network.

The packets are then assigned to different behavior aggregates (BA). Each BA is identified by a single DiffServ CodePoint (DSCP). Users request a specific performance level per packet by marking the DiffServ field of each packet with a specific DSCP value which specifies the Per-Hop-Behavior (PHB) within the provider's network. Packets are forwarded within the core of the network according to the PHB.

The four components of DiffServ are [12]:

- **Services** - Characteristics of packet transmission in one direction over a path in a network are defined by a service. DiffServ can be provided by two approaches:
  - **Quantitative DiffServ** - QoS is specified in deterministically or statistically quantitative terms of throughput, delay, jitter and/or loss.
  - **Priority based DiffServ** - Services are specified in terms of a relative priority of access to network resources.
- **Conditioning Functions and PHB** - A user and a service provider must have a service level agreement (SLA) in place that specifies the supported service classes and the amount of traffic allowed in each class. Individual packets have DiffServ (DS) fields that indicate the desired service and these DS fields can be marked at hosts or at the access router or at the edge router in the service provider network. Packets are classified, policed and possibly shaped at the ingress of the service provider network according to the rules derived from the SLA. Between domains, service provider networks, DS fields may be remarked, if so defined in the SLA between the two service providers. These traffic control functions at hosts, access routers or edge routers are generically called traffic conditioning. Per hop behavior (PHB) are defined to allocate buffer and bandwidth resources at each node among traffic streams. PHB is applied to a DiffServ behavior aggregate and a DiffServ- compliant node.
- **DS CodePoint** – DS field means the type of service (TOS) field in IPv4 and the traffic class byte in IPv6. Six bits of this DS field are used as a codepoint (DSCP) to select the PHB for a packet at each node.
- **A node mechanism for achieving PHB** – Buffer management and packet scheduling mechanisms are used in nodes to achieve a certain PHB. PHBs are defined as behavior characteristics relevant to service provisioning policies instead of particular implementation mechanisms. Various implementation mechanisms may be suitable for a particular PHB group.

5 Conclusions

The issues to be considered in network dimensioning for VoIP are bandwidth, delay, jitter, desequencing and packet loss.

With a small number of conversations the link bandwidth must be sized as all conversations were active at the same time and with a large number of conversations the link can be sized as only a bit more than a half of the conversations were active at the same time.

Buffering in the network increases jitter and therefore reduces interactivity. It is good practice to dimension VoIP links assuming no buffering in the network.

When the bandwidth required by VoIP is calculated for a low packet loss and no buffering is assumed in the network, the network delay and jitter are minimized. The receiving side must correct the remaining network jitter and the desequencing of packets.

In an IP network with mixed voice and data traffic, some mechanism must be used to ensure that the bandwidth calculated for VoIP is not used by other real-time traffic or non real-time traffic. There are two basic approaches to achieve this: Integrated Services (IntServ) and Differentiated Services (DiffServ). IntServ is a stateful approach where resources are reserved in the network before data starts to flow along the reserved path. DiffServ is a stateless approach where real-time traffic is marked to get preferred treatment in the network.

References

[1] Hersent, Olivier; Gurle, David; Petit, Jean-Pierre: IP Telephony, Packet-based multimedia communications systems; Great Britain, 2000, www.awl.com/cseng/, ISBN 0-201-61910-5

[2] Postel, Jon: Internet Protocol, RFC 791, September 1981

[3] Postel, Jon: User Datagram Protocol, RFC 768, 28 August 1980

[4] Schulzrinne, Henning; Casner, Stephen L.; Frederick, Ron; Jacobson, Van: RTP: A Transport Protocol for Real-Time Applications, RFC 1889, January 1996

[5] Blake, Steven; Black, David L.; Carlson, Mark A.; Davies, Elwyn; Wang, Zheng; Weiss, Walter: An Architecture for Differentiated Services, RFC 2475, December 1998

[6] Mankin, A.; Baker, Fred; Braden, Bob; Bradner, Scott; O'Dell, Michael; Romanow, Allyn; Weinrib, Abel; Zhang, Lixia; Resource ReSerVation Protocol (RSVP), Version 1 Applicability Statement, Some Guidelines on Deployment, RFC 2208, September 1997

[7] Trends in the Internet Telephony, http://www.fokus.gmd.de/research/cc/glone/projects/ipt/ (11 March 2001)

[8] IETF Integrated Services (IntServ) Working Group charter, http://www.ietf.org/html.charters/intserv-charter.html (11 March 2001)

[9] IETF Differentiated Services (DiffServ) Working Group charter, http://www.ietf.org/html.charters/diffserv-charter.html (11 March 2001)

[10] Speech Property-Based FEC (SPB-FEC), http://www.fokus.gmd.de/research/cc/glone/products/voice/spb-fec/ (11 March 2001)

[11] Adaptive Packetization / Concealment (AP/C), http://www.fokus.gmd.de/research/cc/glone/products/voice/apc/ (11 March 2001)

[12] Li, Bo; Hamdi, Mounir; Jiang, Dongyi; Cao, Xi-Ren: QoS-Enabled Voice Support in the Next-Generation Internet: Issues, Existing Approaches and Challenges; IEEE Communications Magazine, April 2000

[13] TELECOMMUNICATIONS AND INTERNET PROTOCOL HARMONIZATION OVER NETWORKS ETSI PROJECT – TIPHON, http://webapp.etsi.org/tbhomepage/TBDetails.asp?TB_ID=291&TB_NAME=TIPHON (12 March 2001)

[14] Padhye, Chinmay; Christensen, Kenneth J.; Moreno, Wilfrido: A New Adaptive FEC Loss Control Algorithm for Voice Over IP Applications; IEEE 2000

TRIP, ENUM and Number Portability

Nicklas Beijar
Networking Laboratory, Helsinki University of Technology
P.O. Box 3000, FIN-02015 HUT, Finland
Nicklas.Beijar@hut.fi

Abstract

This paper describes the problem of locating terminals using E.164 numbers, and the problem of selecting a suitable gateway for calls from an IP telephony network to the public switched telephone network (PSTN). Generally, these are the problems of mapping the name of a destination into an address, and to find the best route to the destination in a combined IP and PSTN network. Number portability is closely related to these problems. Due to number portability the address of a destination is changed without changing its name. Number portability may also change the optimal route to a destination.

Two protocols are being developed by the Internet Engineering Task Force (IETF) to solve these problems. The Telephony Routing over IP (TRIP) protocol solves the gateway location problem by distributing routing information between entities on the IP network. The tElephony NUmbering Mapping (ENUM) provides a solution to the terminal location problem based on DNS. ENUM maps an E.164 number into an URI, which is used to locate the end point. Both protocols aim to add a part to the architecture in order to make a global hybrid PSTN-IP network possible. They also aim to enable number portability in the IP network and between the two types of networks.

In this paper, we will introduce the concept of terminal and gateway location. We describe how the current protocols locate terminals and gateways, and what the problems with the current solutions are. The TRIP and ENUM protocols are presented in detail and scenarios based on the protocols are described. Solutions to number portability are presented and some problems are discussed.

Keywords: Voice over IP, IP Telephony, TRIP, ENUM, Number portability, Routing, Address mapping

1 Introduction

When IP telephony was introduced, it was mainly used in small private networks, which were connected to the public switched telephone network (PSTN) through a single gateway. As IP telephony matured, a vision of global public IP telephony became popular. In this scenario the IP and PSTN networks are interconnected with a large number of publicly available gateways. In order to make connectivity between all IP- and PSTN-terminals possible, the problem of terminal and gateway location must be solved. In these problems, addressing is central. Number portability allows users to change operators and locations without changing the telephone number. To smoothen the transition to an IP-based telephone network, number portability is also required between PSTN and IP-networks.

The main signaling protocols for IP telephony are Session Initiation Protocol (SIP) [1] and H.323 [2]. The architectures that they define are similar, although different names are used for the network elements. Calls can be established between IP telephony terminals directly, but usually the call setup signaling passes through a gatekeeper (in H.323) or signaling server (in SIP). The elements have similar functions in the two signaling protocols, so in this paper we will use the name signaling server for both. Some important functions of the signaling server are address translation and location of the destination terminal. For calls between an IP terminal and a terminal on the PSTN, a gateway is used to convert signaling and code the voice stream between the circuit switched and the packet network.

To identify the destination of a call on the PSTN, the caller dials the receiver's telephone number in E.164 format [3]. The telephone number is analyzed digit by digit to locate the path through switches in the PSTN towards the destination. Although the number dialed by the user traditionally is used for routing, a different address is used for routing in many cases.

The European Telecommunications Standards Institute (ETSI) defines the concept of names and addresses as follows: A name is a combination of alpha, numeric or symbols that is used to identify end-users. An address is a string or combination of digits and symbols which identifies the specific termination points of a connection/session and is used for routing. [4]

The main difference between these is that a name is an identifier for the end user, while an address is a locator. An address should typically have some form of structure that allows aggregation for routing purposes. In the Internet, the name is a domain name. The

domain name is mapped into an IP address, which is used for routing. In the telephone network, E.164 numbers have traditionally been used as both names and addresses. However, due to number portability their roles have been separated. The number that the user dials, which can be regarded as a name, is then mapped into a routing number, which is an address. The dialed number is usually referred as a directory number. It is also worth noting, that in many cases entities that functionally are names are called addresses.

To transform the name into an address some type of mapping method is needed. For the mapping of host names into IP addresses, the Domain Name Service (DNS) [15],[16] is used. DNS is a distributed directory service based on DNS servers. Each server knows the mapping of a range of hosts, or the address to a server that has more detailed information. The parts of the domain names are analyzed in hierarchical order and the mapping request is forwarded to more specific DNS servers until the mapping can be completed.

2 The current situation

Today IP telephony is used in mainly two situations: either as a private branch of the PSTN within an organization or for calls between terminals on the Internet. The first case involves gateways, which connects the private IP telephony network with the public switched telephone network. The second case does not usually involve gateways, since it would require publicly available gateways and the existence of a billing system. Additionally IP telephony is used to inexpensively transport long distance calls between PSTN callers through the IP network.

2.1 Locating the destination

In most of today's IP telephony applications, the IP telephony network acts as a branch of the PSTN. The gateway together with the signaling server (or gatekeeper) works like a PBX from the viewpoint of the PSTN. The called E.164 numbers are translated to IP addresses by the signaling server. Calls to and from external numbers are routed through the gateway. In such small networks with only a few gateways, the number translation and gateway selection can be performed by a single signaling server. The mappings are usually configured in the signaling server. The users do not necessary notice that IP telephony is used, since they use E.164 numbers as normally. These types of networks are, however, very limited in size and cannot be considered in a larger deployment of IP telephony. In this paper, we will mainly discuss the use of larger IP telephony networks.

For calls over the public Internet, the situation is more complicated. IP terminals are located using their IP address or host name. The signaling protocols allow various formats of addresses to be used by the users. Users prefer to use E.164 or e-mail type addresses that are familiar from traditional telephony or e-mail, respectively. To set up a call, the name of the destination is mapped to an IP address by a signaling server. The signaling server can be manually configured with the mappings for its local terminals. More usually however, the terminals must register to the signaling server. Based on registration the mapping is created. The server maintains a database of mappings for its registered clients.

The SIP architecture also includes a network element named location server. The location servers store the mappings on the behalf of the signaling servers. A location server may be used by a number of signaling servers. The location server may also be integrated with a signaling server. In this way we can generalize to say that the location server stores the mapping, even though the location server and signaling server in some cases are the same element. In case of separate servers, the information is accesses with some directory access protocol.

In a public IP telephony network there is a large number of signaling servers. There is currently no method to distribute the mappings between different servers. Because of this lack of distribution method, mappings can only be used for calls between phones registered to the same server. Thus, E.164 numbers do not work for calls between terminals registered to different signaling servers or location servers. Note that calls are always possible when the address is given as an IP address.

The SIP protocol also supports the use of names given as Universal Resource Locators (URL) [18]. The URL specifies the user, the host and other parameters. This "user@host" format can be handled in a similar way as email addresses are handled by SMTP [5]. An IP address of a signaling server for the domain is located using DNS. Thus, the host name does not have to be a complete host name. The call can be further forwarded by proxy or redirect servers. [1]

The most popular H.323 client Microsoft Netmeeting uses directory servers to locate users. These directories are propriety solutions, named ULS and ILS. Similar solutions are used by many other clients on the market. Still, there are significant drawbacks in this type of solution. The directories have a limited capacity and they do not exchange information with each other. Furthermore, many of them use non-standardized access protocols. [7]

2.2 Locating the gateway

For calls to the PSTN a gateway must be used. Today, the gateways are in most cases manually configured

into the signaling server. A signaling server has a set of available gateways to use for external calls. For private internal IP telephony networks, external numbers are usually recognized by a preceding “0”.

In SIP the call can be set up using a gateway specified in the URL. The destination is then given as “number@gateway”. This requires the user to know that the destination is on the PSTN and also which gateway should be used. If the gateway is down or if all lines are busy, the user must manually select another gateway. Another method is to let the signaling server choose the gateway, whereas calls can be made by only giving a number. The server selects one from its list of available gateways. The H.323 protocol works in a much similar way.

2.3 Number portability

Number portability allows a user to change service providers, location or service type without changing the telephone number. Service provider portability is mandatory in many countries. The introduction of IP telephony adds a new type of number portability: between different network types. [8]

Today number portability is only implemented on the PSTN. The implementation of number portability differs in different countries. Common to all implementations, is that the directory number dialed by the customer is mapped to either a routing number or a routing prefix. A routing number is a hierarchical routing address, which can be digit-analyzed to reach the correct country, network provider, end-office switch and subscriber line. A routing prefix forms a routing number by adding some digits in front of the directory number. The routing number replaces the hierarchy that is lost, since the directory number space becomes flat due to number portability.

Most number portability solutions utilize Intelligent Network (IN) functions. In these solutions, the mapping database is stored in the Service Data Functions (SDF) elements. Depending on the implementation, the call may have to pass through the previous operator’s network before it reaches the current destination network. [8]

At the time being, there is no specific solutions for number portability across the network types. A number that is moved to the IP network can be handled in a normal way in the number portability solutions on the PSTN. The number portability databases are updated with a routing number that directs calls to a gateway. Limited number portability can be implemented in IP telephony network using redirect and proxy servers. Calls to a moved numbers are forwarded to the new destination by the previous signaling server. However, this feature does more correspond to call forwarding than number portability. The forwarding works with both calls to IP terminals and calls to PSTN destinations.

3 Problem description

3.1 Naming

Traditionally E.164 numbers have been used on the telephone network and e-mail type addresses of format “user@domain” on the Internet. The signaling protocols SIP and H.323 allows using multiple types of names, including both the above methods as well as IP addresses. For Internet users, who have a keyboard available, textual names are preferred since they are easy to remember and deduce.

However, the problem arises when the networks are interconnected. Callers on the PSTN have no keyboard and a scheme for entering characters using number keypad would be too complicated. This limits the PSTN users to entering numeric names. Consequently, an IP terminal must have a telephone number to be accessible from the PSTN. The problem was recognized by TIPHON, which chose to equip IP-terminals with an E.164 number. For calls within the IP network, other types of addressing can be used. Unfortunately, this would require the user to know on what type of network the destination is. When IP telephony is largely deployed, customers do not necessary even know the underlying technology of their own connection.

As we saw in section 2.1, E.164 numbers can currently only be used between host registered to the same signaling server. Using some propriety protocol, mapping can be distributed between smaller groups of servers, but there is no protocol for global distribution.

3.2 Problem categories

The name entered by the user, usually given as an E.164 number, must be mapped to at least one routing address. For calls from IP telephony terminals to other IP telephony terminals, the host address of the destination must be found. For calls over the network boundary to the PSTN, a gateway must be located. Also in the opposite direction, from the PSTN to the IP network, a gateway must be selected.

The TRIP framework [1] divides the problem into three subproblems:

1. Given a phone number corresponding to a specific host on the IP network, determine the IP address of the host. This is required for calls from the PSTN to the IP network, but also for calls within the IP network if E.164 numbers are used. The mapping may be variable, for example if DHCP is used.

2. Given a phone number corresponding to a terminal on the PSTN, determine the IP address of a gateway capable of completing calls to that phone. The choice is influenced by a number of factors, such as policies, location, availability and features. This is called the gateway location problem.
3. Given a phone number corresponding to a user of a terminal on the PSTN, determine the IP address of an IP terminal owned by the same user. This type of mapping may be used if the PC services as an interface for the phone, for example for delivering a message to the PC when the phone rings.

For calls from the PSTN to the IP network, the selection of gateway is performed using normal routing in the switched circuit network, which is static. On longer sight, it would also be necessary to dynamically select a gateway for these calls. This gives us a fourth subproblem.

3.3 The address mapping problem

To establish a call to a terminal on an IP network, the destination IP address must be known. Alternatively the terminal can be identified by a host name, which is translated to an IP address by DNS. As terminals are equipped with an E.164 number, a new mapping is required: from an E.164 name to an IP address. The address mapping problem usually refers to the task of locating terminals on the IP network.

When the switched circuit network and IP telephony networks are interconnected, new call scenarios arise. Since the originating network and destination network can be of two types, there are four basic call scenarios: PSTN-PSTN, PSTN-IP, IP-PSTN and IP-IP. When calls are setup, the first task is to determine the type of the destination network. A mapping from E.164 name to network type is required.

The required mappings could be solved with some type of directory. At a minimum, the mapping from E.164 number to network type and IP address must be supported. The directory must be scaleable too store large amounts of mappings, possibly for all telephones in the world. It must be capable to reply to a high rate of lookups, for each call that is set up. In practice, the directory must therefore be distributed. The directory must also propagate updates rather quickly when the information changes.

Additionally the mapping is expected to be used with several different services. In addition to voice calls, the IP network allows for video conferencing and e-mail among others. Some method of locating the available contact modes and services is desired.

3.4 Routing and number portability

For economic or quality related reasons a transit network of different type can be used, giving two more call scenarios: PSTN-IP-PSTN and IP-PSTN-IP. Even when only two network types are used, the transit network must be selected. It is usually more cost effective to hand over calls to IP destinations to the IP network near the origination point. On the other hand, the voice quality is better if the call uses PSTN most of the path. Possibly the caller could choose whether to route the calls via IP or PSTN using carrier selection mechanisms. Typically this would imply the use of a prefix to select carrier. [23]

The call can thus propagate through several network types. Each time the call goes from one network type to another, it has to pass a gateway where the media stream is converted. The conversions cause delay and jitter, which decrease the quality. Therefore, unnecessary media conversions should be avoided. It would be good to know the type of the destination network already in the originating network.

With number portability numbers may move from one provider's network to another, and even between network types. If a number belonging to a number block of a PSTN operator moves to an IP network, calls from IP subscribers may unnecessarily be routed through the PSTN.

3.5 The gateway location problem

As the usage of IP telephony grows and the number of gateways increases, the management of gateways and routes between the IP- and PSTN networks becomes increasingly complex. In a situation where the IP network approaches the size of the PSTN, a large part of the calls will pass through one or even several gateways on their path. For calls from the IP network to the PSTN, the caller must locate a gateway that is able to complete calls to the desired destination. There may be several available gateways, and selecting the most suitable one is a nontrivial process.

Currently the gateway must be selected by the user or by the signaling servers. The selection and configuration of gateways to use involves manual work. The list of available gateways must be configured into the signaling servers and updated when new gateways become available. Additionally, gateways may become blocked when all lines are in use. The signaling server does not know which gateways are accessible.

Connectivity to the PSTN means that every gateway is able to connect to nearly any terminal on the PSTN. The number of available gateways can thus be very large. The selection of which gateway to use is influenced by a number of factors. Firstly, the location of the gateway is important. For example, there is no

reason to use a gateway in a country far away to connect parties in the same city. To minimize usage of resources it is important that the gateway is near the path between the parties.

Secondly, business relationships are important. The gateway service involves costs when calls are completed to PSTN destination. Gateway providers, in most cases, want to charge for using their gateways. Because of this, the usage of gateways may be restricted to the groups of users that have some type of established relationship with the gateway provider. The end user will probably not pay for the gateway service directly. Instead, the end user may have a relationship with an IP telephony service provider (ITSP). The ITSP may have own gateways or use the gateways of a separate gateway provider. All these policies and relationships influence in the selection of gateway.

Additionally, the end user may have requirements on the gateway. The end user may prefer a certain provider or require a specific feature. The caller may use a specific signaling protocol or media codec that is supported by only some gateways.

Keeping in mind that also the gateway capacity is limited, it is obvious that an automatic method for gateway selection is required. Since the selection is largely driven by policies, some type of global directory of gateways is not suitable. Instead, a protocol for exchanging gateway information between the providers would be a better solution.

4 Telephony Routing over IP

To solve the gateway selection problem, the Internet Engineering Task Force (IETF) working group IP Telephony (IPTEL) began working on a protocol for distributing gateway information between gateway providers and IP telephony providers. The protocol was first called Gateway Location Protocol (GLP) but after finding the problem larger than merely locating gateways, the protocol was renamed to Telephony Routing over IP (TRIP). The most important documents of the work are the TRIP framework [6] and the draft protocol specification [9].

The working group found that a global directory for gateway information is not feasible. The selection of gateway is in large part driven by the policies of the parties along the path of the call. Gateway information is exchanged between the providers and depending on policies, made available locally and propagated to other providers. The providers create their own databases of reachable phone numbers and the routes towards them. These databases can be different for each provider.

TRIP is modeled after the Border Gateway Protocol 4 (BGP-4) [10]. TRIP is like BGP-4 an inter-domain routing protocol driven by policies. The nodes of TRIP are the location servers (LS), which exchange information with other location servers. The information includes reachability information about telephony destination, the routes towards these destinations and properties of the gateways connecting the PSTN and IP network.

TRIP uses the concept of Internet Telephony Administrative Domains (ITAD) in a similar way as BGP-4 uses autonomous systems. The location servers that are administered by a single provider form an ITAD. The ITAD may contain zero or more gateways. The border of the ITAD does not have to correspond to the border of an autonomous system. The main function of TRIP is to distribute information between ITADs, but TRIP also contains functions for inter-domain synchronization of routing information. It is not required that all ITADs in the world are connected. Groups of ITADs can be formed that exchange information with TRIP.

TRIP connects location servers with administratively created peer relationships. The location server forwards the information received from one peer to the other peers. Hereby the location servers in one ITAD learn about gateways in the other ITADs. The location server selects the routes to use in its own domain, and the routes to forward to neighboring domain according to its local policies. The information can be modified according to the policies before it is forwarded. In this way, the provider can control the type of calls passing through the domain.

The location servers collect information and use it to reply to queries about routes to destinations. The query protocol is not defined by TRIP. Any directory access protocol can be used, for example LDAP [11].

4.1 Operation of TRIP

The TRIP protocol, the structure and operation of a node, and the implementation details are specified in the TRIP specification draft [9].

TRIP location servers process three types of routes:

1. External routes received from external peers.
2. Internal routes received from another location server in the same ITAD.
3. Local routes which are locally configured or received from another routing protocol.

The routes are stored in the Telephony Routing Information Base (TRIB), whose structure is depicted in Figure 1. The TRIB consists of four distinct parts.

1. The Adj-TRIBs-In store routing information that has been learned from other peers. These routes are

the unprocessed routed that are given as input to the decision process. Routes learned from internal location servers and from external location servers are stored in separate Adj-TRIBs-In.
2. The Ext-TRIB stores the preferred route to each destination, as selected by the route selection algorithm.
3. The Loc-TRIB contains the routes selected by applying the local policies to the routes in the internal peers' Adj-TRIBs-In and Ext-TRIB.
4. The Adj-TRIBs-Out store the routes selected for advertisement to external peers.

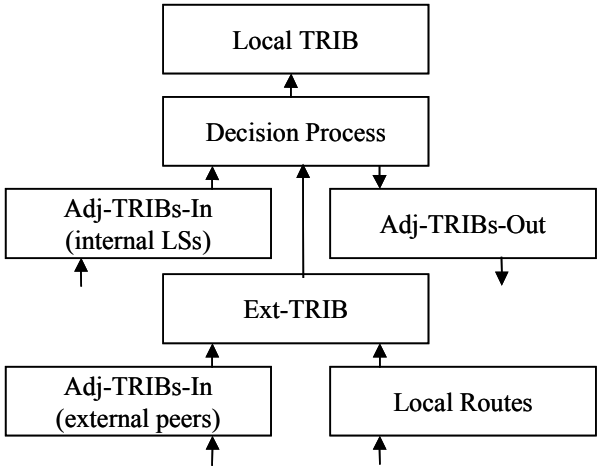


Figure 1: Structure of a TRIP node

TRIP uses the same state machine and the same messages as BGP-4. The messages are the OPEN message for establishing peer connection and exchanging capability information, the UPDATE message for exchanging route information, the NOTIFICATION message for informing about error conditions, and finally the KEEPALIVE message for ensuring that the peer node is running.

The routing information is transmitted in attributes of the TRIP messages. The specification includes a set of mandatory well-known attributes. In addition to the well-known and mandatory attributes, optional attributes can be added to allow for expansion. Gateways have many properties that may need to be advertised, so the expected large number of expansion attributes must be handled correctly. An attribute flag indicates how a location server handles a message that it does not recognize. The flag can take a combination of the values optional, transitive, dependent, partial and link-state encapsulated.

The specification [9] defines the basic set of attributes shown in Table 1. Additional attributes are defined in separate drafts. An authentication attribute is defined in [12] and a service code attribute is defined in [13].

Table 1: The basic set of TRIP attributes

Name	Description
Withdrawn routes	List of telephone numbers that are no longer available.
Reachable routes	List of reachable telephone numbers.
Next hop server	The next signaling server on the path towards the destination.
Advertisement path	The path that the route advertisement has traveled.
Routed path	The path that the signaling messages will travel.
Atomic aggregate	Indicates that the signaling may traverse ITADs not listed in the routed path attribute.
Local preference	The intra-domain preference of the location server.
Multi exit disc	The inter-domain preference of the route if several links are used.
Communities	For grouping destinations in groups with similar properties.
ITAD topology	For advertising the ITAD topology to other servers in the same ITAD.
Authentication	Authentication of selected attributes.

The advertisements represent routes toward a gateway through a number of signaling servers. A route must at least contain the following attributes: withdrawn routes, reachable routes, next hop server, advertisement path and routed path. For an advertised route, the withdrawn routes attribute is empty. The reachable routes attribute contains the list of telephone number ranges belonging to this route, and the corresponding application protocol. The next hop server is the next server that signaling messages are sent to. For the final hop, it contains the address of the gateway. The advertisement path is the path that this advertisement has traveled through and the routed path is the path for the signaling. These paths are lists of ITADs. They are mainly used by the policy to select routes containing, or not containing specific ITADs.

4.2 TRIP for gateways

The TRIP framework [9] leaves the question open, how the location servers learn about the gateways. Usually the register message of SIP has been suggested. However, the draft [14] points out the weaknesses of using the register message and suggests that a subset of TRIP could be used to export routing information from gateways and soft switches to location servers. TRIP manages the needed information transfer and keep-alives more efficiently than other protocols and can better describe the gateway properties. Two new attributes are proposed: circuit capacity for informing about the number of free PSTN circuits, and DSP capacity for informing about the amount of available DSP resources. Because of their dynamic nature, these

are only transmitted to the location server that manages the gateway, and are not propagated.

A more lightweight version of TRIP can be used in the gateways. Since the gateway does not need to learn about other gateways, it operates in send-only mode. It neither needs to create any call routing databases. This stripped down version, called TRIP-GW, is still interoperable with normal TRIP nodes. Nevertheless, due to scalability problems it is recommended that location servers peering with gateways run a separate TRIP instance for TRIP-GW peers.

5 Telephone Number Mapping

While TRIP is carrying routes to destinations on the PSTN, a method for locating terminals on the IP network is still required. This problem is simpler than the gateway location problem, since the amount of information describing a terminal is less than the information about a gateway. TRIP could be used also for this purpose, but the complexity of it is not needed. A simpler directory can be used. It has been suggested that Domain Name System (DNS) [15], [16] could be used. An IETF working group called ENUM (tElephone NUmber Mapping) was established to specify the number mapping procedures.

DNS is used to map domain names into IP addresses. By constructing a domain name from the E.164 number, the DNS system can be used to map telephone numbers into IP addresses. More generally, the result of an ENUM lookup is a Uniform Resource Identifier (URI) [18], which contains the signaling protocol and the host name. An additional DNS lookup is thus required to map the host name to an IP address. The procedure is described in RFC 2916 [17], the main document specifying the ENUM service.

ENUM uses the domain "e164.arpa" to store the mapping. Numbers are converted to domain names using the scheme defined in [17]. The E.164 number must be in its full form, including the country code. All characters and symbols are removed, only the digits remain. Dots are put between the digits. The order of the digits is reversed and the string ".e164.arpa" is added to the end. This procedure will map, for example, the number +358-9-4515303 into the host name "3.0.3.5.1.5.4.9.8.5.3.e164.arpa".

DNS stores information in different types of records. The Naming Authority Pointer (NAPTR) record [19] is used for identifying available ways to contact a node with a given name. It can also be used to identify what services exist for a specific domain name. The fields of the NAPTR record are shown in Table 2. ENUM defines a new service named "E.164 to URI", which maps one E.164 number to a list of URIs. The mnemonic of the service is "E2U". ENUM can be used in conjunction with several application protocols, and can for example, map a telephone number to an email address.

Table 2: Fields of the NAPTR record

Name	Description
Order	The order in which records are processed if a response includes several records.
Preference	The order in which records are processed if the records have the same order value.
Service	The resolution protocol and resolution service that will be available if the rewrite of the regexp or replacement field is applied.
Flags	Modifiers for how the next DNS lookup is performed.
Regexp	Used for the rewrite rules.
Replacement	Used for the rewrite rules.

Figure 2 shows some example NAPTR records with the E2U service. These records describe a telephone number that is preferably contacted by SIP and secondly by either SMTP or using the "tel" URI scheme [20]. The result of the rewrite of the NAPTR record is a URL, as indicated by the "u" flag. The own resolution methods of SIP and SMTP are used. In case of SIP, the result is a SIP URI, which is resolved as described in [1]. In case of the "tel" scheme, the procedure is restarted with a new E.164 number.

```
$ORIGIN 3.0.3.5.1.5.4.9.8.5.3.e164.arpa.
  IN NAPTR  10   10  "u"  "sip+E2U"
     "!^.*$!sip:nbeijar@tct.hut.fi!"    .
  IN NAPTR  100  10  "u"  "mailto+E2U"
     "!^.*$!mailto:nbeijar@tct.hut.fi!" .
  IN NAPTR  100  10  "u"  "tel+E2U"
     "!^.*$!tel:+35894515303!"     .
```

Figure 2: Example NAPTR records

The draft [21] describes a telephone number directory service based on ENUM. The model is divided into four levels.

The first level is a mapping of the telephone number delegation tree into authorities, to which the number has been delegated. The hierarchical structure of DNS is used, and the mapping may involve one or several DNS queries, which are transparent from the user's point of view. The delegation maps the hierarchy of the

E.164 number to the DNS hierarchy, using the country codes, area codes and other parts of the number. The first level mapping uses name server (NS) resource records in DNS.

The second level is the delegation from the authority, to which the number has been delegated, to the service registrar. The registrar maintains the set of service records for a given telephone number. Since there may be several service providers for a given number, the registrar has the role to manage service registrations and arbitrate conflicts between service providers. The second level uses the DNAME and CNAME records of DNS to provide redirection from the designated authority to the service registrar. The delegated authority and the service registrar can be the same entity, which is anticipated especially in the early stages of ENUM deployment.

The third level is the set of service records, which indicate what services are available for a specific telephone number. There can be multiple records for the same service, indicating competitive or redundant service providers. The NAPTR type of records is used at the third level. The response to a client's query is a set of NAPTR records, and the client is responsible for selecting the service to use for the intended action. A URI is obtained by rewriting the query using the rewrite rule. The URI can be an LDAP directory server, a H.323 gatekeeper, a SIP signaling server or a specific end point address.

Finally, a fourth level can be provided if necessary. This level provides specific attributes for the services that are only known by the provider of the service. Such attributes can be needed for placing calls, routing messages or validating capabilities. The attributes can be obtained through a SIP query to a signaling server or a LDAP query to a directory server. The level is service specific and dynamic, and should therefore be possible with minimal coordination between the directories of competing providers.

6 Scenarios

In this section, some scenarios based on ENUM and TRIP are presented. First the different types of resource records used by ENUM are presented through an example. Then two call setup situations are analyzed. The draft [22] describes how ENUM can be used in different call setup situations where interworking between the PSTN and IP-based networks is necessary. By additionally using the TRIP framework [4] and the ENUM model [17, 21] , we construct examples of how the protocols are used. Since any final call setup procedure is not defined, these examples only represent one possible approach for interworking between the networks.

6.1 Call setup using ENUM

To illustrate the use of ENUM, we will study a call setup situation, where the DNS records of Figure 3 are used. The figure shows the DNS configuration for the top level delegations, the national delegations, a service provider and a service registrar.

Sample top level delegations from ITU:

```
3.3.e164.arpa    IN NS ns.FR.phone.net.  ;France
8.5.3.e164.arpa  IN NS ns.FI.phone.net.  ;Finland
```

Sample national delegations:

```
5.4.9.8.5.3.e164.arpa.   IN NS
ns.ServiceProviderX.net.
```

Sample service provider's configuration:

```
1.5.4.9.8.5.2.e164.arpa. DNAME
1.5.4.9.8.5.2.ns.hut.fi.
```

Sample service registrar configuration:

```
*.1.5.4.9.8.5.2.ns.hut.fi.
    IN NAPTR 100 10 "u" "ldap+E2U"\
    "$!ldap://ldap.hut.fi/cn=\1!" .
```

Figure 3: Configuration of DNS records

The described service enables an end-user to discover the various methods by which the recipient can be reached. The service is hosted by the recipient's corporation.

When a call is setup to the telephone number +35894515303, the number is first translated into the domain name "3.0.3.5.1.5.4.9.8.5.3.e164.arpa" according to the ENUM rules. Using the NS records in the top-level and national authorities' databases, the service provider is located. In this example, the number block +3589451xxxx is delegated to the service provider. The service provider provides a non-terminal redirection pointer to the corporation, which is the service registrar for the number block +35894515xxx. The query for the reachme service returns the NAPTR record. The client then applies the regular expression and gets an LDAP URI of "LDAP://ldap.hut.fi/cn=35894515303". The client uses LDAP with the reachme schema to determine the available communications methods.

6.2 Calls from PSTN to IP-based network

The call setup scenario for a call from PSTN to an IP based network is depicted in Figure 4. The originating customer, who resides on the PSTN, dials an E.164 number. The PSTN operator forwards the call to an appropriate gateway. The selection of gateway depends on several factors. This is a gateway location problem

similar to that on the IP network, but there are currently no corresponding solutions like TRIP. The draft [22] leaves the question open.

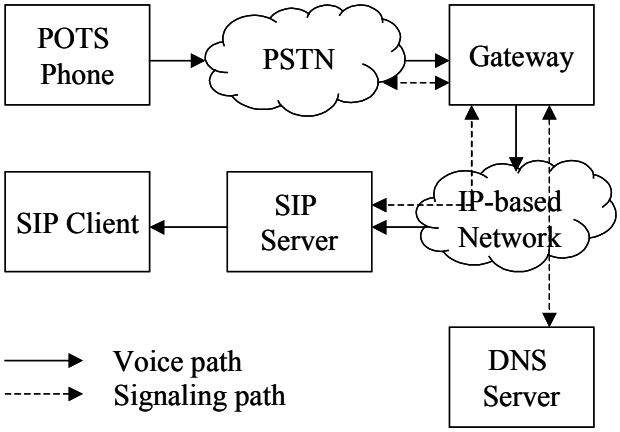


Figure 4: Call from PSTN to IP-based network

The gateway, which contains ENUM functionality, looks up the number in DNS. The dialed number is mapped into an URI. If necessary, the country and area codes are added to the number by the gateway. The DNS returns any service records that are associated with the URL. The record may be a SIP URI such as "sip:nbeijar@sipserver.hut.fi". The gateway makes a DNS query for the host, in this case "sipserver.hut.fi" to get the IP-address of the signaling server. The SIP call can then be established to the user agent of the given user.

6.3 Calls from IP-based network to PSTN

A call setup scenario for a call from an IP-based network to the PSTN is illustrated in Figure 5. The originating customer dials an E.164 number. Since a customer may dial a local number or a national number, the client must be capable of supply any missing digits. Here the caller uses a SIP client, but any other signaling protocol can be used. The client must contain ENUM functionality. A DNS request is constructed from the dialed digits according to the ENUM specification.

When the client looks up the name in DNS, the DNS returns any NAPTR service records associated with the URL. Since the destination is on the PSTN, the query only returns one record containing the URI in the "tel" format. For example the URI "tel:+35894515303" might be returned. The SIP client initiates an INVITE to the SIP server using the given URI. The SIP server queries a location server with LDAP or any other front end protocol suggested in the TRIP framework [4]. The location server has learned about available gateways using TRIP. The location server returns the IP address of a suitable gateway and the call is routed to the gateway by the SIP server. The gateway then completes the call through the PSTN.

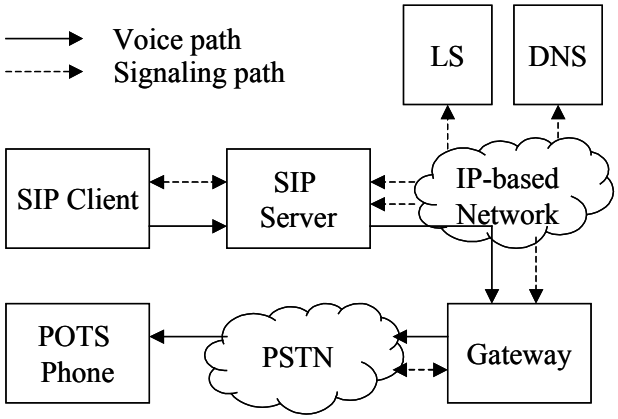


Figure 5: Call from IP-based network to PSTN

7 Solutions for number portability

Number portability requires a mapping between name and address. Generally numbers can be moved by changing the mapping. The described protocols TRIP and ENUM both provide a mapping between name (telephone number) and address (URI, next step signaling server or gateway).

7.1 ENUM and number portability

ENUM provides a solution for number portability for numbers moving within the IP network and to the PSTN. This allows users to change IP service providers without having to change their telephone number [22]. The directory service solution defined in [21] describes number portability on three of the conceptual levels of ENUM.

If the number is delegated to another authority, the corresponding update is performed in ENUM by changing the name server records to point to the new authority. The number is thus moved to another service provider or to a portability authority.

The service registrar can be reassigned, for example if the customer wants to also change the service registrar in conjunction with the change of service provider. The DNAME or CNAME records are then updated to point to the new service registrar. New service specific NAPTR records are created by the new service registrar.

Most frequently the movement of a number can be accomplished by changing the NAPTR records. This happens when a specific service is moved from one provider to another, for example when switching telephony providers. The service registrar coordinates the deletion of the old records and insertion of records for the new service provider.

7.2 Interworking and number portability

ENUM also solves number portability for hybrid PSTN-IP networks. The draft [22] separates three scenarios:

1. The number moves within the PSTN.
2. The number moves between PSTN and IP.
3. The number moves within the IP network.

For each scenario, the call setup procedure from both PSTN and the IP network is described.

For calls originating from the PSTN, the first scenario is already handled by today's number portability. The second scenario is solved by changing the number portability mapping to direct the call to a gateway. The third scenario is solved by ENUM as was described.

For calls established on the IP network, the first scenario may lead to inefficient routing. As the number moves, within the PSTN, the most suitable gateway probably changes. As a result, the DNS information must be updated. It is still not defined how this is done, and how it can be automated. Alternatively, if the DNS contains routing addresses (such as LRN) for PSTN destinations, these must be updated to point to the new operator and new gateway. Otherwise calls may be routed to the wrong operator. If the gateway do not have routing addresses available, an IN query must be performed by the gateway or at a later stage.

In scenario 2 for IP originated calls, it would be enough to update the type of URI returned by DNS. A "tel:"-based URI would be replaced by an URI for a SIP or H.323 terminal, or vice versa. Also the third scenario is solved by updating ENUM information.

The question about whether to store routing or directory numbers of PSTN terminals in DNS has been discussed in IETF working groups. It is also unclear how to know which terminals reside on the PSTN. In current plans, mainly mappings for IP terminals are stored in DNS. It is assumed that calls to unknown E.164 numbers are routed to the PSTN. This may create unnecessary traffic and gateway blocking due to wrongly dialed numbers.

7.3 CTRIP

As we saw in the two first scenarios, both the mappings in ENUM and in the IN databases on the PSTN must be updated in some cases. The question how the update is coordinated and how the information is transferred is still unresolved. Moreover, the information of TRIP must be updated in some cases. There is still no solution how to coordinate information in ENUM, TRIP and the IN databases.

It seems to be necessary to automate the distribution of numbering information between the network types and between the protocols. To solve the problem, a counterpart to TRIP is being developed in the Networking Laboratory at Helsinki University of Technology. The suggested protocol, named Circuit Telephony Routing Information Protocol (CTRIP), automates the distribution of routing information between operators and network elements. Information is exchanged with other protocols in Numbering Gateways. [24]

8 Conclusions

Although the signaling protocols provide basic mechanisms for locating terminals and gateways, new protocols are required for distributing routing information in order to make a global IP based telephone network possible. TRIP provides a solution for the gateway location problem by distributing information about gateways and reachable PSTN destinations between location servers. ENUM defines a directory of name to address mappings, which is used to locate terminals on the IP network. Both are based on tried solutions: TRIP is based on BGP-4 and ENUM uses the existing DNS system.

Number portability is generally implemented by modifying the mapping between name and address. In the PSTN the Intelligent Network implements the mapping functions. On the IP network the mappings of ENUM and TRIP can be modifying to realize number portability.

When the two network technologies, PSTN and IP, are interconnected new problems arise. The information in ENUM, TRIP and IN must be kept updated to avoid wrongly or inefficiently routed calls. Currently the update is performed manually, and the process is uncoordinated between service providers. This becomes a burden, especially when number portability causes increased update frequencies. Also the risk of wrong and incompatible information is high. An automated approach for synchronizing information between the protocols is needed.

The protocols are still under development. The basic ENUM specification has reached RFC standards track stage but TRIP is still an Internet draft. Commercially available implementations are not available. The protocols' suitability for IP telephony in real networks is still not verified.

Yet, the need for standardized protocols for distributing IP telephony routes is high and the future for TRIP and ENUM seems promising. The signaling protocols alone cannot be used to form a global IP-based network, and the described protocols provide the required solution. However, as we have seen some new parts are required to make the architecture complete.

References

[1] Handley, M. Schulzrinne, H., Schooler, E., Rosenberg J.: SIP: Session Initiation Protocol, March 1999, IETF RFC 2543

[2] International Telecommunications Union Telecommunication Standardization Sector, Study group 16: Packet-based multimedia communications systems, February 1998, ITU-T Recommendation H.323

[3] International Telecommunications Union Telecommunication Standardization Sector: The international public telecommunication numbering plan, Geneva, May 1997, ITU-T Recommendation E.164

[4] European Telecommunications Standards Institute: The Procedure for Determining IP Addresses for Routeing Packets on Interconnected IP Networks that support Public Telephony, DTR 4006, 2000

[5] Postel, Jonathan: Simple Mail Transfer Protocol, August 1982, IETF RFC 821

[6] Rosenberg, J., Schulzrinne, H.: A Framework for Telephony Routing over IP, June 2000, IETF RFC 2871

[7] Mensola, Sami: IP-verkon kommunikaatio-palveluiden hallinta, November 1998, Master's Thesis

[8] Foster, Mark, McGarry, Tom, Yu, James: Number Portability in the GSTN: An Overview, March 2000, draft-foster-e164-gstn-np-00.txt

[9] Rosenberg, J., Salama, H., Squire, M.: Telephony Routing over IP (TRIP), November 2000, draft-ietf-iptel-trip-04.txt

[10] Rekhter, Y., Li, T.: Border Gateway Protocol 4 (BGP-4), March 1995, IETF RFC 1771

[11] Yeong, W., Howes, T., Kille, S.: Lightweight Directory Access Protocol, March 1995, IETF RFC 1777

[12] Rosenberg, J., Salama, H.: Authentication Attribute for TRIP, December 2000, draft-ietf-iptel-trip-authen-00.txt

[13] Peterson, J.: The ServiceCode Attribute for TRIP, November 2000, draft-jfp-trip-servicecodes-00.txt

[14] Rosenberg, J., Salama, H.: Usage of TRIP in Gateways for Exporting Phone Routes, July 2000, deaft-rs-trip-gw-01.txt

[15] Mockapetris, P.: Domain names – concepts and facilities, November 1987, IETF RFC 1034

[16] Mockapetris, P.: Domain names – implementation and specification, November 1987, IETF RFC 1035

[17] Faltstrom, P.: E.164 number and DNS, September 2000, IETF RFC 2916

[18] Berners-Lee, T., Fielding, R.T., Masinter, L.: Uniform Resource Identifiers (URI): Generic Syntax, August 1998, IETF RFC 2396

[19] Mealling, M., Daniel, R.: The Naming Authority Pointer (NAPTR) DNS Resource Record, September 2000, IETF RFC 2915

[20] Vaha-Sipila, A.: URLs for Telephone Calls, April 2000, IETF RFC 2806

[21] Brown, A.: ENUM Service Provisioning: Principles of Operation, October 2000, draft-ietf-enum-operation-01.txt

[22] Lind, S.: ENUM Call Flows for VoIP Interworking, November 2000, draft-line-enum-callflows-01.txt

[23] Rosbotham, Paul: WG4 FAQ, TIPHON temporary document (discussion)

[24] Raimo Kantola, Jose Costa Requena, Nicklas Beijar: Interoperable routing for IN and IP telephony, Computer Networks, Volume 35, Issue 5, April 2001