GMD Report
95

GMD –
Forschungszentrum
Informationstechnik
GmbH

Berthold Butscher, Georg Carle,
Jiri Kuthan, Mikhail Smirnov,
Heinrich J. Stüttgen, Lars Wolf (Eds.)

Proceedings of the 1st IP-Telephony Workshop (IPTel 2000)

April 12-13, 2000 in Berlin, Germany

GI/ITG

April 2000

GMD – Forschungszentrum Informationstechnik GmbH
Schloß Birlinghoven
D-53754 Sankt Augustin
Telefon +49 -2241 -14 -0
Telefax +49 -2241 -14 -2618
http://www.gmd.de

Anschriften der Herausgeber/Addresses of the editors:
Berthold Butscher
Dr. Georg Carle
Jiri Kuthan
Prof. Dr. Mikhail Smirnov
Institut für Offene Kommunikationssysteme
GMD – Forschungszentrum Informationstechnik GmbH
Kaiserin-Augusta-Allee 31
D-10589 Berlin
E-mail: {butscher, carle, kuthan, smirnov}@fokus.gmd.de

Dr. Heinrich J. Stüttgen
C&C Research Laboratories Heidelberg
NEC Europe Ltd
Adenauerplatz 6
D-69115 Heidelberg
E-mail: stuttgen@ccrle.nec.de

Prof. Dr. Lars Wolf
Universität Karlsruhe
Computing Center
Zirkel 2
D-76128 Karlsruhe
E-mail: Lars.Wolf@rz.uni-karlsruhe.de

ISSN 1435-2702

Preface

**The 1st IP-Telephony Workshop (April 12th-13th, 2000 in Berlin),
SIG on Communication and Distributed Systems (KuVS) of German Informatics Society
(GI) and German Society for Technical Informatics (ITG**

The Internet telephony has been gaining momentum in the last years. Lots of research and development efforts have been aiming at provisioning voice services over the Internet. The Internet telephony is expected to enable a new generation of telecommunication services and to reduce costs by having a single infrastructure for both data and voice. The objective of the First IP Telephony Workshop is to bring together researchers, developers, vendors and service providers working in the IP telephony area to participate actively in a discussion on recent deployment experiences, innovative results and future directions.

Submissions concerning with almost any aspect of the IP telephony are reflected in these Proceedings:

1. Basic Technologies: Signalling protocols (SIP, H.323, ...), Gateways, IP-Telephony PBXs, Inter-Gateway-Communication, Audio-/Video-Encoding for the Internet telephony, QoS and the Internet Telephony, Software-Architectures for the IP Telephony;
2. IP-Telephony Services: IP-Telephony Applications and Added-value Services, Look-up Services for the IP-Telephony, Internetworking between the IP Telephony and Intelligent Networks, Service Composition, Configuration and Provisioning, Security;
3. Business Deployment: Business Models, IP Telephony Charging; Deployment reports on Reliability, Acceptance, Performance/QoS, etc.

Keywords:

IP Telephony, Voice over IP, Signaling, SIP, H.323, QoS, Multimedia, Conferecing, Media Gateways

Organised by:

GMD FOKUS, Berlin

NEC CCRL, Heidelberg

University of Karlsruhe

Supported by:

Deutsche Telekom T-Nova, Berlin

Vorwort

1. IP-Telefonie Workshop (12. -13. April 2000 in Berlin),
Eine Veranstaltung unter Beteiligung der Fachgruppe Kommunikation und Verteilte Systeme der Gesellschaft für Informatik (GI) und der Informationstechnischen Gesellschaft im VDE (ITG)

In den letzten Jahren hat das Thema 'IP-Telefonie' bereits große Beachtung gefunden. Viele Arbeiten, im Forschungs- als auch im Entwicklungsbereich, zielen darauf ab, Unterstützung für die Sprachkommunikation über das Internet anzubieten.
Durch derartige IP-Telefonie-Techniken sollen sowohl Kosteneinsparungen als auch neue Dienste möglich werden. Noch besteht aber ein Mangel an Erfahrungen mit dem Einsatz dieser Verfahren. Außerdem sind weitere Forschungs- und Entwicklungsarbeiten notwendig, um leistungsfähige Dienste und eine geeignete Infrastruktur bereitstellen zu können. Im Rahmen dieses Workshops sollen der Stand der Arbeiten und bisherige Erfahrungen mit IP-Telefonie diskutiert werden.
Beiträgen, die sich mit allen Aspekten der IP-Telefonie beschäftigen können sind abgespiegelt hierebei:

1. Basis-Technologie: Signalisierungsprotokolle (H.323, SIP,...), Gateways, IP-Telefonie-Anlagen, Inter-Gateway-Kommunikation, Audio-/Video-Kodierung fuer IP-Telefonie, Dienstqualitätsunterstützung für Internet-Telefonie, oftware-Architekturen für IP-Telefonie;
2. P-Telefonie-Dienste: Anwendungen und Mehrwertdienste basierend auf IP-Telefonie, Verzeichnisdienste für IP-Telefonie, Interworking von IP-Telefonie und Intelligenten Netzen, Komposition und Erbringung von IP-Telefoniediensten, Sicherheitsaspekte;
3. Business-Aspekte: Business-Modelle, Tarifierung von Internet-Telefonie, Erfahrungen beim praktischen Einsatz von IP-Telefonie, Zuverlässigkeit, Akzeptanz, Performance/QoS.

Schlüsselwörter:

IP Telefonie, Paketorientierte Sprachübertragung, Signalisierung, SIP, H.323, QoS, Multimedia, Konferenzdienste, Media Gateways

Organisation des Workshops

GMD FOKUS, Berlin

NEC CCRL, Heidelberg

University of Karlsruhe

Unterstützt von:

Deutsche Telekom T-Nova, Berlin

Programme Committee:

W. Bauerfeld (T-Nova), F. Brockners (Cisco Systems GmbH), G. Carle (GMD FOKUS),
J. Glasmann (TU Muenchen), O. Haase (NEC CCRLE), B. Ip (Siemens),
J. Kuthan (GMD FOKUS), T. Magedanz (IKV++ GmbH), P. Noll (TU Berlin),
J. Ott (Uni Bremen), R. Reinema(GMD TKT), H. Sanneck (GMD FOKUS),
H. Schulzrinne (Columbia Uni., USA), H. Sinnreich (MCI WorldCom, USA),
M. Smirnow (GMD FOKUS), R. Steinmetz (TU Darmstadt), H. Stuettgen (NEC CCRLE),
H. Wermescher (Infonova GmbH, Austria), R. Wittmann (TU Braunschweig),
L. Wolf (Uni Karlsruhe), A. Wolisz (TU Berlin)

Executive Committee:

General Chair:	Berthold Butscher	GMD FOKUS
Technical Program Co-Chairs:	Heinrich Stüttgen	NEC CCRLE
	Michael Smirnow	GMD FOKUS
	Lars Wolf	Uni Karlsruhe
	Georg Carle	GMD FOKUS
Local Arrangements Chair:	Jiri Kuthan	GMD FOKUS

Table of Contents

Day 1: April 12th

Day 2: April 13th

Quality of Service and Enhanced Services

Architecture and Implementations

Demonstrations

Invited Talk

Henning G. Schulzrinne

Assoc. Professor, Dept. of Computer Science & Dept. of Electrical Engineering
Columbia University, New York, NY

Internet Telephony: A Second Chance

Henning Schulzrinne, Columbia University

It is rare that one gets to re-engineer a fundamental part of the communications infrastructure that has existed for more than a hundred years. We now have this once-in-a-lifetime chance, but must resist the temptation to simply recreate the same old network, just using packets instead of circuits. Below, we give some examples of how to emphasize *Internet* telephony rather than Internet *telephony*. We also describe our perception where the open issues are and possible approaches.

First, this design principle implies that we should restrict any PSTN-specific features and assumptions limited to gateways and translation software, rather than making Internet devices aware of these legacy technologies. Such legacies include E.164 telephone numbers, voice-only orientation, the use of voice prompts or messages and in-band signaling such as DTMF.

Where sensible, services inspired by the PSTN should be made to work across all Internet services, not just telephony. For example, emergency call services ("911" in the US, 110/112 in Germany) should work the same way whether invoked from a chat tool, email or an Internet application. Locating gateways, as implemented in TRIP [1], [2], may be applicable as a wide-area service location protocol for both electronic and physical services. Also, dynamic carrier selection, available on a per-call basis in the PSTN, needs to be made available for QOS-controlled IP services.

In the long run, it is likely that Internet telephony service will not be a stand-alone offering, but rather be part of a large set of applications that do not look like a telephone at all. Internet phone calls will be initiated from chat applications, distributed games, virtual reality environments, web pages and applets embedded in email. SIP, for example, accommodates this, making it easy for web pages to contain SIP URLs for one-click-dialing and allowing SIP responses to contain web pages or redirect calls to any other URL, such as email, web page or chat. Indeed, it has been suggested that chat and Internet telephony are so closely related that it makes sense to use a single signaling protocol for both [3], [4]. In that model, text chat is just one of many possible session types, including traditional telephony, multi-player games or conferencing.

Much of the complexity of the current PSTN arises from its charging model. The PSTN charging model is neither sufficient nor necessary for the new environment, except when charging for gatewayed calls into the legacy PSTN. Already, advertising-supported phone calls are becoming popular, as the cost per impression of about 0.6 to 6 US cents approaches the cost of providing a minute of domestic service. For higher-quality and video services, bandwidth-based charging, independent of the application, needs to be developed, possibly based on congestion-adaptive pricing models [5] that make it possible to offer affordable high-quality video service at least during off-hours.

Providing assured quality-of-service is probably more of an administrative than a technical problem at this point. Voice service, in particular, is a good candidate for differentiated services [6], as traffic engineering is relatively straightforward. In particular, the aggregated model [7] where RSVP or similar resource reservation protocols provide admission control for traffic classes. Simple prioritization for voice packets works well as long as VoIP is the major QOS-assured traffic class.

Probably the major challenge faced by Internet telephony is moving from the current Internet reliability of about 99% or 99.5% to 99.999%, i.e., no more than five minutes of unavailability per year. This requires a different mindset, not just protocol and technology fixes, as upgrades have to be done while the system is running and every component has to be engineered to have a hot standby. A particular problem for Internet telephony is that it requires a large number of components, including gateways, proxy servers, DNS, DHCP and resource reservation, each subject to independent failures, thus, simplicity and re-use of core infrastructure services is needed.

Many system failures are caused by misconfiguration. Particularly for Internet telephones, devices need to be able to be bought, plugged into an Ethernet socket and then function, without any further manual intervention. Configuration using DHCP [8] or SLP [9] work well in single-provider LANs, but may not be sufficient where a single access infrastructure such as CATV is shared by multiple operators.

In a few years, most telephone devices will be wireless. Next-generation networks such as 3G will push IP closer to the end system, but it would simplify the overall architecture if a single mobility protocol can handle both the cases of discontinuous and continuous mobility [10], [11].

Unified messaging, combining email, fax and voicemail into a single user interface, will be made much easier with the use of IP-based delivery. For example, MGCP [12] or RTSP [13] can be used to generate voice prompts or record messages, then delivered via SMTP and retrieved via IMAP or POP.

Unlike traditional telephony, where services are only available in a few pre-canned varieties, Internet telephony offers the opportunity to have service providers, administrators and end users customize their telephony services. So far, mostly call filtering and routing have been made programmable [14], [15], [16], [17],. but there are opportunities for programming media interactions.

In summary, Internet telephony should be viewed as an opportunity mostly for system integration, not for inventing fundamentally new Internet architectures. It will succeed not by replicating the existing phone network (except in its reliability), but by creating an open platform for experimentation and creation of new services.

REFERENCES

[1] J. Rosenberg and H. Schulzrinne, "A framework for telephony routing over IP," Internet Draft, Internet Engineering Task Force, Nov. 1999. Work in progress.

[2] J. Rosenberg, H. Salama, and M. Squire, "Telephony routing over IP (TRIP)," Internet Draft, Internet Engineering Task Force, Jan. 2000. Work in progress.

[3] C. Vermeulen, "Presence info data format (PIDF)," Internet Draft, Internet Engineering Task Force, Dec. 1999. Work in progress.

[4] J. Rosenberg and H. Schulzrinne, "SIP for presence," Internet Draft, Internet Engineering Task Force, Nov. 1998. Work in progress.

[5] X. Wang and H. Schulzrinne, "RNAP: A resource negotiation and pricing protocol," in *Proc. International Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV)*, (Basking Ridge, New Jersey), pp. 77–93, June 1999.

[6] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss, "An architecture for differentiated service," Request for Comments (Informational) 2475, Internet Engineering Task Force, Dec. 1998.

[7] Y. Bernet, R. Yavatkar, F. Baker, B. Davie, P. Ford, B. Braden, J. Wroclawski, M. Speer, E. Felstaine, and L. Zhang, "A framework for integrated services operation over diffserv networks," Internet Draft, Internet Engineering Task Force, Mar. 2000. Work in progress.

[8] G. Nair and H. Schulzrinne, "DHCP option for SIP servers," Internet Draft, Internet Engineering Task Force, Feb. 2000. Work in progress.

[9] J. Veizades, E. Guttman, C. Perkins, and S. Kaplan, "Service location protocol," Request for Comments (Proposed Standard) 2165, Internet Engineering Task Force, June 1997.

[10] F. Vakil *et al.*, "Mobility management in a SIP environment requirements, functions and issues," Internet Draft, Internet Engineering Task Force, Mar. 2000. Work in progress.

[11] E. Wedlund and H. Schulzrinne, "Mobility support using SIP," in *Second ACM/IEEE International Conference on Wireless and Mobile Multimedia (WoWMoM'99)*, (Seattle, Washington), Aug. 1999.

[12] D. Cromwell, "A syntax for the MGCP audio package," Internet Draft, Internet Engineering Task Force, Nov. 1998. Work in progress.

[13] H. Schulzrinne, A. Rao, and R. Lanphier, "Real time streaming protocol (RTSP)," Request for Comments (Proposed Standard) 2326, Internet Engineering Task Force, Apr. 1998.

[14] B. Pagurek, J. Tang, T. White, and R. Glitho, "Management of advanced services in H.323 internet protocol telephony," in *Proceedings of the Conference on Computer Communications (IEEE Infocom)*, (Tel Aviv, Israel), Mar. 2000.

[15] J. Lennox, J. Rosenberg, and H. Schulzrinne, "Common gateway interface for SIP," Internet Draft, Internet Engineering Task Force, May 1999. Work in progress.

[16] L. Slutsman, G. Ash, F. Haerens, and V. Gurbani, "Framework and requirements for the internet intelligent networks (IIN)," Internet Draft, Internet Engineering Task Force, Mar. 2000. Work in progress.

[17] J. Lennox and H. Schulzrinne, "CPL: a language for user control of internet telephony services," Internet Draft, Internet Engineering Task Force, Mar. 1999. Work in progress.

Keynote 1: IP Telephony - Standardisation and Deployment

Stefan Gessler
(NEC CCRLE, Heidelberg)
Herwart Wermescher
(INFONOVA, Austria)

IPTelephony - Standardisation and Deployment

Stefan Gessler* and Herwart Wermescher†

*NEC Europe Ltd., Germany
Stefan.Gessler@ccrle.nec.de

†INFONOVA GmbH, Austria
herwart.wermescher@infonova.at

For quite a long time, the Internet played a significant role only in the academia or in technology oriented research departments. The development of the World Wide Web can be regarded as the ignition of the first stage of a rocket 'Internet', which brought the Internet in everyone's mind. Now we face the ignition of the second stage, the introduction of telephony services over IP-based networks.

Unusual for the Internet community is the influence of industry and service providers in the exploration phase of this new technology. However, for researchers it is indispensable to consider the requirements from the industry, in particular in the case of the telephony service, where a well-developed and widely established classical technology and infrastructure has to be integrated.

Since this area covers a wide range of aspects, a couple of bodies are concerned with standardisation for IPTelephony issues. The variety of existing bodies and promoted standards, recommendations and agreements has a bewildering effect to an observer from outside.

The goal of this talk is to give an overview of the existing bodies, which are involved in the standardisation process for IPTelephony technology. It addresses their charter and work plan, the structure, participating organisations and companies, and the relationships to other bodies. Particularly the ETSI TIPHON is presented in detail. ETSI TIPHON is essentially a European forum but has a global scope. It probably comprises the largest extent among all IP Telephony bodies.

As introduction a brief summary of IPTelephony standards will also be given.

In the second part an IP-Telephony deployment project TTT-Services (TEN TELECOM TIPHON-Services) is presented. TTT-Services is a project of an industrial consortium with the goal to deploy global IP-Telephony services (e.g. UPT) based on TIPHON specifications by 22.11.00. The project is run in the framework of the TEN-TELECOM programme of the European Commission DG XIII.

Major European telephone carriers as well as manufacturers are currently undertaking actions to enable global, seamless, interoperable IP-Telephony applications for the mutual benefit of the participants.

The closeness of TTT-Services and ETSI Project TIPHON enables two organizations to focus on their individual strengths: TIPHON is best positioned to develop technical specifications required in the delivery of IP telephony services. The charter of TTT-Services is to deploy commercial services based on TIPHON specifications. By combining these strengths, a complementary, integrated approach can be achieved for the global deployment of IP Telephony based services in the business marketplace.

Service Platforms, Tools and Interworking

Programming SIP Services

Anders Kristensen, Anders Byttner, Roman Kurmanowytsch
HP Labs, Bristol
Filton Road, Stoke Gifford
BS34 8QZ Bristol, U.K.
{ak,andbyt,romkur}@hplb.hpl.hp.com

***Abstract* --As the number of communication modalities available to people increase, the ability for service providers and end users to author and provision communications services will become increasingly important. Programmability of Internet Telephony services will arguably need to be more like Web services than traditional telephony service environments. We have proposed a Java API based on the concept of Java servlets as an extension mechanism for SIP servers. This paper gives an overview of this API and our prototype implementation of it, including a description of how we support the Call Processing Language on top of it.**

I. Introduction

The convergence of telecommunications and computers is promising to dramatically redefine both fields. Traditional fixed line and mobile telephony on one side and email, Web, chat, presence, games, and streamed media on the other will combine to provide a wealth of new communications services. This will dramatically increase the level of reachability of people as well as the number of communication options available to them. In this brave new world of communication services, programmability is important both as a way of providing these services and as a way for end users to control them [4].

A number of protocols have been proposed for Internet telephony signaling but here we shall concern ourselves only with SIP, the Session Initiation Protocol [1]. Just like programmability of Web servers span a wide range of solutions, we will likely see a variety of approaches to programming SIP servers. A number of considerations come into play when evaluating different approaches. One important dimension is to what degree extension code is trusted by the server running it. Obviously, it is of paramount importance for a server to protect itself against accidental or malicious extension programs. Another dimension is ease-of-use. If ordinary end-users are going to configure their communications services it needs to be fairly straightforward. These two aspects, safety and simplicity, are interdependent, and it has been argued that they point to the need for two levels of programmability–one for trusted, advanced developers or system administrators, and one for untrusted end users [5].

We agree with the need for two such levels and are experimenting with solutions for both of them. We have addressed the need for a full fledged extension mechanism through the definition of a Java API which is discussed in the following section. Another approach is that of SIP CGI [5].

The Call Processing Language (CPL) has been proposed as an Internet Telephony scripting language for end users [3]. The design of CPL was informed in part by the mail filtering language Sieve, and has the attractive properties of being high-level, protocol-independent, simple, safe, and easily extensible. CPL scripts cannot contain loops and doesn't allocate memory or manipulate pointers, and so can do very little damage to the execution engine. Section III describes how we implemented CPL support on top of the more powerful Java API.

II. The SIP Servlet API

Java has many properties which makes it well-suited as an extension language. This has been exploited in the HTTP Servlet API, which allows Java objects, known as servlets, to dynamically generate content from within a Web server. We have proposed a *SIP Servlet API* based on similar principles [2]. The HTTP servlet API is often compared with CGI, and in this context the advantage most frequently cited is that servlets doesn't require a separate process per request and thus performs much better. This is also the case for SIP servlets.

The main purpose of the API is to enable programmability of SIP servers by extending control over message processing to SIP servlets. Figure 1 illustrates the basic model.

The figure shows how the servlet API is implemented by a servlet *engine* which in turn depends on the services of the underlying SIP stack. The servlet engine is responsible for mapping incoming client requests to servlets. The specifics of how this is done is considered part of deployment and is not reflected in the API as such.

There is some architectural flexibility in the relationship between SIP server and servlet engine. They may be colocated in the same process or they may run in separate

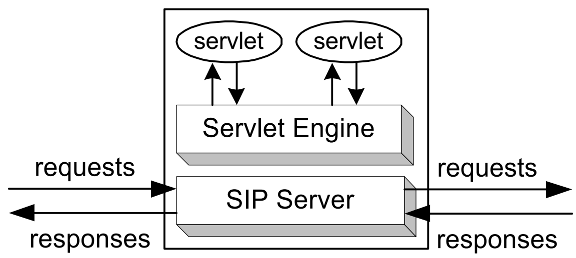


Figure 1: Basic SIP Servlet model.

processes, possibly on different hosts, in which case a custom protocol is needed for communication between them. This flexibility is also present in the HTTP Servlet API and is a useful feature for improving scalability and reliability of service platforms.

A. Overview

The API provides convenient access to common SIP abstractions and can be considered to be structured into the following parts (plus a few miscellaneous classes and interfaces):

1) Addressing

Three interfaces are concerned with addressing: *SipURL*, *SipAddress* representing To and From header fields, and *Contact* representing values of the Contact header field.

2) SIP Messages

Interfaces *SipRequest* and *SipResponse* (not surprisingly) represents SIP requests and responses, and both extend *SipMessage*. A SipMessage in turn has a reference to a *SipTransaction* object. Servlets have access to all parts of both requests and responses, although the API limits how certain *system headers* are accessed (Call-ID, From, To, CSeq, Via, Record-Route, Content-Length). The body of messages is available either as an uninterpreted byte array, or as a parsed object structure, and allows servers to handle MIME multipart messages.

3) Server Abstractions

The *SipServletContext* interface represents the SIP server to servlets, and extends *SipFactory* which allows creation of various types of objects, and *ContactDatabase*, which provides access to location information independent of the actual mechanism used by the server for storing and accessing this information, be it a relational database, a directory, or something else.

4) Servlets

The *SipServlet* interface must be implemented by SIP servlets and in addition to defining various life cycle methods (`init` and `destroy`) defines `gotResponse` and `gotRequest` methods invoked by the server in order to let the servlet do its thing. An abstract class *SipServletAdapter* is provided as a convenience class and has default implementations for all methods of SipServlet.

Servlets can perform the following operations (see also Figure 1):

- respond to requests
- proxy requests, possibly to multiple destinations
- receive and forward responses
- initiate requests of its own

The ability to proxy requests and forward responses allows servlets to operate in proxies, performing address translation type services in the call-setup phase.

The ability to respond to requests allows servlets to operate as user agent servers, most frequently rejecting or redirecting callers, but also as a server accepting a call, establishing media streams through which to interact directly with callers, e.g. by operating as a voice response unit.

Finally the ability exists for servlets to initiate calls. This capability is possibly the more controversial and might not be granted non-discriminately to all servlets.

B. Security Features

There are a number of aspects to security. For a start, language features like strong typing, exception handling, and the lack of pointers makes Java a relatively safe programming language.

Secondly, the availability of a wide range of APIs means there is little need for allowing servlets to execute arbitrary code on the host platform.

And most interestingly, the Java security model allows the server to implement fine grained access control. For example, a particular servlet may not be allowed to proxy requests, in which case attempt to do so will result in a runtime security exception.

C. Open Issues

A number of issues are currently unresolved, e.g.

- there is a need to standardize mechanisms for mapping requests to servlets along with a notation for expressing such mappings
- some provisioning for executing multiple servlets on incoming messages may be needed
- the definition of a deployment description language, probably based on XML. This will allow configuration, mapping, and security information needed for deployment to be associated with servlets in a standardized manner, again similar to the HTTP Servlet API.

It is our intention to address these issues in a future version of the API.

III. Implementing CPL on Top of the Servlet API

The difference in levels of trust and control extended to SIP servlets and CPL scripts can be reflected in implementations in a straightforward and pleasing manner by implementing the CPL interpreter as a SIP servlet. This section outlines how we have done this.

The HP SIP server consists of a number of layers. The components that make up individual layers can be configured to make the network element act as different types of servers, e.g. as a registrar, redirect, or proxy server, and to control the set of services provided by a server. At the bottom we always have the core stack. The core stack provides a number of services which must be part of any SIP stack, e.g. parsing, configuration, connection, call state, and timer management, etc. On top of the core stack a number of modules perform SIP message processing. By configuring different modules into network elements we can easily change the personality of SIP servers. Figure 2 shows part of a typical configuration for a SIP proxy.

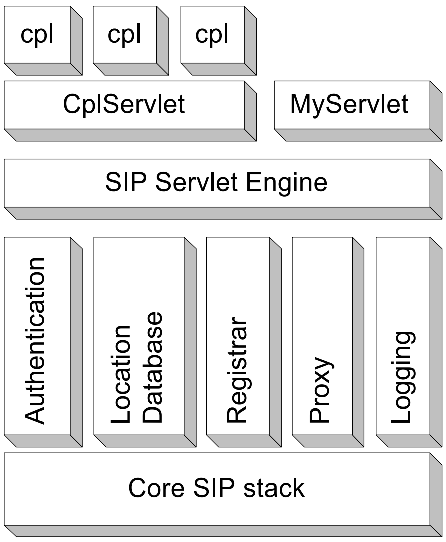


Figure 2: High-level architectural view showing how the SIP Servlet API and CPL are supported on the HP SIP proxy.

The proxy is configured with location database, registrar, and proxy modules. The module API is our internal API for fully trusted components and is our equivalent to the module API of the Apache Web server.

On top of those components we have our prototype implementation of the SIP Servlet API. The servlet engine primarily needs to interact with the proxying function but also needs access to the core services, the location database, etc.

We have then implemented a CPL interpreter as a SIP servlet. This `CplServlet` manages its own scripts and performs its own mapping from requests to scripts but from the servlet engines point of view is no different from any other servlet. This approach is essentially equivalent to how Web page scripting using Java Server Pages are commonly implemented as a `JspServlet` on Web servers. The fact that servlets have a superset of information and control available to them compared with CPL scripts makes this approach viable.

Likewise, it should be straightforward to implement a SIP CGI capable server as a `CgiServlet`.

IV. Configuring vs. Authoring Services

The purpose of CPL is to provide a degree of control over call setup to end users without compromising the safety of servers. An alternative to CPL which seems likely to become popular is to allow people to subscribe to and configure services via a Web interface. This is equally safe and employs a model well known to most people. It may be a less general solution but might well suffice for most people's needs. Maybe such a Web based SIP configuration service would internally generate CPL scripts, in which case the Web based CPL authoring service could exist separately from any CPL capable SIP server.

V. Conclusion

We expect service programmability to become increasingly important in the future and to extend more control to end users in a more direct way than ever before. We have proposed the SIP Servlet API as a powerful server extension mechanism. We have demonstrated how this can form the basis for a CPL interpreter or can be used to provide services directly, e.g. via a Web interface.

References

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

[2] A. Kristensen and A. Byttner, "The SIP Servlet API", Internet Draft, Sep. 1999, <draft-kristensen-sip-servlet-00.txt>. Work in progress.

[3] J. Lennox and H. Schulzrinne, "CPL: A Language for User Control of Internet Telephony Services", Internet Draft, Feb. 26, 1999.

[4] J. Nielsen, "A New Ideology for Messaging", netWorker, Vol. 3.4, Dec. 1999.

[5] J. Rosenberg, J. Lennox, and H. Schulzrinne, "Programming Internet Telephony Services", *IEEE Internet Computing Magazine*, May/June 1999.

A Service Platform for Internet Telephony

Stefan Gessler, Oliver Haase, Andreas Schrader

Computer & Communication Research Laboratories Heidelberg, NEC Europe Ltd
email: {Stefan.Gessler|Oliver.Haase|Andreas.Schrader}@ccrle.nec.de

Abstract— **Inevitably, two formerly separated kinds of communication networks –** ***public switched telephone networks (PSTN)*** **and** ***packet data communication networks*** **– are meeting under the umbrella of IP telephony. In this paper we present I²N (Intelligent Internet Telephony) as a novel platform for IP telephony, which takes the best of the network centric approach of PSTN and the edge centric approach of packet data networks. I²N provides various layers of a comprehensive IP telephony system, from basic call signalling, via access to user directories and support of various aspects of mobility, to the rapid integration of value-added services. Together with the integrated AQUARIUS QoS framework, I²N is perfectly suited to realise user-tailored communication applications with high quality media support. Interworking with related standards is provided by multi-level gateway technology.**

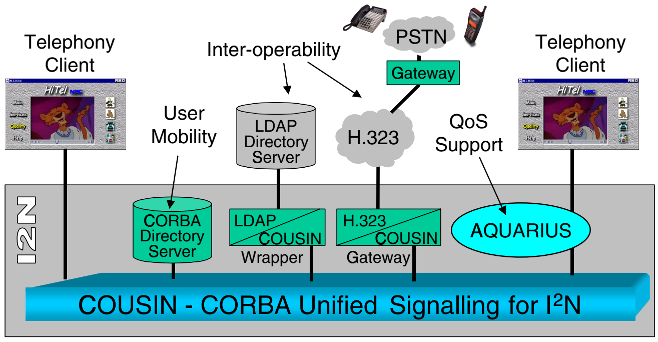


Fig. 1. General overview of I²N platform components

I. Introduction

One particular strength of the network centric approach of the PSTN is excellent support of *security*, *privacy*, and *service quality*. The edge centric approach, well-known from the Internet, offers an enormous flexibility for quick and easy development of new, possibly user-tailored services, creating new business models. In this paper we present I²N (Intelligent Internet Telephony) as a novel platform for IP telephony, which perfectly integrates the network centric and the edge centric approach, exploiting the best of both. This platform provides various layers of a comprehensive IP telephony system, from basic call signalling, via access to user directories and the support of various aspects of mobility, to the rapid integration of value-added services.

Fundamental for the design layout was the decision to rely on CORBA's remote method invocation (RMI) mechanism. This highly abstract communication paradigm results in an architecture which is easy to understand, fast to implement, and hence most cost-effective. Even in terms of performance, we can show the competitiveness of our approach. Our platform comprises a complete set of all necessary components (see fig. 1).

Active Directories: These keep track of each user's current location. We have realised LDAP-like CORBA directories, which map personal addresses to device addresses, store personal profile information, and refer to a user's individual set of services. *Call Signalling*: The I²N platform holds an instance of a signalling state component for every registered user. A telephony application on top of I²N can be implemented merely as a state-less collection of user interfaces. Running the signalling components under the control of the platform supports secure execution of critical services and enables accounting & billing facilities. *QoS Support*: The architecture includes a complete QoS framework supporting the adaptive transmission of real-time multimedia streams. *Value-Added Services*: We promote a combination of CORBA and Java. Each service consists of a downloadable Java part (e.g. the GUI for service configuration, or the modified call signalling state machine) and a stationary part at the provider's site. Since these parts communicate via CORBA IIOP, the latter one can be implemented in any programming language preferred by the service provider. *Telephony Application*: For demonstration and proof of concept we have built a thin Java telephony client which can be downloaded through the WWW. *Gateways to other Technologies*: A novel approach like this must not neglect interoperability. For this purpose, we provide interworking with other IP telephony protocols (e.g. H.323, SIP) on the three levels of media transmission, call signalling, and name resolution.

There are a number of approaches existing or under development which cover different aspects of IP telephony, e.g. the IETF Session Initiation Protocol (SIP) [11], [7], the ITU-T protocol suite H.323 [8] and the related architecture of ETSI TIPHON [3], as well as some proprietary architectures, e.g. AT&T's TOPS [2]. The main focus of most of these approaches is interoperability with PSTN (through gateway technology). Consequently, the forthcoming service integration concepts show a strong similarity with the Intelligent Networks (IN) [9]. Without doubt, IN is very useful, proven and appropriate for circuit switched, network centric technologies. However, the Internet has completely different characteristics: it is connection-less, open, programmable, edge-centric, and offers transportable software through the Java programming language. Hence, it is worthwhile re-thinking about appropriate service platforms in the Internet.

II. Active Directories

For the support of user and service mobility, *directories* are the basic supporting technology. We have implemented a network of LDAP-like CORBA directories, each of which providing fully transparent access to the entire distributed information base. Our performance measurements show the competitiveness of CORBA's general purpose IIOP protocol compared with the dedicated LDAP protocol [5], [6].

In I^2N , we use the DNS hierarchy of the Internet to distribute the global naming space into admininstrative domains. Each subscribed user belongs to a particular domain (*home domain*), for which one directory server is responsible, containing a personal entry for each subscriber consisting of

- the subscriber's current location, as described in section III;
- user defined information about the subscriber, e.g. the email address, a small picture of him, the profession, and a cookie, i.e. some free text;
- a set of references to subscribed services. Each reference to a service actually consists of two Java URLs pointing to classes, one of which is downloaded to the subscriber's current end device on service configuration, and the other one on service invocation (see section V).

Whenever an I^2N user registers with the platform, the registry of the *hosting domain* delegates the registration to its directory server (see section III). Through the distribution mechanisms of the directory service, a foreign, mobile user will be registered with the directory server of his home domain.

As can be seen, the directory is the core enabling technology for *user* as well as *service mobility*. Service mobility is crucial to provide anytime anywhere access to personalised services.

III. Call Signalling

As already discused in section I, call signalling is running under the control of I^2N to ensure security, and to enable accounting & billing. To become more concrete, a user's signalling state machine (SSM) which is capable of handling incoming and outgoing call requests, is running *remotely on a distinguished signalling server.* Two peer SSMs as well as a user's telephony application and his SSM are communicating via CORBA RMI.

In fig.2 a mobile caller is dialling a callee, who happens to be logged in his home domain. When the caller registers with the platform (1), the registry of the hosting domain creates an SSM for his telephony application (2). The object reference of the newly created SSM is stored

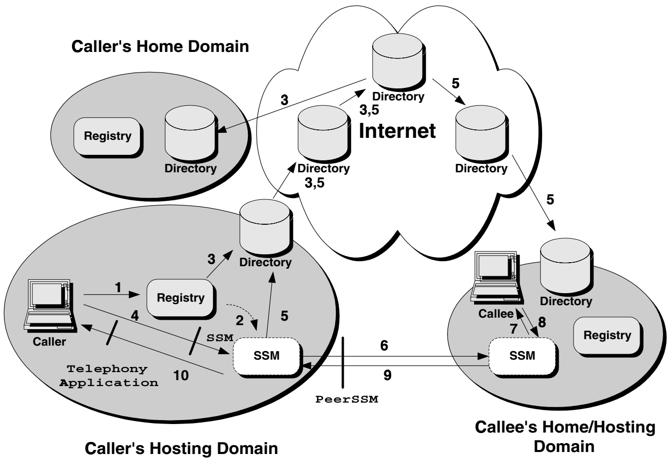


Fig. 2. Call Setup using SSMs and the appropriate CORBA interfaces

with the user's home directory server (3). The registration of the *callee* is done analogously.

All call requests to and call indications from other parties go through the SSM, which acts as a signalling proxy for the respective telephony application. An SSM provides some remote methods to the telephony application as well as some remote methods to other peer SSMs. The interface `SSM` comprises the first set of methods, while the interface `PeerSSM` consists of the methods to be invoked by peer SSMs. Each I^2N telephony application must implement the interface `TelephonyApplication`. This ensures that the SSM is allowed to invoke the appropriate call-back methods to indicate incoming requests or other users' responses to previously signalled call requests.

In the above example, the caller invokes a call request through the SSM interface (4). The SSM looks for the right object reference by a directory query (5) and forwards the request to the remote SSM via the `PeerSSM` interface (6). The callee's SSM uses the `TelephonyApplication` interface to indicate the call to the user (7). The response is handled analogously (8, 9, 10).

IV. QoS Support

Due to the connection-less nature of IP-based networks, no service qualities can be guaranteed. In wireless access networks (e.g. GSM/GPRS, UMTS) there is even a significantly high loss of packets. Therefore QoS mechanisms are mandatory for an acceptable provision of audio and video services on the Internet.

AQUARIUS (**A***daptive* **Qu***ality of Service* **Ar***chitecture for* **I***ntelligent* **U***niversal* **S***ervices*) is a comprehensive framework, consisting of distributed, hierarchically organised QoS components. AQUARIUS components are located at end-systems as well as at intermediate nodes. They provide adaptive scaling, filtering and transcoding for a broad range of audio/video codecs from low bandwidth speech codecs (e.g. GSM) to high quality video streaming (e.g. H.261/H.263). The network administrative *static, global* QoS policies are stored in the CORBA Directory Servers (in accordance with the IETF QoS Policy Framework [4]). The user's *dynamic, local* QoS wishes are mapped to appropriate network parameters within distributed QoS Brokers. With his knowledge about the underlying network layer, available codecs, etc., the QoS Broker chooses the appropriate QoS strategy and configures all other AQUARIUS QoS components. Through the usage of Java Media Technology (JMF [12]) AQUARIUS is able to download, install and maintain QoS components on demand to react to changing network characteristics. AQUARIUS is perfectly able to use underlying network QoS technologies, like the IETF *IntServ* [10] and *DiffServ* [1] architectures. The AQUARIUS QoS Brokers support all kind of multimedia requirements by choosing an appropriate compromise between resource reservations (via RSVP; mainly in corporate managed LANs), appropriate Classes of Service (CoS) (via DiffServ Codepoints; mainly in the backbone) and adaptive scaling mechanisms.

V. Value-Added Services

The rapid and easy introduction of new, user-tailored intelligent services will be the key issue for the market acceptance of IP telephony. We tackle this goal by splitting a service into a *stationary* part and *two movable* parts. The stationary part runs at the service provider's site and executes the essential computations of the service. Examples are databases or Web server with product information for e-commerce services. One movable service part is used for the *configuration* of a service, e.g. to specify the trigger conditions for services running on the callee site (e.g. *call distribution*), or to choose the personal 'look & feel' of the service. One illustrative example for a configuration service part is a GUI for the user's preselection of some frequently-called persons for an *abbreviated dialling* service. This list is transmitted to, and stored within, a server in the provider's domain. The second movable service part allows for the actual *invocation* of a service. To continue the *abbreviated dialling* example, the invocation of the service causes

- the list of preselected persons to be displayed as buttons on the screen;
- the basic functionality *call request* (see section III) to be bound to these buttons.

Both movable service parts (service configuration and invocation) must be implemented in Java, whereas the stationary part can be implemented in an arbitrary language. Either part of a service can be missing; e.g. a *call forwarding* service does not need an invocation part because this service is invoked passively when a request indication arrives.

A service is described through its *input* and its *output*. We have defined some standard types, such as `Person` or `PersonalAddress`. If a certain service (component) s_2 requires, e.g. *input* of type t_2, and service (component) s_1 produces *output* of type t_1 which is a subtype of t_2, then s_1 and s_2 can be concatenated to a new complex service (s_1, s_2). For instance, the abbreviated dialling service delivers the personal address of the selected person, which can be used to configure the call forwarding service.

VI. Telephony Application

Since the I²N architecture provides all necessary signalling and media facilities, applications using I²N can be built as state-less collections of graphical user interfaces, either for the active request of a call or as call-back methods for incoming indications. As a prototype we have developed *HiTel (**H**eidelberg **i**ntelligent **Tel**ephony)* as a Java applet which can be downloaded from the Web.

Fig. 3. HiTel telephony client

In the default version, HiTel supports only basic call signalling. On the other hand, by subscribing additional value-added and AQUARIUS QoS services, HiTel can be seen as a virtually user-tailored telephony client, which is perfectly suited for the demands of the user without changing one single line of code in the applet itself.

As indicated in section V, additional intelligent services can be provided by arbitrary service providers. HiTel provides mechanisms to *subscribe, configure, maintain* and *invoke* these services from within the telephony client. The *subscription of a service* can be invoked through the provider's HTML page. After subscription the downloadable parts of the service will be transported to and evaluated within HiTel. To *configure a service*, HiTel provides mechanisms to invoke the configuration interface of the service. To *maintain services*, the list of currently subscribed services can be viewed and modified. To *invoke a service* from within HiTel, HiTel provides an appropriate interface in its service management interface, as well as service invocation buttons on the main window of the client to allow direct access to frequently used services (see fig.3). The configuration of these buttons will be stored with the user's personal entry at his home directory server (see section II).

HiTel also provides mechanisms to chain services. Services can be concatenated dynamically, if their respective input and output types fit match, see section V. For instance, as soon as a certain name within a phonebook is chosen, the user is provided with a list of all services which get a parameter of type `Person` as input (e.g. call forwarding configuration, info page invocation, abbreviated dialling configuration, etc).

For the integration of QoS configuration and media render interfaces, HiTel is able to use the AQUARIUS system via the respective APIs. The AQUARIUS Media API can be used for example to capture audio or stream RTP/RTCP flows, but also to aquire a panel to render received video data. However, the provision of QoS can be envisaged as one particular value-added service, which can be provided by different competing QoS providers.

With the mechanisms described above, users can arbitrarily configure the HiTel system according to their personal preferences.

VII. Gateways to other Technologies

I^2N is designed as an integrated IP telephony system comprising various aspects of telephony systems. Individual aspects are also covered in some way by other existing IP telephony systems. The aimed flexibility requires us to provide inter-working facilities to those systems as well. Moreover, we do not intend to provide all desirable functionality within I^2N itself, but to rely on existing services. Inter-working or direct access is achieved by connecting the different systems via gateways. Therefore, gateways on four levels of inter-operability are under development.

1. *Call signalling* gateways, providing inter-working between I^2N call signalling and signalling of other IP telephony systems, such as H.323 or SIP.
2. *IP telephony service control* gateways. These provide I^2N service access to other IP telephony systems, and access to foreign services from within I^2N .
3. *PSTN service access* gateways. These gateways directly inter-connect I^2N services and PSTN service entities on a peer-to-peer level.
4. *Value-added service* gateways. Similar to the previous category, these connect I^2N service entities with Non-PSTN (e.g. Web based) services.

The reader may have noticed that no gateways between PSTN signalling and I^2N signalling are planned. However, placing a call from I^2N to PSTN and vice versa can be performed over a slight detour using an I^2N /H.323 as well as an H.323/PSTN gateway. This technique avoids the provision of own signalling gateways to PSTN.

VIII. Conclusion

In this paper we have presented a novel integrated architecture for IP telephony services. Our approach covers the complete set of necessary functionality, ranging from basic call signalling, to the support of mobility via active directories and mechanisms for the subscription of user-tailored value-added services. The architecture also includes the AQUARIUS QoS framework for the support of adaptive media transmission.

Through the usage of the CORBA communication paradigm, the I^2N architecture combines network layer and programming language independence. With the provision of the Java based HiTel telephony client we demonstrated the ability to use downloadable value-added services to enhance the overall functionality on demand. Gateways allow for interoperability with other telephony architectures, like H.323, SIP, and PSTN.

The future deployment of IP based telephony systems will mainly depend on the availability of services, which are not possible or not seen so far in the traditional telephone networks. By providing mechanisms to download new value-added service on demand, new business models can be identified for service providers. The same argument holds for the provision of Quality of Service, which can be interpreted as one special instance of such a value-added service. An open question remains in the automatic detection of service incompatibilities (e.g. conflicts in the necessary modification of signalling state machines).

Future research activities can be identified in the area of accounting and billing. Through the introduction of new business roles for value-added service and QoS providers, appropriate mechanisms are essential to handle the negotiation of prices and to control the registration of actual service usage. Another very important aspect is security. To protect the privacy of users from unsocial behaviour of other users' applications and to protect the service providers from illegal use of resources, appropriate mechanisms are needed. The analysis of the possibilities and restrictions of small end devices, e.g. embedded systems or stand-alone IP-Phones without operating systems, is also a challenging open issue.

References

[1] A. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss. *RFC2475: An Architecture for Differentiated Services*. IETF.

[2] N. Anerousis et al. TOPS: An Architecture for Telephony over Packet Networks. *IEEE Journal on Selected Areas in Communications*, 17(1), January 1999.

[3] ETSI. *ETSI technical specification TS 101 313: Telecommunications and Internet Protocol Harmonization Over Networks (TIPHON)*, Feb 1999.

[4] S. Gai, J. Strassner, D. Durham, S. Herzog, H. Mahon, and F. Reichmeyer. *Internet Draft: QoS Policy Framework Architecture*. IETF, February 1999.

[5] O. Haase, A. Schrader, K. Geihs, and R. Janz. CORBA Directories for Virtual Home Environments. In *Proc. SoftCOM'99 Intl Conference on Software in Telecommunications and Computer Networks*, pages 215 – 224, Split - Rijeka, Croatia; Trieste - Venice, Italy, Oct 1999. ISBN 953-6114-32-1.

[6] O. Haase, A. Schrader, K. Geihs, and R. Janz. Mobility Support with CORBA Directories. In T. Znati and R. Simon, editors, *Proc. CNDS'00 Intl Conference on Communication Networks and Distributed Systems Modelling and Simulation*, pages 20–29, San Diego, USA, Jan 2000. ISBN 1-56555-179-6.

[7] IETF. *RFC2543: SIP - Session Initiation Protocol*, March 1999.

[8] ITU-T. *Recommendation H.323: Packet-based multimedia communications systems*, Feb 1998.

[9] T. Magedanz and R. Popescu-Zeletin. *Intelligent Networks*. International Thomson Computer Press, 1996.

[10] D. Clark R. Braden and S. Shenker. *RFC1633: Integrated Services in the Internet Architecture: An Overview*. IETF, June 1994.

[11] J. Rosenberg, J. Lennox, and H. Schulzrinne. Programming Internet Telephony Services. *IEEE Network Magazine*, 13(3):42–49, May/June 1999.

[12] Sun. *The Java Media Framework Version 2.0 API*. http://java.sun.com/products/java-media/jmf.

A Service Platform for Internet Telephony

{Stefan.Gessler|Oliver.Haase|Andreas.Schrader}@ccrle.nec.de

C&C Research

Heidelberg NEC

Motivation

PSTN:

- network centric
- security,
- privacy,
- service quality

Internet:

- edge centric
- flexibility,
- service-orientation
- adaptivity

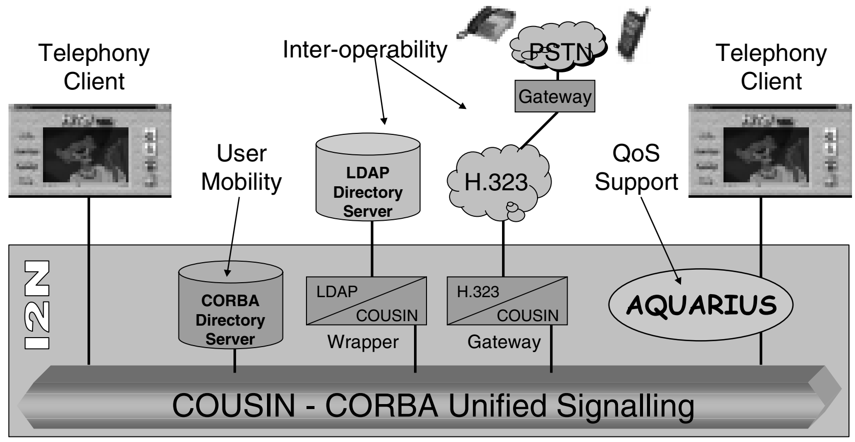
I²N (Intelligent Internet):

- telephony platform *combining both* approaches
- comprising different component to support *various layers* of a telephony system

C&C Research

Heidelberg NEC

Motivation (cont'd)



C&C Research

Heidelberg NEC

Active Directories

Usage of LDAP-like CORBA Directories for storage and retrieval of

- **current location of a subscriber (inherent *user mobility*); the location is updated automatically whenever a subscriber registers or unregisters with the system;**
- **user defined personal information, e.g. postal address, email address, picture, profession, etc.;**
- **set of subscribed services (URLs to Java service proxies)**

The CORBA Directories provide *fully transparent*, *type safe* access to a *globally distributed* information base.

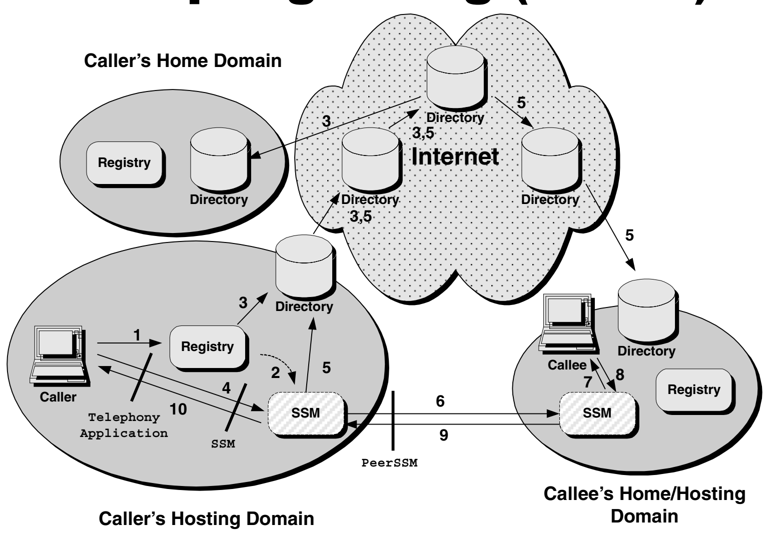
C&C Research

Heidelberg NEC

Call Setup Signalling

- On registration of a user, a new signalling state machines (SSM), running on a *distinguished server*, is created
- The address of this SSM is stored in user's personal directory entry
- SSMs are running remotely to put them under *network control*
- SSM can be envisaged as a *signalling proxy* of the application
- two peer SSMs communicate via CORBA RMI
- telephony application and its SSM also communicate via CORBA RMI
- the application must provide certain callback methods to its SSM; this is ensured through a CORBA interface which must be implemented by each application

Call Setup Signalling (cont'd)



QoS Support

AQUARIUS

Adaptive Quality of Service Architecture for Intelligent Universal Services

- Middleware between Applications and Operating System
- Distributed Collection of QoS Broker, QoS Manager and QoS Control entities
- Real-time Media Capturing, Coding, Streaming, Rendering (JMF and native)
- QoS Scaling, Filtering, Transcoding Mechanisms
- User and Management QoS Policies
- QoS Parameter Interface Provision
- Downloadable Codecs (Software Radio)
- Support of network QoS technologies (DiffServ, IntServ)
- Smoothly integrated in I^2N but suitable for any Multimedia Application

Heidelberg NEC C&C Research

Value-Added Services

- services are split into a *movable* (service proxy) and a *stationary* part
- movable part
  + must be implemented in Java
  + can provide a `use` and a `config` method
  + communicate with the stationary part over CORBA RMI
  + expose their input and output types through Java *reflection* mechanisms; these types can be used to - either *statically* or *dynamically* - combine services to *complex* services
- stationary part
  + can be implemented in an *arbitrary* language (due to CORBA)
  + is typically a database, Web server, user directory, ...

Heidelberg NEC C&C Research

Value-Added Services (cont'd)

list of subscribed services

PhoneBook service provides `use`, but no `config` method

methods for maintainance of services list

frequently used *fantastic 4* services, accessible from main card

Heidelberg NEC

C&C Research

Telephony Application

HiTel - Heidelberg Intelligent Telephony

- Prototype Development for demonstrating I²N Concepts
- Java Application (Applet), Swing Graphical Interfaces, JMF Media Panels
- Stateless Collection of Graphical User Interfaces
- User-tailored Service integration by downloading transportable Service code
- Mechanisms to subscribe, configure, maintain, invoke and withdraw VAS
- 'Fantastic-4' Fast Service Invocation Interface
- Concatenation of Services to Chains by Type Filtering Mechanisms
- HTML Browser for Third Party Service Provider Access
- Context-sensitive Help Functionality

Heidelberg NEC

C&C Research

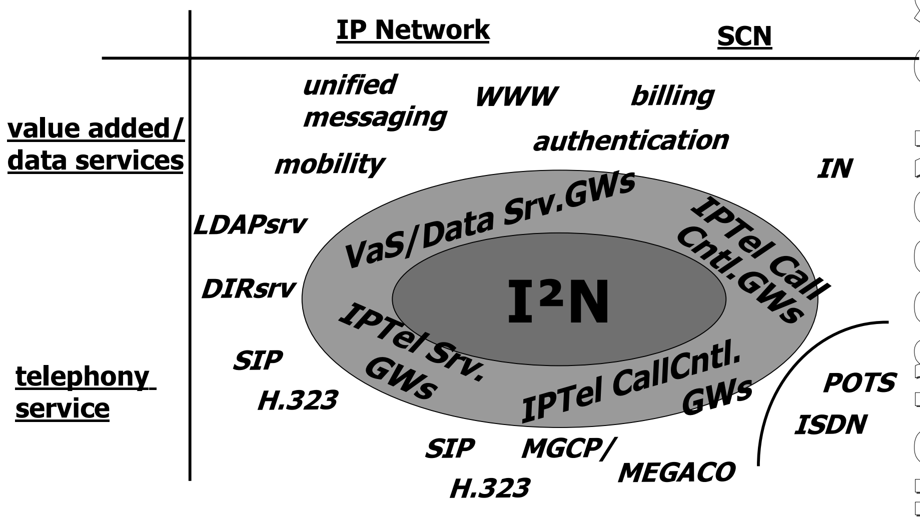
Telephony Application (cont'd)

HiTel - Heidelberg Intelligent Telephony

C&C Research

Heidelberg

Multi-Level Gateways



C&C Research

Heidelberg

Conclusion

- I^{2}N is an integrated, CORBA-based architecture covering
  - basic call signalling
  - mobility support through active directories
  - provision of value added services
  - QoS support through AQUARIUS' adaptive media transmission
- I^{2}N's service concept demonstrates the usefulness of Java technology for downloadable services (proxies)
- Due to the generic service interface, today's HiTel telephony application is able to deal with tomorrow's (user-tailored) services
- Even *QoS provision* can be envisaged as one particular VAS

C&C Research

Heidelberg NEC

Interworking Between SIP/SDP and H.323

Kundan Singh and Henning Schulzrinne
Dept. of Computer Science
Columbia University
New York, USA
{kns10,hgs}@cs.columbia.edu

Abstract—**There are currently two standards for signaling and control of Internet telephone calls, namely ITU-T Recommendation H.323 and the IETF Session Initiation Protocol (SIP). We describe how a signaling gateway can allow SIP user agents to call H.323 terminals and vice versa. Our solution addresses user registration, call sequence mapping and session description. We also describe and compare various approaches for multi-party conferencing and call transfer.**

Keywords— **Internet telephony, Interworking, SIP, SDP, H.323, Signaling gateway.**

I. INTRODUCTION

IT appears likely that both the Session Initiation Protocol (SIP) [1], [2], together with the Session Description Protocol (SDP) [3], and the ITU-T recommendation H.323 in its various versions [4], [5] will be used for setting up Internet multimedia conferences and telephone calls. For example, currently H.323 is the most widely used protocol for PC-based conferences, due to the widespread availability of Microsoft's NetMeeting tool, while carrier networks using so-called soft switches and IP telephones seem to be built based on SIP. Thus, in order to achieve universal connectivity, interworking between the two protocols is desirable. This paper describes approaches to achieving this.

The ITU-T Recommendation H.323 [4] defines packet-based multimedia communication systems and is based heavily on previous ITU-T multimedia protocols. In particular, H.323 call signaling is inspired by H.320 [6] for ISDN, and call control by H.324 [7] for GSTN terminals. SIP [1], developed in the IETF, builds on a simple text-based request-response architecture similar to other Internet protocols such as HTTP [8] and RTSP [9]. With the exception of conference control, SIP provides a similar set of basic services as H.323 [10], [11].

Interworking between the protocols is made simpler since both operate over IP (Internet Protocol) and use RTP (Real time Transport Protocol [12]) for transferring real-time audio/video data, reducing the task of interworking between these protocols to merely translating the signaling protocols and session description. Since no media data needs to be translated, a single gateway can likely serve thousands of end systems.

Interworking between SIP and H.323 requires transparent support of signaling and session descriptions between the SIP and H.323 entities. We call the server providing this translation a SIP-H.323 *signaling gateway* (SGW). We refer to the set of terminals speaking H.323 and SIP as the H.323 and SIP *networks*, respectively, even though they are likely to be intermingled on the same IP network. We use the term *native network* to refer to the network used by a particular terminal, while the *foreign network* is the network whose access is mediated by the SGW. For an H.323 terminal, a SIP terminal is in a foreign network.

When addressing a terminal using another signaling protocol, there are two approaches. First, the user can explicitly identify the protocol as part of the address, for example, by inventing some form of H.323 URL[1] such as `h323:alice@columbia.edu`. If, for example, an H.323 URL is used by a SIP terminal, it would then be the responsibility of the SIP terminal to find the appropriate SGW.

Alternatively, a terminal using a particular signaling protocol sees all other terminals as being native, and does not know or care that a particular address refers to a terminal in the foreign network. Indeed, an address could well change between being native and foreign, depending on what equipment the owner of the address happens to be using. This approach is preferable, but requires that user registrations are exported into the foreign network. Depending on the type of information sharing between H.323 or SIP elements and the SGW, different architectures are possible to provide the transparent address resolution and call establishment, as we will discuss below.

A. *Outline of the rest of the paper*

The remainder of the paper is organized as follows. In Section II, we list the problems in translating SIP to H.323

This work was supported by a grant from Sylantro Corp.

[1] Such a URL scheme was proposed by Cordell [13] in an expired Internet draft.

and vice versa. Section III describes and compares different approaches to address user registration. In Section IV, we describe a mechanism to map SIP addresses to H.323 addresses. Call sequence mapping between SIP and H.323 is described in Section V. Section VI gives an insight into translating multi-party conferencing and call transfer. Finally, we describe our current implementation and future work in Section VIII.

II. Background

A. Protocol overview

H.323 includes various other subprotocols: H.225.0 [14] for connection setup and media transport (RTP), resource access and address translation, H.245 [15] for call control and capability negotiation, H.332 [16] for large conferences, H.235 [17] for security, H.246 [18] for interoperability with the PSTN, H.450.x [19], [20], [21] for supplementary services like call transfer.

In H.323, a simple call is established as follows. If a user (say Alice) wants to talk to another user (Bob), Alice first sends an admission request to its gatekeeper. The *gatekeeper* acts as a management entity in H.323, which grants access to resources, controls bandwidth and maps user names to IP addresses, among other things. The gatekeeper finds out the IP addresses at which Bob can be reached and informs Alice. After that, Alice establishes a TCP connection to the IP address of Bob. This is followed by ISDN-like *call signaling* procedure. Alice sends a Q.931 [22] SETUP message and Bob responds with a Q.931 CONNECT message. Once the first stage of Q.931 signaling is complete, H.245 takes over. H.245 messages are used to negotiate terminal capabilities, i.e., the support for various audio/video algorithms. The H.245 OpenLogicalChannel procedure is used for opening different unidirectional media channels. A *media channel* is defined as a pair of UDP channels, one for RTP and the other for RTCP. Audio and video packets are encapsulated in RTP and sent from one end system to the other. Depending on the version of H.323, Q.931 and H.245 steps can be combined in various ways.

SIP sets up calls with an INVITE message and a response from the called party. Both INVITE and the response contain a *session description* indicating terminal capabilities, typically, but not necessarily, encoded using SDP. Proxy and redirect servers are responsible for translating between user names and the called party's IP address.

B. Call setup translation

Three pieces of information are needed for establishing an call between two endpoints, namely the signaling destination address, local and remote media capabilities, and local and remote media transport addresses at which the endpoint can receive the media packets. In H.323, this information is spread over different stages of the call setup, while SIP conveys it in an INVITE message and its response.

Translating a SIP call to an H.323 call is straightforward. The SGW gets all three pieces of information in the SIP INVITE message and can split it across multiple stages of the H.323 call establishment. However, in the reverse direction, from H.323 to SIP, the different stages of H.323 call establishment have to be merged into a single SIP INVITE message. We describe and compare various approaches in Section V. The H.323v2 (version 2.0) Fast Connect procedure is a step towards simplifying the multistage signaling of H.323. However, it is optional and an H.323v2 entity is required to support the traditional multistage signaling. Thus, we describe call setup both with and without Fast Connect.

C. User registration

SIP-H.323 translation also has to solve the user registration problem. User registration involves mapping of user names, phone numbers or some other human-understandable identifier such as email addresses to network addresses. By allowing users to be reached by location-independent identifiers, User registration provides personal mobility. For instance, a call destined at *sip:bob@mydomain.com* reaches user Bob no matter what IP address he might currently be using.

In SIP, proxy and redirect servers access a location server, often a registrar that receives user registration information. A server at *mydomain.com* will map all the addresses of the form *sip:xyz@mydomain.com* to the appropriate IP addresses, depending on where *xyz* is currently logged in. In H.323, the same functionality is performed by the H.323 gatekeeper. The SGW should use the user registration information available in both networks to resolve a user name to an IP address. The SGW can contain a SIP registrar server, an H.323 gatekeeper or neither, as discussed in Section III.

D. Session description

An SGW also must map session descriptions between the two signaling protocols. H.323 uses H.245 for session description. H.245 can negotiate media capabilities, provide conference floor control, and establish and tear down media channels. In H.245, media capabilities are described as a set of capability descriptors, listed in decreasing order of preference. A *capability descriptor*, also called a simultaneous capability set, is a set of alternative capability sets, where each alternative capability set contains a list of algorithms, only one of which can be used at any given time. For instance, a capability descriptor $\{[a_1, a_2][v_1, v_2][d_1]\}$ has three alternative capability sets: $[a_1, a_2]$, $[v_1, v_2]$, and $[d_1]$. It indicates that the terminal can support audio, video and data simultaneously. Audio can use either codec a_1 or a_2, video codec v_1 or v_2, and data format d_1.

SIP can, in principle, use any session description format. In practice, however, SDP is used exclusively. SDP lists media types and the supported encodings for each. Unlike H.245, SDP cannot express cross-media or intermedia constraints, however. For example, SDP cannot indicate that for a particular media type, the other side can only choose subset A or subset B of the listed codecs, but not codecs from both subsets. Similarly, SDP cannot express that certain audio codecs can only be used in conjunction with certain video codecs.

Thus, a SIP media capability can be easily described in H.245, however the reverse is more complicated. One approach is to carry multiple SDP messages in the message body of SIP INVITE requests and responses, using the "multipart" content type. Each SDP message then represents one capability descriptor of the H.245 capability set. In Section V we describe how sending multiple SDP messages can be avoided.

E. Multi-party conferencing

Ad-hoc conferencing among SIP and H.323 end systems is not possible without modifying one or both of these protocols. Ad hoc conferencing is defined as the one in which the participants do not know in advance whether the call will be point-to-point (two-party) or multi-party. The participants can switch from a point-to-point call to a multiparty conference or vice-versa during the call. It is possible for the participants to invite a third party in the conference or for the third party to join the conference. Both SIP and H.323 individually support ad hoc conferencing. In SIP, conference topology can be a full mesh with every participants having a signaling relationship with every other participant or a centralized bridged conference (star topology) in which every participant has a signaling relationship with the central conference bridge [23], [24]. It is possible to switch from a mesh to a bridged conference. In H.323, conferences are managed by central entity called a *Multipoint Controller* (MC). An MC can be part of an H.323 terminal, gateway, gatekeeper, or MCU (Multipoint Control Unit). H.323 conferences have inherently a star topology with every participant having an H.245 control channel with the MC. The MC is responsible for deciding the common media capabilities for the conference, conference floor control, and other conferencing functions. All the participants are required to obey the media capabilities given by the MC. Because of the difference in the topology of the conferences in the SIP and H.323 (star like in H.323 and full mesh or star like in SIP), the transparent support of multiparty conferencing cannot be achieved without modifying the protocols. However, with some simplifying assumptions, basic conferences can be set up, as described in Section VI.

F. Call services

Advanced call services like call forwarding and call transfer are supported by both SIP and H.323. H.323 uses H.450.x for these supplementary services. SIP has support for blind transfer, operator assisted transfer, call forwarding, call park and directed call pickup [23]. These services are not yet widely deployed, so that translation is not critical at this moment. Section VI describes some of the issues related to this.

G. Security and quality of service

Other problems in SIP-H.323 translation include security and quality of service (QoS). Both, SIP and H.323, individually support these. However, translating from the open architecture of SIP, where security and QoS is independent of the connection establishment, to H.323, where security and QoS go hand-in-hand with the call establishment, remains an open issue.

III. Architecture for user registration

In this section, we describe different architectures for user registration and address resolution. *User registration servers* are the entities in the network which store user registration information. SIP registrars and H.323 gatekeepers are user registration servers. It simplifies locating users independent of the signaling protocol if the SGW has direct access to user registration servers. The user registration server forwards the registration information from one network, to which it belongs, to the other.

A. Signaling gateway contains SIP proxy and registrar

Our first approach combines an SGW with a SIP registrar and proxy server, as shown in. Fig. 1(a). In this approach the registration information is maintained by the H.323 gatekeeper(s). Whenever the SIP registrar receives a SIP REGISTER request, it generates a registration request (RRQ) to the H.323 gatekeeper, translating a SIP

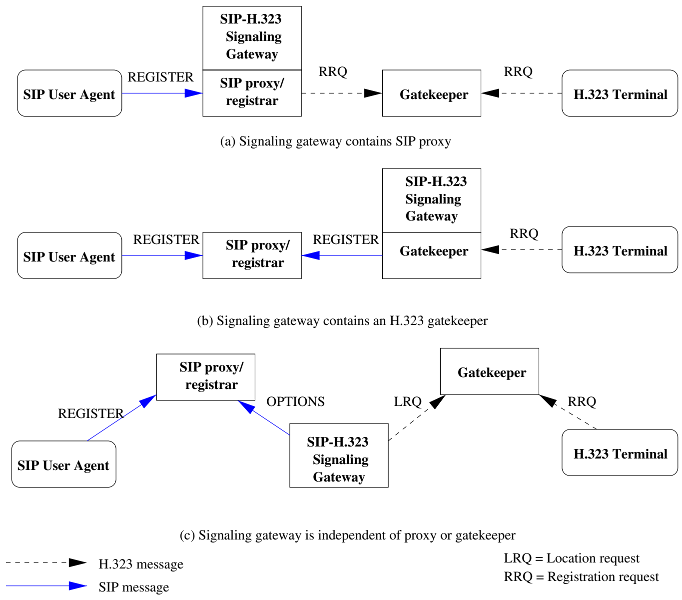


Fig. 1. Architectures for user registration

URI into H.323 Alias Address. H.323 users register via the usual H.225.0 procedure. Since the SIP registration information is also available through the H.323 gatekeeper(s), any H.323 entity can resolve the address of SIP entities reachable via the SIP server/signaling gateway. In the other direction, if a SIP user agent wants to talk to another user, who happen to reside in the H.323 network, it sends a SIP INVITE message to the SIP server. The SIP server multicasts H.323 location requests (LRQ) to the H.323 gatekeepers. The gatekeeper to which the H.323 user is registered responds with the IP address of the H.323 user. Once the SIP server knows that the address belongs to the H.323 world, it can route the call to the destination.

One drawback of this approach is that the H.323 gatekeepers are burdened with all the registrations in the SIP network.

This approach only makes those SIP addresses handled by the registrar available to the H.323 zone. Typically, a registrar is responsible for a single domain, e.g., `columbia.edu`. Thus, each H.323 zone would have to have an SGW. If an H.323 user wants to call a SIP terminal, first the H.323 terminal locates, using DNS TXT records, [25, p. 57] the appropriate gatekeeper[2], which in turn uses the registration information conveyed by the SGW to discover that this address is actually located in the SIP network.

B. Signaling gateway contains an H.323 gatekeeper

This architecture, shown in Fig. 1(b) is similar to the previous approach except that the SIP proxy server maintains the user registration information from both networks. Any H.323 registration request received by the H.323 gatekeeper is forwarded to the appropriate SIP registrar, which thus stores the user registration information of both the SIP and H.323 entities.

To the SIP terminal, H.323 terminals simply appear as SIP URLs within the same domain. (See Section IV on how H.323 addresses are translated to SIP URLs.) If an H.323 entity wants to talk to a user who happens to reside in the SIP network, it sends an admission request (ARQ) to its gatekeeper. The gatekeeper multicasts the location request (LRQ) to all the other gatekeepers. The GK-SGW server captures the request and tries to find out if the address belongs to a SIP user. It does so by sending a SIP OPTIONS request, which does not set up any call state. If the address is valid in the SIP network and the user is currently available to be called, the SGW responds with the location confirmation (LCF), letting the H.323 terminal know that the destination is reachable.

This approach has the similar drawback as the previous approach (Section III-A) in that the proxy has to store all H.323 registration information.

However, this approach has the advantage that even if some H.323 gatekeepers are not equipped with a SGW, the address resolution works: If an H.323 gatekeeper cannot resolve a called address, it multicasts a location request (LRQ) to the other gatekeepers in the network. As long as at least one H.323 gatekeeper exists with the SIP-H.323 signaling translation capability, the SIP user can be located from the H.323 network. Note that the previous approach (Section III-A) required that all the SIP registrars/proxy servers must be equipped with SGWs.

C. Signaling gateway is independent of proxy or gatekeeper

In the third approach, shown in Fig. 1(c), the signaling gateway is not colocated with either an H.323 gatekeeper or an SIP proxy server. User registration is done independently in the SIP and H.323 networks. However, when a call reaches the SGW, the SGW queries the other network for user location. Here, we assume that the SGW is capable of interpreting and responding to the location request (LRQ) from the H.323 network.

The address resolution mechanism works as follows. Suppose the SIP user Sam wants to talk to Henry, an H.323 user. Henry has registered with its own gatekeeper in the H.323 network and the gatekeeper knows Henry's IP address, conveyed via RRQ. When Sam contacts the SIP proxy with Henry's name, the SIP proxy has no registration for Henry, but is configured to contact the SGW in case the called party is in the H.323 network. The SGW,

[2] It is not clear how widely implemented this approach is.

in turn, multicasts the location request (LRQ) for Henry to all gatekeepers. If there is no positive response from the gatekeepers of the H.323 network within a timeout period, the SGW concludes that the address is not valid in the H.323 network and the branch fails.

In the other direction, Henry sends an admission request (ARQ) to its gatekeeper. Since this gatekeeper does not have the address mapping for Sam, it multicasts the location request (LRQ) for Sam to the other gatekeepers in the network. In addition, the SGW is tuned to receive the LRQ. The SGW then uses the SIP OPTIONS request (as in Section III-B) to find out if Sam is available in the SIP network and informs the GK if the request succeeds. This is followed by H.323 call establishment between Henry and the SGW and a SIP call between the SGW and Sam.

The SGW should support direct H.323 connections. For instance, a SIP user (Sam) should be able to call an H.323 user (Henry) through the signaling gateway (say `sip323.columbia.edu`) by placing a call to `sip:henry@sip323.columbia.edu`. Similarly, the H.323 user should be able to reach a SIP user (`sip:sam@mydomain.com`) by establishing a Q.931 TCP connection to the signaling gateway and providing the destination address or the remote extension address in the Q.931 SETUP message as `sip:user1@mydomain.com`. The direct connection does not involve user registration and the caller is expected to know that the destination is reachable via the signaling gateway.

IV. ADDRESS TRANSLATION

While user registration exports identities into the foreign network, address translation is performed by the SGW to create valid SIP addresses from H.323 addresses and vice versa. In SIP, addresses are typically SIP URIs of the form sip:*user@host*, where *user* names can also be telephone numbers. However, SIP terminals can also support other URLs schemes, for example "tel:" URLs for telephone numbers [26] or H.323 URLs [13]. Generally, SIP terminals proxy calls to their local server if they do not understand the particular URL scheme, in the hope that the server can translate it.

In H.323, addresses (ASN.1 AliasAddress) can take many forms, including unstructured identifiers (h323-ID), E.164 (global) telephone numbers, URLs of various types, host names or IP address, and email addresses (email-ID). Local user names and host names appear to be most common. For compatibility with H.323 version 1.0 entities, the h323-ID field of H.323 AliasAddress must be present.

For SIP-H.323 interoperability, there should be a consistent and unique way of mapping a SIP URI to an H.323 address and vice-versa. Translating a SIP URI to an H.323 AliasAddress is easy: We simply copy the SIP URI verbatim into the h323-ID. The user and host parts of SIP-URI are used to generate an email identifier, "*user@host*", which is stored in the email-ID field of AliasAddress. The transport-ID parameter is copied from the host part of SIP-URI if the latter is given numerically. The e164 field is extracted from the user part of SIP address if it is marked as a telephone number.

Translating an H.323 AliasAddress to a SIP address is more difficult since multiple representations (e.g., e164, url-ID, transport-ID) need to be merged into a single SIP address. In the easiest case, the alias contains a url-ID with a SIP URI, in which case it is simply copied into the SIP message. Otherwise, if the h323-ID can be parsed as a valid SIP address (e.g., "Alice <sip:alice@host>" or "alice@host") it is used. Next, if the transport-ID is present and it does not point to the SGW itself, then it forms the host and port portions of the SIP URI. Finally, if the H.323 alias has an email-ID, it is used in the SIP URI prefixed with "sip:" URI scheme.

Note that the translated address may not necessarily be valid. On the H.323 side, it may be desirable to configure a gatekeeper to route all calls that are not resolvable within the H.323 network to the SGW, which would then attempt a translation to a SIP URI. This would allow H.323 terminals to reach any SIP terminal, even those not cross-registered.

V. CONNECTION ESTABLISHMENT

Once the user knows that the destination is reachable via the signaling gateway, the connection is established. A point-to-point call from Alice to Bob needs three cruicial pieces of information, namely the logical destination address (A) of Bob, the media transport address (T) at which each of the users is ready to receive media packets (RTP/RTCP) and a description of the media capabilities (M) of the parties. Alice should know A, T and M of Bob and Bob needs to know Alice's T and M. The difficulty in translating between SIP and H.323 arises because A, M, and T are all contained in the SIP INVITE request and its response, while H.323 may spread this information among several messages.

A. Using H.323v2 Fast Connect

If the H.323v2 Fast Connect procedure is available, the protocol translation is simplified because fast start establishes call in a single stage, with a one-to-one mapping between H.323 and SIP call establishment messages. Both the H.323 SETUP message with fast start and the SIP INVITE request have all three components. If the call succeeds, both the H.323 CONNECT message with Fast Connect, and the SIP 200 response, including the session description, have the required components (M and T of the call destination).

Since Fast Connect is optional in H.323v2, an H.323 entity must be able to handle calls without the Fast Connect feature for backward compatibility. In particular, the SGW must accept a non-Fast Connect call from the H.323 side. In the other direction, the SGW should try to use H.323v2 Fast Connect, but must be prepared to switch to the multi-stage call establishment procedure if the response from the H.323 entity indicates that this is not supported.

B. Call translation without using Fast Connect

Translating a SIP call to an H.323 call is straightforward even without Fast Connect. The SGW uses A, M and T for the Q.931 and H.245 phases. The responses from the H.323 side are collated and forwarded to the SIP side, as shown in Fig. 2.

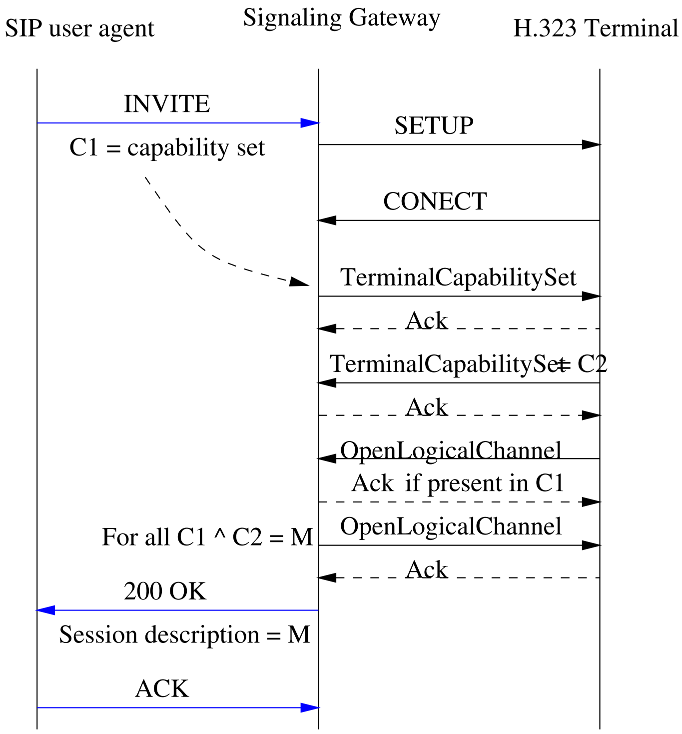


Fig. 2. Call from SIP terminal to H.323 terminal without Fast Connect

A multi-stage H.323 call can be translated to a SIP call in a variety of ways. One obvious approach is to accept the H.323 call without informing the SIP user agent. The H.323 call proceeds between the H.323 terminal and the SGW as if the SGW is just another H.323 terminal. The signaling gateway may get the media capabilities of the SIP user agent using the SIP OPTIONS message. Media capabilities of the H.323 terminal are obtained via H.245 capability negotiation. Once the logical channels are established from the SGW to the H.323 terminal, the SGW knows M and T and can place a SIP call by sending an INVITE. The media transport address from the 200 response is conveyed to the H.323 terminal while acknowledging the OpenLogicalChannel requests of the H.323 terminal.

While this approach is pretty simple, it has the disadvantage that the SGW accepts the call without even asking the actual destination, leading to caller confusion if the SIP destination is not reachable.

This problem can be solved if the SGW sends a SIP INVITE without session description or a session description without media transport information when receiving the Q.931 SETUP message from the H.323 terminal. Only after the SIP user agent has accepted the call, the SGW forwards the confirmation (Q.931 CONNECT) to the H.323 terminal. The rest of the call establishment proceeds as before, except that the SIP OPTIONS message is not needed because the 200 response from the SIP user agent describes the media capabilities.

The media capabilities of the H.323 terminal are received in the H.245 TerminalCapabilitySet message and are forwarded to the SIP user agent as part of the ACK message or via an additional INVITE. The media capabilities of the SIP user agent are found in the session description of the 200 response to the INVITE request.

The different interpretations of media capabilities by H.245 and SDP potentially causes problems during the call. In SDP, a receive media capability of G.711 and G.723.1 means that the sender can switch between these algorithms at any time during a call without explicitly informing the receiver. However, in H.245, the sender chooses an algorithm from the capability set of the receiver and explicitly opens a logical channel for that algorithm. The sender cannot switch dynamically to another algorithm without informing the receiver. The sender has to close the previous logical channel and re-open it with new algorithm. Alternatively, the receiver can use H.245 ModeRequest to request the sender to use a different algorithm.

This problem can be addressed by having the RTP/RTCP packets from SIP to H.323 be intercepted by the SGW. If the SGW detects a change in coding algorithm, it initiates the required H.245 procedures. However, this approach is not advisable, as it scales poorly.

Another approach limits the media description sent to the SIP side to only one algorithm per media (or per alternative capability set). This can be achieved by maintaining a maximal intersection of the SIP and H.323 terminal capability sets. A maximal intersection of two capability sets is a capability set which is a subset of both the capability sets and no other superset is a subset of those capability sets. The operating mode, that is, the selected algorithms for the call, is derived from the intersection of the two capability sets by selecting one algorithm per alternative capability set. If the SIP side sends additional INVITE requests during the call to change media parameters, the SGW simply recalculates the operating modes.

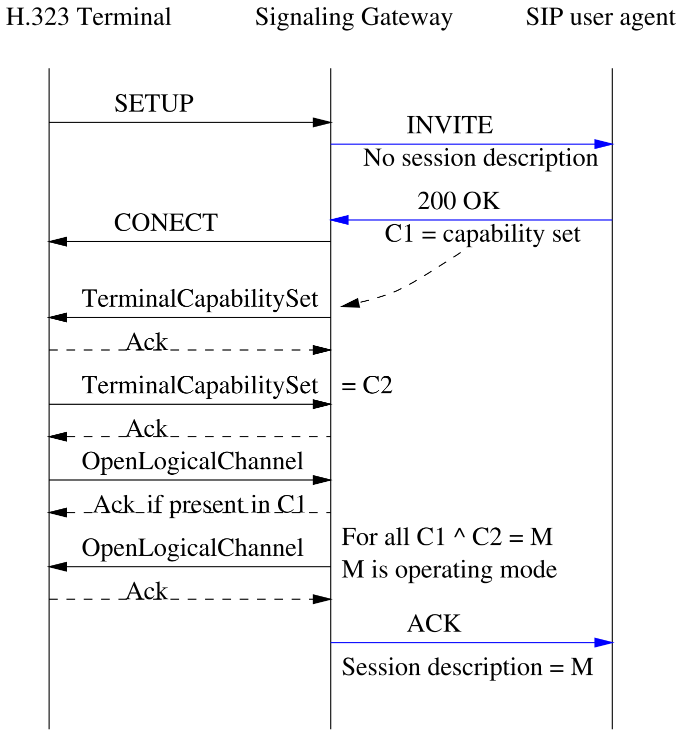


Fig. 3. Call from H.323 to SIP terminal call without Fast Connect

Finding maximal intersection of capability sets is described in [27]. As an example, let the SIP capability set be {[PCMU,PCMA,G.723.1][H.261]} and H.323 capability set be {[PCMU,PCMA,G.729][H.261]} {[G.723.1][H.263]} (i.e., the SIP user can support PCMU, PCMA or G.723.1 audio and H.261 video, whereas the H.323 user can support either one of the PCMU, PCMA, G.729 audio with H.261 video or G.723.1 audio with H.263 video). The maximal intersection as calculated by the SGW is {[PCMU,PCMA][H.261]} {[G.723.1]}. The signaling gateway derives an operating mode by selecting a capability descriptor from the maximal intersection and selecting one algorithm per alternative capability set (e.g., {PCMU,H.261}). The signaling gateway conveys only the PCMU audio and H.261 video to the SIP user agent. If the SIP side sends additional INVITE with a different capability set ({[G.729,G.723.1][H.261]}, the new maximal intersection becomes {[G.729][H.261]}{[G.723.1]}. The signaling gateway derives a new operating mode ({G.729,H.261}) and initiates the H.245 procedure to change the PCMU audio to G.729.

VI. Translating advanced services

Both SIP and H.323 support advanced services like multi-party conferencing and call transfer. In this section we propose possible approaches for translating these services.

A. *Multi-party conferencing*

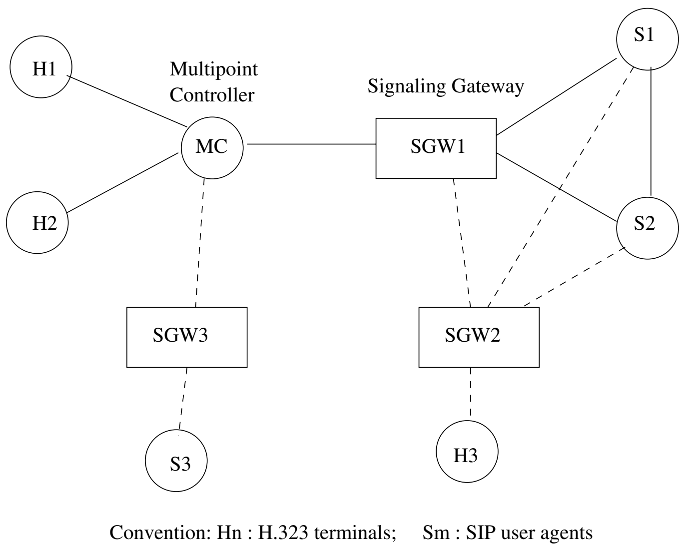


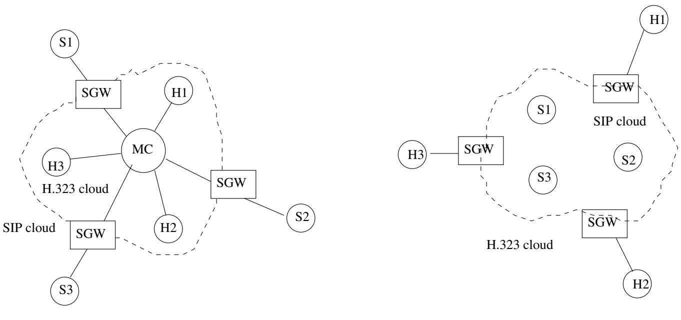
Fig. 4. Ad-hoc conferencing among SIP and H.323 endpoints

A transparent support for multi-party conferencing can be achieved by having the SGW mirror the endpoint(s) in each direction. Fig. 4 shows a scenario in which two H.323 terminals (H1 and H2) and two SIP user agents (S1 and S2) are involved in a conference. From the H.323 side, the signaling gateway (SGW1) looks like a single H.323 terminal. From the SIP side, the signaling gateway acts as a single SIP user agent.

This approach fails if S1 invites another H.323 user H3 via a different signaling gateway (SGW2). How will the other participants such as H2 know that H3 has joined the conference? Alternatively, if H1 invites a SIP user, S3, S2 will not know of the presence of S3. One way for the participants to know about the existence of the other participants is to rely on the RTP/RTCP packets. This goes against the idea of H.323 conferencing where H.245 messages are used to convey the existence of new participants.

We can solve this problem by forcing all invitations to pass through the SGW. Fig. 5(a) shows a conference managed by an MC where H.323 terminals are directly connected to the MC and SIP user agents are connected through signaling gateways. A SIP user agent is allowed to only invite other SIP UAs through the SGW, so that the

SGW can update the MC state. In a SIP-centric architecture, Fig. 5(b), the H.323 terminals take part in the conference through the signaling gateways.



(a) H.323 centered conference

(b) SIP centered conference

Fig. 5. Different conferencing architectures

We recommend a SIP-centered architecture because the SIP conferencing model is more general, allowing full mesh with distributed control or centralized bridged conferences. In general, translating services is greatly simplified if an operator adopts a primary signaling protocol, with services offered only in that protocol. Terminals using another protocol are restricted to making calls through the SGW.

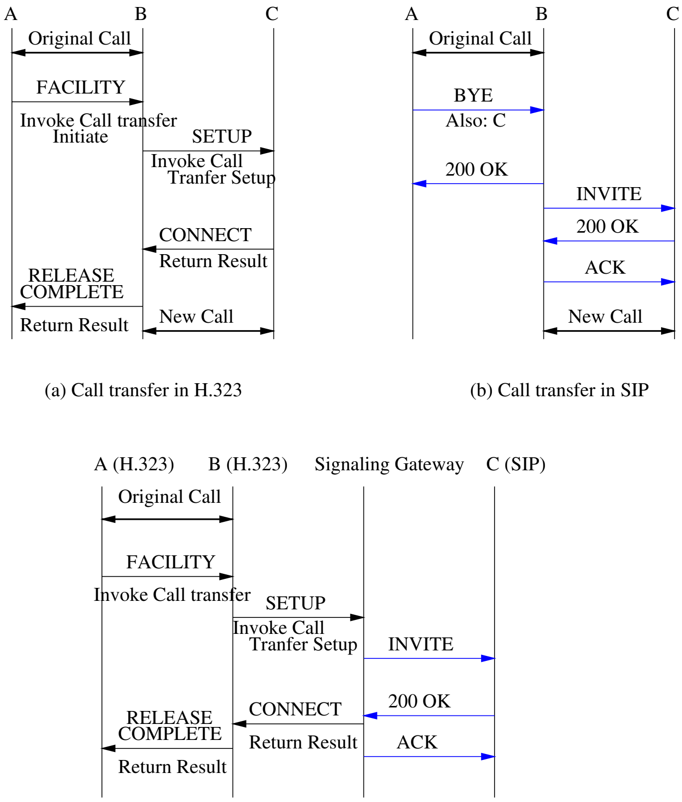
Supporting H.332 loosely coupled conferences is straightforward, since SDP is used in that context.

B. *Call transfer*

Call transfer is one of the many supplementary services needed for internet telephony. The idea is to transfer a call between two entities (say, A and B) to a call between B and C. Fig. 6 shows the message sequence in H.323 and SIP and a possible translation when A and B are H.323 terminals and C is a SIP user agent.

A difference between SIP and H.323 arises because of the different philosophies of protocol extension. H.323 designers identify a supplementary service such as call transfer, call forwarding, call hold and define a new set of messages to accomplish it. This results in different procedures for different advanced services (e.g., H.450.2 for call transfer, H.450.3 for call diversion, H.450.4 for call hold). In SIP, crucial information needed for call services is identified and is encapsulated in new message headers (e.g., Also, Replaces, Requested-By). Different call services are then designed using these building blocks.

A number of open issues remain when translating advanced services, including whether all call parameters can be translated and how security and authentication are to be handled.



(a) Call transfer in H.323

(b) Call transfer in SIP

(c) Call transfer in mixed network. A and B are H.323 terminals and C is a SIP user agent.

Fig. 6. An example of call transfer mapping

VII. Related work

The problem of interworking between SIP and H.323 has only recently started to attract attention, with ETSI TIPHON and ITU now likely to get involved.

Details of the SIP-H.323 interworking described here can be found in [27]. Agboh [28] and Kausar and Crowcroft [29] address the problem of interworking, but do not solve the issues of registration and media capability translation.

VIII. Conclusion and future work

We have described a framework for interworking between SIP and H.323. The challenges include call sequence mapping, address translation and mapping session descriptions.

Ad-hoc conferencing among SIP and H.323 participants is not possible without modifying one or both of these protocols. The problem can be made tractable by keeping an SGW aware of all call state changes.

H.323 has picked up a number of features from SIP, such as Fast Connect or, more recently, UDP-based signaling. It is possible that further convergence may occur, although not without fundamental changes to either SIP or H.323.

We have implemented a basic signaling gateway using the OpenH323 library and a SIP signaling stack developed locally and demontrated a simple audio call setup between SIP user agents and Microsoft NetMeeting.

We have yet to address the issue of multistage translation, where two H.323 users communicate via a SIP gateway. It is not yet clear how common such a scenario would be, given direct network connectivity between the two parties.

IX. Acknowledgments

We would like to thank the members of the `sip-h323` mailing list (sip-h323@eGroups.com) for their comments.

References

[1] M. Handley, H. Schulzrinne, E. Schooler, and J. Rosenberg, "SIP: session initiation protocol," Request for Comments (Proposed Standard) 2543, Internet Engineering Task Force, Mar. 1999.

[2] Henning Schulzrinne and Jonathan Rosenberg, "Internet telephony: Architecture and protocols – an IETF perspective," *Computer Networks and ISDN Systems*, vol. 31, no. 3, pp. 237–255, Feb. 1999.

[3] M. Handley and V. Jacobson, "SDP: session description protocol," Request for Comments (Proposed Standard) 2327, Internet Engineering Task Force, Apr. 1998.

[4] International Telecommunication Union, "Packet based multimedia communication systems," Recommendation H.323, Telecommunication Standardization Sector of ITU, Geneva, Switzerland, Feb. 1998.

[5] James Toga and Joerg Ott, "ITU-T standardization activities for interactive multimedia communications on packet-based networks: H.323 and related recommendations," *Computer Networks and ISDN Systems*, vol. 31, no. 3, pp. 205–223, Feb. 1999.

[6] International Telecommunication Union, "Narrow-band visual telephone systems and terminal equipment," Recommendation H.320, Telecommunication Standardization Sector of ITU, Geneva, Switzerland, May 1999.

[7] International Telecommunication Union, "Terminal for low bit-rate multimedia communication," Recommendation H.324, Telecommunication Standardization Sector of ITU, Geneva, Switzerland, Feb. 1998.

[8] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, and T. Berners-Lee, "Hypertext transfer protocol – HTTP/1.1," Request for Comments (Draft Standard) 2616, Internet Engineering Task Force, June 1999.

[9] H. Schulzrinne, A. Rao, and R. Lanphier, "Real time streaming protocol (RTSP)," Request for Comments (Proposed Standard) 2326, Internet Engineering Task Force, Apr. 1998.

[10] Henning Schulzrinne and Jonathan Rosenberg, "A comparison of SIP and H.323 for internet telephony," in *Proc. International Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV)*, Cambridge, England, July 1998, pp. 83–86.

[11] Ismail Dalgic and Hanlin Fang, "Comparison of H.323 and SIP for IP telephony signaling," in *Proc. of Photonics East*, Boston, Massachusetts, Sept. 1999, SPIE.

[12] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RTP: a transport protocol for real-time applications," Request for Comments (Proposed Standard) 1889, Internet Engineering Task Force, Jan. 1996.

[13] P. Cordell, "Conversational multimedia URLs," Internet Draft, Internet Engineering Task Force, Dec. 1997, Work in progress.

[14] International Telecommunication Union, "Media stream packetization and synchronization on non-guaranteed quality of service LANs," Recommendation H.225.0, Telecommunication Standardization Sector of ITU, Geneva, Switzerland, Nov. 1996.

[15] International Telecommunication Union, "Control protocol for multimedia communication," Recommendation H.245, Telecommunication Standardization Sector of ITU, Geneva, Switzerland, Feb. 1998.

[16] International Telecommunication Union, "H.323 extended for loosely coupled conferences," Recommendation H.332, Telecommunication Standardization Sector of ITU, Geneva, Switzerland, Sept. 1998.

[17] International Telecommunication Union, "Security and encryption for H-Series (H.323 and other H.245-based) multimedia terminals," Recommendation H.235, Telecommunication Standardization Sector of ITU, Geneva, Switzerland, Feb. 1998.

[18] International Telecommunication Union, "Interworking of h-series multimedia terminals with H-Series multimedia terminals and voice/voiceband terminals on GSTN and ISDN," Recommendation H.246, Telecommunication Standardization Sector of ITU, Geneva, Switzerland, Feb. 1998.

[19] International Telecommunication Union, "Generic functional protocol for the support of supplementary services in h.323," Recommendation H.450.1, Telecommunication Standardization Sector of ITU, Geneva, Switzerland, Feb. 1998.

[20] International Telecommunication Union, "Call transfer supplementary service for H.323," Recommendation H.450.2, Telecommunication Standardization Sector of ITU, Geneva, Switzerland, Feb. 1998.

[21] International Telecommunication Union, "Call diversion supplementary service for H.323," Recommendation H.450.3, Telecommunication Standardization Sector of ITU, Geneva, Switzerland, Sept. 1997.

[22] International Telecommunication Union, "Digital subscriber signalling system no. 1 (dss 1) - isdn user-network interface layer 3 specification for basic call control," Recommendation Q.931, Telecommunication Standardization Sector of ITU, Geneva, Switzerland, Mar. 1993.

[23] H. Schulzrinne and J. Rosenberg, "SIP call control services," Internet Draft, Internet Engineering Task Force, June 1999, Work in progress.

[24] Henning Schulzrinne and Jonathan Rosenberg, "Signaling for internet telephony," Technical Report CUCS-005-98, Columbia University, New York, New York, Feb. 1998.

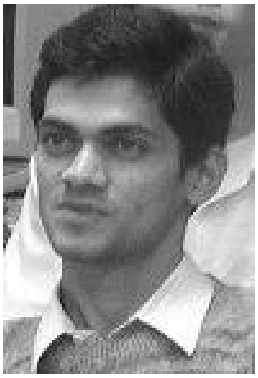
[25] Olivier Hersent, David Gurle, and Jean-Pierre Petit, *IP telephony*, Addison Wesley, Reading, Massachusetts, 2000.

[26] A. Vaha-Sipila, "URLs for telephone calls," Internet Draft, Internet Engineering Task Force, Dec. 1999, Work in progress.

[27] K. Singh and H. Schulzrinne, "Interworking between SIP/SDP and H.323," Internet Draft, Internet Engineering Task Force, Jan. 2000, Work in progress.

[28] Charles Agboh, "A study of two main ip telephony signaling protocols: H.323 signaling and sip; a comparison and a signaling gateway specification," M.S. thesis, Unversite Libre de Bruxelles (ULB), Facuts des Science, Dpartment Informatique, Brussels, Belgium, 1999, supervised by Eric Manie.

[29] Nadia Kausar and Jon Crowcroft, "An architecture of conference control functions," in *Proc. of Photonics East*, Boston, Massachusetts, Sept. 1999, SPIE.



Kundan N. Singh received a B.E.(Hons) degree in Computer Science from Birla Institute of Technology and Science in India and is con-tinuing his studies towards an M.S. degree in the same field at Columbia University in New York City. As a research assistant in the Internet Real-time Lab at Columbia University, he is doing research on internet telephony, SIP-H.323 signaling gateway and unified messaging systems.

Henning G. Schulzrinne received a B.S. degree from the Darmstadt University of Technology in Germany, an M.S. degree from the University of Cincinnati in Ohio, and a Ph.D. from the University of Massachusetts in Amherst, all in electrical engineering. An associate professor of computer science and electrical engineering at Columbia University in New York City, Dr. Schulzrinne's research interests include internet telephony, internet multimedia control and transport and performance evaluation.

Interworking Between SIP/SDP and H.323

Henning Schulzrinne
Dept. of Computer Science
Columbia University
New York, USA
hgs@cs.columbia.edu

SIP vs H.323

SIP

- Text based request response
- SDP (media types and media transport address)
- Server roles: registrar, proxy, redirect

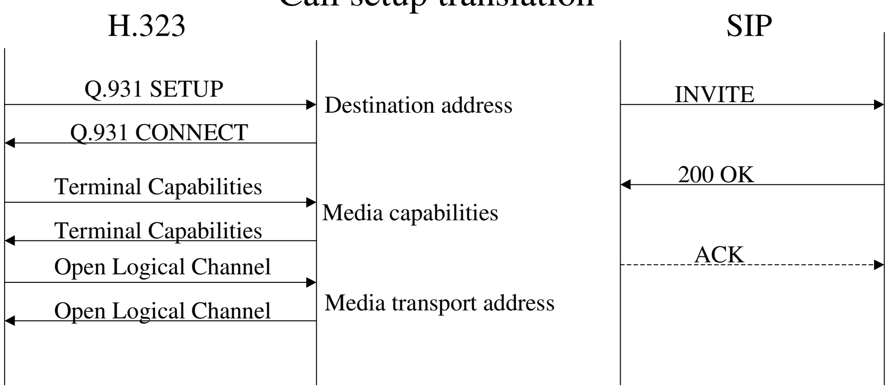
H.323

- Binary ASN.1 PER encoding
- Sub-protocols: H.245, H.225 (Q.931, RAS, RTP/RTCP), H.450.x...
- H.323 Gatekeeper

Both use RTP/RTCP over UDP/IP

Interworking Problems

Call setup translation



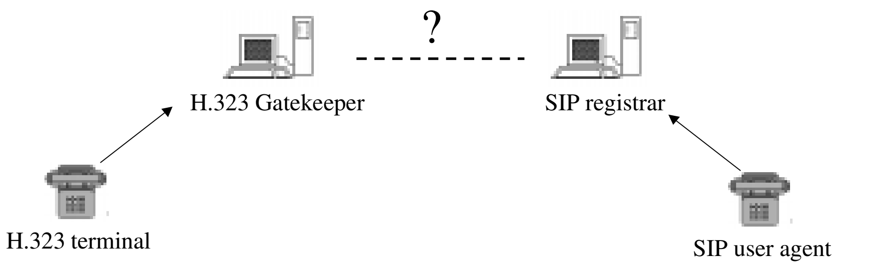
- Multi-stage dialing
- H.323v2 Fast-start is optional

Interworking Problems

User Registration

H.323

SIP



- Location independent user identifier ?
- Use information from both networks

Interworking Problems

Media Description

H.323/H.245

Supports inter-media constraints

{ [G.711 Mu law, G.711 A law][H.261 video]} { [G.723.1] [no video] }

SIP/SDP

List of alternative set of algorithms.

audio G.711 Mu law, G.723.1, G.728
video H.261

- Translation in both directions
- Algorithm selection by end-systems

Interworking Problems

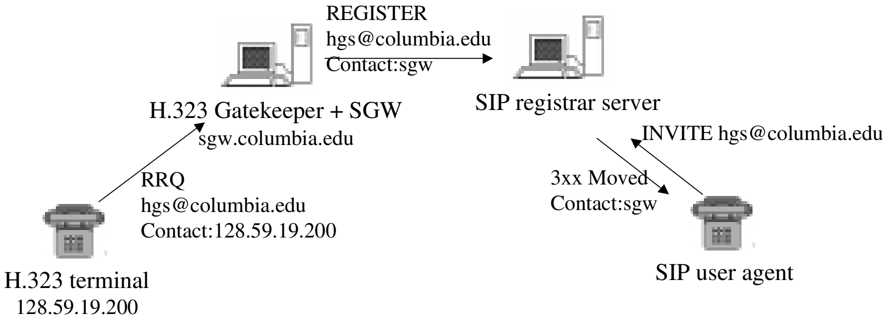
Advanced Services

- H.323 Conferencing: centralized signaling control, MC (Multi-point Controller)
- Supplementary services: H.450.x

- SIP Conferencing: centralized bridged + decentralized distributed
- New headers : Also, Requested-By, Replaces

User registration

- Registration info to foreign network
- Three ways: SGW + GK, SGW + proxy/registrar, SGW

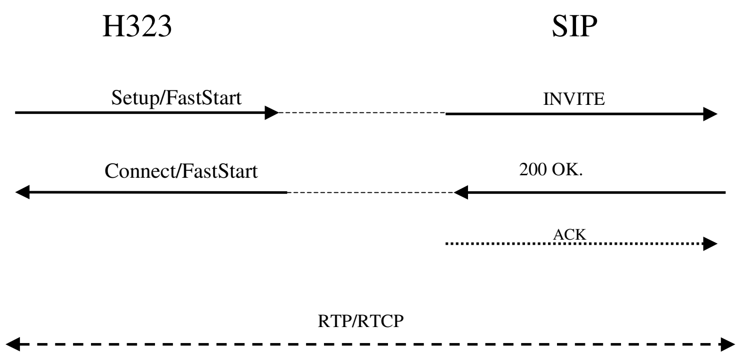


- Independent SGW preferable - use SIP OPTIONS and H.323 LRQ

Call Setup

with H.323v2 Fast Start

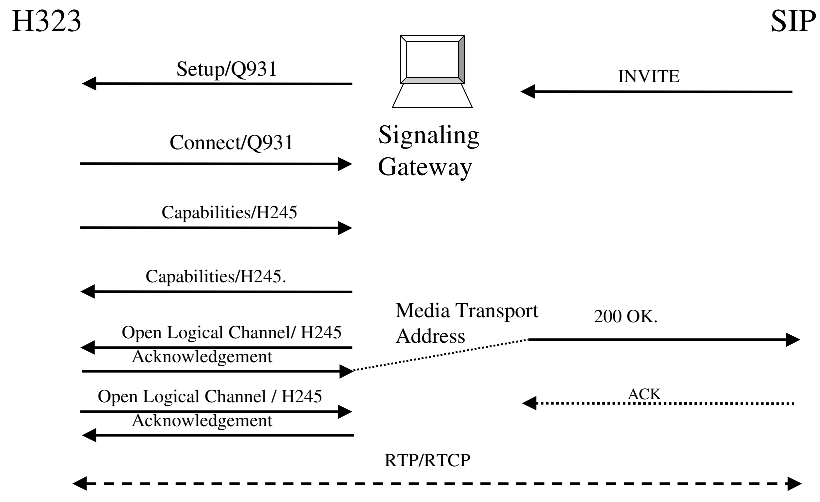
One-to-one mapping between SIP and H.323 messages.



Reverse direction is similar

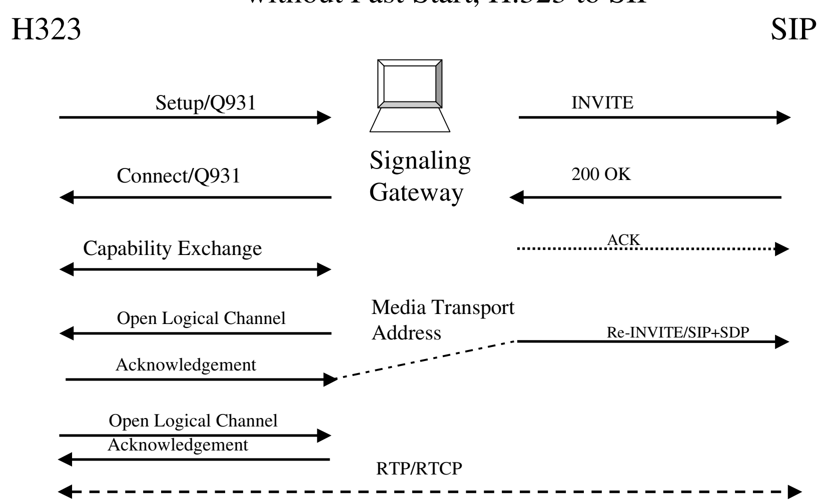
Call Setup

without Fast Start, SIP to H.323



Call Setup

without Fast Start, H.323 to SIP



Capability Set

- Capability set in each direction
- Maximal intersection and current operating modes
- Re-INVITE or change in H.323 mode or logical channels, whenever it changes

Example:
C1 = { [PCMU, PCMA, G.723.1] [H.261] }
C2 = { [PCMU, PCMA, G.729] [H.261] }
C1 $\cap$ C2= {[PCMU, PCMA] [H.261]}
operating modes = [audio=PCMU,video=H.261]

Conclusion and Future Work

- Ad-hoc conferencing
- SIP centered or H.323 centered conferencing
- Basic call setup $\Rightarrow$ other supplementary services

- Our demonstration setup (openh323+Columbia stack) for basic audio call
- IETF, ITU and ETSI TIPHON
- Convergence between SIP and H.323 in newer versions

Business Applications and Products

Mehr als kostengünstige Telefonie - Kundenorientierte integrierte Sprache & IP Kommunikationslösungen

Laura Liess

T-Nova, Technologiezentrum Darmstadt

e-mail: laura.liess@telekom.de

1. Einführung

Die ersten IP-Telefonie-Standards und Produkte sind jetzt schon ein paar Jahre alt. In dieser Zeit haben ISPs und klassische Netzbetreiber verschiedene IP-Telefonie-Piloten aufgesetzt, manche von diesen nutzen die IP-Telefonie für Ferngespräche.

Trotzdem ist der vorausgesagte rasante Massenerfolg ausgeblieben. Es gibt dafür einige wichtige Gründe:

1. Die einzige treibende Kraft für die Produktentwicklung und für die Piloten der Netzbetreiber waren Kosteneinsparungen durch "Trunk- Replacement". Allerdings sind die Preise für PSTN Sprachverbindungen inzwischen drastisch gesunken, so dass Investitionen in eine neue Technologie für die reine Übertragung von Sprache nur bedingt und kurzfristig wirtschaftlich sein könnten.

2. Endkunden, insbesondere Geschäftskunden, erwarten einen Sprachdienst in gewohnter PSTN Qualität, die Verfügbarkeit der üblichen ISDN-Leistungsmerkmale, **aber insbesondere neue Dienste.** Die IP-Telefonie-Technologie, die heute auf dem Markt verfügbar ist, ist noch nicht in der Lage, diese Requirements zu erfüllen.

3. Der PC mit Kopfhörer, bis jetzt das Endgerät für Internet-Telefonie, ist aus zwei Gründen nur bedingt dafür geeignet:

   - Die Kunden ziehen das herkömmliche "allways-on"- Telefon dem PC vor
   - Die Sprachcodierung in der Client-Software ist meistens zu langsam und verursacht zu hohe Verzögerungszeiten.

2. Die Rolle der Geschäftskunden und die wachsende Bedeutung der Value Added und Advanced Services

Der wichtigste Grund für die Integration der Sprache in IP- Netze ist die Möglichkeit der Integration von Sprache mit Computer-Applikationen. Die offene Natur der IP-Netze und die Verschiebung der Intelligenz aus dem Netz an die Peripherie (Endgeräte und Server) ermöglicht eine klare Trennung zwischen Transport und Diensten. Diese Trennung wird neue Geschäftsmodelle mit sich bringen und wird neue Dienste und eine bessere Kundenorientierung ermöglichen.

Aufgrund der hohen Kosteneinsparungen, die durch eine Erhöhung der Effizienz der Geschäftsprozesse erreicht werden können, sind die Geschäftskunden an Diensten interessiert, die ihnen dazu verhelfen.

Firmen benutzen heute für Sprachkommunikation PBX-basierte Corporate Networks. Die PBXe waren lange Zeit sehr nützlich und CTI schafft einen gewissen Grad an Integration mit Computerapplikationen. Allerdings verfügen PBX-basierte Corporate Networks nicht über weltweite Remote access Möglichkeiten und viele nützliche PBX-Leistungsmerkmale sind zwischen verschiedenen Standorten nicht verfügbar, da in der Praxis Q.Sig herstellerspezifisch ist. Ausserdem sind die Möglichkeiten, die Behandlung ankommender Anrufen individuell zu gestallten, sehr eingeschränkt.

Auf dem Geschäftskundenmarkt werden Firmen in Zukunft ein einziges IP-Netz haben anstelle von separaten Netzen für Daten und Sprache. PBXe werden nicht mehr notwendig sein, da IP-Telefone direkt an das IP-Netz angeschlossen sein werden.

In der nächsten Zeit werden Firmen Kommunikationssysteme brauchen, die über verschiedene Arten von Endgeräten zugänglich sind:

Telefone, Computer, Laptops, Mobiltelefone, Pager, Palmtops. Die verschiedenen Teilnehmer in dem Businessprozess eines Unternehmens, Entwickler, Management, Vertrieb, Kunden und Reseller brauchen Zugang zu dem Unternehmensnetz, unabhängig davon, wo sie sich gerade befinden. Dienste dürfen nicht auf einzelne Länder oder Standorte begrenzt sein. Weltweite Erreichbarkeit unter einer Kennung und Mobilität werden immer wichtiger. Allerdings müssen auch die Möglichkeiten geschaffen werden, dass die Menschen bestimmen können, wann, von wem und wie sie erreicht werden sollen.

Business Kommunikation braucht Lösungen, die Sprache, Computer und Internet Applikationen nahtlos integrieren. Diese Lösungen müssen allen Teilnehmern an dem Businessprozess eines Unternehmens Dienste wie Sprache, weltweit verfügbare PBX-like Leistungsmerkmale, E-mail, Unified Messaging, Multimedia Konferenzen, Application-Sharing, WWW, Directories, E-Commerce, Multimediakatalogen, Instant Messaging and Presence zur verfügung stellen. Spiele, Video-on Demand, Mobilität aber auch neue, kundenspezifische Dienste. Weltweite verteilte Web Call Centers müssen fähig sein, intelligentes, kriterienbasiertes Call Routing nach Expertise, Verantwortung und Tageszeit, als auch Dienste wie "click-to-call-back" und virtual customer care zu unterstützen.

Aus diesen Gründen werden Geschäftskunden in dem Fortschritt der IP-Telefonie die treibende Rolle spielen.

3. IP-Centrex als Platform für neue Kommunikations- Dienste, das zukünftige Geschäftsmodell der Netzbetreiber

Für die klassischen Netzbetreiber ist die Integration von Sprache in die IP- Netze eine gute Chance für neue Einnahmequellen. Wegen des raschen Preisverfalles für reine Voice-Übertragungsdienste werden die Gewinnmöglichkeiten mit dem klassischen Telefoniedienst in Zukunft weiter schrumpfen.
Value- Added integrierte Voice/Internet-Dienste für Businesskunden bieten die Möglichkeit an, neue Geschäftsmodelle zu gestalten, bei denen Kunden und Netzbetreiber profitieren.

In dem PSTN-Modell sind die Dienste und die Intelligenz im Netz angesiedelt, in den Vermittlungsystemen oder in den Intelligent-Networks. Die nächste Generation der Kommunikationsdienste wird auf dem Internet-Modell basieren, d.h. die Intelligenz ist teilweise in den Kundenendgeräten und teilweise in Server angesiedelt. Server gehören aus technischer Sicht zur Peripherie, allerdings aus der Sicht der Dienste sind Server Teil des Netzes.

Die Kombination zwischen API-basierter Programmierung und offenen Standards vereinfacht die Entwicklung, das Testen und die Modifikation von Diensten. Service Provider werden mehr und schneller neue Dienste auf den Markt bringen, da die Verluste bei einem eventuellen Misserfolg des Dienstes erheblich reduziert werden. Ausserdem werden Dienste wegen der niedrigen Investitionen bei einer viel niedrigeren Marktpenetration als früher als erfolgreich gelten. Aus diesem Grunde werden Netzbetreiber auch Spezial- und Nischenmärkte adressieren können.

Die Netzbetreiber werden in Zukunft von eigenen geschützten Servern, die sogenannte IP-Centrex Plattform, neue integrierte Dienste anbieten. In Kombination mit IP-VPNs, die sicheres Remote Access zum Intranet und Corporate Soft-PBX für Aussendienst- und Heimarbeiter, Lieferanten, Kunden möglich machen, ist IP-Centrex ist eine Chance für neue Businessmodelle, die Netzbetreiber wahrnehmen sollten. Sonst riskieren sie, die 20% der Top-Kunden zu verlieren, die 80% des Gewinnes bringen.

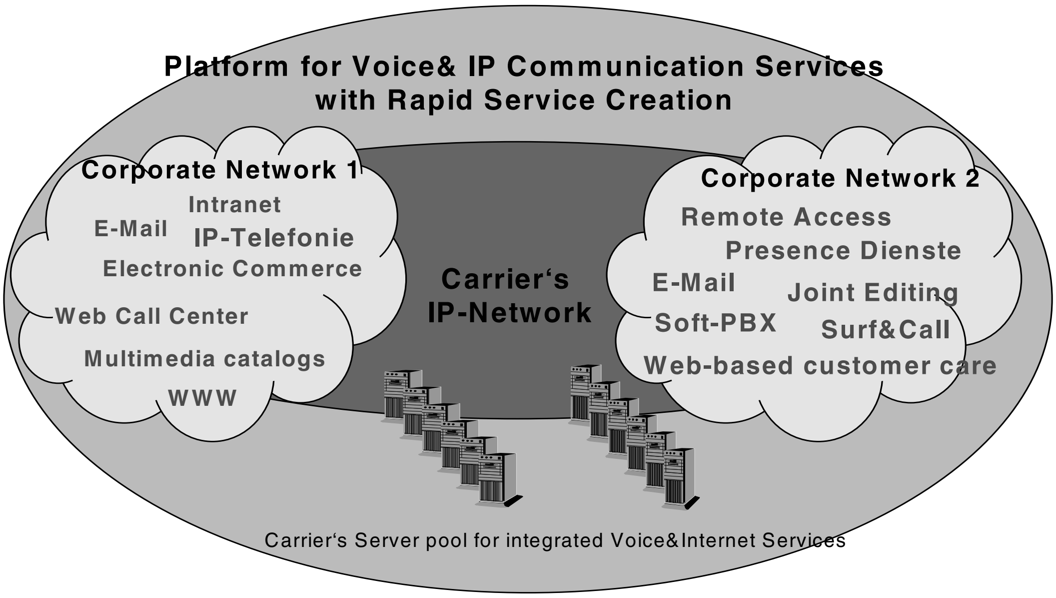


Abbildung 1: IP-Centrex

4. DIE IETF PROTOKOLLE FÜR VOICE UND MULTIMEDIA OVER IP ALS TECHNISCHE GRUNDLAGE FÜR IP-CENTREX UND NEUE DIENSTE

Es ist schwer vorherzusagen, welche „die integrierten Sprache-/Datendienste” der Zukunft sein werden. Die erfolgreichsten werden wahrscheinlich in einem „try and miss”- Prozess entstehen. Für den Erfolg ist aber maßgebend, daß diese Dienste kostengünstig, schnell, flexibel und gut skalierbar implementiert werden können.

Das bedeutet , daß für das Voice-over-IP Call Control eine Technologie verwendet werden sollte, die möglichst einfach und nahtlos in die Internet-Protokollwelt passt und leicht zu implementieren ist.

Die IETF Protokolle für Multimedia Conferencing und Telefonie wurden unter Berücksichtigung dieser Requirements entwickelt und sind damit eine ideale Ausgangsbasis für neue integrierte Sprache-/Datendienste. Das Session Initiation Protocol (SIP), verwendet für Aufbau, Management und Abbau von interaktiven Kommunikations Sessions, inklusive Voice Sessions, ist ein Internet-freundliches Protokoll, dem HTTP-Protokoll (für WWW verwendet) sehr ähnlich. SIP verwendet die im Internet übliche URL-Adressierung, daher können schon im Netz verfügbare Mechanismen wie DNS und LDAP benutzt werden. Zusammen mit dem Text-basierten Session Description Protocol (SDP) erlaubt SIP eine nahtlose Integration mit klassischen Internet Diensten wie WWW, E-Mail or Chats.

Die meisten klassischen ISDN/IN Dienste können mit SIP schnell und einfach implementiert werden (“Implementing Intelligent Network Services with the Session Initiation Protocol” von Schulzrinne, Lennox, La Porta), ISUP/Q.Sig Informationen können einfach als MIME in SIP Nachrichten transportiert werden.

SIP ist ein Internet Application Layer Protokoll der über eine sehr hohe Flexibilität verfügt, so dass es nicht nur für Call Control, sondern auch für anderre Dienste wie Mobility (draft-itsumo-hmmp-00.txt) oder Instant Messaging and Presence.

Besonders wichtig ist die Integrationfähigkeit mit E-Commerce. Diese ist dadurch gewährleistet, daß E-Commerce ebenfalls auf ein HTTP-ähnliches Protokoll basiert, das Internet OpenTrading Protocol (IOTP), das von IETF und dem Open Trading Protocol Consortium gemeinsam entwickelt wurde.

Das SIP-Protokoll entwickelt sich z. Zt. sehr schnell, getrieben insbesondere von Netzbetreibern die in der neuen Technologie die Chance für neue Dienste und damit neue Ein-

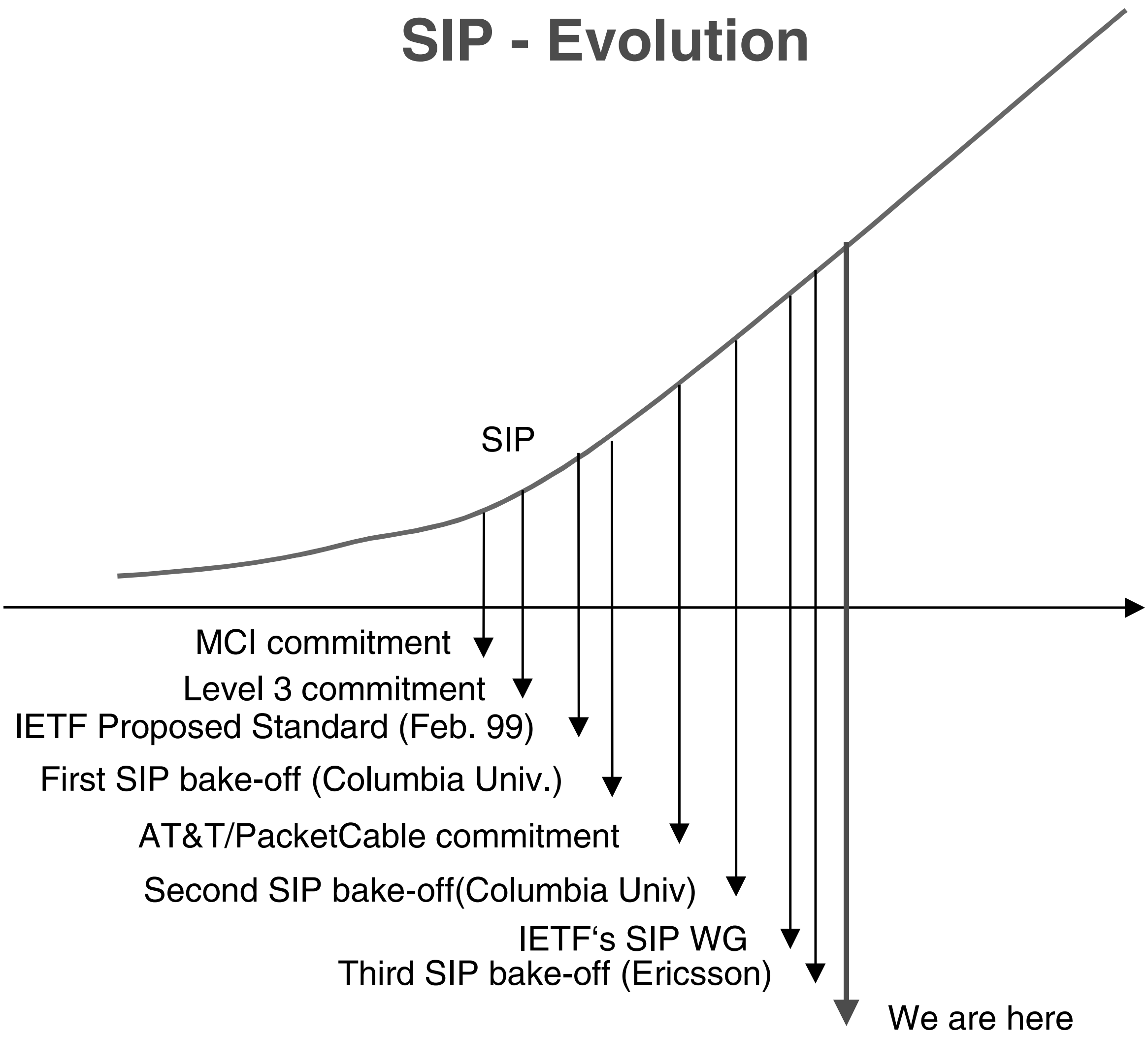


Abbildung 2: SIP-Evolution

nahmequellen sehen: MCIWorldcom, AT&T, Level3 Communication.

Die Abbildung zeigt die schnelle SIP Entwicklung und die wichtigsten Ereignisse diesbezüglich.

Anbei einige Dienste die basierend auf SIP Telefonie einfach zu realisierbar sind:

- problemloses Zusammenschalten von Standorten weltweit (z.B. nach Firmenkäufe), ohne Verlust von Leistungsmerkmalen wie aus der PBX-Welt bekannt
- weltweitverteilte Sprach- und Multimedia-Call-Center mit Kriterien (Zuständigkeit, Expertise, Tageszeit)-basiertem Call-Routing zu Händlern, Experten, Operators
- Mobilitätsdienste
- Instant Messaging and Presence mit integrierter Sprache
- WWW mit integrierter Sprache
- E-Commerce mit integrierter Sprache

Das Gebiet der neuen Dienste basierend auf SIP ist z.Zt. noch wenig untersucht worden, aber viele Netzbetreiber haben das Potential schon erkannt.

5. Netzbetreiber Requirements und Hersteller Aktivitäten

Anders als ISPs, die PC2Phone und PC2PC-Dienste Internetnutzern schon in einem frühem Technologiestadium über das Best-Effort Internet zu niedrigen Preisen angeboten haben, sind Netzbetreiber an kommerziellen Voice-over-IP Dienste und hauptsächlich auf

Geschäftskunden (inklusive SOHO-Kunden), fokussiert.
Aus diesem Grunde haben Netzbetreiber besonders hohe Anforderungen an die VoIP-Technologie. Im Rahmen der IETF-Working-Groups und der Industriekonsortien wie Softswitch Consortium und der Distributed-Call-Signalling-Groups, entwickeln Netzbetreiber und Hersteller die Protokolle und Netzarchitekturen für Carrier-Class Realtime Services, inklusive Voice over IP. In diesem Zusammenhang ist das Thema Quality of Service (QoS) / Class of Service (CoS) von besonderer Bedeutung.

Ein kommerzielles Sprachdienst kann, unabhängig von der verwendeten Technologie, nicht ohne Quality of Service (QoS) angeboten werden.

QoS für IP-Telefonie hat zwei Aspekte:

1. **QoS für die Mediaübertragung**

Anders als in leitungsvermittelten Netzen, wo das Prinzip "alles oder nichts" gilt, sind in IP-Netzen verschiedene Class of Service (CoS) möglich. RSVP wird im Accessbereich benutzt, um die notwendige/gewünschte Bandbreite zu reservieren. Die RSVP-Requests werden in den Edge-Routers an den Common Open Policy Service (COPS) übergeben. Das Common Open Policy Service, bestehend aus dem Policy Server (Policy Decision Point) und dem Policy Enforcement Point (co-located mit dem Edge Router) prüft, ob die Requests berechtigt sind. Der Edge-Router agiert als Agregator und übergibt die RSVP-Requests in die DiffServ-Flows des Backbone-Netzes.

Viele Netzbetreiber werden in nächster Zeit QoS/CoS Support in die IP-Netze implementieren und den Geschäftskunden über Virtual Private Networks (IP-VPNs) Premium Class Services zur Verfügung stellen. Damit wird ein IP-Telefonie-Dienst mit der Qualität des traditionellen PSTN-Sprach-dienstes möglich sein.

2. **QoS Signalling**

Für IP-Telefonie mit QoS sind eine schnelle Übertragung der Signallisierung und eine Koordination zwischen dem IP-Telefonie-Signallisierungsprotokoll und den Protokollen für Ressource Management unerlässlich. Es ist noch unklar, ob diese Koordination so erfolgen muss, dass die IP-Telefonie eine eins-zu-eins Abbildung des PSTN-Sprachdienstes sein muss. Ein Beispiel ist die Ressourcenverfügbarkeit wenn das Telefon klingelt, wie es im PSTN der Fall ist.

Der Distributed Call Signaling Vorschlag für QoS Signalling geht von einer eins-zu-eins Abbildung des PSTN-Sprachdienstes in die IP-Netze aus. Allerdings hat dieses Modell den Nachteil, dass wesentlich mehr Nachrichten notwendig sind als mit dem reinrassigen SIP.

Ausser QoS sind folgende Aspekte von besonderer Bedeutung für die Netzbetreiber:

- Accounting/ Charging/Billing
- Security
- Skalierbarkeit
- Interdomain-Interoperabilität
- Support für PBX/IN-ähnliche Dienste und personal mobility
- Integrierte Sprache/ Internet-Dienste
- Rapid Service Creation
- User Acces zu Service Logic (WG IPTEL)
- Interoperabilität zu PSTN/ISDN/IN (WG PINT, SPIRITS, MeGaCo und SigTran)

Entsprechende Mechanismen befinden sich z.Zt. noch in der Entwicklung, einige im Rahmen der IETF SIP Working Groups. Allerdings ist es sehr wichtig, daß auch nach den entsprechenden Erweiterungen die Protokolle ihre ursprüngliche Einfachkeit nicht verlieren.

Trotz der schnellen Fortschritte bezüglich der Protokolle sind auf Gesamtkonzeptebene viele Zusammenhänge noch unklar. Daher sind Beiträge wie die der DCS-Gruppe oder draft-sinnreich-interdomain-sip-qos-osp-00.txt, die einen Überblick über die in einem Carrier Grade IP Telefoniedienst involvierten Funktionen und deren Zusammenspiel geben, von großer Bedeutung.

Einige Hersteller haben auf die Veränderungen in der IP-Telefonie Landschaft schon reagiert. Es sind inzwischen die ersten SIP-Produkte am Markt verfügbar, Vorversionen werden z. Zt. von den Netzbetreibern getestet.

Ein wichtiger Aspekt ist hier die Interoperabiliät zwischen den Produkten verschiedener

Hersteller. Schnelle Einführung neuer, innovativer Dienste ist nur dann möglich, wenn die Netzbetreiber herstellerunabhängig sind und nach dem Internet-Modell agieren können. Das bedeutet, dass neue Komponenten (HW und SW) problemlos in das vorhandene Netz eingebunden werden können.

Ein wichtiger Schritt in diese Richtung war die SIP-Bake/Off Gründung. Im Rahmen der SIP-Bake/Off's haben die Entwickler die Möglichkeit, SIP-Interoperabilitätstests durchzuführen. Das letzte SIP-Bake/Off in Richardson hat mit 33 Testteams die Erwartungen übertroffen.

Mehr als kostengünstige Telephonie

Kundenorientierte integrierte Sprache- & IP-Kommunikationslösungen basierend auf SIP

T-Nova

Technologiezentrum
Laura Liess
laura.liess@telekom.de

Inhalt

- Bisherige Entwicklung in der IP Telephonie
- Die wachsende Bedeutung der Value Added Services für Kunden und Netzbetreiber
- Die IETF Protokolle für Voice und Multimedia over IP als technische Grundlage für neue Businessmodelle und Dienste
- Netzbetreiber Requirements und derzeitige Aktivitäten

T-Nova

Technologiezentrum
Laura Liess
laura.liess@telekom.de

1

Bisherige Entwicklung der IP-Telephonie

- „Trunk-Replacement" war der bisherige Businesscase, der allerdings wegen den stark sinkenden Preise für Übertragungskapazität nur bedingt Investitionen in einer neuen Technologie rechtfertigt.

- Geschäftskunden erwarten QoS, die gewohnten PBX Leistungsmerkmale und insbesondere neue Dienste. Dies ist mit den heute am Markt verfügbaren Produkten nur eingeschränkt möglich.

T-Nova

Technologiezentrum
Laura Liess
laura.liess@telekom.de

Was bedeutet IP-Telephonie

- Transport der Sprache und Daten über ein Netz
- Offene Internet Protokolle für alle Dienste
- Ein einziges Managementsystem
- Integration der Sprache mit Computer und Internet Applikationen
- Verschiebung der Intelligenz aus dem Netz in die Peripherie (Endgeräte und Server)
- Trennung zwischen Transport und Dienste

⇒

- Kosteneinsparungen
- neue Kommunikations-dienste
- Interoperabilität
- Rapid Service Creation
- neue Geschäftsmodelle

T-Nova

Technologiezentrum
Laura Liess
laura.liess@telekom.de

2

Die Rolle der Geschäftskunden für die Integration der Sprache- und Internet Diensten

- Effiziente Unternehmenskommunikation bedeutet effiziente Geschäftsprozesse
- Neue, integrierte Dienste können die Effizienz der Kommunikation in Unternehmen deutlich erhöhen
- Businesskommunikation braucht Lösungen, die
  - Sprache, Computer und Internet Applikationen nahtlos integrieren
  - allen Teilnehmern im Businessprozess zur Verfügung stehen
  - weltweit verfügbar sind
  - auf die Bedürfnisse des Kunden zugeschnitten sind

T-Nova

Technologiezentrum
Laura Liess
laura.liess@telekom.de

IP -Centrex als zukünftiges Geschäftsmodell der klassischen Netzbetreiber

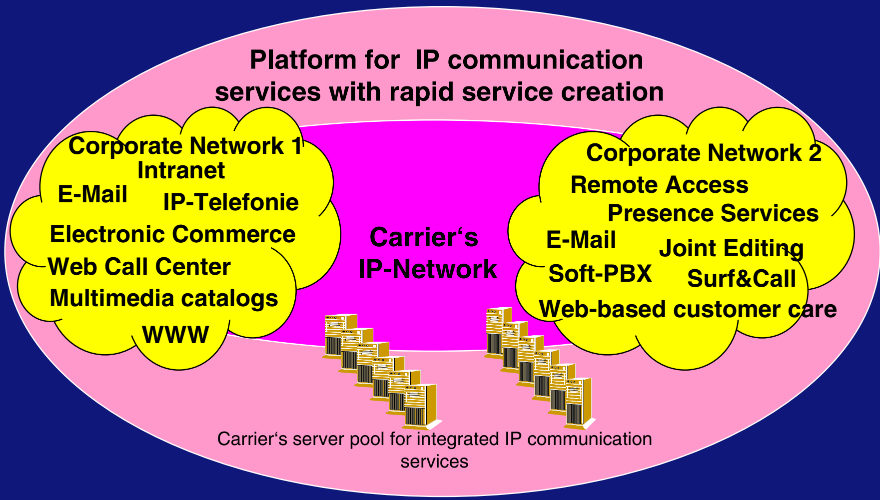
- Die sinkenden Preise für Transport werden dazu führen, dass die Bedeutung von neuen kundenspezifischen Kommunikations- und E-Commerce-Diensten in den Geschäftsmodellen der Netzbetreiber stark zunimmt.
- Offene Internet Standards beschleunigen die Entwicklung, das Testen und die Modifikation solcher Dienste
- Neue Dienste werden oft kurzlebig sein und deren Einführung kann auch nach dem „try and error" Prinzip erfolgen
- Die Netzbetreiber können von eigenen geschützten Servern IP-Centrex Dienste anbieten
- IP-Centrex über IP-VPN's ermöglicht weltweite Verfügbarkeit aller Dienste

T-Nova

Technologiezentrum
Laura Liess
laura.liess@telekom.de

3

Netzbetreiber Plattform für integrierte Sprach- & Internet Dienste (IP-Centrex)



T-Nova

Technologiezentrum
Laura Liess
laura.liess@telekom.de

Erfolgsfaktoren für das IP-Centrex Geschäftsmodell

- Skalierbarkeit
- Kosteneffizienz
- Rapid Service Creation
- Flexibilität

T-Nova

Technologiezentrum
Laura Liess
laura.liess@telekom.de

4

Die IETF Protokolle für IP Telephonie und Multimedia Conferencing (SIP, SDP, RTSP) als technische Grundlage für neue Dienste (1)

- Textbasiert
- HTTP-ähnlich aufgebaut
- URL's für die Adressierung
- Nahtlose Integration mit klassischen Internet Protokollen wie SMTP und HTTP
- Nahtlose Integration mit IOTP (Internet Open Trading Protocol) für E-Commerce
- Sehr flexibel (z.B. das Header-Konzept)
- Gut skalierbar

T-Nova

Technologiezentrum
Laura Liess
laura.liess@telekom.de

Die IETF Protokolle für IP Telephonie und Multimedia Conferencing (SIP, SDP, RTSP) als technische Grundlage für neue Dienste (2)

- Einsetzbar auch für andere Dienste wie
  - **globale Mobilität**
    (Schulzrinne und Wedlund „Mobility Support using SIP", Internet-Drafts: draft-itsumo-hmmp-00 und draft-itsumo-sip-mobility-req-01)
  - **Instant Messaging and Presence**
- Die meisten klassischen PBX/IN Dienste können mit SIP problemlos implementiert werden
  (Schulzrinne, Lennox, La Porta „Implementing Intelligent Network Services with the Session Initiation Protocol")
- Ermöglichen kundendefinierte Dienste
  (Call Processing Language)

T-Nova

Technologiezentrum
Laura Liess
laura.liess@telekom.de

5

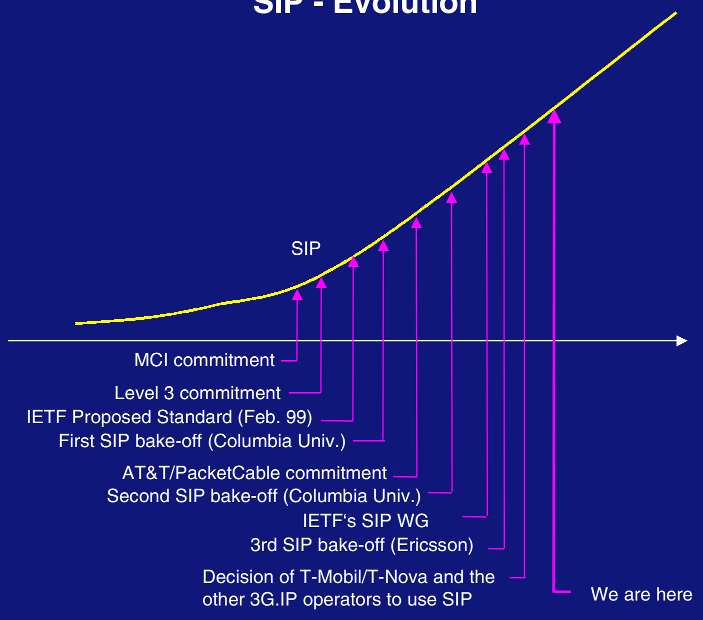
Beispiele

- Zusammenschalten von Standorten weltweit ohne Verlust von Leistungsmerkmalen, wie aus der ISDN-Welt bekannt (ISUP ist ETSI/ANSI/länderspezifisch, Q.Sig ist hersteller-spezifisch)
- weltweit verteilte Sprach- und Multimedia Call Center mit kriterienbasiertem Call Routing (Zuständigkeit, Expertise, Tageszeit)
- Userdefiniertes Call Routing, abhängig von persönlichen Kriterien (z.B. Einträge im Kalender)
- Number Portability
- Instant Messaging mit integrierter Sprache

T-Nova

Technologiezentrum
Laura Liess
laura.liess@telekom.de

SIP - Evolution



T-Nova

Technologiezentrum
Laura Liess
laura.liess@telekom.de

6

Anforderungen an Carrier Grade IP-Telephonie Netze und laufende Aktivitäten

- **QoS (Media und Signaling)**
  - PacketCable's (DCS) Ansatz
  - MCIWorldCom's Ansatz
  - Ansatz für Mobile Networks ?
- **Accounting, Charging, Billing** (IETF AAA, GMD-Fokus, Herstellerkonzepte)
- **Security** (SIP WG)
- **Call Flows für Supplementary Services** (SIP WG)
- **Interworking mit PSTN** (INFO-Method, SIP-T, PINT, SPIRITS, MGCP, ENUM)
- **Interdomain Interoperability** (draft-sinnreich-interdomain-sip-qos-osp-01)
- **Call Flows für 3G Mobile Networks** (Columbia University, ITSUMO-Group, 3G.IP..)

T-Nova

Technologiezentrum
Laura Liess
laura.liess@telekom.de

Predicting Internet Telephony Call Setup Delay

Tony Eyers
TITR
University of Wollongong
Australia
tony@elec.uow.edu.au

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

Abstract—**Internet telephony has been the focus of much recent effort by ITU and IETF standards bodies, with initial, albeit small-scale deployment in progress. While Internet telephony voice quality has been studied, call setup delay has received little attention. This paper outlines a simulation study of Internet Telephony Call Setup delay, based on UDP delay/loss traces. The focus is signaling transport delay, and the variations arising from packet loss and associated retransmissions. Of particular interest are the differences arising from H.323 signaling, which uses TCP, and SIP, which can use UDP with additional error recovery. Results show that during high error periods, H.323 call setup delay significantly exceeds that of SIP. We also consider PSTN/Internet telephony interworking, and show that high blocking rates are likely if either H.323 or SIP are used across the public Internet.**

I. Introduction

Internet telephony is experiencing significant growth, prompted initially by low-price long distance calls [1]. Longer term growth will be motivated by the greater service flexibility offered by IP networks, compared to the Public Switched Telephone Network (PSTN) [2]. This flexibility arises from the increased signaling capability of IP end systems compared to current handsets, as well as the ability to support multiple media types. To be widely accepted however, Internet telephony QOS must match or exceed that of the PSTN. From a user's perspective, the quality of service consists of the reliability and amount of time of setting up the call and then the audio and video quality of the actual call. While the latter aspect has received considerable attention, experience has shown that Internet telephony call setup times can be much longer than the essentially instantaneous call setups that have become routine for the PSTN. This paper attempts to predict call setup delays over the public Internet.

Over the 120 year history of telephony, signaling performance has improved markedly. For example, Gherardi and Jewett [3] note that call setup time dropped from 4 minutes (!) in 1923 to 1.2 minutes in 1928. Duffy and Mercer [4] reports that in 1978, the average time between end of dialing and ringback was about 10.9 s. In 1998, AT&T [5] claimed a call setup time of less than two seconds for toll calls, and 2.5 seconds for calls requiring database lookups. Since Internet telephony signaling uses the same high-speed backbone links as for data while most SS7 systems are still connected by 64 kb/s links, call setup delay could be significantly less for Internet telephony. We will explore this in detail below.

Call setup delay (also known as post-dialing delay or post-selection delay [6]) is defined as the interval between entering the last dialed digit and receiving ringback. Another, related, measure is the time between entering the last dialed digit and when the callee's phone starts to ring. We will refer to this delay as the *dial-to-ring delay*, as there does not seem to be a standard designation. In a traditional phone system, there is no acoustic feedback between dialing and ringback, so that an excessive delay until ringback may lead the caller to believe that "something is wrong" and abandon the call. Internet telephony has the advantage that it can provide additional feedback during call setup, before ringback. For example, SIP servers can send any number of *provisional responses* that indicate the progress of address translations or other network actions, as discussed in Section III. E.721 [6] recommends an average delay of no more than 3.0, 5.0 or 8.0 s, for local, toll and international calls, respectively. The 95th percentiles are set at 6.0, 8.0 and 11.0 s, respectively.

The importance of the dial-to-ring delay depends on the type of call. For example, if a fax machine is "blast-faxing" to a number of receivers, the call setup time becomes an important component of the achievable throughput. It is similarly important for short data calls like checking email, although the connection setup delay of modem calls appears to be dominated by the modem training time and PPP delays that often take ten seconds or more. For completed calls, that is, about 70% of all calls [4], it takes the callee about on average 8.5 s to pick up the phone, so that reducing the call setup time much below a second probably yields limited improvement for human-to-human calls. Traditional benchmarks for signaling performance cannot distinguish between these different uses of the telephone system, but the distinction may be important if Internet telephony is primarily used for human-to-human contact, with data and fax using other mechanisms.

Another important signaling delay is post-pickup delay (or, more formally, answer-signal delay [6]), which roughly measures the delay between the time the callee picks up the receiver and the time the caller receives indication of this. The actual definition in E.721 only considers the message transfer delay, not any delays incurred in the end systems. If the speech path from callee to caller only gets cut through when this message reaches the caller, the first "hello" of the callee may get lost, leading to confusion. While this paper does not provide measurements for this particular delay, we will describe how it relates to post-dial delay. E.721 [6] recommends average answer-signal delays of 0.75 s for local, 1.5 s for toll and 2.0 s for international connections, with 1.5 s, 3.0 s, and 5.0 s as 95% values.

The design and performance modeling of circuit switched networks has been an active research area for most of the 20th century. In particular, the advent of common channel signaling (SS7) has prompted many performance studies [7]. The result has been a robust PSTN with tightly engineered QOS, particularly with regard to call setup delay. Associated with this is a series of ITU recommendations which specify performance targets for signaling transport [8] and call processing [9]. These underpin the signaling network engineering (e.g., [10]) and switch design needed to ensure call setup delay performance.

The recent arrival of Internet Telephony (ca. 1996) and the much more varied network infrastructure have precluded the same depth of performance study. Some Internet Telephony call setup delay targets have been proposed [11], based on ITU recommendations for the PSTN. However, delay targets for Internet telephony call setup components encompassing both IP signaling transport and server delay are not yet in place. Indeed, it appears unlikely that an Internet standard for signaling delay will emerge, except possibly in connection with ensuring delay targets for SS7 networks. Instead, customers may make signaling delay part of their service level agreement (SLA) with a carrier.

Internet Telephony uses new end-to-end signaling protocols, such as H.323 [12] and SIP [13], with IP networks providing signaling message transport. We present a simulation study of Internet Telephony call setup delay, based on these protocols and Internet delay traces. The purpose is threefold: to determine call setup delays arising from signaling transport within the public Internet, to compare the relative performance of SIP and H.323, and to investigate blocking probabilities arising from PSTN/Internet telephony interworking. A key finding is that TCP error control, used in H.323[1], can significantly increase call setup delay compared to the UDP-based approach commonly used in SIP.

Section II outlines previous Internet telephony performance studies, then reviews ITU recommendations for call setup delay, and their applicability to Internet telephony. Section III presents H.323 and SIP call setup procedures, highlighting the different packet loss recovery techniques. Section IV describes the Internet delay traces used for this study. Section V presents the simulation methodology. Results in section VI compare H.323 and SIP over a variety of paths. Discussion and conclusions are in sections VII and VIII, respectively.

II. Prior Work

A major thrust of Internet telephony research has been protocol development. H.323 [12] and SIP [13], [15], [16] have emerged as the key peer-to-peer call setup protocols, with G.729 and G.723.1 the leading low-bit-rate audio codecs. Current protocol issues include billing, address resolution [2] and resource allocation [17]. QOS arising from the established and developing protocols remains a key issue.

Initial Internet telephony QOS research has considered voice quality arising from deployment of the new codecs over the public Internet. Kostas *et al.* [18] generate UDP trace records over six months, and, from the mean delay and standard deviation, conclude that acceptable voice quality would generally be available over the intra-USA paths considered. Maxemchuk and Lo [19] extend this work by incorporating compensation within the codecs for lost and delayed voice packets. They conclude that acceptable performance is usually available within the USA, but that voice quality on international calls is often poor.

Call setup delay is a key and easily discernable QOS parameter, with multiple components, e.g. dial-to-ring and post-pickup delay (as outlined). These delays comprise processing in transit switches and end systems, and signaling transfer delay. In the PSTN, these delays correspond to ISUP/MTP processing delays and SS7 queueing/propagation delays respectively [20].

To date there appears to have been no Internet telephony call setup studies incorporating signaling transfer delays. Elwalid *et al.* [21] consider processing delays, with a queueing analysis on an H.323-based switch used to determine the intra-server delay, i.e. the call setup delay distribution within the switch. The 99th percentile of this delay is, seemingly arbitrarily, bounded at 1.5 seconds, with

[1] The use of UDP for H.323 signalling transport is discussed in [14]

this bound used to determine the maximum server load. Signaling message transfer delay between switches is not considered. In this paper we take the opposite approach, by modeling the signaling transfer delay for SIP and H.323 messages. The total call setup delay also includes server call processing and the translation between domain names and IP addresses via DNS. The server call processing delay can vary widely, depending on whether the server, for example, makes calls to networked databases or processes per-call scripts [22]. Given the variability of both components, we do not attempt to characterize them here. From our experience, a basic SIP redirection operation takes between 10 and 100 ms, depending on whether an external process in invoked or not.

Lin *et al.* [11] propose Internet telephony call setup delay targets based on the ITU Q.725 targets for the PSTN. The ITU figures for signaling transfer point (STP)[2] message transfer delay, maximum number of signaling hops and ISUP message transfer delay are combined to provide mean and 95th percentile Internet telephony call setup delay targets. The delay targets in [11] and [21], while both pertaining to post-dial delay, are different. In particular, the 99th percentile is specified in [21], even though no source is listed for this figure. While the delay targets in [11] are based on ITU sources, they do not appear to incorporate queueing delay for signaling messages (ITU figures for this are in E.733 [23]). In addition, [11] does not mention ITU recommendation E.721 [6], which provides mean call setup delay targets. However, the figures in E.721 and [11] are similar. It appears therefore that firm guidelines for Internet Telephony call setup delay are still to be determined, and that Internet Telephony signaling transfer delays have yet to be considered. This latter objective is the focus of this paper.

While ITU delay targets provide a starting point for Internet telephony, there are clear differences. Directly mapping SS7 signaling transport delay and ISDN cross-switch delay to Internet telephony implies that the ratio of these two components is the same for the PSTN and the Internet. Given the difference between SS7 and IP message transport, it seems more appropriate to define specific targets for each. There is another constraint on Internet call setup delay, which applies when interconnecting with ISDN switches. These switches may abandon a call if a reply from a setup attempt (i.e., an IAM signaling message) is not received within two seconds [24]. Hence, for PSTN/Internet interworking, an additional Internet call setup delay target is required, which keeps this loss rate within acceptable bounds.

[2] An SS7 "router"

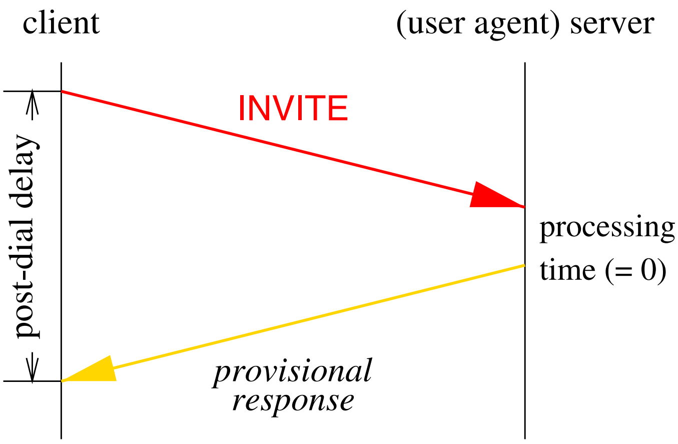


Fig. 1. SIP call setup (initial portion)

III. H.323 and SIP Call Setup

Both H.323 and SIP are *peer-to-peer* signaling protocols, used by Internet telephony end systems to establish multimedia sessions. H.323 and SIP allow a variety of call setup mechanisms, ranging from a single message exchange between caller and callee (e.g., H.323v2 Fast Connect), to more complex calls which traverse a number of servers before reaching their destination. Fundamental to all these call types are provisions for reliable message transfer in the face of packet losses, which, in turn, determine the upper bounds for call setup delay. We outline the H.323 and SIP call setup procedures, focusing on the error recovery techniques.

A. SIP

Figure 1 shows part of a basic SIP call setup. A Client sends an INVITE call setup message to a User Agent Server (callee). Usually, the UAS returns one or more provisional response messages indicating receipt of the INVITE request and call progress. This is roughly equivalent to the ISDN IAM/ACM message exchange, with the delay representing the post-dial delay. This simple call setup, comprising the reliable exchange of an INVITE and provisional response messages (with the post-dial delay shown in the figure), is a key element of our comparative study.

Figure 2 shows a more complex SIP call, where the client first queries a *redirect server*, whose response contains either the address of the final destination or that of another redirect server. The rest of the call continues as in Figure 1, i.e. the INVITE/provisional response message exchange. More complex SIP call types are outlined in [16].

The signaling messages in Figures 1 and 2 are generally sent via UDP, although SIP also supports TCP [13]. An application-level timeout and retransmission scheme

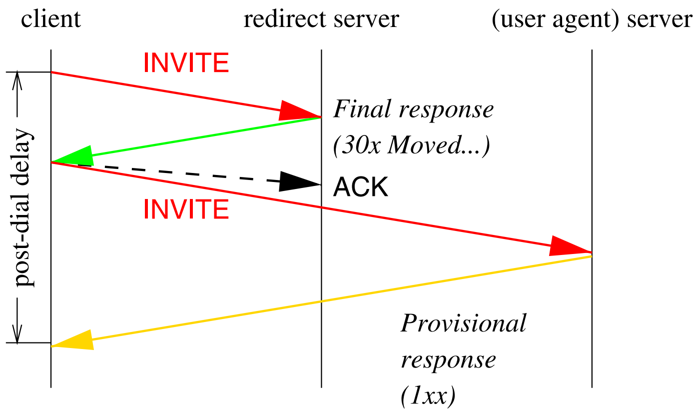


Fig. 2. SIP call setup with redirect server

recovers from UDP errors. An INVITE message is retransmitted until the first provisional response is received, first after 500 ms, then again after one additional second, then two seconds and finally every four seconds[3]. INVITE message transmissions cease after seven attempts. The server simply transmits a provisional response for each INVITE received, without any timers. When the call is answered, redirected or fails for some reason, the UAS transmits a *final response*. The final response is retransmitted with the same spacing as the INVITE, until the caller sends an ACK message. The post-pickup delay is determined by how long it takes for the first final response to reach the caller.

B. H.323

Figure 3 shows the simplest H.323 call setup, the Fast Connect option available in H.323v2. This comprises a TCP connection setup, then a Setup/Connect message exchange. The post dial delay equals the SIP one shown in Figure 1, plus the TCP connection setup time. The additional delay resulting from the TCP connection setup forms a key part of our investigation. An UDP based H.323 call setup option is proposed in [14], however this option is not part of the current H323v.2 standard.

A wide variety of other H.323 scenarios are possible [12], some using a gatekeeper for address resolution, connection admission control and call signaling. A general comparison of H.323 and SIP appears in [25]. For this study, we consider the Fast Connect option only, the aim being to highlight the delay differences between this call type and the corresponding SIP one. A significantly more involved call setup mechanism requiring several TCP connection is specified in the 1998 version of [12].

As these differences arise principally from the use of TCP in H.323, we review briefly TCP connection setup and error control. TCP connection setup requires an exchange of SYN messages, followed by an ACK to complete the three way handshake, as shown in Fig. 3. Data transfer then begins, which, in this case, comprises the H.323 SETUP/CONNECT message exchange. The SYN and SETUP/CONNECT messages time out if not acknowledged. The timeout value increases exponentially, usually by a factor of two, each time a given message is retransmitted. TCP timeout values are generally determined by the measured round trip delay and delay variance. However, the default TCP timeout values apply for new connections, such as those shown here.

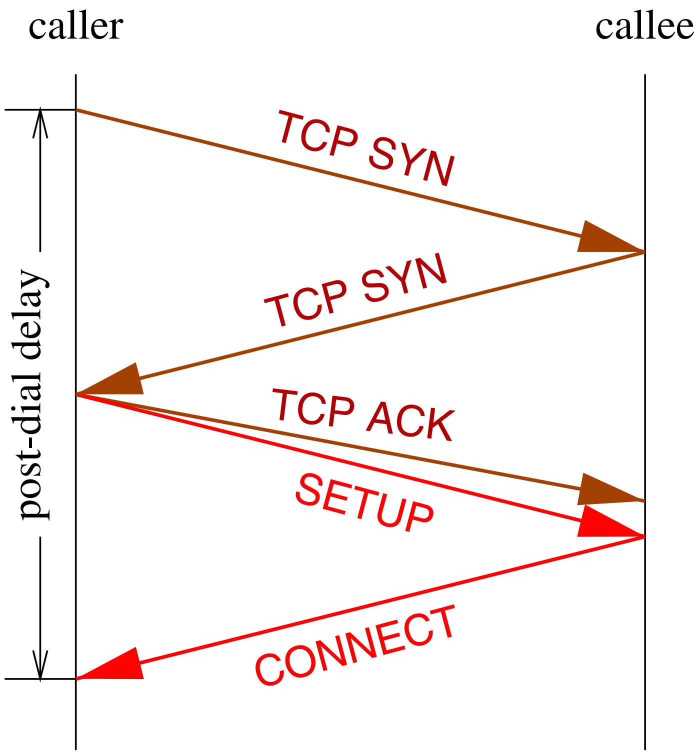


Fig. 3. H.323 Fast Connect call setup

RFC 1122 [26] specifies an initial timeout of three seconds[4], however some implementations start at six seconds [27]. Either value is too high for Internet telephony.

Concerns about the suitability of TCP error control for signaling are raised in [28], which highlights delays arising from TCP timer granularities (generally set to 500 ms). The TCP delayed acknowledgment mechanism, designed to reduce traffic loads, adds more delay [29]. Clearly TCP needs tuning for Internet telephony signaling, with lower initial timeout values, such as the SIP ones, and immediate acknowledgments. Some systems, such as Solaris using `ndd`, allow the system-wide tuning of these parameters.

[3]The value of 500 ms was chosen since it represents a reasonable upper bound for interactive voice communications.

[4]The Solaris operating system, for example, uses a three-second time out.

To avoid biasing the results by operating system settings, we assume these lowered initial timeout values.

In many cases, H.323 uses *gatekeeper-routed* signaling, where signaling is propagated along a chain of TCP connection from client through one or more gatekeepers to the destination. Since the number of such gatekeepers are hard to predict, we ignore them in our model. Also, if these gatekeepers exchange signaling messages regularly, they may already have an existing TCP connection, so that they can avoid the TCP connection setup overhead. Our model would then apply.

IV. Internet Delay Traces

Packet traces have received much attention in recent years. The major focus has been on interarrival distributions, which have displayed long range dependencies which are at odds with traditional traffic models [30]. The scope of trace results has been extended by the IETF IP Performance Measurement Working Group (IPPM), which has developed metrics and techniques for one way delay and loss measurements [31]. These techniques underpin the Surveyor project [32], run by Advanced Networks and Services, which provides the Internet delay and loss statistics used in this paper.

The Surveyor project, which began in 1997, provides continuous monitoring of UDP delay and loss between selected sites. There are currently 38 of these, mostly in the USA, with some in Europe and the Pacific region. UDP packets of 40-byte length are sent at exponentially distributed intervals with a mean of 500 ms [33]. Using GPS receivers for synchronization, the receiver measures the one way delay with 50 μs resolution, while the packet headers allow loss detection. Results are collated at Advanced Networks, with delay and loss histograms available at the Web site. We have used the individual trace results for our simulations, which essentially provide a delay sample or loss indication every 500 ms.

The Surveyor database provides far more extensive trace measurements than the internally generated ones used in other Internet Telephony studies, e.g. [18]. The wide choice of routes available from the Surveyor Project allow extensive experimentation, using real network data. In particular, the results in section 6 are based on many different Surveyor routes.

V. Simulation Methodology

The simulation estimates the call setup delay distribution experienced by SIP and H.323 system operating over the public Internet. The SIP and H.323 call setup message exchanges outlined previously are modelled, using the Internet delay and loss figures gathered by the Surveyor project. Unfortunately, we cannot simply map signaling requests to a corresponding Surveyor sample, since the spacing of the Surveyor samples is not uniform, with additional gaps due to packet losses. We approximate the network delay behavior by assuming that the instantaneous UDP delay is the one experienced by the most recent Surveyor sample corresponding to the simulated transmission time of the SIP or H.323 request or response. If this sample was lost, then the most recent delay sample before that is used.

The simulation aims to capture the effect of UDP burst errors. A two-state error model is used, which operates as follows: The number of UDP packet losses in the last 200 samples (E200) and the last 20 samples (E20) is recorded. This corresponds to the previous 100 seconds and 10 seconds, respectively. If the number of errors in the last 20 samples is zero or one, then a “good” error state is assumed. The UDP error probability used in the simulation is E200/200, the mean error rate over the previous 100 seconds. Otherwise, a “bad” error state is assumed, where the error probability is E20/20, the mean error rate over the last 10 seconds.

While two-state error models are commonly used, the heuristic presented here and the values chosen are arbitrary. The results which follow test the sensitivity of this error model.

The simulation uses these extrapolated UDP delay and loss statistics to determine the respective H.323 and SIP call setup delay distributions. We assume that the one way delay obtained from the UDP statistics can be applied to TCP packet transfers. The H.323 results consider the Fast Connect call setup shown in Fig. 3. The SIP results encompass a simple call setup (as in Fig. 1), and a redirect server interaction, followed by a simple call setup, as in Fig. 2). For the SIP calls, the path between the client and user agent server (UAS) may traverse multiple stateless proxies.

Delay distributions are generated as if calls were made over one hour, on a specified day, according to the Surveyor data. A one hour delay distribution is based on around six million simulated calls.

VI. Results

The scope of the Surveyor database allows the gathering of results over extended periods, providing insight into average and worst case performance of H.323 and SIP. In particular, we highlight TCP delays in H.323, and investigate delay increases arising from more complex SIP calls which incorporate redirect servers and intermediate proxies.

Similar to circuit switched network engineering, we identify an Internet telephony “busy hour”. While a de-

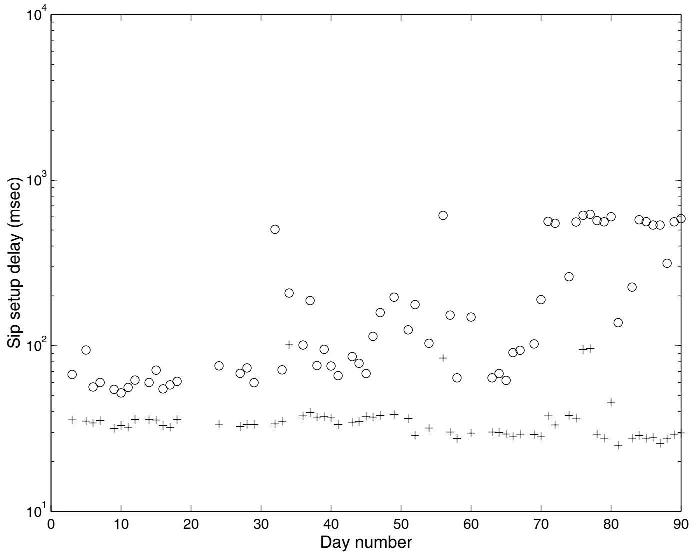


Fig. 4. Minimum and 95th percentile SIP setup delay, New York → Boston

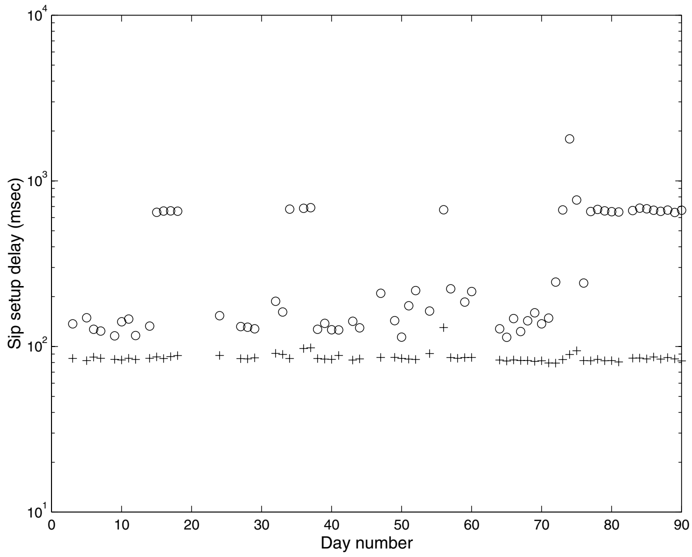


Fig. 6. Minimum and 95th percentile SIP setup delay, New York → West Coast

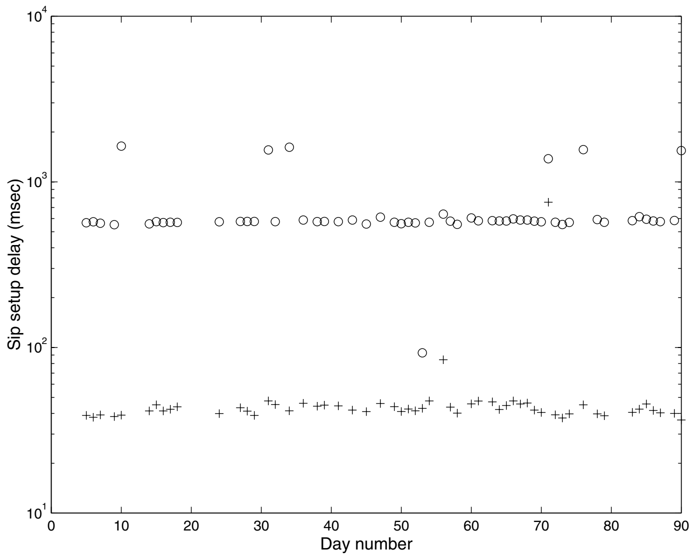


Fig. 5. Minimum and 95th percentile SIP setup delay, New York → Chicago

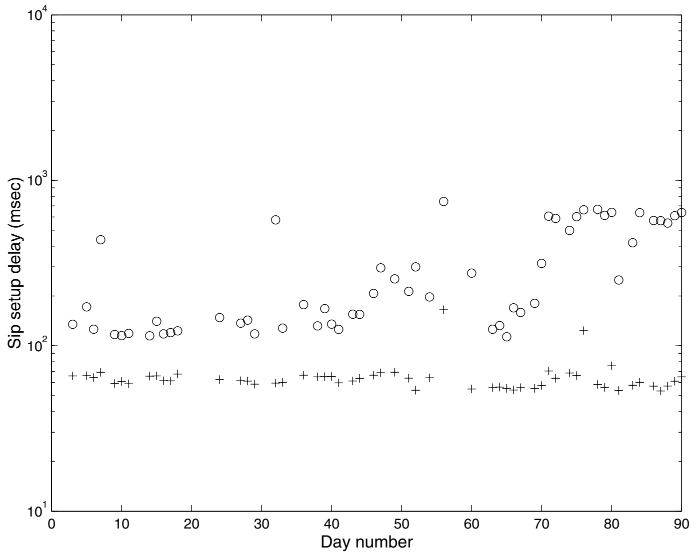


Fig. 7. Minimum and 95th percentile SIP setup delay, New York → Boston, redirected in Washington, D.C.

tailed analysis of Internet busy hours is beyond the scope of this paper, the Surveyor trace histograms show 16:00 hours (Eastern Time) to be a reasonable busy hour choice. The results here cover the first 90 business days of 1999, and consider one hour each day, starting at 16:00 hours.

We investigate three paths within the USA. All originate at the Advanced Networks headquarters in New York, and extend to either Boston (Harvard University), Chicago (University of Chicago) or the West Coast (NASA AMES near Sunnyvale, California). They are 306, 1158 and 4128 km away from New York, with one-way propagation delays of 1.5, 5.8 and 20.6 ms, respectively.

We consider the following scenarios:

- A one-hop SIP call setup over each of the three paths, that is, an INVITE/provisional response exchange between the source and destination (Fig. 4 to 6).
- A SIP call over each path, which first queries a redirect server at George Washington University, in Washington, D.C. (328 km from New York), then exchanges an INVITE/provisional response, as before (Fig. 7 to 9).
- Fig. 10 extends the SIP call in Figure 9, by passing the INVITE/provisional response messages through a stateless proxy in Boston.
- Fig. 11 to Fig. 12 consider a one-hop H.323 call setup ("fast connect"), and represent the H.323 equivalent of the SIP calls in figures 4 to 6. The difference is the H.323 TCP

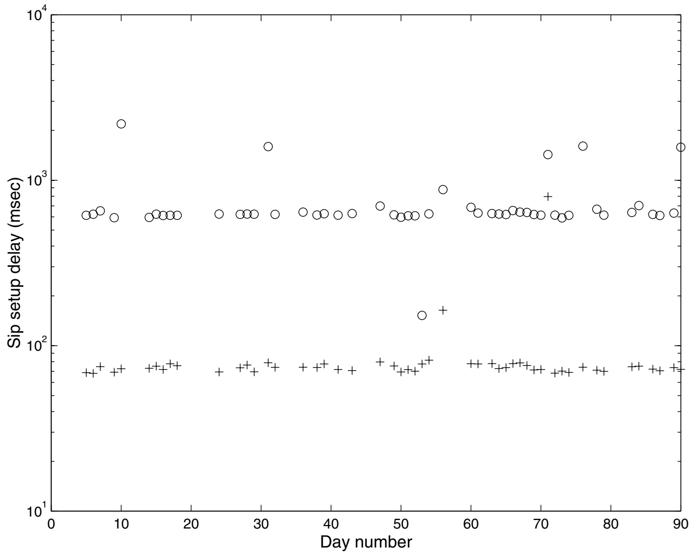


Fig. 8. Minimum and 95th percentile SIP setup delay, New York → Chicago, redirected in Washington, D.C.

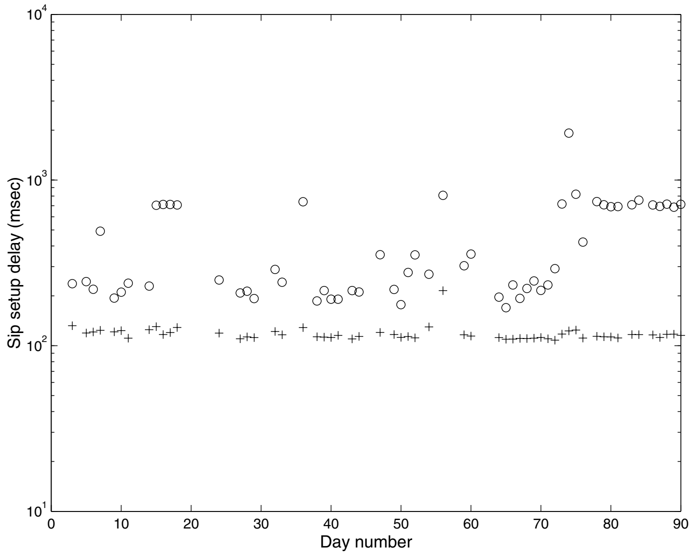


Fig. 9. Minimum and 95th percentile SIP setup delay, New York → West Coast, redirected in Washington, D.C.

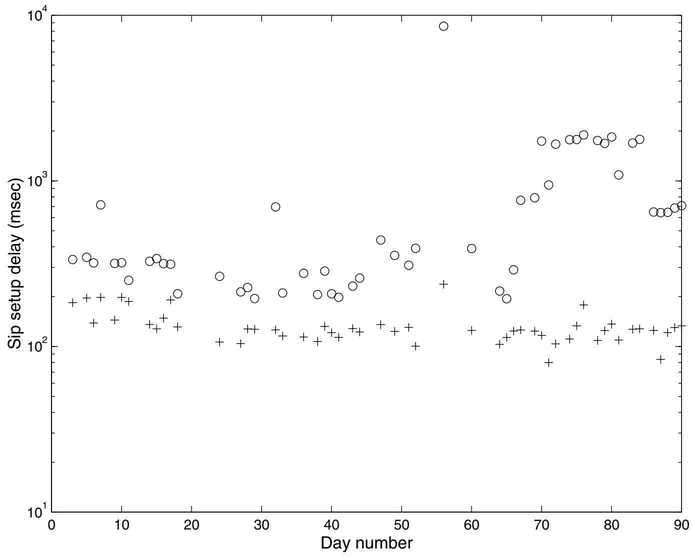


Fig. 10. Minimum and 95th percentile SIP setup delay, New York → West Coast via Boston, redirected in Washington, D.C.

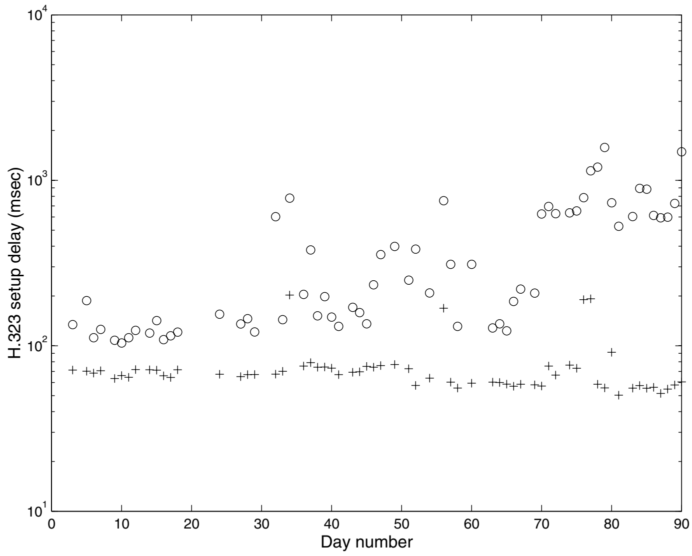


Fig. 11. Minimum and 95th percentile H.323 setup delay, New York → Boston

connection establishment, shown in Figure 3. The TCP timeout values used in the simulation are the same as the SIP ones and thus do not represent the much longer values likely to be encountered by "stock" operating systems.

The plots show, for each day, the minimum call setup delay (a "+") and the 95th delay percentile (an "o").

A key aim of this study has been to investigate the effect of UDP burst errors, as captured by the Surveyor traces. As indicated, the simulation maintains two error windows, which indicate "good" and "bad" error states. As mentioned earlier, the default window sizes are 200 samples and 20 samples respectively. The sensitivity to these parameters is tested in figures 14 and 15, which repeat the New York/Chicago SIP call from Fig. 5. Figure 14 keeps the "good" window at 200 samples, but reduces the "bad" window size to four samples ($\equiv$ 2 seconds). Fig. 15 measures the mean UDP error rate over the entire hour, and uses that value for all calls, ignoring error correlation.

Section II indicated a hard call setup delay limit of 2 seconds for PSTN/Internet telephony interworking since ISDN switches abandon call attempts which exceed this limit. If we require that no more than 1% of such calls fail during the busy hour, we need to ensure that the 99th delay percentile is below 2 seconds. As the simulation results represent signalling delays only, we assume a con-

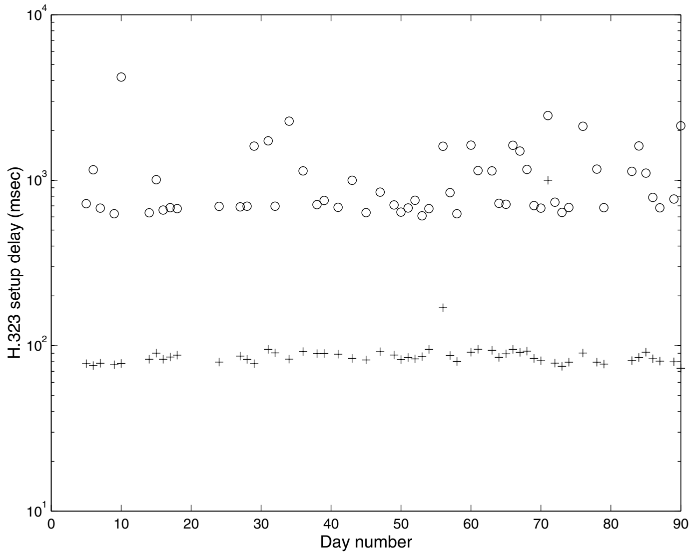


Fig. 12. Minimum and 95th percentile H.323 setup delay, New York → Chicago

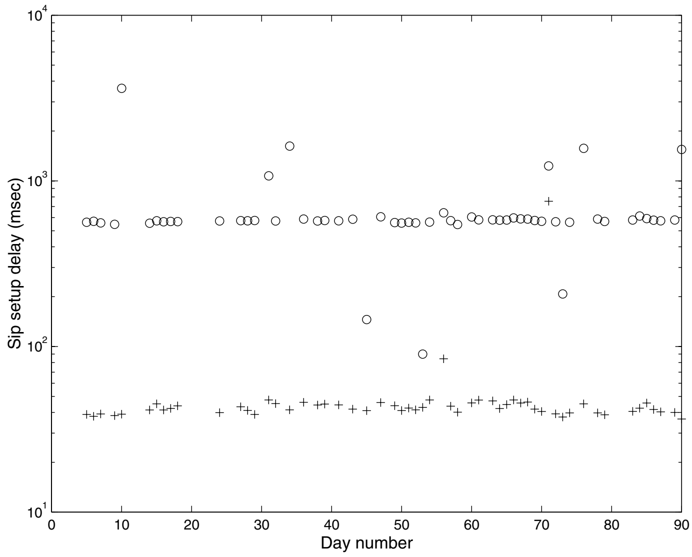


Fig. 14. Minimum and 95th percentile SIP call setup delay, New York → Chicago, with error window of 4 samples

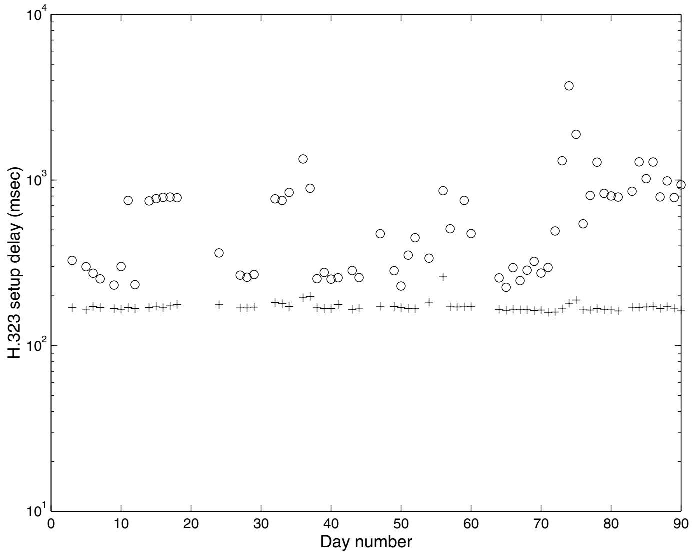


Fig. 13. Minimum and 95th percentile H.323 setup delay, New York → West Coast

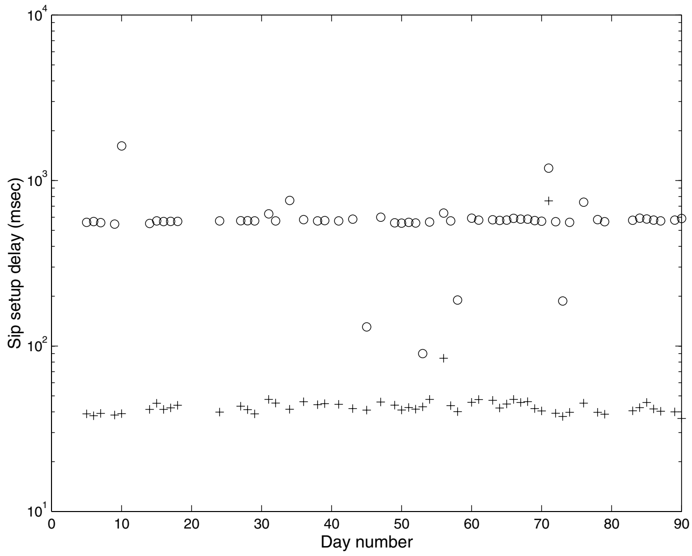


Fig. 15. Minimum and 95th percentile SIP call setup delay, New York → Chicago, with average error

stant delay of 500 ms for other call processing tasks such as DNS lookups, resulting in a delay budget of 1.5 s. Table I shows the percentage of days for which this limit is not achieved, for one-hop SIP and H.323 calls between the destinations shown. (The H.323 figures are shown below the corresponding SIP figure.)

VII. Discussion

Due to the high speed links used in these paths (T3 and above), the minimum delay results essentially show the round trip message propagation delay [33]. Figures 4 to 6 show this increasing from around 30 ms for New York - Boston to 40 ms for New York - Chicago, and around 80 ms for New York – West Coast.

The 95th percentile results in Figures 4 to 6 show two causes of increasing delays, namely increased queueing delays and retransmissions. The queueing delay is essentially the difference between the minimum delay, i.e. the propagation delay and the associated 95th percentile. Figures 4 to 6 show this difference to be between 100 and 200 ms.

In Figures 4 to 6, the highest 95th percentile delays are mostly around 600 ms, due to the 500 ms SIP initial timeout, propagation and queueing delays. In Figures 5 and 6, some 95th percentile results are around 1.6 s. Here, two timeouts occur in the same message exchange, a 500 ms timeout, then a 1 s timeout. The Chicago route experiences the most errors, with almost all days showing 95th percentiles of around 600 ms.

Figures 7 to 9 add the SIP redirect server interaction, increasing the number of one-way paths from two to four. This produces a 30 ms increase in the minimum delay. The 95th percentile results due to queueing delay only (i.e., those less than 500 ms) roughly double. This is essentially the convolution of the queueing delays on each path. However, the 95th percentile results due to retransmissions (those above 500 ms) are largely unchanged from those in Figures 4 to 6.

The retransmission timeout delay clearly dominates the 95th percentile results. Figure 10 investigates this delay further, by routing the New York-West Coast messages via Boston. Comparing these results with Figure 9 (i.e., without the Boston leg), we see increases in the minimum delay of around 30 to 50 ms. However the 95th percentile results are worse, with many delays in the range of 1.6 s. This more complex SIP call, while not greatly increasing the minimum delay, worsens the 95th delay percentile, and hence the perceived call setup delay QOS.

Clearly the number of paths traversed during a call setup determines the message loss probability, and hence the delay due to retransmissions. In this context we consider the H.323 results in Figures 11 to 12. These essentially repeat the SIP results in Figures 4 to 6, with an additional message exchange for the TCP connection setup. Hence we see a doubling of the minimum delay. Figures 11 and 12 show the 95th percentile results arising from retransmission timeouts (i.e., those around 600 ms) to be largely the same as the SIP ones in Figures 4 and 6. Fig. 12, the New York–Chicago route, however shows substantially worse 95th percentile results than the corresponding SIP ones in Fig. 5. In particular, the number of days with a 95th percentile of one second or above (indicating multiple timeouts in the message exchanges) increases by a factor of four.

Delay targets for Internet telephony signalling transport have yet to be established, as indicated. However a 95th percentile of less than a second appears reasonable (and lies well within the ranges outlined in [11]). This target is achieved for almost all the simple SIP call setups (Figures4 to 6). The H.323 calls also achieve this limit on some paths. However the TCP delays shown here are "best case", with timeout values less than the default ones, and without the additional delays arising from timer granularity. As Fig. 10 shows, call setup delay targets are less likely to be achieved over the public Internet for more complex SIP and H.323 call types.

While these results indicate delay trends, they do not measure availability. Gaps in the results, particularly in Fig. 10, are due, in part, to a lack of Surveyor results for those days. The simulation also dropped days which exhibited

- a gap of more than 5 seconds between trace files, or
- a gap of more than 60 seconds between trace records.

The aim was to avoid biasing results by including gaps which may have arisen from faults in the measuring equipment, rather than actual path unavailability. Hence the results here apply to "good" days, with continuous path availability. It is possible that the delay results from some of the missing days are far worse than the ones shown here. The next stage of this project will investigate Internet telephony availability.

Figures 14 and 15 test the sensitivity of the error model outlined in Section V. Moving the "bad state" error window from 20 samples (Fig. 5) to four samples (Fig. 14) produces almost no change in the results. Fig. 15 ignores error bursts, by using the mean error rate over the entire hour for all calls. While the Figure 15 results are similar to the Fig. 5 ones, the "bad" error days (i.e., with a 95th percentile around 1.6 seconds) are not detected. Hence, while the two-state error model shows bursty error effects, as desired, the results appear to be insensitive to the window size chosen.

	Bos.	Chi.	West	Wash.	Colorado
New York	20.3	77.2	32.3	9.1	15.4
	28.2	94.7	40.0	20.0	18.5
Boston		1.6	31.5	0.0	5.4
		1.6	31.5	0.0	10.8
Chicago			34.3	5.2	28.6
			34.3	6.9	61.4
West Coast				33.3	45.3
				36.7	57.3
Washington State					6.6
					6.6

TABLE I
PERCENTAGE OF DAYS WHERE PSTN/INTERNET TELEPHONY BLOCKING PROBABILITY EXCEEDS 1%, FOR SIP (TOP ROW) AND H.323 (BOTTOM ROW)

Table I shows that, for PSTN/Internet Telephnoy interworking, reasonable blocking targets (here, 1%) are not likely to be achieved. The majority of the source/destination pairs show many days (more than 20%) when the 1% blocking probability is not reached. The H.323 results, i.e. the lower entry in each box, are generally worse than the SIP ones, due to the TCP connection setup delays. However, neither SIP nor H.323 provide satisfactory performance. Hence, from the perspective of blocking probability, the best-effort Internet appears not well suited for PSTN interworking. For this application, dedicated IP signalling capacity is more appropriate.

VIII. Conclusions

This paper has considered Internet call setup delays, focusing on the delay component arising from signalling transport. While initial Internet telephony call setup delay targets have been proposed elsewhere, individual targets for the signalling component are still needed.

Drawing on delay and loss traces from the public Internet, our simulation study has shown that, for the paths considered, acceptable SIP call setup delay is available for simple call types. More complex SIP calls, which traverse multiple paths, display variable delay performance. Our results show that the TCP connection setup associated with H.323 calls substantially increases call setup delay over errored paths, even after tuning TCP implementations for more rapid retransmission.

If large Internet telephony gateways dominate Internet telephony, the number of signaling paths for each such gateway will likely be small. In those cases, substantially better signaling performance can be achieved if retransmission timers are tuned, based on previous calls, for the round-trip delays to the destination or, if the request inter-arrival rate is less than a third of the SIP time out value, by using TCPs fast retransmit.

While acceptable call setup delay performance is at times available over the public Internet, our results show that unacceptable blocking rates are likely when interconnecting with the PSTN.

Acknowledgements

The authors would like to thank the staff at Advanced Networks and Services, and in particular Sunil Kalindindi, for providing access to the data used in this project.

References

[1] D. Clark, “A taxonomy of internet telephony applications,” in *Proc. of 25th Telecommunications Policy Research Conference*, (Washington, DC), Sept. 1997.

[2] C. A. Polyzois, K. H. Purdy, P.-F. Yang, D. Shrader, H. Sinnreich, F. Mnard, and H. Schulzrinne, “From POTS to PANS – a commentary on the evolution to internet telephony,” *IEEE Network*, Vol. 13, pp. 58–64, May/June 1999.

[3] B. Gherardi and F. B. Jewett, “Telephone communication system of the united states,” *Bell System Technical Journal*, Vol. 9, pp. 1–100, Jan. 1930.

[4] F. P. Duffy and R. A. Mercer, “A study of network performance and customer behavior during-direct-distance-dialing call attempts in the USA,” *Bell System Technical Journal*, Vol. 57, no. 1, pp. 1–33, 1978.

[5] AT&T, “AT&T sets the industry standard for network reliability,” Mar. 1998. http://www.att.com/network/standrd.html.

[6] International Telecommunication Union, “Network grade of service parameters and target values for circuit-switched services in the evolving isdn,” Recommendation E.721, Telecommunication Standardization Sector of ITU, Geneva, Switzerland, May 1999.

[7] V. A. Bolotin, P. J. Kuhn, C. D. Pack, and R. A. Skoog, “Common channel signaling networks: Performance, engineering, protocols and capacity management,” *IEEE Journal on Selected Areas in Communications*, Vol. 12, pp. 377–544, Apr. 1994. Special issue.

[8] International Telecommunication Union, “Message transfer part signalling performance,” Recommendation Q.706, Telecommunication Standardization Sector of ITU, Geneva, Switzerland, Mar. 1993.

[9] International Telecommunication Union, “Signalling performance in the telephone application,” Recommendation Q.725, Telecommunication Standardization Sector of ITU, Geneva, Switzerland, Mar. 1993.

[10] R. A. Skoog, “Engineering common channel signaling networks for ISDN,” in *Twelfth International Teletraffic Congress*, Vol. 2, (Torino), pp. 1–7 (2.4A,), June 1988.

[11] H. Lin, T. Seth, A. Broscius, and C. Huitema, “VoIP signaling performance requirements and expectations,” Internet Draft, Internet Engineering Task Force, June 1999. Work in progress.

[12] International Telecommunication Union, “Visual telephone systems and equipment for local area networks which provide a non-guaranteed quality of service,” Recommendation H.323, Telecommunication Standardization Sector of ITU, Geneva, Switzerland, May 1996.

[13] M. Handley, H. Schulzrinne, E. Schooler, and J. Rosenberg, “SIP: session initiation protocol,” Request for Comments (Proposed Standard) 2543, Internet Engineering Task Force, Mar. 1999.

[14] International Telecommunication Union, “H.323 annex E: call signalling over UDP,” Recommendation H.323E, Telecommunication Standardization Sector of ITU, Geneva, Switzerland, Sept. 1998.

[15] H. Schulzrinne and J. Rosenberg, “Internet telephony: Architecture and protocols – an IETF perspective,” *Computer Networks and ISDN Systems*, Vol. 31, pp. 237–255, Feb. 1999.

[16] H. Schulzrinne and J. Rosenberg, “The session initiation protocol: Providing advanced telephony services across the internet,” *Bell Labs Technical Journal*, Vol. 3, pp. 144–160, October-December 1998.

[17] P. Goyal, A. Greenberg, C. Kalmanek, B. Marshall, P. Mishra, D. Nortz, and K. K. Ramakrishnan, “Integration of call signaling and resource management for ip telephony,” *IEEE Network*, Vol. 13, pp. 24–33, May/June 1999.

[18] T. J. Kostas, M. S. Borella, I. Sidhu, G. M. Schuster, J. Grabiec, and J. Mahler, “Real-time voice over packet-switched networks,” *IEEE Network*, Vol. 12, pp. 18–27, Jan. 1998.

[19] N. F. Maxemchuk and S. Lo, “Measurement and interpretation of voice traffic on the internet,” in *Conference Record of the International Conference on Communications (ICC)*, (Montreal, Canada), June 1997.

[20] International Telecommunication Union, “Telephone network and ISDN quality of service, network management and traffic engineering,” Recommendation E.723, Telecommunication Standardization Sector of ITU, Geneva, Switzerland, 1992.

[21] A. I. Elwalid, G. G. Freundlich, P. M. Gerhardt, H. Hagirahim, K. G. Ramakrishnan, and D. Tse, “An overview of the multime-

dia communications exchange (mmcx) and its performance characterization," *Bell Labs Technical Journal*, Vol. 2, Spring 1997.

[22] J. Rosenberg, J. Lennox, and H. Schulzrinne, "Programming internet telephony services," Technical Report CUCS-010-99, Columbia University, New York, New York, Mar. 1999.

[23] International Telecommunication Union, "Methods for dimensioning resources in signalling system no. 7 networks," Recommendation E.733, Telecommunication Standardization Sector of ITU, Geneva, Switzerland, 1988 Nov.

[24] Bellcore, "Lssgr: Switching system generic requirements for call control using the integrated services digital network user part (isdnup)," Tech. Rep. GR-317-CORE, Bellcore, Morristown, New Jersey, Dec. 1997. Issue 2.

[25] H. Schulzrinne and J. Rosenberg, "A comparison of SIP and H.323 for internet telephony," in *Proc. International Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV)*, (Cambridge, England), pp. 83–86, July 1998.

[26] R. T. Braden, "Requirements for internet hosts - communication layers," Request for Comments (Standard) 1122, Internet Engineering Task Force, Oct. 1989.

[27] W. R. Stevens, *TCP/IP illustrated: the implementation*, Vol. 2. Reading, Massachusetts: Addison-Wesley, 1994.

[28] T. Seth, A. Broscius, C. Huitema, and H. Lin, "Performance requirements for signaling in internet telephony," Internet Draft, Internet Engineering Task Force, Nov. 1998. Work in progress.

[29] M. Allman, "On the generation and use of TCP acknowledgments," *ACM Computer Communication Review*, Vol. 28, pp. 4–21, Oct. 1998.

[30] A. Feldmann, A. C. Gilbert, W. Willinger, and T. G. Kurtz, "The changing nature of network traffic: Scaling phenomena," *ACM Computer Communication Review*, Vol. 28, pp. 5–29, Apr. 1998.

[31] V. Paxson, G. Almes, J. Mahdavi, and M. Mathis, "Framework for IP performance metrics," Request for Comments (Informational) 2330, Internet Engineering Task Force, May 1998.

[32] Advanced Networks and Services, "The Surveyor Project home page," 1999. http://www.advanced.org/csg-ipmm.

[33] S. Kalidindi and M. J. Zekauskas, "Surveyor: An infrastructure for internet performance measurements," in *Proc. of INET*, (San Jose, California), June 1999.

Applications and Services for Voice/Data Convergence

Sigrid Schneiders
SIEMENS AG, D-81359 Munich, Germany
E-mail: Sigrid.Schneiders@icn.siemens.de Fax: +49 89 722 36320

Abstract

The rapid rises of the Internet and of data traffic in general are causing network operators to make large investments in data networks. They are looking into opportunities for synergies between voice and data through network convergence and into opportunities for increased revenues from their investments through service convergence.

An evolutionary strategy to achieve carrier-grade services across heterogenous networks is analyzed. The network architecture, protocols, characteristics of key network elements and suitable implementation concepts are discussed.

It is demonstrated how the re-use of existing call control capabilities from voice switches and the proper use of commercial IT-platforms as adjuncts achieves maximum flexibility in the migration path and enables network operators to maximize their service revenues.

1 Driving forces of voice/data convergence

The past few years have seen the rapid rise of the Internet to become a second universal network alongside the telephone network. Already data traffic – predominantly generated by the Internet – exceeds that of the telephone network. Network operators respond to this trend by making large investments in data networks and by looking for synergy opportunities through converging voice and data networks. The increase in dial-up Internet traffic with its long call holding times leads to congestion in the voice networks, which have been designed and engineered for the typical voice traffic patterns. As TDM-based networks are not optimal for data transport, the objective is clearly, to offload Internet traffic as early as possible from the circuit switched network onto a packet switched data network. This requires to move the gateways for transition from TDM to packet (the Remote Access Server – RAS) further out towards the access or to install xDSL technology, which also gives the user higher bandwidths.

Traffic patterns in voice and data change faster than in the past, due to new competitors and decreasing tariffs, which gives subscribers more flexibility to change their communication habits and their service providers rapidly. Also the busy hours for voice and data is at different times during the 24-hour period. For these reasons, a solution to serve voice and data through a converged network promises to offer a cost efficient and flexible infrastructure and to allow a better utilization of network resources. While traditional data networks were not able to offer adequate voice services, upcoming technologies are going to make voice over packet networks feasible, especially technologies for support of Quality of Service (QoS) in packet networks and powerful Digital Signal Processors (DSPs) for voice codec processing.

As competition increases in both, voice and data, network operators have to stay competitive not only in their price and cost structures, but also in their service offerings. Innovative service offerings are becoming a key differentiator against the competition, especially services, which allow to combine voice, data and multimedia to increase productivity and convenience. Only service providers, who can respond to their customer's needs fast and with customized solutions, will be able to thrive in the new competitive environment.

Even though data traffic is overtaking voice traffic, voice services are still the main revenue source for carriers and should therefore not be disrupted by a re-shaping of networks towards convergence. In the process of voice/data convergence, voice revenues mainly have to finance the investments in converged networks and services. Additional revenues from voice/data and multimedia services will then augment these revenues, whereas revenues from pure data transport (bit transport) alone will not be able to pay back for the necessary investments in network and service infrastructure.

Consequently, any solution for voice/data convergence has to fulfill these criteria:

- Lead towards an infrastructure, which is equally efficient for transport of voice, data and multimedia streams,

- Preserve the existing voice services and their robustness and stability as a revenue generator through all stages of the migration (including connectivity and signaling to other voice networks),
- Provide high bandwidth access to voice and data services in a cost efficient way as a prerogative for the use of advanced services,
- Facilitate the introduction of converged voice, data and multimedia services, allowing fast and easy deployment of customized solutions.

2 Network architecture principles

In the world of voice/data convergence network architectures are no longer solely determined by the transport technology. Actually, different transport networks will coexist for the foreseeable future. While the circuit switched PSTN/ISDN with its huge embedded base will not be replaced in a short period of time, packet-based switching and routing technologies will also exist in different flavors: ATM networks, classical IP-routed networks, networks employing Multi Protocol Label Switching (MPLS).

In order to guarantee communication between subscribers on different types of networks continually and to provide interworking between networks new generations of mediation systems are required.

Media gateways convert all kinds of media streams between networks:

- Voice, fax and multimedia (e.g. video telephony) between circuit and IP networks (CODEC),
- Voice, fax and multimedia between circuit and ATM networks (ATM Adaptation Layer - AAL),
- Modem and ISDN data between circuit and IP networks (RAS),

Obviously it would neither be economical nor manageable to have the intelligence for services and call control and the required signaling protocols embedded in all these networks and media gateways. To cope with this situation a new paradigm has evolved, namely the separation of service intelligence and call control from the underlying transport network. This direction is acknowledged by a number of standards bodies and for a worldwide (e.g. IETF, ITU-T, MSF). The architecture arising from this concept is shown in Fig. 1.

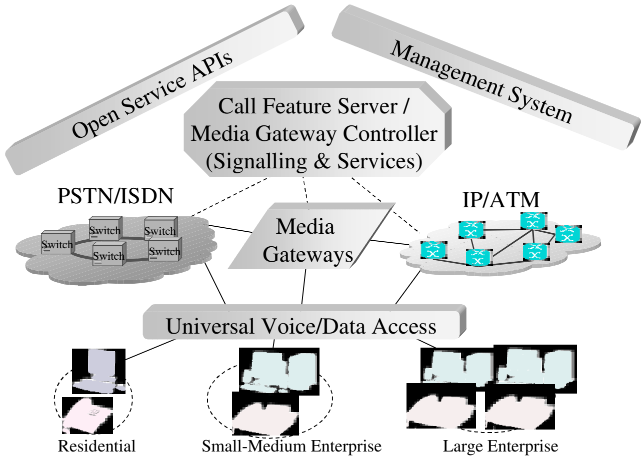


Fig. 1: Voice/data network architecture

In this architecture centralized servers, called Call Feature Servers (CFS) or Media Gateway Controllers (MGC) process call control logic and signaling, including voice/multimedia service features, voice/ISDN signaling, SS7 and IN, SIP/H.323. The CFS/MGC controls the IP/ATM transport networks and the media gateways using a Media Gateway Control Protocol (MGCP or Megaco).

Moreover Call Feature Servers also control devices for universal voice/data access and provide services and features to subscribers connected to this new breed of access systems. Universal voice/data access integrates POTS/ISDN, xDSL, IP, ATM, Voice over DSL, Frame Relay and Leased line access and provides direct access for all services to any network, be it circuit switched, IP-based or ATM. Access aspects of voice/data convergence are discussed in more depth in Ref. /1/.

The CFS/MGC is the enabling element to provide the full set of existing voice services and features also for voice and multimedia communications over packet networks. It allows the network operator to carry voice traffic from the PSTN/ISDN across packet networks without reduction in functionality and also to serve VoIP subscribers (e.g. LAN-PBXs).

Additionally, to allow fast and flexible deployment of customized services encompassing voice, data and multimedia, Application Programming Interfaces (APIs) have to be deployed. APIs for converged voice/data services enable the network operator to integrate service components from different vendors into new, network-wide services, designed and customized by him. Open APIs provide IP-based interfaces to the service functions of the communication networks, allowing developing service logic on IT-servers. Service APIs based on state-of-the art distributed processing IT-technology are an outstanding tool for service providers to maintain and expand their market share of the telecommunication market of the future.

3 SURPASS Applications for Voice/Data Convergence

3.1 Migration Strategy

In order to fulfill the criteria for a voice/data convergence solution, which have been established in section 1, and also to protect as much as possible the network operator's investments in his embedded base, it becomes necessary to look more closely into the implementation of the network architecture.

Those key elements are as discussed before:

- Media gateway,
- Call Feature Server / Media Gateway Controller,
- Open service APIs.
- Management system

The target architecture for next generation networks will be one that combines the benefits of TDM and packet worlds to deliver a new breed of services, at minimal costs and with highest reliability.

The following table summarizes the key benefits of applications in TDM and Internet domains:

Characteristics of TDM Applications	Characteristics of IP Packet Networks
Quality Speech quality is consistently excellent	**Time to Market** The IT connectivity possible with IP based network applications en-ables aggressive time to market for new appli-cations for IP networks
Reliability TDM networks provide highest service availability. Users expect the PSTN to be always available	**Personalization** Internet services are highly personalized. User profiles and preferences are maintained by the service
Security The PSTN network is extremely secure, it provide "authentication by wire" for each terminal, making fraud virtually impossible	**Wide range of Applications** Due to simplicity and the common availability of IT skills, there is a huge application base for the Internet
Intelligent Network TDM network concentrate the service intelligence in the network, provided by switching nodes and Intelligent Network (IN) components in the network	**Intelligent Terminals** In the Internet, servers, host applications and user terminals provide a massive processing power. Applications are clearly separated from transport of IP packets.

3.2 SURPASS Strategy

SURPASS introduces a network architecture for merging the advantages of a packet-oriented, multimedia-capable network with the complete voice intelligence of traditional real-time networks. At the core of SUPASS is **SURPASS hiQ** the centralized server for voice call, feature and signaling control, combined with an open service platform for deployment of innovative voice-data user services.

SURPASS hiQ also controls gateway products at the edge of the data network, the **SURPASS hiG** family of media gateways.

3.3 Trends and Chances

The trend for convergence of voice and data networks is made possible by the definition of protocols and standards by the IETF, ITU, ATM Forum and other standardization bodies to allow for real time applications over packet networks.

Originally a hobbyist application Voice over IP has grown from being a provider of "free", poor quality long distance call towards an enabler for a new generation of converged voice/data applications. VoIP brings Internet enhancements to telco applications with the following promises:

- Personalized service
- Easy creation of new applications via open API interfaces
- Minimization of service provisioning and management costs by self provisioning Web Interfaces
- Infrastructure Savings by unified access for voice and data
- Cost savings for cabling and networking infrastructure for SME premises (Soft PBX)
- Mobility and roaming via open IP interfaces
- Service delivery to dumb terminals as well as intelligent terminals

3.4 Application Overview

It is a key element of the SURPASS product family to provide service providers with end to end application solutions addressing all major end user market segments. In addition SURPASS enables service providers to develop applications running over TDM or packet networks based on the open API interfaces.

The SURPASS application family offers the following common characteristics:

- Carrier grade availability and scalability
- Based on commercial HW and SW platforms
- Services support TDM as well as packet networks
- Smooth integration with PSTN and ISP infrastructure (e.g. switches, modem pools, IP backbone)
- Optimized for integration into heterogeneous networks
- Web enabled service administration
- Modular application approach allows to bundle and combine SURPASS applications
- Support for PC based intelligent terminals
- Standards based architecture (H.323, SIP, MGCP)

The following table gives an overview about the application segments SURPASS is addressing:

3.4.1 Subscriber Services

The focus of subscriber services is expand the capabilities of the residential or SOHO subscriber line to offer productivity enhancing features based on IP enhancements. The business model for subscriber services is typically based on a flat, periodical subscription charge or on a per use charge.

IP enhancing the subscriber line via H.323 based Voice over IP and Web connectivity provides the following applications:

- User Control for Incoming Calls on PC
- Web enabled outbound calling
- Telephony Feature Control over Web Interfaces

User Control for Incoming Calls on PC (CCIB)
The increased usage of the Internet for residential and SOHO users associated with long hold times has lead to the problem of reachability during an Internet session. Siemens was the first in the market to offer users to receive calls on their PC while surfing in the Internet or checking e-mail.

The Pop-up window the SURPASS application also gives users the control about incoming calls, e.g. they are able to redirect calls to voice mail on a call by call basis.
Users surveys in the context of deployed CCIB applications has also shown that users like the mobility option of CCIB in that they can redirect their home line to any Internet PC worldwide.

Web enabled outbound Dialing (Click-to-Phone™)
In addition to controlling incoming call, the Siemens SURPASS subscriber line applications suite also provides the option to call any number in the PSTN. The application provides carrier grade security and charging features as well as a personalized, web based user interface for the service. In order to minimize the effort for the user to use the service only a standard H.323 client SW (MS NetMeeting) and a web browser are required.

3.4.2 Commerce Services

The growth of e-Commerce in the Internet has surpassed any forecasts. However despite the massive interest hardly any e-Shop is generating profits today. Users surf extensively, but they don't buy.
Analysis of customer behavior have shown that potential customers have several concerns:

- Payment methods not reliable
- Products offered not fully understood
- No direct customer support available (only e-mail)
- Complicated navigation in shops

The SURPASS solution addresses in particular the issue of missing customer support. In addition SURPASS enables to expand the reach of e-commerce offering to any phone user (TeleCommerce).

Click to Dial
Is the SURPASS solution to make e-commerce sites more customer friendly in that human assistance is always only one mouse click away. This is achieved by placing a call button on the e-commerce Web site that allows connecting the user directly to a call center. The connection is established depending on the user's preference via voice over IP or the PSTN. As an option the SURPASS Click-to-Dial solution allows the call center agent to push web pages to the users PC in order to point the potential customer to additional information.

3.4.3 Multimedia Applications

The flexible bandwidth allocation of packet networks and the media rich nature of Internet applications allows for a new breed of media processing and convergence in voice and video applications. The focus of Multimedia application is on video, graphics, text and audio information and the conversion of the media types according to the access terminal used.

VoxPortal

VoxPortal is an example for a Multimedia application, focussing on the audio rendering of e-mail, HTML pages and VXML content. The VoxPortal is able to feed dynamically generated audio streams, combined with text to speech SW to H.323 as well as VoIP terminals.

VoxPortal also manages user interaction via a flexible IVR engine. The unique feature of VoxPortal is its VXML based architecture that allows using commercial HTTP servers to host the content and to utilize CGI interfaces to dynamically access databases.

Business Communication

One of the drivers for true end to end IP applications is going to in the area of business communication. The convergence of voice and data originated in the enterprise domain, where LANs and PBX networks are beginning to merge. Initially for small and medium sized locations this trend will later also affect large-scale enterprise networks. The drivers for converged networks in the enterprise area are:

- Cost reductions for unified cabling
- Integration of telephony into workflow applications
- Availability of intelligent, networked PC on any desktop
- Advances in LAN QoS technology

Conclusion

A strategy, which exploits optimally the strength of both, existing voice service logic and open IT-platforms, maximizes the operator’s revenues from existing and new, advanced services.

A Siemens survey of end users in the residential as well as business customer segment has indicated that these new generation services are a key for the future of today’s service providers:

- Residential users are prepared to spend an average 20% premium for a package of productivity enhancing Voice Data convergence applications
- Residential and Business users are prepared to change the operator if a new operator provides a more innovative service package than the current one

As a conclusion, new voice data services have the potential to compensate for lost revenues. Residential as well as business users are willing to pay a premium for the right mix of services.

Migration from Switched Circuit Networks to Packet Network

Heinz Schwarze and Hartmut Weik
Alcatel SEL AG
Lorenzstr.10, D-70435 Stuttgart, Germany

A. INTRODUCTION

At the dawn of the third millennium IP data networks have become a worldwide proliferated communication medium. Today, voice is already being transported over various data networks. However, this is just the start. Media Gateways and controlling Call Servers are about to be introduced, so that the data network infrastructure will not only be able to carry basic voice calls. The full range of value-added voice services complemented by innovative multimedia services is the challenging goal.

1st. Objectives

The benefit of today's still rather scare voice over IP applications is primarily cost reduction. Therefore voice is, first of all, carried over the long distance data network infrastructure. An example is voice over IP interconnection of private networks. Considering the extraordinary price-cuttings for telecommunication over the past few years, it is very unlikely that in future price reduction remains the only argument in a competitive telecommunication market. What is urgently needed is improved functionality for new revenue making services

2nd. Enablers and new perspectives

Progress in voice and video coding, as illustrated in Figure 1 and Figure 2, in conjunction with the emerge of powerful , low cost Digital Signal Processors (DSP) are surely the necessary prerequisites. Meanwhile, protocol stacks for call setup and media control in packetised networks are also available. Figure 3 provides an overview.

However, from the technical point of view the reference architecture for migration, as illustrated in Figure 4 indicates that the pure inband media conversion in the Media Gateway, like voice from the Switched Circuit Network to voice over IP, is only one part of interworking. The pivotal point for bringing new sophisticated communication services to data networks is the Media Gateway controlling interface. A rather promising Media Gateway to Media Gateway Controller interface specification is shortly called MEGACO or H.248. Establishment and control of calls in IP data networks furthermore requires reliable transport of signaling information. The Internet Engineering Task Force (IETF) currently defines a generic protocol on top of UDP for the transport of signaling protocols like No.7, DSS1 or ISUP. It is called Simple Control Transmission Protocol SCTP (Figure 4).

An important objective of the migration is to carry the intelligence of the existing switched circuit networks to data packet networks. This can efficiently be done by a Media Gateway controlling Call Server, which makes ample use of the already proven circuit switched call-handling functionality. Packet networks compared with circuit switched networks are capable to offer more flexibility in bandwidth allocation and management. This is an excellent prerequisite for efficient introduction of innovative multimedia communication.

3rd. Network Configuration and significant feature requirements

The key to the cannibalization of the Switched Circuit Networks through IP is the handling of different types of access networks. The solutions of today only support pure IP at the customer premises. The challenge is to provide an access, which can also handle POTS and ISDN lines for Switched Circuit Networks. This scenario is illustrated in Figure 5 and Figure 6. In legacy Switched Circuit Networks the above function is supported by the local exchange. The point is that new operators don't want to install widely 'old fashioned' local exchanges. So one solution is to provide a device that can be connected to the access equipment, to handle the voice traffic and to offload IP traffic. In addition functions of the interconnection to other operators have to be provided. This device can be used either from Incumbent Local Exchange Carriers (ILECs) to replace their existing local exchanges, or from Competing LECs to install new equipment.

Access network types that may be connected to the V5.2 interface are countless. Examples include

- Analogue Telephone Access
- ISDN Basic Rate Access
- ISDN Primary Rate Access
- XDSL Access
- Fiber to the Curb Access
- Microwave Access (Pt-to-Pt, Pt-to-Mpt)
- Hybrid Fiber Coax Access
- Power Line Access

A future IP migrated network configuration is illustrated in Figure 6. It is flexible enough to serve all V5.2 compatible Access Networks (AN), the IP Network and further existing and future core networks.

An Access Gateway must, first of all, support remote access for narrow band (e.g. via modem) and broad band (e.g. via ADSL) subscriber to Internet services. Furthermore for IP telephony the Access Gateway must provide capabilities for monitoring and controlling end points and resources for endpoint connections. This includes for example conversions from circuit switched voice to packetized RTP voice streams. In particular for efficient support of multimedia conferencing the Access Gateway must provide in addition to point to point configurations also point to multipoint configurations.

All call signaling for voice and multimedia calls is passed from the Access Gateway directly to the Call Server. The signaling for voice calls carried in the V5.2 Communication Channel is transported without modification to the Call Server. However, the reception of DTMF digits from analogue terminals and tone/announcement insertion is required in the Access Gateway.

The Call Server terminates all call signalling for voice and multimedia calls. This includes set up and control of calls with multiple connections for voice, video and data. It is primarily the Call Server that brings new communication services to the IP data network. Candidate Call Service features include:

- Trunk cross functionality, e.g. voice over IP backbone networks
- Multiple connections per call, e.g. for multimedia communication
- Local exchange supplementary services, e.g. number identification, call waiting
- Point to multipoint communication, i.e. who hears/sees whom
- Value added services, e.g. premium rate, freephone, number portability

4th. Interfaces and Protocols

On top of the well-known Internet transport protocols various protocols specific for IP telephony need to be supported. Examples include: Gateway Control Protocol, Call Signalling, Control protocol for multimedia communication and the Bearer Independent Call Control protocol.

Figure 7 illustrates an explanatory basic call set up sequence for a H.323 terminal originated call setup to the V5.2 interface. The setup is triggered from a H.225.0 setup message from the H.323 terminal. The Media Gateway Controller (MGCP) passes the setup to the V5.2 Interface. When the called party goes off hook the Media Gateway Controller (MGC) receives the Answer Message and instructs the Media Gateway to forward the media stream from a specific trunk ID through the IP network through RTP streams with a respective encoding. The set up sequence is completed when message 19 (H.225.0 Connect) arrives at the H.225.0 terminal.

5th. The Prototype

The Prototype as shown in figure 5 is used to demonstrate the network concept. So the prototype includes existing access network devices which are capable to terminate ADSL, POTS and ISDN lines.

The Access Gateway is based on a family of highly scalable, reliable multiservice access concentrators. Digital Signal Processor (DSP) technology on the Access Gateway automatically recognizes and compresses voice and data traffic using a full range of compression, conversion, echo cancellation and encoding schemes, including G.711, G.723.1 and G.729A for voice and T.37 and T.38 for fax. The DSPs are also used for the provisioning of tones and announcements towards the telephone lines.

The Access Gateway scales to over 2000 ports per shelf and four shelves per rack. In addition a variety of other interfaces, including channelized DS-3, E1, ISDN PRI, ATM (DS3 and OC3C/STM-1), frame relay and 100 Mbit/s Fast Ethernet are supported.

The Call Server Software is derived from software of a public exchange. In this software the V5 Device Handlers are mediated in an adaptation layer to control physical devices (the Access Gateways) via the Media Gateway Control Protocol MGCP.

The first version Access Gateway/Call Server prototype can be characterized by:

- A start with high performance, future safe platform (e.g. high port density access gateway)
- Support of narrow band PC internet dial in
- Support of ISDN to ISDN telephone call over IP backbone
- Support of supplementary services like number identification

6th. Step by Step adding Intelligence to IP Networks

A low cost entry into the Voice over IP market may be voice trunking as illustrated in Figure 9. This scenario may in addition support H.323 terminals. Introduction of new and innovative services can make ample use of functions, which have already proven reliability in communication networks. Examples include:

- Extended Gatekeeper Functions
- Service Management Center Functions
- Call Server Functions
- Extended Intelligent Network Functions
- Service Creation Environment Functions

Rapid creation of new services furthermore may be opened to competitive third parties. This opens the door for a plethora of imaginative new services and applications (Figure 10).

7th. Summary and Outlook

Today, introduction of IP telephony is still in a starting phase. VoIP gateways are beginning to be deployed for backbone replacement of incumbent telephone operators and for new competing operators. With the demonstration of the prototype it can be shown, that the demarcation line between circuit switched and packet oriented network can be moved closer to the access. The concept of the universal access gateways allows the planning and deployment of networks using IP as a common platform for all type of services. The multiple interworking and gateway capabilities of the access gateway/call server combination allows for a smooth transition from today's PSTN to an IP network including next generation telephony provisioning.

References

[1] EN 300 347-1 V5.2 interface for the support of Access Network (AN), Part1: V5.2 interface specification

[2] Draft ITU-T H.248 Gateway Control Protocol

[3] ITU-T H.225.0, Call signaling and media stream packetization

[4] ITU-T H.245
Control protocol for multimedia communication

[5] IETF RFC 2543 SIP Session Initiation Protocol

[6] IETF RFC 1889 RTP A Transport Protocol for Real-Time Application

Standard	Coding	Bitrate [kbit/s]	MOS	Complexity	Delay [ms]
G.711	PCM	64	4,3	1	0.125
G.726	ADPCM	32	4,0	10	0.125
G.728	LD-CELP	16	4,0	50	0.625
GSM	RPE-LTP	13	3,7	5	20
G.729 A	CSA-CELP	8	4,0	15	15
G.723.1	A-CELP MP-MLQ	6,3 5,3	3,8	25	37.5
US Dod	LPC-10	2,4	synthetic	10	22,5

Source: BT Journal 1997

- G.723.1 has been selected by International Teleconferencing Group (IMTC) as primary Speech Coder
- G.728 is used in private Networks
- US Government selected LPC-10 as new Coder for secure voice communication

Figure 1: The Enabler: Efficient Voice Coding

Standard	Coding	Bitrate	Resolution	Frame Rate
MPEG-2	DCT	4 - 9 Mbit/s	720 x 576	25 Hz
H.261	DCT	n x 64 kbit/s	352 x288	10 Hz
H.263	DCT	8 - 20 kbit/s	176 x144	8,33 Hz
MPEG-4	DCT/DWT	5 kbit/s to 4 Mbit/s	176 x 144 to 720 x 576	5 to 25 Hz

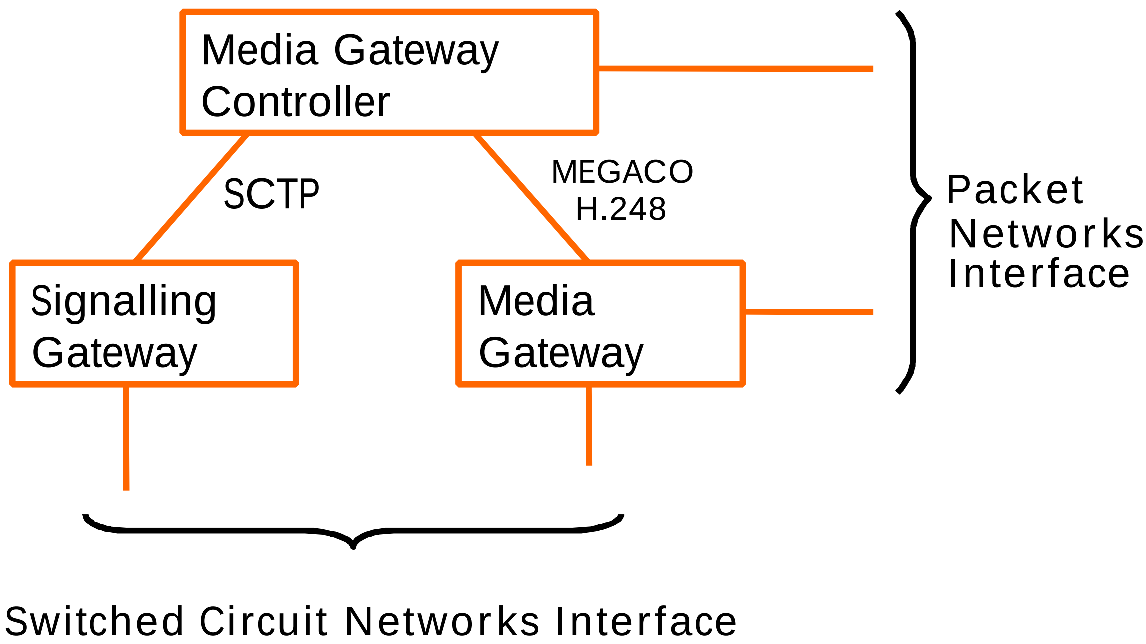
DCT - Dicrete Cosinus Transformation

DWT - Discrete Wavelet Transformation

Figure 2: Offers new Possibilities: Efficient Video Coding

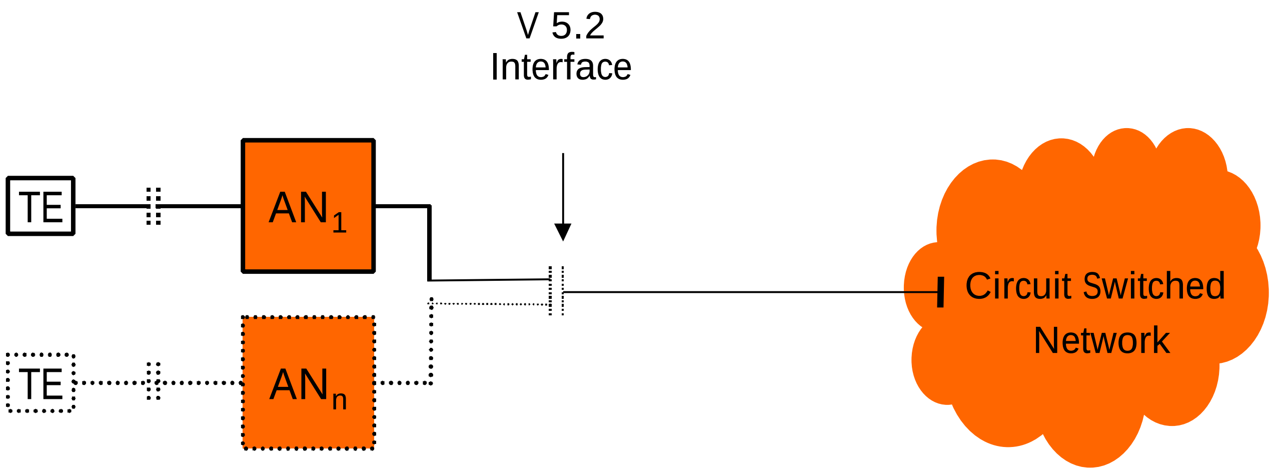
	H.320	H.324m	H.323	SIP
Audio codec	G.711, G.722, G.723, G.728, G.729	G.723 AMR	G.711, G.722 G.723, G.728 G.729	Not specified
Video codec (optional)	H.261, H.262, H.263	H.261, H.263 MPEG4	H.261 H.263	Not specified
User data appl. (optional)	T.120, etc.	T.120, etc.	T.120, etc.	Not specified
Multiplexing	H.221	H.223	H.225 (RTP)	Not specified
Call Control	Q.931	Basic Call (CS)	H.225 (Q.931)	SIP
EndToEnd Control	H.242/H.243	H.245	H.245	SDP, SAP
Transport	64kbps unrestricted	CS data (Bearer Service 30)	RTP/UDP/IP for audio/video, TCP/IP or UDP/IP for data and control	RTP/UDP/IP for audio/video, TCP/IP or UDP/IP for data and control

Figure 3: Protocol Overview



SCTP - Simple Control Transmission Protocol (SCTP), MEGACO - Internet Engineering Task Force (IETF) Working Group

Figure 4: Gateway Reference Architecture



TE - Terminal Equipment, AN - Access Network

Figure 5: Existing V5.2 Network Configuration

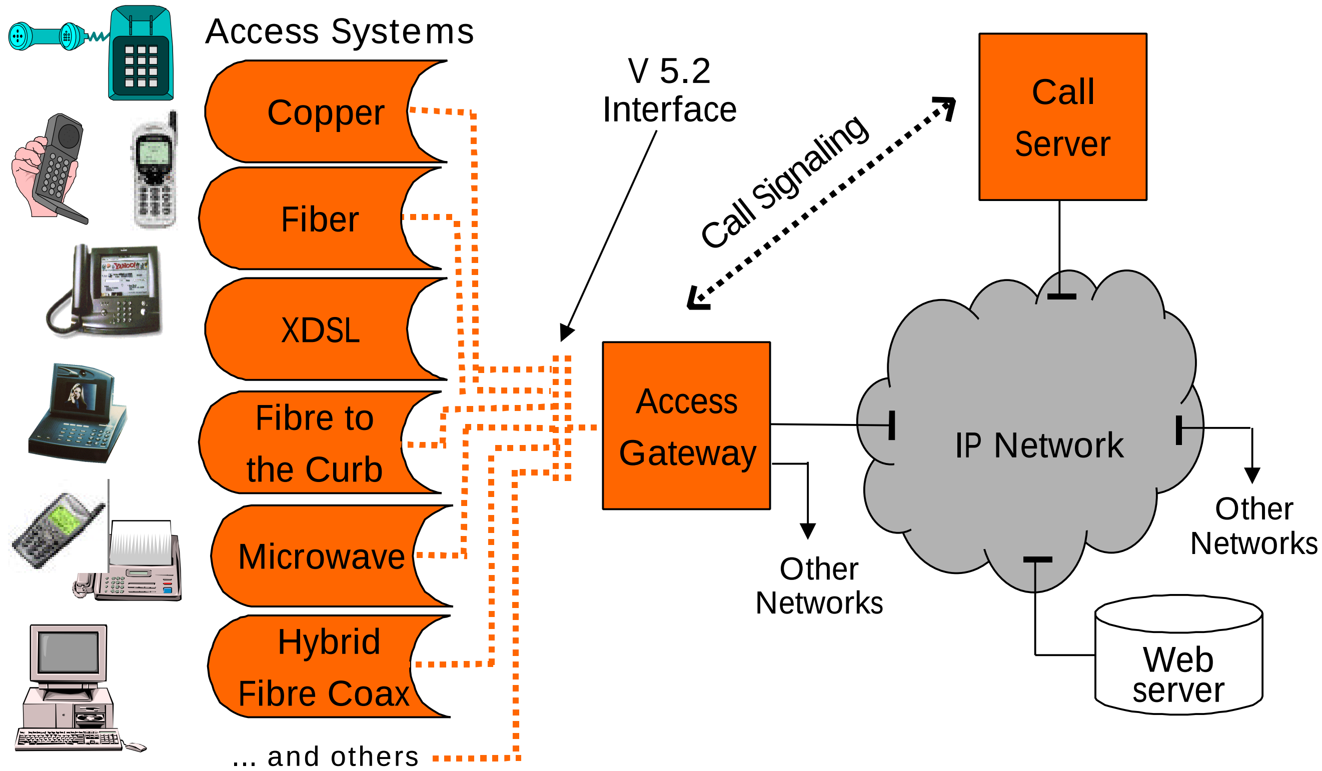


Figure 6: Future IP Migrated Network Configuration

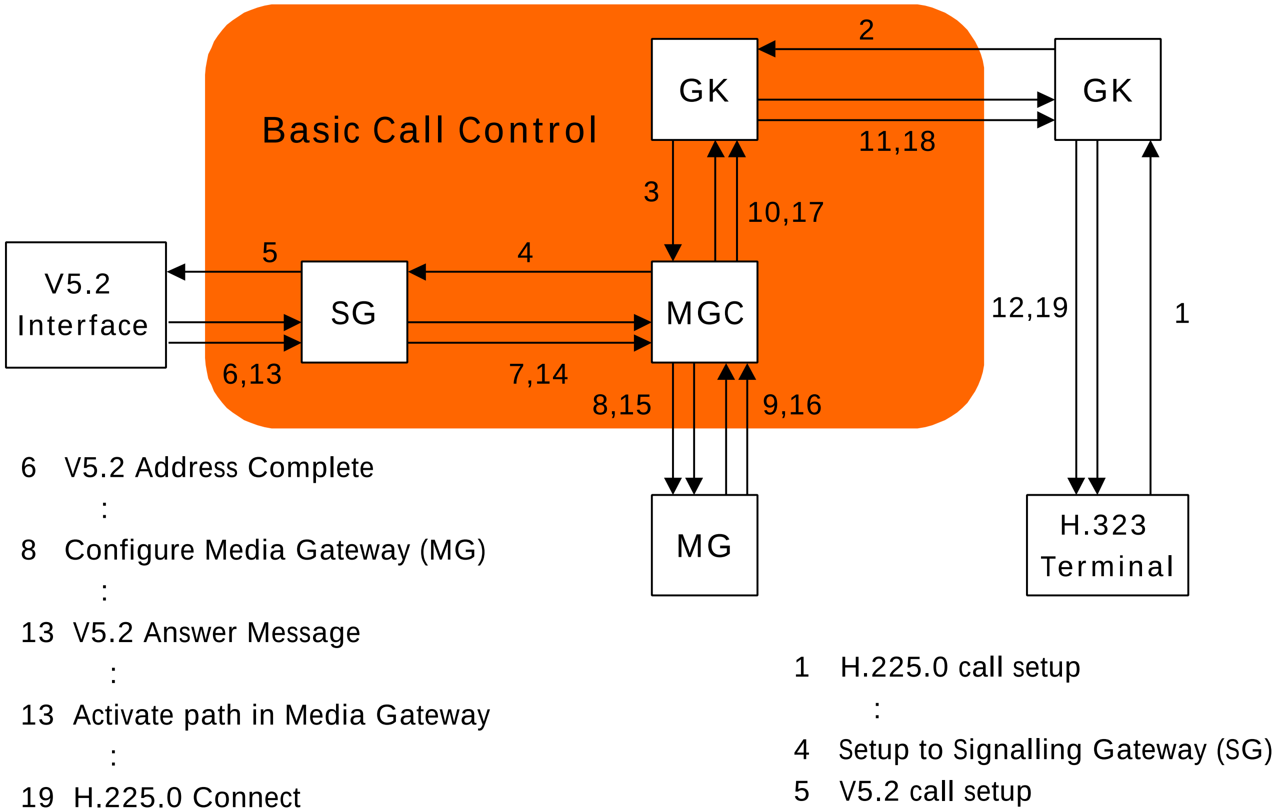


Figure 7: Explanatory Basic Call Setup Sequence

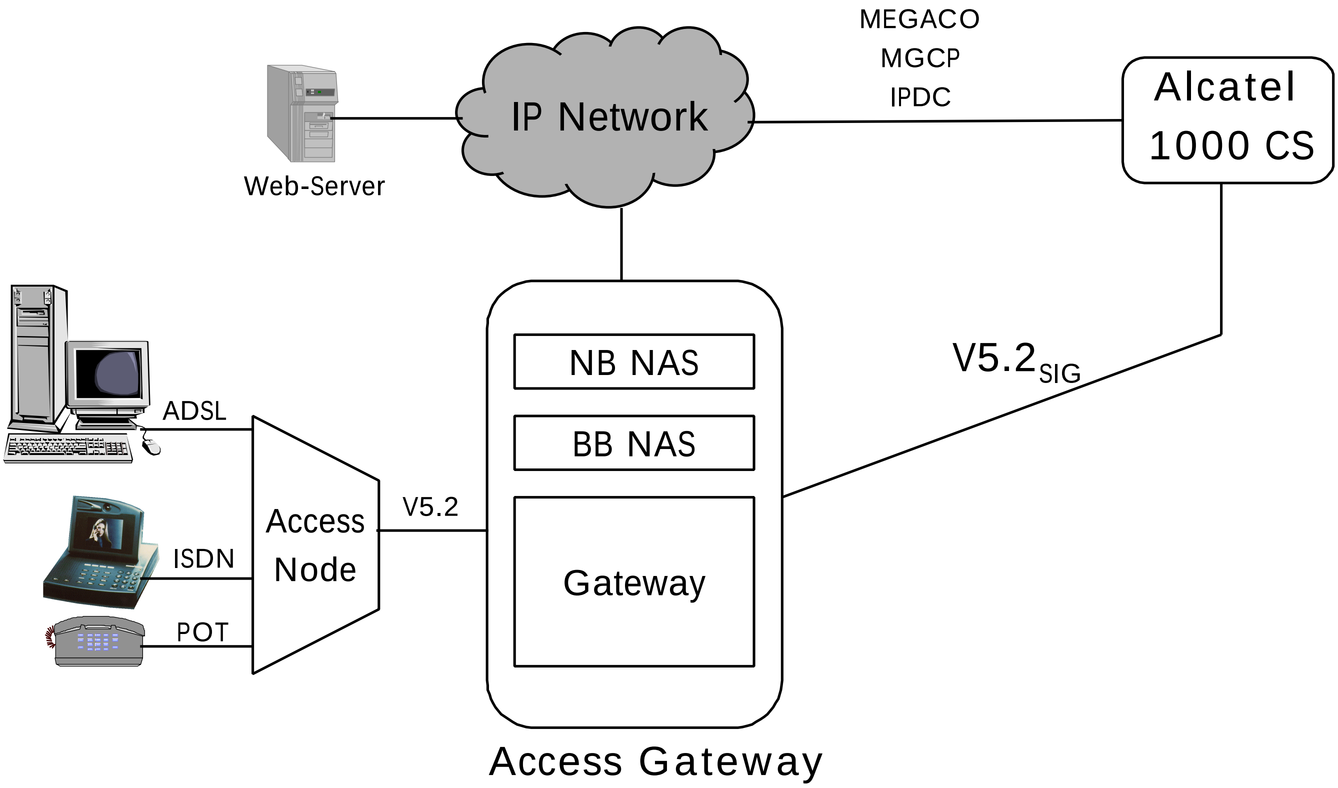


Figure 8: Prototype Configuration

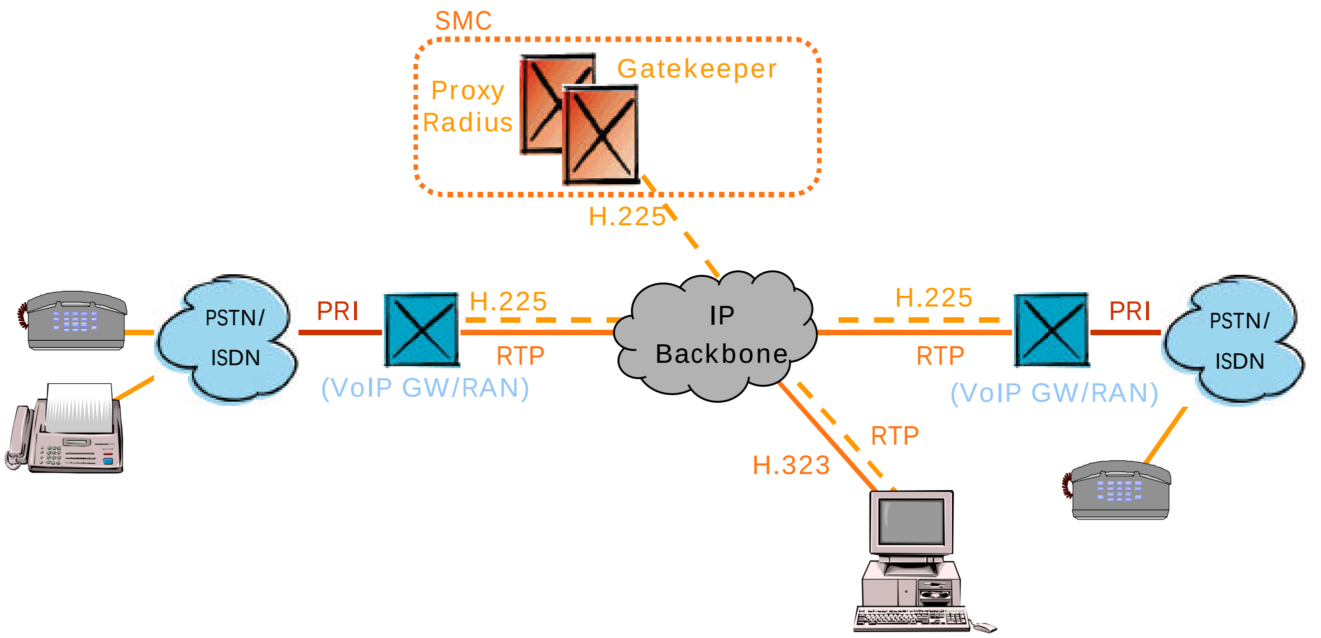


Figure 9: Basic Voice over IP Service

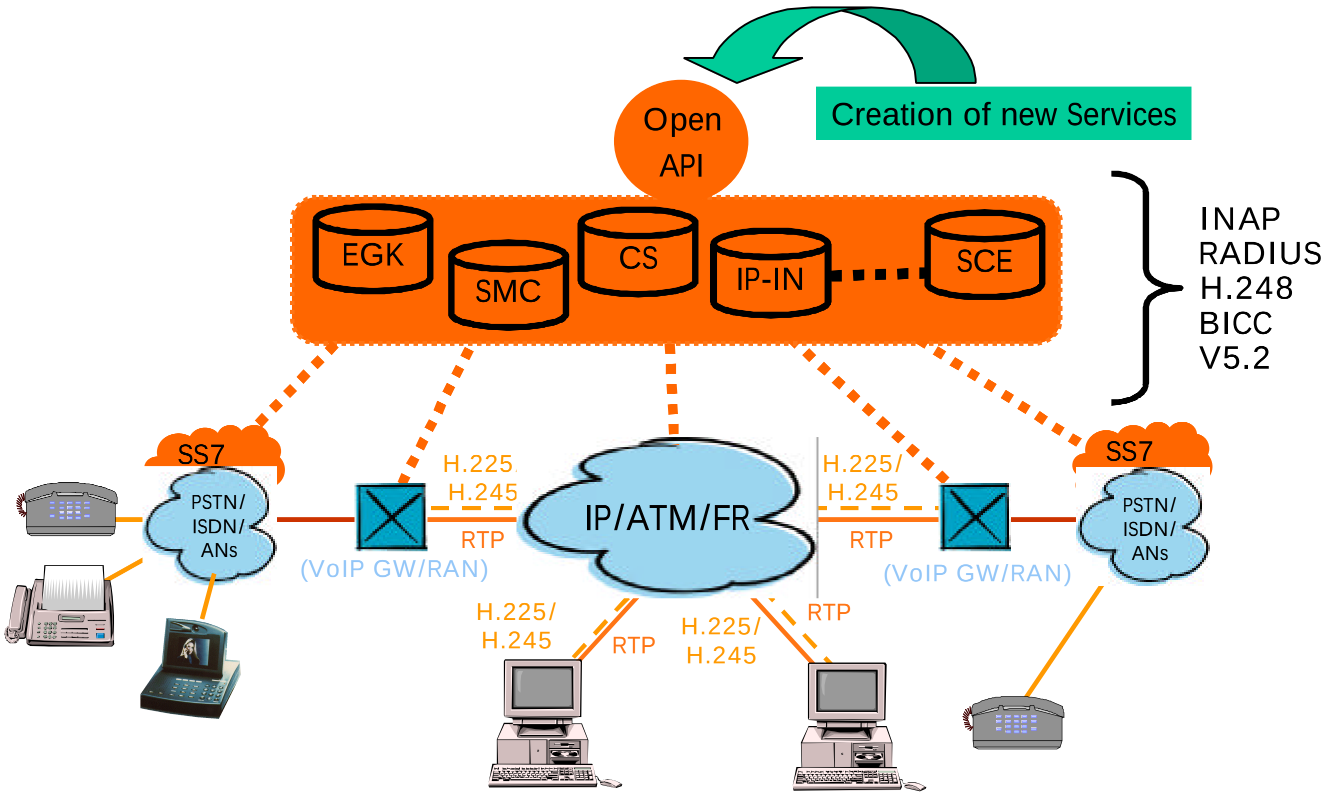


Figure 10: Step by Step Additional Network Intelligence

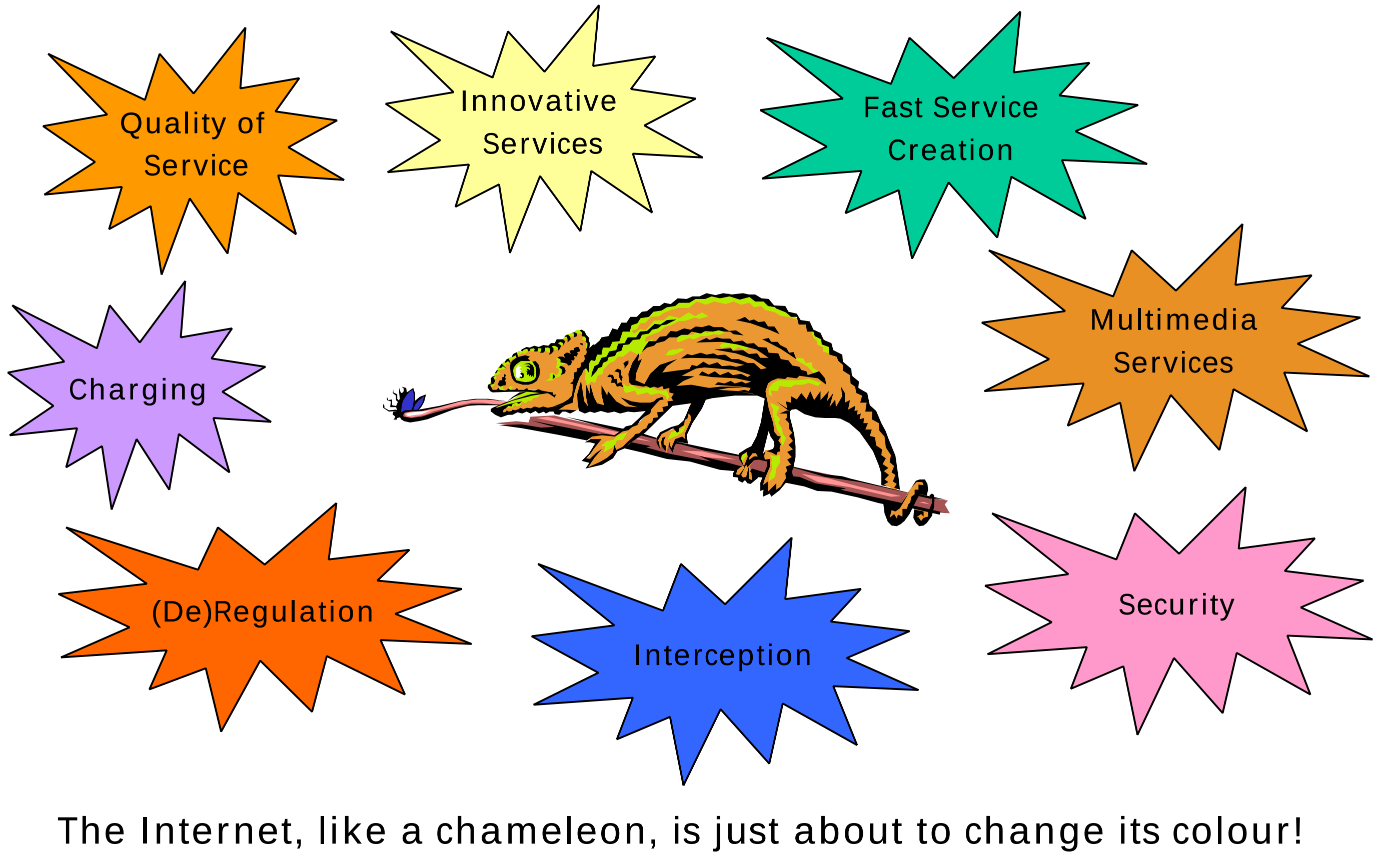


Figure 11: The Chameleon Approach

Keynote 2: Voice over IP Challenges and Opportunities

Wilhelm Wimmreuter
Siemens AG, München

VoIP Challenges & Opportunities

Wilhelm Wimmreuter
Siemens AG ICN
Hofmannstrasse 51, Munich, Germany

Abstract--

On the turn to next generation information networks, we will have to ask ourselves the question, how telephony will look like in future.

This, of course, will lead straight to the convergence of voice and data. Voice over the Internet then becomes the next question considering a given convergency. Trying to give at least a bit of an answer is the goal of this presentation. Of course this is only a small glimpse into the world of challenges and opportunities of next generation applications and voice networks.

To summarize the promises of converged networks, a look to the goals of convergence is essential.

Index terms—**Voice over IP, VoIP, Challenges, Opportunities**

A. CONVERGENCE PROVIDES:

1.. Shared Resources

provide shared transport, switching and access facilities for both, voice and data. At present, sharing of resources is typically done in the backbone and for low speed data in the access. We will get flawless resource sharing throughout the network in future.

2.. Reduced Costs for Management of Networks & Services

Management is a considerable cost factor on the operational side of networks. Avoiding media transitions on the way from "end to end".

3.. New Services and their Global Distribution

At present, services are typically centralized in the Central-Office and on Intelligent-Networks where Signaling-Control-Processors handle and control services. Even more, these services are traditionally bound to a given operator and rarely interwork between them. The Internet however allows the propagation of these and new services to arbitrary distributed servers located anywhere on the net. Amongst others, this will further open the market of services, even for companies not owning a network at all.

1) End Users have Services at their Fingertips from anywhere in the World.

Of course, the bill has to be paid by the end user. In exchange, they will receive the freedom to select services from any service provider. Thus, the users overall expenses might go down and the excess money might be spent in new services lurking around.

B. DRIVING SERVICES

Even today, services are the driving force to deliver telecom features a user is in need of. Now, as Internet-Telephony looks around the corner, we have to find a way to integrate this technology into the "day to day" communication environment.

New devices, such as IP-Phones, telephony enabled workstations, Central-Offices and IP enabled PABX's shall at least provide the same functionality as today's phones. Conferencing solutions and Call-Centers on the other hand are to be expected to tremendously enrich their feature set with the migration to the net. We shall not forget about traditional services such as Call-Waiting, Forwarding etc. That must be available in the new environment as well.

New services around voice and data convergence will deliver collaborative work to everyone and other services that will be invented soon.

This is in particular to be considered when thinking about being permanently connected to the net. Here, being always online enables a potential shift of paradigm providing the purchase of different network bandwidth on demand, instant notification and delivery of information at almost any place.

All this requires the combination of new and existing technology for the transitional phase until a fully deployed IP environment takes place. Services such as PSTN-Internet-Interworking (PINT) provide for this requirement. Thus, for the foreseeable future, and to protect investment, "Steam Phones" must have their place in IP-Tel.

C. VOIP CHALLENGES

At present, justifiable business models for the provision of high bandwidth IP networks are "elusive" when solely considering IP telephony. A long lasting business model can only be achieved considering new services provided by convergency. Otherwise arbitrage business, dealing with the margins of traditional phone companies, will disappear in a couple of years.

Business models for billing of VoIP have yet to be defined and must enable the billing for millions of subscribers. An additional challenge is the replacement of traditional settlement between operators by the deployment of Clearing-Houses.

Network and subscriber management is still to be unified. This creates the need to manage network elements as well as subscriber profiles and their policies. Nowadays, almost all of these subsystems are managed individually and that must be changed if VoIP systems will have a bite of the traditional telecom cake.

Security changes from "trust by wire" to "trust by authentication". Due to the fact, that network elements as well as users will be exposed to the public Internet, the paradigm where all elements are connected by private wires, without interconnection to the public network will fade away. We can not focus on "Trust by Wire" in future. We need to deploy new means of trust by authentication for users as well as network elements. The technology therefore is available; we simply must make it scalable to earn the fruits of convergency.

The current lack of scalability and the possible migration of ownership of information and services make operators of traditional telecom services think twice about jumping on this next generation network bandwagon. It even forces new business models where access-, transport-, service- and content-providers must cooperate to get advantage of convergence.

Services will propagate throughout the network and to the customer premises. This shift of paradigm migrates services out of the traditional central office or intelligent network systems onto the open Internet. Thus it has to be considered, that network operators at present receive income from services and for providing Network Access. They may have to share service income with others in future.

D. VoIP Opportunities

Facing above challenges will open a wealth of opportunities too. This is in fact the rational that "End to End IP throughout the global network" will provide a tremendous reduction of management costs for operators that have to build up a single mindset only to run and manage a next generation network.

New services can be provided by operators without their own networks, which means, that suppliers might sell their services to almost everyone that builds up a business case for providing a certain service to the user or to operators. Providers may even sell services to end users that run the code on their customer premises equipment. Such services then will be available everywhere and can be accessed from all around the world.

Deploying the new security infrastructure imposed by the change from "Trust by Wire" to "Trust by Authentication" enables the deployment of entirely new services globally

Another consequence of these new services will be the need to provide global directories of services. These directories will turn out to be another great opportunity to face.

The Internet is blamed not to keep up with the availability of today's redundant telephone systems. However, highly available services can be provided on the net in different ways archiving the same.

Finally one thing is for certain, the number of services will grow, and grow, and ...

Wilhelm Wimmreuter works for the R & D center of the Public Communication Networks Group of Siemens AG in Munich.
His present activities are in the architecture, standardisation, design, and integration of Internet services for next generation communication systems.

SIEMENS

Voice over IP
Challenges and Opportunities

IP-Tel 2000

Berlin, Apr. 2000

Wilhelm Wimmreuter
SIEMENS AG ICN SIB
mailto: Wilhelm.wimmreuter@icn.siemens.de

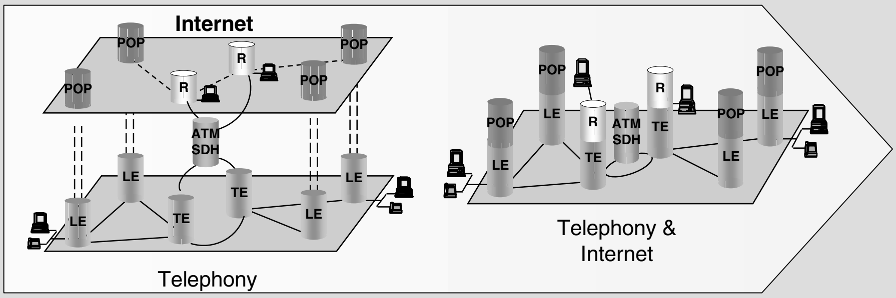
SIEMENS

Voice over IP
Challenges & Opportunities

- Introduction
  - Towards converged Voice and data networks?
  - Services
- Challenges
- Opportunities
- Evolution of Networks

SIEMENS

Towards Converged Voice & Data Networks



Convergence:

- Provides shared Access, Transport, and Services
- Shared Resources reduce Costs for Management of Networks & Services
- New Services and their Global Distribution; Voice & Data
- End Users have Services at their Fingertips from Anywhere in the World

Information and Communication Networks

Voice over IP Challenges & Opportunties/ Mar.1.2000 / © Siemens AG 2000

SIEMENS

VoIP Driving Services

- IP-Telephony is lurking around the corner, waiting for integration
  - IP Phones / Workstations, CO, PABX ...
  - IP Conferencing, Call Centers
  - IP Telephony services; Call Waiting Internet ...
  - IP Collaborative Work... And others to be invented
- Always - On: Enables a potential paradigm shift, providing the possibility to purchase different levels of network bandwidth as needed, whilst always being connected.
- Combining new with existing technologies for the transitional phase to a fully deployed IP network, by using, for example PINT

For the foreseeable future, to protect investment, "Steam Phones" must have their place in IP-Telephony.

Information and Communication Networks

Voice over IP Challenges & Opportunties/ Mar.1.2000 / © Siemens AG 2000

SIEMENS

VoIP Challenges

- Justifiable business models for the provision of high bandwidth IP networks are `elusive´ when solely considering IP telephony
- Business models for billing of VoIP have yet to be defined
- Network and subscriber management is still to be unified
- Security changes from "trust by wire" to "trust by authentication"
- Incumbent network operator resistance
- Services will propagate throughout the network and to the customer premises
- Network operators at present receive income from services and for providing Network Access. They may have to share service income with others in future.

SIEMENS

VoIP Opportunities

- End to End IP throughout the global network (Reduction of management costs ...)
- New services can be provided by operators without their own networks
- Selling services to virtually anyone (Operators and Users)
- Services will be available everywhere
- Security infrastructure enables new services globally
- Global directories of services is a new opportunity
- Highly availability can be provided in different ways

The number of services will grow, and grow, and ...

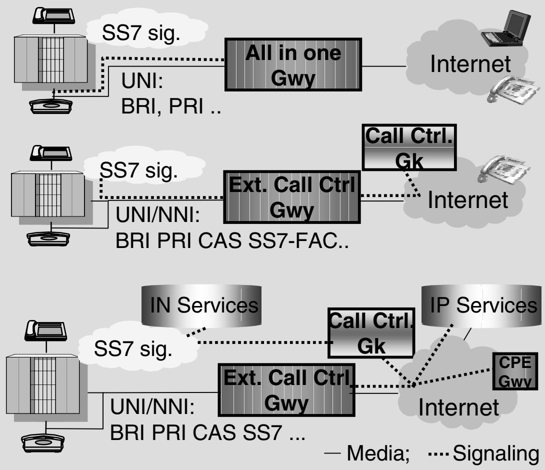
SIEMENS

VoIP evolution of interworking ... towards Shared-Services

Embedded Call Control
UNI I/F
IX1000, 3Com

External Call Control
UNI & NNI:
IX 2110, & others

Shared Services & Ext. Call Ctrl.
UNI & NNI:
IX 2110, Castle,
Surpas, SRS3000, & others
(existing Gwys reused, MGCP...)



SIEMENS

VoIP reliably delivering the future

- The world is looking for IP based networks
- There will be big changes in service provision and usage
- For new operators IP networks are "the way to go"; Existing operators need a smooth migration path. We aim to provide for both.

We provide the integrated solution...
...for the whole story

Quality of Service

DELAY AND DISTORTION BOUNDS FOR PACKETIZED VOICE CALLS OF TRADITIONAL PSTN QUALITY

Jan Janssen, Danny De Vleeschauwer and Guido H. Petit
Alcatel, Network Strategy Group
Francis Wellesplein 1, B-2018 Antwerp, Belgium
{jan.janssen, danny.de_vleeschauwer, guido.h.petit}@alcatel.be

Abstract **--When voice is to be transported over packet-based networks with particular quality guarantees, bounds on the mouth-to-ear delay and distortion should be adhered to. In this paper, the E-model is used to calculate such (delay and distortion) bounds when traditional (circuit-switched) telephone quality is aimed for.**

Index Terms **-- delay, distortion, E-model, quality, voice**

A. INTRODUCTION

For traditional Public Switched Telephone Network (PSTN) calls, which do not suffer from distortion, the key factor that determines the quality is the mouth-to-ear delay, defined as the delay incurred from the moment the talker utters the words until the instant the listener hears them. The mouth-to-ear delays that can be tolerated depend on the level of echo disturbing the voice call [1, 2].

Voice calls can also tolerate some distortion, that is, the voice signal heard by the listener does not need to be an exact copy of the voice signal produced by the talker. In contrast to circuit-switched calls (which do not suffer from distortion), distortion is likely to be introduced in packetized voice calls by the codec that compresses the voice signal or by the loss of voice packets.

As such, controlling both the mouth-to-ear delay and distortion is key to offering high-quality packetized voice calls. In this paper, we will use the E-model to calculate the mouth-to-ear delay and distortion bounds that should be respected for packetized voice calls of traditional PSTN quality.

B. E-MODEL

The E-model [3, 4, 6, 7] is a computational tool to predict the subjective quality of a telephone call based on its characterizing transmission parameters. It combines the impairments caused by these transmission parameters into a rating *R*, which ranges between 0 and 100 and can be used to predict subjective user reactions such as e.g. the Mean Opinion Score (MOS). The model was developed such that its results are in accordance with the results of extensive subjective laboratory tests.

The *R*-scale was defined so that impairments are approximately additive in the *R*-range of interest, i.e., $R = R_0 - I_s - I_d - I_e + A$. The first term R_0 groups the effects of (background and circuit) noise. The second term I_s includes impairments that occur simultaneously with the voice signal, such as those caused by quantization, by too loud a connection and by too loud a side tone. The third term I_d encompasses delayed impairments, including impairments caused by talker and listener echo or by a loss of interactivity. The fourth term I_e covers impairments caused by the use of special equipment. For example, each low bit rate codec has an associated impairment value. This impairment term can also be used to take the influence of packet loss into account. The fifth term *A* is the expectation factor, which expresses the decrease in the rating *R* that a user is willing to tolerate because of the "access advantage" that certain systems have over traditional wire-bound telephony. As an example, the expectation factor *A* for mobile telephony is 10.

ITU-T draft Recommendation G.109 [5] states that a rating *R* in the ranges [90,100], [80,90], [70,80], [60,70] and [50,60] corresponds to best, high, medium, low and poor quality, respectively. A rating below 50 indicates unacceptable quality. Throughout this paper, these quality classes are color coded according to Table 1.

As far as quality is concerned, a packetized voice call introduces more delay and distortion than a traditional PSTN call.

First, the delay for packetized voice calls, where the most important contributions are encoding, packetization, propagation, queuing, service, dejittering and decoding delay, is larger than for a traditional circuit-switched voice call, where the mouth-to-ear delay is mainly made up of the propagation delay and switching delay.

Second, in contrast to circuit-switched voice calls, as a result of voice compression and packet loss during transport or in the dejittering buffer, the distortion of packetized voice calls is not negligible.

We have studied the impact of the one-way mouth-to-

Table 1: Speech quality classes

R -value range	100 - 90	90 - 80	80 - 70	70 - 60	60 - 0
speech transmission quality category	best	high	medium	low	(very) poor

PSTN quality

ear delay (via I_d) and the distortion (via I_e) on the quality of a packetized voice call. Other factors may also impair the quality of a packetized voice call (via R_0 and I_s), but as these factors are not fundamentally different from a traditional PSTN call, they were not considered in this paper. Furthermore, as the objective was to make a fair comparison between the quality of packetized voice calls and traditional PSTN calls, the expectation factor A was set to 0.

Consider a packetized voice call between two parties, referred to as party 1 and party 2 (see Fig. 1). Using the E-model, we calculated how party 1 will judge the call, that is, what rating R will be assigned to it. The influence of delay was studied first, followed by the influence of distortion.

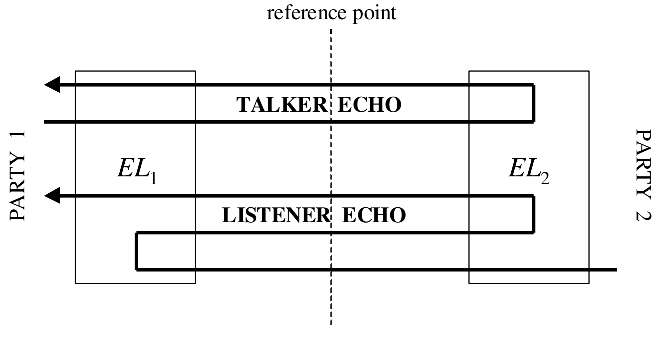
1st. Influence of mouth-to-ear delay

If the voice signal party 1 hears is delayed, the rating R decreases by an amount equal to the impairment I_d associated with the mouth-to-ear delay. This impairment is the sum of 3 contributing impairments, caused by talker echo, listener echo and loss of interactivity.

First, talker echo disturbs party 1, who hears an attenuated and delayed echo of his own voice. This echo is caused by a reflection close to party 2 with attenuation or echo loss EL_2 (measured with respect to a certain reference point) [7].

Second, listener echo also disturbs party 1, who hears the original signal from party 2 followed by an attenuated echo of this signal. This echo is determined by a reflection close to party 1 with attenuation EL_1, followed by a reflection close to party 2 with attenuation EL_2.

Echo may occur in the 4-to-2-wire hybrid (if the packetized voice call is terminated over a local PSTN) or in the callers' terminal equipment. For PSTN calls



EL – Echo Loss

Figure 1: Talker and listener echo

from traditional handsets, where echo is mainly caused by the hybrids, a typical value for the echo loss is 21 dB [7]. The same value is valid for packetized voice calls terminated over a local PSTN to traditional handsets. Handsfree phones are likely to have a lower echo loss value due to acoustic echo. When there is no hybrid (e.g. in packet-based access networks), the echo loss only depends on the acoustic echo introduced in the used terminals. Multimedia PCs are expected to have rather low echo loss values, while well-designed IP-phones will have high echo loss values. The echo losses EL_1 and EL_2 can be increased by using an echo controller, which should be deployed as close to the echo source as possible. A simple echo controller can increase the echo loss by 30 dB. Perfect echo control, in which the echo losses EL_1 and EL_2 increase to infinity, can be achieved at moderate computational cost.

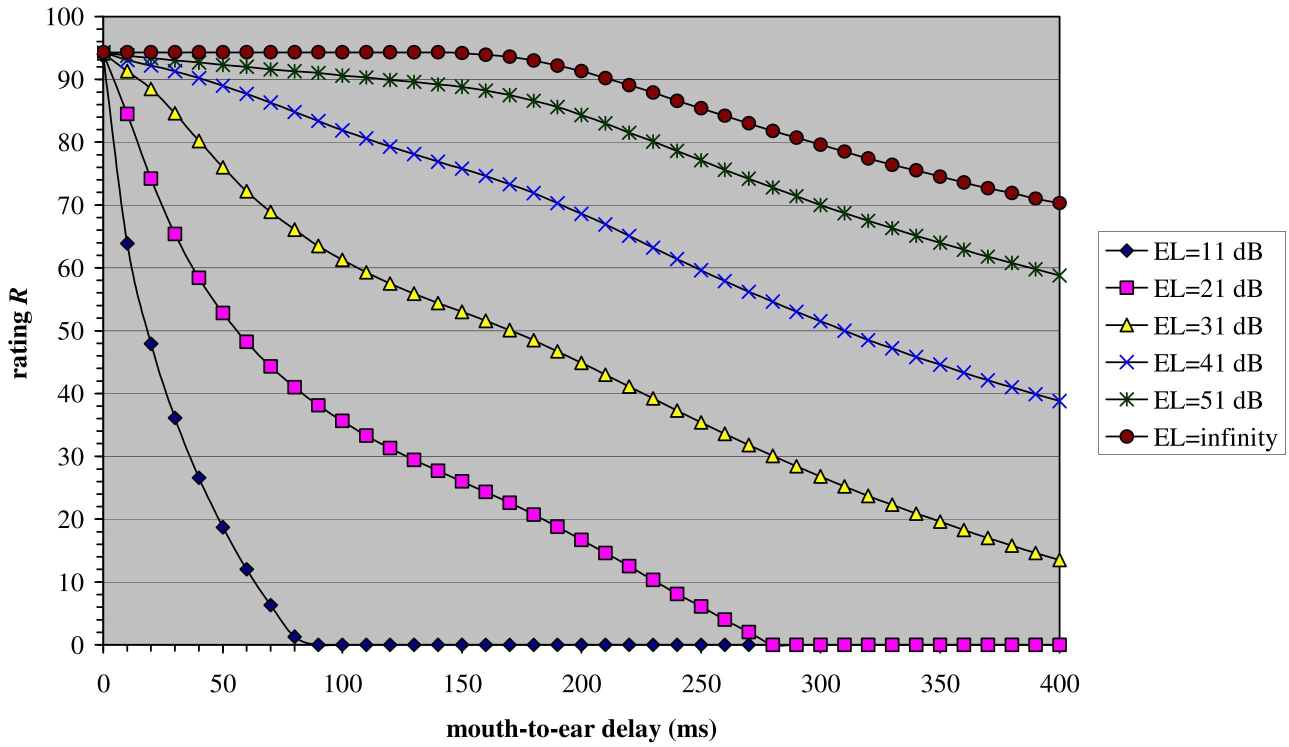
The third delay-related factor that may disturb party 1 is the loss of interactivity. If the mouth-to-ear delay is too large, an interactive conversation becomes impossible.

We have used the E-model, which takes all these impairments into account, to calculate the rating R given by party 1 to undistorted voice calls, i.e., calls transported without packet loss in the G.711 format. Fig. 2 shows the influence of the mouth-to-ear delay on the rating R for different echo loss values. The latter are assumed to be equal at both end points ($EL_1 = EL_2$). The impairment associated with delay is strongly influenced by this echo loss value.

Observe that the rating R is a non-increasing function of the mouth-to-ear delay. The intrinsic quality of a voice call is defined as the rating R associated with a mouth-to-ear delay of 0 ms. The intrinsic quality of a packetized voice call transported without packet loss in the G.711 format corresponds to $R = 94.3$. Fig. 2 shows that if echo is perfectly controlled ($EL_1 = EL_2 = \infty$), this voice call retains its intrinsic quality up to a mouth-to-ear delay of 150 ms.

ITU-T Recommendations G.114 [1] and G.131 [2] specify the following tolerable mouth-to-ear delays for traditional PSTN calls:

- Under normal circumstances (i.e. if the echo loss is at least 21 dB), echo control is needed if the mouth-to-ear delay is larger than 25 ms.
- When the echo is adequately controlled:
  - a mouth-to-ear delay of up to 150 ms is acceptable for most user applications,
  - a mouth-to-ear delay between 150 ms and 400 ms is acceptable, provided that one is aware of the impact of delay on the quality of the user applications, and



EL – Echo Loss

Figure 2: The rating *R* as a function of the mouth-to-ear delay for undistorted voice and for various echo loss values

- a mouth-to-ear delay above 400 ms is unacceptable.

It can be seen from Fig. 2 that for an echo loss of 21 dB, the rating *R* drops below 70 at a mouth-to-ear delay of 25 ms. For calls with perfect echo control, the rating *R* drops below 70 at a mouth-to-ear delay of 400 ms. Hence, ITU-T Recommendations G.114 and G.131 ensure that traditional PSTN calls have a rating *R* of at least 70. Also, the interactivity bound of 150 ms can be observed in Fig. 2 for infinite echo loss.

2nd. Influence of distortion

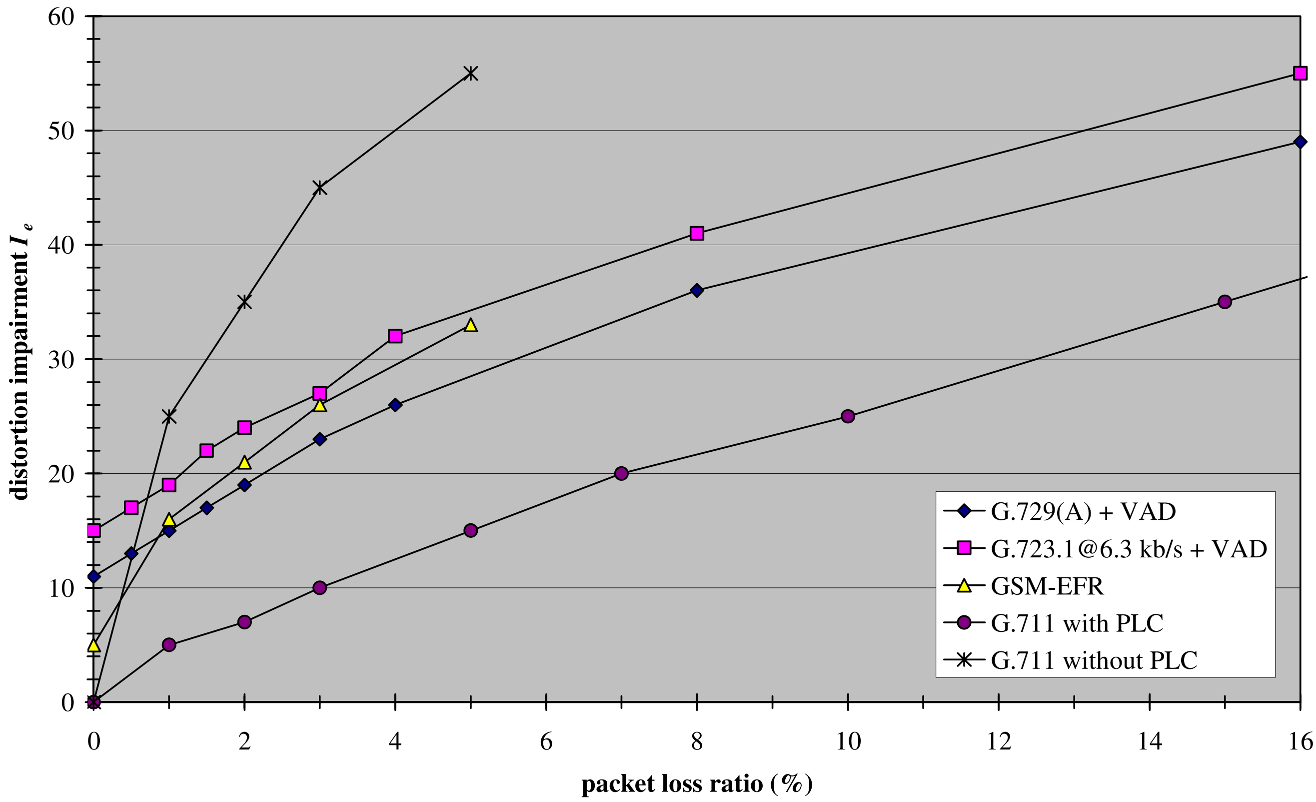
If the voice signal party 1 hears is distorted, the rating *R* decreases by an amount equal to the distortion impairment I_e. This impairment has two sources: encoding of the voice signal from party 2 and packet loss during the transport of voice packets from party 2 to party 1.

Table 2 summarizes the distortion impairment I_e and intrinsic quality (expressed in terms of *R*) associated with each standard codec [8]. It follows immediately that traditional PSTN quality ($R \geq 70$) cannot be attained with the G.726/G.727 codecs at 16 and 24 kb/s, while the intrinsic quality of the GSM-HR codec is very close to the acceptable limit. These particular codecs should be avoided.

The distortion impairment I_e associated with a codec increases as the packet loss ratio increases. Fig. 3, based on [8], shows this effect for 4 codecs, assuming that voice packets are lost at random. This figure deals only with one specific packetization interval per codec (10 ms for G.711, 20 ms for G.729 and GSM-EFR, 30 ms for G.723.1). Similar results are not yet known for other packetization intervals. The sensitivity to packet loss depends on whether a Packet Loss Concealment

Table 2: Distortion impairment and intrinsic quality of standard codecs

origin	standard	type	codec bit rate (kb/s)	I_e	intrinsic quality *R*
ITU-T	G.711	PCM	64	0	94.3
	G.726, G.727	ADPCM	16	50	44.3
			24	25	69.3
			32	7	87.3
			40	2	92.3
	G.728	LD-CELP	12.8	20	74.3
			16	7	87.3
	G.729(A)	CS-ACELP	8	10	84.3
	G.723.1	ACELP	5.3	19	75.3
		MP-MLQ	6.3	15	79.3
ETSI	GSM-FR	RPE-LTP	13	20	74.3
	GSM-HR	VSELP	5.6	23	71.3
	GSM-EFR	ACELP	12.2	5	89.3



VAD – Voice Activity Detection
PLC – Packet Loss Concealment

Figure 3: Distortion impairment as a function of the packet loss

(PLC) technique is used by the codec. More precisely, for codecs with PLC the impairments increase slower than for codecs without PLC. In contrast to the G.711 codec, most low bit rate codecs (i.e. G.729, G.723.1 and GSM-EFR) have a built-in PLC scheme. However, a PLC scheme can be implemented on top of the G.711 codec.

The voice signal does not need to be transported in the same format end-to-end. Somewhere along the route, voice might be transcoded from one codec format into another. Since all (considered) standard codecs need an 8 kHz stream of uniformly quantized voice samples at the input, the code words of the first codec need to be decoded before the signals can be encoded into another codec format. Consequently, the distortion impairment terms associated with the two codecs should be added to obtain the overall impairment I_e, because, in the E-model, impairments are additive on the *R*-scale. The intrinsic quality associated with all combinations of two codecs can be found in Table 3 (using the color code of Table 1). It turns out that transcoding can be very harmful to the quality of a call and should be avoided.

C. QUALITY BOUNDS

If the mouth-to-ear delay, echo loss and distortion impairment are known, the quality of a packetized voice call (i.e. its rating *R*) can be derived from Fig. 2 as follows. First, identify the curve on Fig. 2 that corresponds to the given echo loss. Then, using this curve, read the rating *R* corresponding to the given mouth-to-ear delay. Finally, subtract the distortion impairment I_e from this rating *R*.

As stated before, if there is no echo control, the echo loss is likely to be (smaller than) 21 dB for packetized voice transport. For that value of the echo loss, the rating *R* drops rapidly as the mouth-to-ear delay increases. Hence, if there is no echo control, there is only a very small delay budget for which traditional PSTN quality ($R \geq 70$) can be guaranteed.

From now on, we assume that perfect echo control is performed [9], in which case the intrinsic quality of the call is attained if the mouth-to-ear delay is kept below 150 ms. This intrinsic quality depends solely on the distortion impairment I_e, which in turn is determined by the codec(s) used and the overall packet loss experienced.

Since the intrinsic quality of an undistorted call is 94.3 and the bound for traditional quality is 70, there is an impairment budget of 24.3, part of which is

Table 3: Transcoding matrix

CODEC	G.711 (64kb/s)	G.726 (40kb/s)	G.726 (32kb/s)	G.726 (24kb/s)	G.726 (16kb/s)	G.728 (16kb/s)	GSM-FR (13kb/s)	G.728 (12.8kb/s)	GSM-EFR (12.2kb/s)	G.729 (8kb/s)	G.723.1 (6.3kb/s)	GSM-HR (5.6kb/s)	G.723.1 (5.3kb/s)
G.711 (64kb/s)	94.3	92.3	87.3	69.3	44.3	87.3	74.3	74.3	89.3	84.3	79.3	71.3	75.3
G.726 (40kb/s)	92.3	90.3	85.3	67.3	42.3	85.3	72.3	72.3	87.3	82.3	77.3	69.3	71.3
G.726 (32kb/s)	87.3	85.3	80.3	62.3	37.3	80.3	67.3	67.3	82.3	77.3	72.3	64.3	68.3
G.726 (24kb/s)	69.3	67.3	62.3	44.3	19.3	62.3	49.3	49.3	64.3	59.3	54.3	46.3	50.3
G.726 (16kb/s)	44.3	42.3	37.3	19.3	0	37.3	24.3	24.3	39.3	34.3	29.3	21.3	25.3
G.728 (16kb/s)	87.3	85.3	80.3	62.3	37.3	80.3	67.3	67.3	82.3	77.3	72.3	64.3	68.3
GSM-FR (13kb/s)	74.3	72.3	67.3	49.3	24.3	67.3	54.3	54.3	69.3	64.3	59.3	51.3	55.3
G.728 (12.8kb/s)	74.3	72.3	67.3	49.3	24.3	67.3	54.3	54.3	69.3	64.3	59.3	51.3	55.3
GSM-EFR (12.2kb/s)	89.3	87.3	82.3	64.3	39.3	82.3	69.3	69.3	84.3	79.3	74.3	66.3	70.3
G.729 (8kb/s)	84.3	82.3	77.3	59.3	34.3	77.3	64.3	64.3	79.3	74.3	69.3	61.3	65.3
G.723.1 (6.3kb/s)	79.3	77.3	72.3	54.3	29.3	72.3	59.3	59.3	74.3	69.3	64.3	56.3	60.3
GSM-HR (5.6kb/s)	71.3	69.3	64.3	46.3	21.3	64.3	51.3	51.3	66.3	61.3	56.3	48.3	52.3
G.723.1 (5.3kb/s)	75.3	73.3	68.3	50.3	25.3	68.3	55.3	55.3	70.3	65.3	60.3	52.3	56.3

consumed by the codec (see Table 2). Once the codec has been chosen, the remainder of the margin can be consumed either by allowing the mouth-to-ear delay to exceed 150 ms or by allowing some packet loss. Tables 4 and 5 give the codec-dependent bounds on the packet loss and mouth-to-ear delay, respectively, assuming only one of these phenomena is allowed to occur. Note that packet loss could be traded off against mouth-to-ear delay (e.g. by varying the dejittering delay), as long as the impairment budget is not exceeded.

D. CONCLUSIONS

The E-model has been used to study the quality of packetized voice calls. With regard to quality, for packetized voice calls more delay and distortion is introduced than for traditional PSTN calls.

Since the tolerable mouth-to-ear delay budget is small for compressed voice, echo control is recommended.

If the echo is perfectly controlled, the quality remains equal to the intrinsic quality up to a mouth-to-ear delay

Table 4: Tolerable packet loss bounds for a mouth-to-ear delay below 150 ms
PLC – Packet Loss Concealment
VAD – Voice Activity Detection

origin	standard	codec bit rate (kb/s)	packet loss bound (%)
ITU-T	G.711 without PLC	64	1
	G.711 with PLC	64	10
	G.729(A) + VAD	8	3.4
	G.723.1@6.3 kb/s + VAD	6.3	2.1
ETSI	GSM-EFR	12.2	2.7

Table 5: Tolerable mouth-to-ear delay bounds when there is no packet loss
NA – traditional quality is Not Attainable

origin	standard	codec bit rate (kb/s)	mouth-to-ear delay bound (ms)
ITU-T	G.711	64	400
	G.726, G.727	16	NA
		24	NA
		32	324
		40	379
	G.728	12.8	212
		16	324
	G.729(A)	8	296
	G.723.1	5.3	221
		6.3	253
ETSI	GSM-FR	13	212
	GSM-HR	5.6	180
	GSM-EFR	12.2	345

of 150 ms. The intrinsic quality depends on the amount of distortion that is introduced.

The intrinsic quality associated with some low bit rate codecs is lower than the traditional PSTN quality. Therefore, these codecs should not be used. For the same reason, transcoding should be avoided if possible.

The margin between the intrinsic quality of a codec and the bound for traditional quality can either be consumed by allowing a mouth-to-ear delay above 150 ms or by allowing some packet loss. The mouth-to-ear delay and packet loss bounds are reported here for the most common codecs. These bounds should be respected by any packetized voice call if traditional quality is to be maintained.

ACKNOWLEDGMENT

This work was carried out within the framework of the project LIMSON, sponsored by the Flemish institute for the promotion of scientific and technological research in the industry (IWT).

REFERENCES

[1] "One-Way Transmission Time", *ITU-T Recommendation G.114*, February 1996.

[2] "Control of Talker Echo", *ITU-T Recommendation G.131*, August 1996.

[3] N.O. Johannesson: "The ETSI Computation Model: A Tool for Transmission Planning of Telephone Networks", *IEEE Communications Magazine*, pp. 70–79, January 1997.

[4] P. Meschkat: "TPE: Transmission Planning (End-to-End) using the E-model (Supporting ETSI Guide 201 050)", Windows Software Tool, Alcatel Telecom, December 1997.

[5] "Definition of Categories of Speech Transmission Quality", *ITU-T Recommendation G.109*, September 1998.

[6] "The E-model, a Computational Model for Use in Transmission Planning", *ITU-T Recommendation G.107*, December 1998.

[7] "Speech Processing, Transmission and Quality Aspects (STQ); Overall Transmission Plan Aspects for Telephony in a Private Network", *ETSI Guide 201 050*, February 1999.

[8] "Provisional Planning Values for the Equipment Impairment Factor I_e", *Appendix to ITU-T Recommendation G.113 (Draft)*, September 1999.

[9] D. De Vleeschauwer, J. Janssen, G.H. Petit: "Delay Bounds for Low Bit Rate Voice Transport over IP Networks", *Proceedings of the SPIE Conference on Performance and Control of Network Systems III*, volume 3841, pp. 40–48, Boston (MA), USA, 20-21 September 1999.

ALCATEL

IPTel 2000, April 12-13, 2000, Berlin

Delay and Distortion Bounds for Packetized Voice Calls of Traditional PSTN Quality

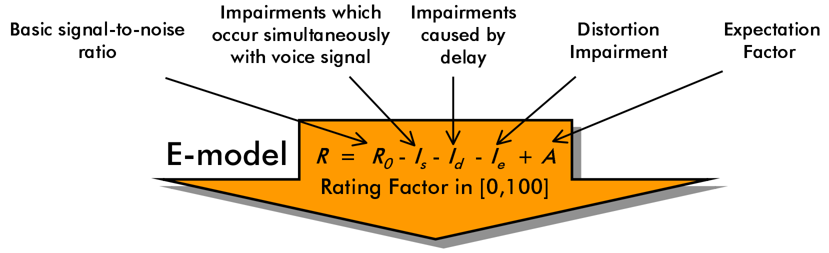
Jan Janssen, Danny De Vleeschauwer, Guido H. Petit
Alcatel, Network Strategy Group, Antwerp, Belgium

ALCATEL

List of contents

The E-model Description

Objective network parameters



Predictions of user reactions:

Mean Opinion Score	Good Or Better	Poor Or Worse	Ter- Minate Early

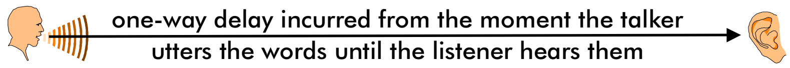
Subjective quality measures

The E-model R_0, I_s and A

- The basic signal-to-noise ratio R_0 (room noise, circuit noise, ...) and the impairments which occur simultaneously with the voice signal I_s (too loud a connection, too loud a side tone, …) are
  - not fundamentally different for circuit-switched or packetized voice calls
  - not considered
- The expectation factor A is defined as the amount of impairment tolerated because of "access of advantage" with respect to wire-bound telephony
  - Example: A=10 for mobile telephony
  - The quality of circuit-switched and packetized voice calls is compared in a fair way: A=0

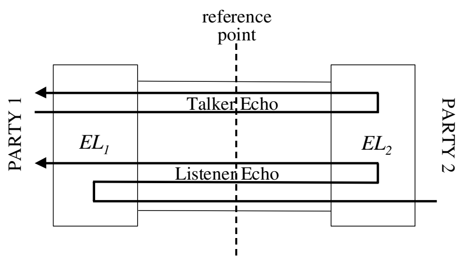
The E-model
Delay impairment I_d

- In the context of packet-based networks, the quality of voice calls is mainly determined by
  1. the impairment associated with the mouth-to-ear delay I_d



     - loss of interactivity
     - talker echo
     - listener echo
  2. the impairment associated with distortion I_e

Delay impairment I_d
Talker and listener echo

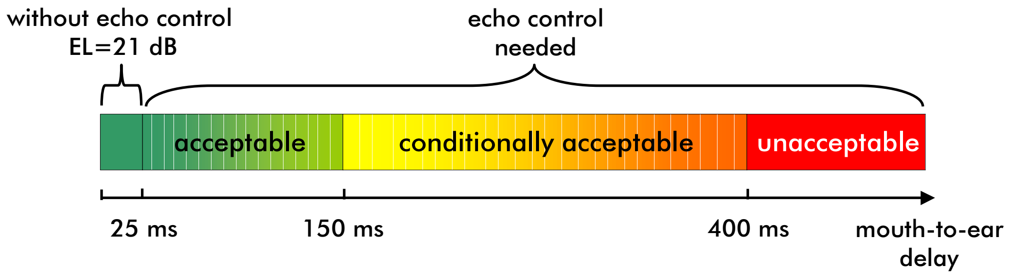


EL (Echo Loss)

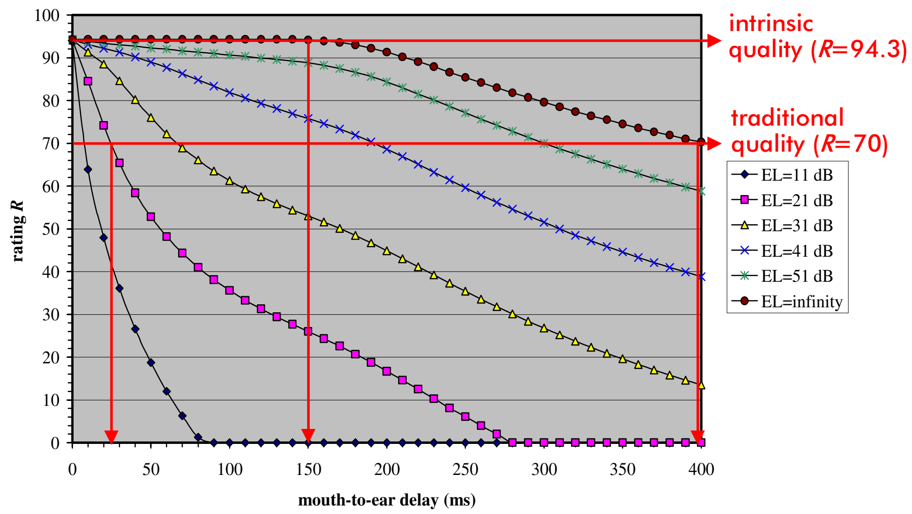
- In the PSTN, EL is typically 21 dB (due to 4-to-2 wire hybrid echo)
- If the packetized voice call is terminated over the PSTN to a traditional phone, EL≈21 dB
- If the packetized voice call is terminated over a packet-based network on
  - a PC, the EL is likely to be smaller (<21 dB) due to acoustic echo in the PC
  - an IP-phone, EL≈40 dB
- Echo control increases the EL by 30 dB, perfect echo control increases EL to infinity

Delay impairment I_d
Standardized delay bounds for undistorted voice

- ITU-T Recommendations G.114 and G.131 specify the following tolerable mouth-to-ear delay bounds for undistorted (analog or G.711@64 kb/s) voice with (hybrid or acoustic) echo



Delay impairment I_d
Results for undistorted voice



ALCATEL

The E-model
Distortion impairment I_e

- In the context of packet-based networks, the quality of voice calls is mainly determined by
  - ❶ the impairment associated with the mouth-to-ear delay I_d
  - ❷ the impairment associated with distortion I_e
    - compression by low bit rate codecs
      - VAD (Voice Activity Detection)
      - transcoding
    - packet loss

ALCATEL

Distortion impairment I_e for standardized low bit rate codecs

- ITU-T Recommendation G.109 (draft)

PSTN quality

R-value range	speech transmission quality category
90 - 100	best
80 - 90	high
70 - 80	medium
60 - 70	low
0 - 60 *	(very) poor

* : *R*-values below 50 are not recommended

origin	standard	type	codec bit rate (kb/s)	I_e	intrinsic quality *R*
ITU-T	G.711	PCM	64	0	94.3
	G.726, G.727	ADPCM	16	50	44.3
			24	25	69.3
			32	7	87.3
			40	2	92.3
	G.728	LD-CELP	12.8	20	74.3
			16	7	87.3
	G.729(A)	CS-ACELP	8	10	84.3
	G.723.1	ACELP	5.3	19	75.3
		MP-MLQ	6.3	15	79.3
ETSI	GSM-FR	RPE-LTP	13	20	74.3
	GSM-HR	VSELP	5.6	23	71.3
	GSM-EFR	ACELP	12.2	5	89.3

- VAD increases I_e by 0 to 1

Distortion impairment I_e Transcoding

- Transcoding = translation of voice from codec format X into codec format Y (via 8 kHz sampled voice)
  - Might occur at the boundary between two networks that do not have the same codec bank
  - In the E-model, impairments are (approximately) additive: the distortion impairments I_e associated with codecs X and Y should be added to obtain the overall distortion impairment factor I_e
    ⇒ Transcoding can be very harmful to the quality of a call and should be avoided if possible (see transcoding matrix on next slide with intrinsic quality levels, i.e., for mouth-to-ear delays<150 ms)

Distortion impairment I_e Transcoding matrix

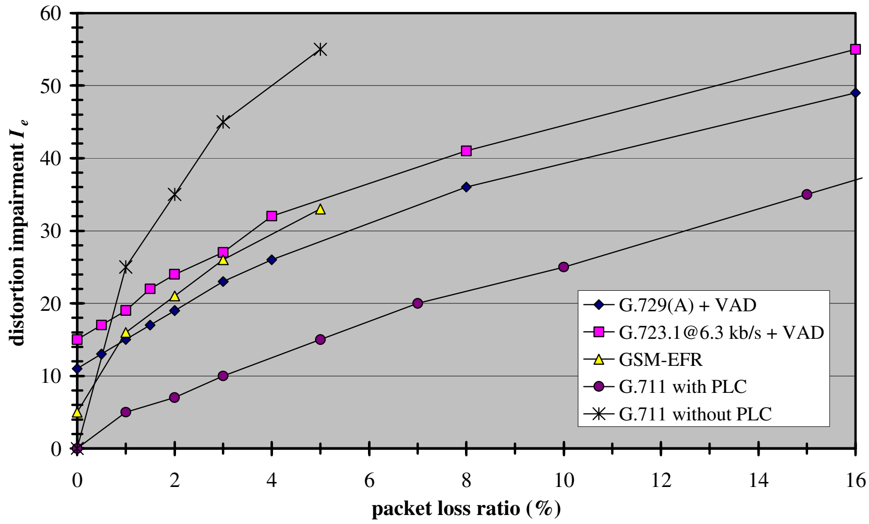
CODEC	G.711 (64kb/s)	G.726 (40kb/s)	G.726 (32kb/s)	G.726 (24kb/s)	G.726 (16kb/s)	G.728 (16kb/s)	GSM-FR (13kb/s)	G.728 (12.8kb/s)	GSM-EFR (12.2kb/s)	G.729 (8kb/s)	G.723.1 (6.3kb/s)	GSM-HR (5.6kb/s)	G.723.1 (5.3kb/s)
G.711 (64kb/s)	94.3	92.3	87.3	69.3	44.3	87.3	74.3	74.3	89.3	84.3	79.3	71.3	75.3
G.726 (40kb/s)	92.3	90.3	85.3	67.3	42.3	85.3	72.3	72.3	87.3	82.3	77.3	69.3	71.3
G.726 (32kb/s)	87.3	85.3	80.3	62.3	37.3	80.3	67.3	67.3	82.3	77.3	72.3	64.3	68.3
G.726 (24kb/s)	69.3	67.3	62.3	44.3	19.3	62.3	49.3	49.3	64.3	59.3	54.3	46.3	50.3
G.726 (16kb/s)	44.3	42.3	37.3	19.3	0	37.3	24.3	24.3	39.3	34.3	29.3	21.3	25.3
G.728 (16kb/s)	87.3	85.3	80.3	62.3	37.3	80.3	67.3	67.3	82.3	77.3	72.3	64.3	68.3
GSM-FR (13kb/s)	74.3	72.3	67.3	49.3	24.3	67.3	54.3	54.3	69.3	64.3	59.3	51.3	55.3
G.728 (12.8kb/s)	74.3	72.3	67.3	49.3	24.3	67.3	54.3	54.3	69.3	64.3	59.3	51.3	55.3
GSM-EFR (12.2kb/s)	89.3	87.3	82.3	64.3	39.3	82.3	69.3	69.3	84.3	79.3	74.3	66.3	70.3
G.729 (8kb/s)	84.3	82.3	77.3	59.3	34.3	77.3	64.3	64.3	79.3	74.3	69.3	61.3	65.3
G.723.1 (6.3kb/s)	79.3	77.3	72.3	54.3	29.3	72.3	59.3	59.3	74.3	69.3	64.3	56.3	60.3
GSM-HR (5.6kb/s)	71.3	69.3	64.3	46.3	21.3	64.3	51.3	51.3	66.3	61.3	56.3	48.3	52.3
G.723.1 (5.3kb/s)	75.3	73.3	68.3	50.3	25.3	68.3	55.3	55.3	70.3	65.3	60.3	52.3	56.3

Distortion impairment I_e
Influence of packet loss

- The distortion impairment I_e associated with a codec increases as the packet loss ratio increases
- 4 codecs (with specific packetization interval)
  - G.711 (10 ms)
  - GSM-EFR@12.2 kb/s (20 ms)
  - G.729@8 kb/s with VAD (20 ms)
  - G.723.1@6.3 kb/s with VAD (30 ms)
- PLC (Packet Loss Concealment)
  - Built-in for all considered low bit rate codecs
  - Can be implemented on top of G.711 codec

IPTel 2000, April 12-13, 2000, Berlin [JJ-DDV-GP] slide 13

Distortion impairment I_e
Influence of packet loss (cont'd)



IPTel 2000, April 12-13, 2000, Berlin [JJ-DDV-GP] slide 14

ALCATEL

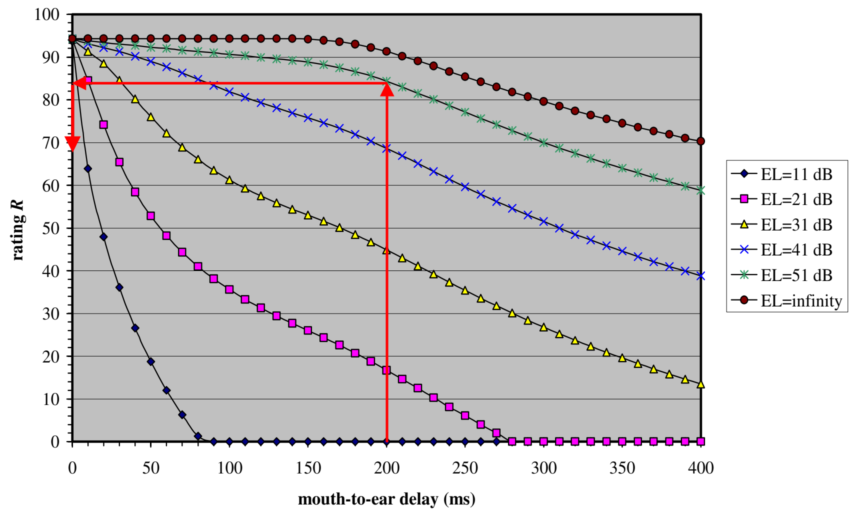
Determination of *R*-factor of a packetized voice call

- Given: mouth-to-ear delay, echo loss and distortion impairment I_e
  ⇨ *R*-rating of a packetized voice call (from the G.711 figure)
  - Identify the curve corresponding to the given EL value
  - Read rating *R* corresponding to given mouth-to-ear delay (*R*=84)
  - Subtract the distortion impairment I_e from this *R*-value (*R*=68)
- Example on next slide
  - Mouth-to-ear delay=200 ms
  - EL=51 dB
  - Distortion impairment I_e=16 (GSM-EFR codec with 1% packet loss)

IPTel 2000, April 12-13, 2000, Berlin [JJ-DDV-GP] slide 15

ALCATEL

Determination of *R*-factor of a packetized voice call (cont'd)



IPTel 2000, April 12-13, 2000, Berlin [JJ-DDV-GP] slide 16

Quality bounds

- Without echo control ⇨ very small delay budgets if traditional quality is aimed for
- Assume perfect echo control
  - Intrinsic quality (for mouth-to-ear delays<150 ms) depends solely on distortion impairment I_e (codec and packet loss)
  - 94.3 (intrinsic quality undistorted voice) - 70 (*R*-factor traditional quality) = 24.3 ⇨ if we aim for traditional voice quality, we have a maximum impairment budget of 24.3 on the *R*-scale

Quality bounds (cont'd)

- Impairment budget of 24.3
  - Part is consumed by impairment of (low bit rate) codec (see slide 11)
  - Remainder can be consumed by
    - allowing some packet loss (for mouth-to-ear delays < 150 ms)

origin	standard	codec bit rate (kb/s)	packet loss bound (%)
ITU-T	G.711 without PLC	64	1
	G.711 with PLC	64	10
	G.729(A) + VAD	8	3.4
	G.723.1@6.3 kb/s + VAD	6.3	2.1
ETSI	GSM-EFR	12.2	2.7

Quality bounds (cont'd)

- allowing a mouth-to-ear delay larger than 150 ms (when packet loss ratio = 0 %)

origin	standard	codec bit rate (kb/s)	mouth-to-ear delay bound (ms)
ITU-T	G.711	64	400
	G.726, G.727	16	NA
		24	NA
		32	324
		40	379
	G.728	12.8	212
		16	324
	G.729(A)	8	296
	G.723.1	5.3	221
		6.3	253
ETSI	GSM-FR	13	212
	GSM-HR	5.6	180
	GSM-EFR	12.2	345

NA (traditional quality is Not Attainable)

- a combination of both (trade-off between packet loss and mouth-to-ear delay)

Conclusions

- Echo control is recommended
- For perfect echo control, quality remains equal to intrinsic quality up to a mouth-to-ear delay of 150 ms
  - Intrinsic quality of some low bit rate codecs is lower than traditional PSTN quality
  - Transcoding should be avoided
  - Margin between intrinsic and traditional quality can be consumed by allowing a mouth-to-ear delay above 150 ms and/or by allowing some packet loss

Questions
Acknowledgments

Jan Janssen (jan.janssen@alcatel.be)

Danny De Vleeschauwer (danny.de_vleeschauwer@alcatel.be)

Guido H. Petit (guido.h.petit@alcatel.be)

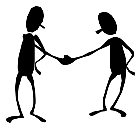
Alcatel, Network Strategy Group

Qos, Traffic and Routing Technologies

Francis Wellesplein 1

B-2018 Antwerp

Belgium



This work was carried out within the framework of the project LIMSON, sponsored by the Flemish institute for the promotion of scientific and technological research in the industry

IP-Telefonie über Differentiated Services

U. Thürmann, M. Zitterbart
Institut für Betriebssysteme und Rechnerverbund, TU Braunschweig
{thuerman|zit}@ibr.cs.tu-bs.de

Abstract— **Neben klassischen Datendiensten entwickeln sich zur Zeit eine Reihe von Echtzeitdiensten imInternet. Insbesondere hat die IP-Telefonie in den letzten Jahren an Bedeutung gewonnen. Es fehlt aber immer noch eine geeignete Infrastruktur, um diesen Anwendungen die geforderte Dienstgüte zur garantieren zu können. Der zur Zeit bei der IETF entwickelte Ansatz der Differentiated Services wird in dieser Arbeit bezüglich seiner Eignung für IP-Telefonie untersucht.**

Keywords—**IP-Telefonie, Differentiated Services, Quality-of-Service, Simulationen**

I. Motivation

Das Internet wird derzeit als globale Kommunikationsinfrastruktur für verschiedenste Anwendungen gesehen. Einer der prominentesten Vertreter aus Sicht der Anwendungen ist sicherlich das WWW. Neben reinen Datenanwendungen sollen aber auch Anwendungen mit Audio- und Videokomponenten unterstützt werden, vom „einfachen“ Telefonieren bis zu Tele-Konferenzen und virtuellen Welten. Hieraus resultieren Anforderungen hinsichtlich einer geeigneten Dienstgüteunterstützung für solche Anwendungen, die im Internet kaum bereitgestellt wird.

In Abbildung 1 ist ein Beispiel für eine IP-Telefonie-Anwendung dargestellt, basierend auf H.323 [1]. IP-Telefone in verschiedenen lokalen Netzen sollen über das Internet bzw. über ein IP-basiertes Intranet verbunden werden und einen vom Telefonnetz her gewohnten Telefondienst anbieten. Hierzu muß unter anderem eine Sprachqualität entsprechend derjenigen im Telefonnetz angeboten werden. Dies erfordert die Unterstützung von Dienstgüte im Intranet bzw. im Internet.

Der Beitrag untersucht den Ansatz der Differentiated Services [2] zur Dienstgüteunterstützung von IP-Telefonie und ist wie folgt strukturiert. In Abschnitt II wird zunächst das den Untersuchungen zugrunde liegende Simulationsmodell skizziert. Ergebnisse der simulativen Experimente diskutiert Abschnitt III. Abschnitt IV schließt den Beitrag mit einer Zusammenfassung und einem Ausblick auf weitere Arbeiten.

II. Basisszenario

Für die simulativen Experimente wurde ein Simulationsmodell in ns-2 realisiert, das die geforderten DiffServ-Mechanismen unterstützt. Die damit aufgebauten Netztopologien bestehen aus einer Reihe von DiffServ-Domänen, an die jeweils vier lokale Netze mit Endsystemen (u.a. IP-Telefone) angeschlossen sind (siehe Abb. 2). Die DiffServ-Domänen bestehen jeweils aus einem Ring von fünf inneren Routern und fünf Border-Routern. Diese ringförmige Struktur ist typisch für große ISPs wie z.B. den DFN. Jeder Border-Router ist mit genau einem der inneren Router und mit einem Router eines benachbarten Netzes verbunden. In den Experimenten wurden in den lokalen Netzen jeweils vier IP-Telefonie-Datenströme etabliert und in Richtung des Border-Routers der nächsten Domäne weitergeleitet (B14 in Abbildung 2). Insgesamt wurde ein Netz mit drei DiffServ-Domänen simuliert (siehe Abb. 3). Die Domänen sind dabei jeweils über einen Link mit einer Datenrate von 10 Mbit/s verbunden.

In allen Routern der DiffServ-Domänen kommen DiffServ-Mechanismen zum Einsatz [2], [3], [4]. In den Border-Routern wird eine Zugangskontrolle durchgeführt. Ankommende Pakete werden vom Classifier für den ihnen zugeordneten Service markiert. Hier wird den IP-Paketen einer Telefonverbindung ein entsprechender DiffServ-Codepoint zugeordnet, der deren weitere Behandlung im Netz bestimmt. Der Traffic-Meter überprüft die von den einzelnen Service-Klassen benutzten Ressourcen und markiert gegebenfalls Pakete als out-of-profile. Für die Warteschlangen werden Class Based Queueing (CBQ) [5] und RIO-Warteschlangen [6] verwendet.

In den Endsystemen des LANs dagegen werden einfache FIFO-Warteschlangen ohne Differenzierung zwischen Paketen verschiedener Verkehrsströme verwendet. Die Router im LAN sind nicht DiffServ-fähig.

Für die Simulation des *Hintergrundverkehrs* wurden ON/OFF-Quellen mit Pareto-Verteilung benutzt [7], um selbstähnlichen Verkehr zu generieren. Jedes lokale Netz sendet und empfängt 40 dieser burstartigen Datenströme. Insgesamt werden also 480 Pareto-verteilte Hintergrundverbindungen etabliert, wobei sich davon 120 an der Stelle der höchsten Konzentration zwischen R14 und B14 überlagern. In den Simulationen mit Differentiated Services wird der Hintergrundverkehr verschiedenen Service-Klassen (Assured Service und Best Effort) zugeordnet. Es wird mehr Hintergrundverkehr erzeugt, als in den Traffic-

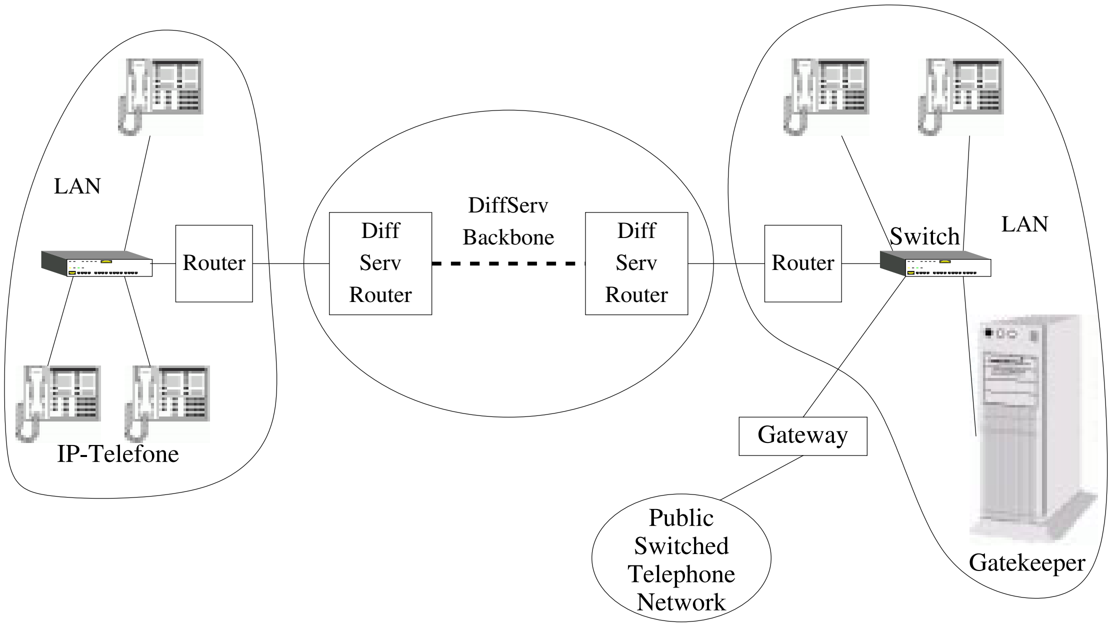


Fig. 1. IP-Telefonie über das Internet

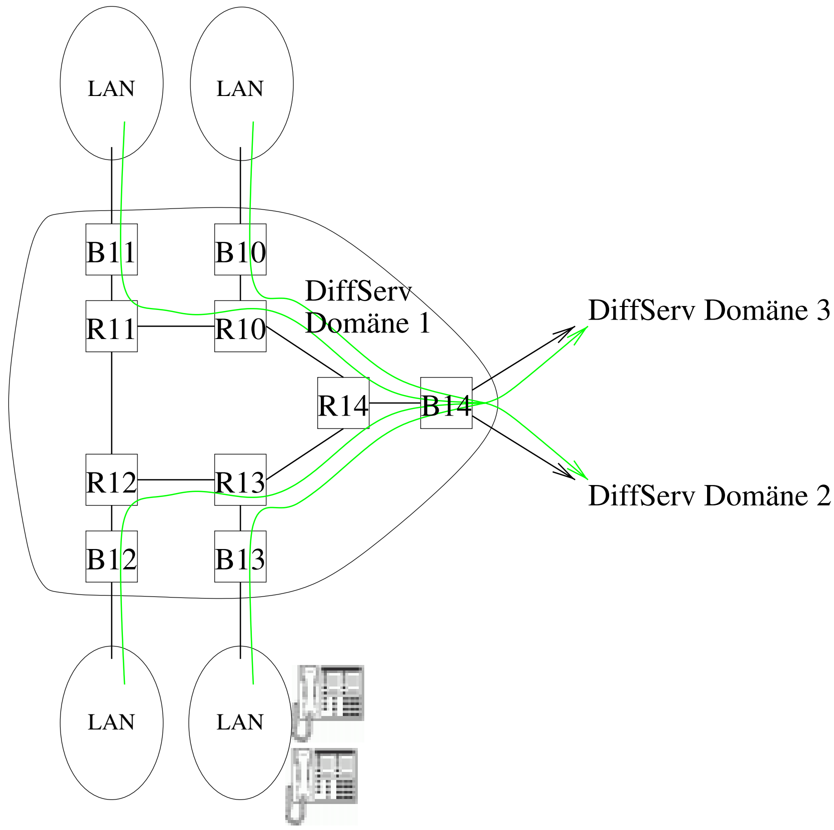


Fig. 2. Aufbau einer DiffServ-Domäne

Metern erlaubt ist, so daß ein Teil der Pakete als out-ouf-profile markiert wird und damit eine höhere Drop-Precedence erfährt. Jedes LAN darf nur jeweils 800 kbit/s in-profile-Verkehr in den Klassen AF1 und AF2 an seine DiffServ-Domäne senden. Der Rest wird als out-of-profile markiert. Zwischen R14 und B14 aggregiert sich dieser Verkehr zu einem Strom von jeweils 3.2 Mbit/s.

Jedes LAN erzeugt neben dem bereits erwähnten Hintergrundverkehr vier Datenströme, die aus den Paketen der Sprachverbindungen bestehen. Von den vier Daten-

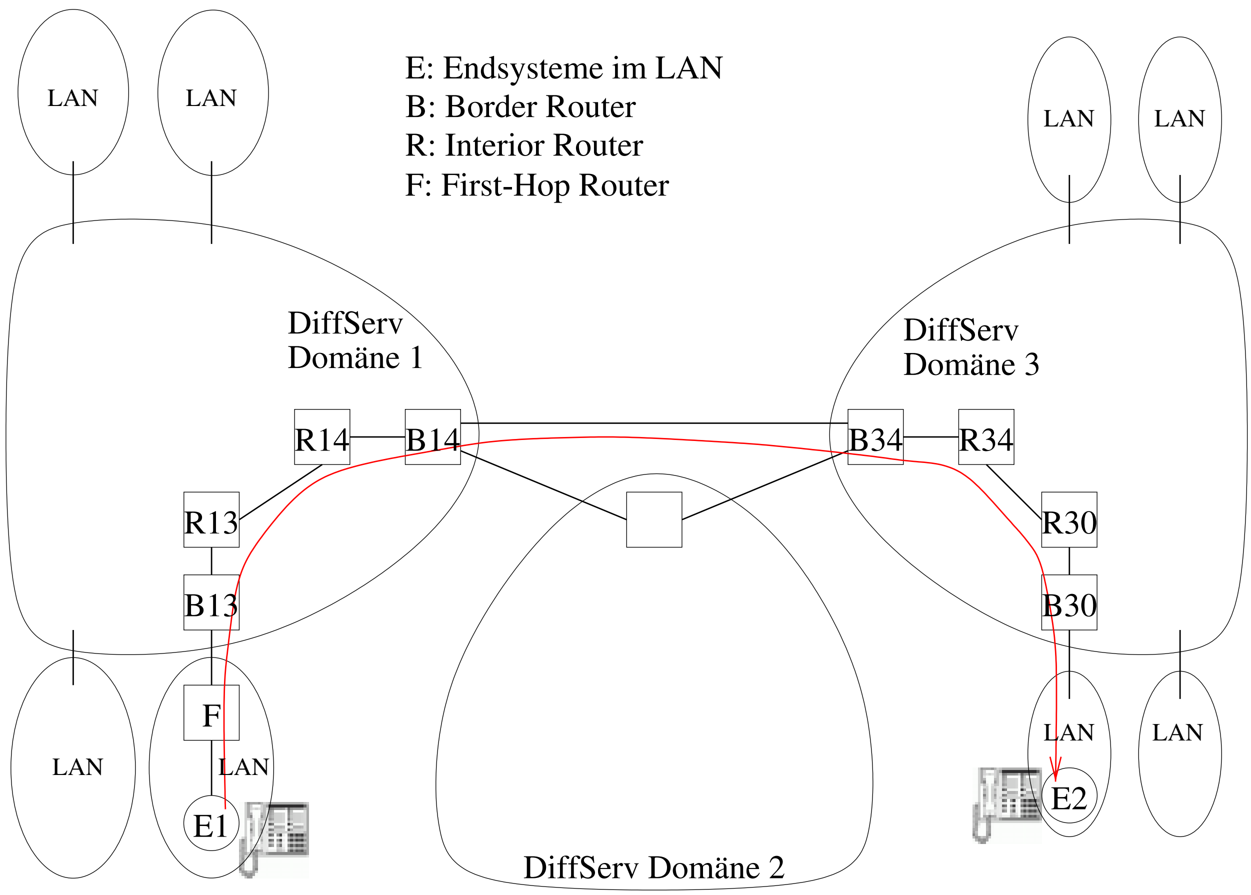


Fig. 3. Simulationsszenario

strömen, die von einem LAN erzeugt werden, gehen jeweils 2 Datenströme zu einem LAN in einer der beiden benachbarten DS-Domänen (vgl. Abbildung 2).

Die höchste Konzentration von Telefonieverkehr tritt in den DiffServ-Domänen zwischen dem Border-Router zu den beiden anderen DiffServ-Domänen und dem daran angeschlossenen inneren Router auf, also z.B. in der DS-Domäne 1 zwischen den Routern B14 und R14 (vgl. Abbildung 3). Es handelt sich um 16 einzelne Verkehrsströme, die zusammen als aggregierter Strom ca. 1.1 Mbit/s Bandbreite benötigen. Der Anteil der Link-Kapazität, die für Premium Service mit Hilfe des CBQ-Verfahrens reserviert wird, ist so bemessen, daß der Telefonieverkehr ohne Verluste und große Verzögerungsschwankungen fließen kann, falls er als Premium-Service-Verkehr klassifiziert wird. Außerdem wird der Premium-Service-Verkehr in den CBQ-Warteschlangen mit hoher Priorität behandelt, so daß die Warteschlangen für Premium Service nie mehr Pakete enthalten sollten als die Anzahl der an dem Router ankommenden Links. Assured Service und Best Effort besitzen die gleiche Priorität, die allerdings niedriger ist als diejenige von Premium Service. Die Zugangskontrolle in den Border-Routern verhindert, daß der Premium Service mehr als die ihm zugeordneten Ressourcen erhalten kann.

Die Zuordnung von Bandbreite auf den 10Mbit/s-Links innerhalb und zwischen den DS-Domänen auf die einzelnen PHBs ist wie folgt konfiguriert:

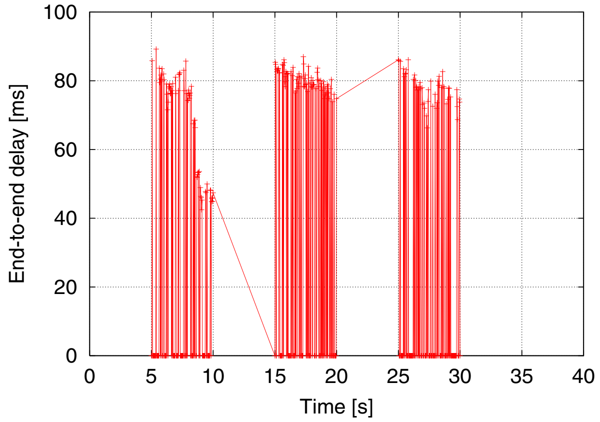
EF (Premium Service)	11%
AF1 (Assured Service)	40%
AF2 (Assured Service)	30%
BE (Best Effort)	5%

Die restliche nicht zugeteilte Bandbreite, sowie ungenutzte, einem PHB zugeteilte Bandbreite, kann durch die CBQ-Mechanismen auf aktive PHBs verteilt werden.

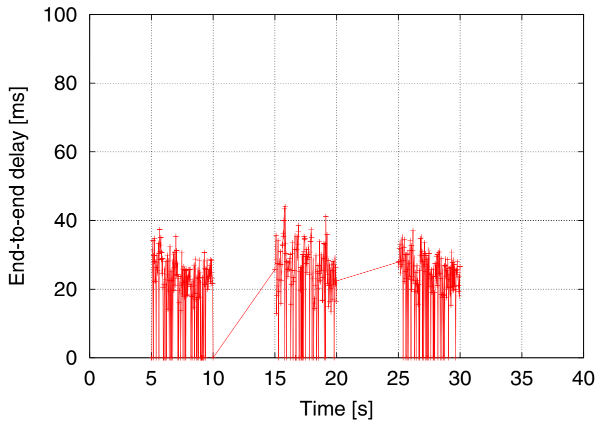
Zur Beurteilung der Qualität von IP-Telefonie-Datenströme wurden drei verschiedene *Sprachverkehrsmuster* untersucht:

- 64 kbit/s CBR.
- gemessener LiveLan-Verkehr
- gemessener NetMeeting-Verkehr

Für den *CBR-Verkehr* werden in Intervallen von 32 ms Pakete einer Länge von 256 Bytes gesendet. Die IP-Telefoniedatenströme senden in drei Intervallen von jeweils 5 s Länge mit Unterbrechungen von 5 s. Es soll hier-

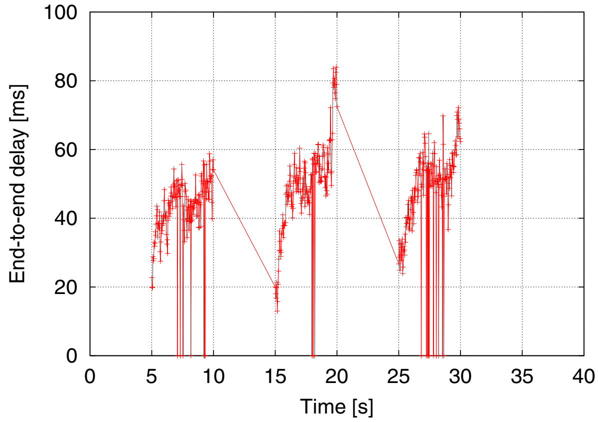


(a) Best Effort mit DropTail/FIFO

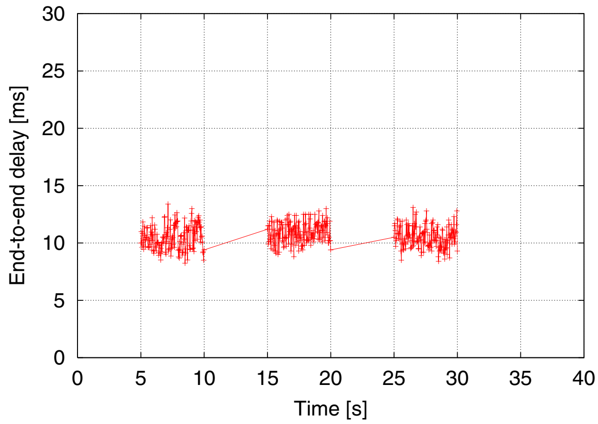


(b) Best Effort mit RED

Fig. 4. Verzögerung mit Best Effort für CBR-Verkehr



(a) Assured Service



(b) Premium Service

Fig. 5. Verzögerung mit DiffServ für CBR-Verkehr

mit das Verhalten des Verkehrs direkt nach dem Einschalten bei evtl.[1] Silence Suppression beobachtet werden.

Bei *LiveLan* werden Pakete von 566 Bytes benutzt. Die Sendeintervalle schwanken zwischen 50 ms und 80 ms, so daß sich eine mittlere Bandbreite von ca. 70 kbit/s ergibt.

Der *NetMeeting-Verkehr* ist deutlich burstartiger als LiveLan. Die Paketgröße beträgt 78 Bytes, die Intervalle zwischen den Paketen liegen zwischen 3.5 ms und 200 ms. Dabei treten meist zwischen 2 und 5 Pakete in sehr kurzen Abständen von 3.5 ms auf, gefolgt von einem Intervall von 70-200 ms. Die mittlere Bandbreite des gemessenen NetMeeting-Verkehrs beträgt ca. 16 kbit/s.

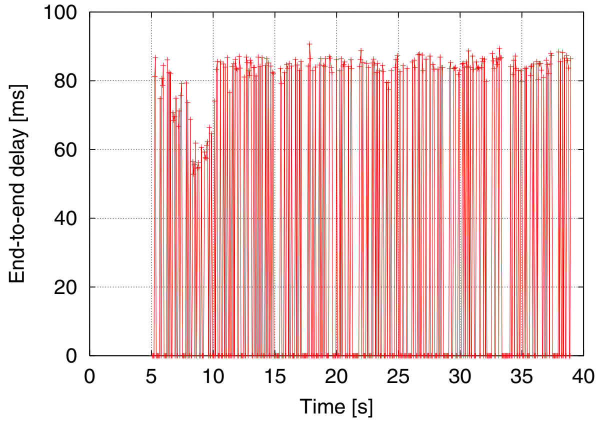
Die Pakete eines IP-Telefonie-Datenstroms werden in den Simulationen mit dem DS-codepoint für Best Effort, Assured Service bzw. Premium Service markiert. Im Falle von Assured Service werden die Pakete alle als in-profile markiert.

[1] H.323 und G.711 spezifizieren keine Silence Suppression für Audio-Datenströme

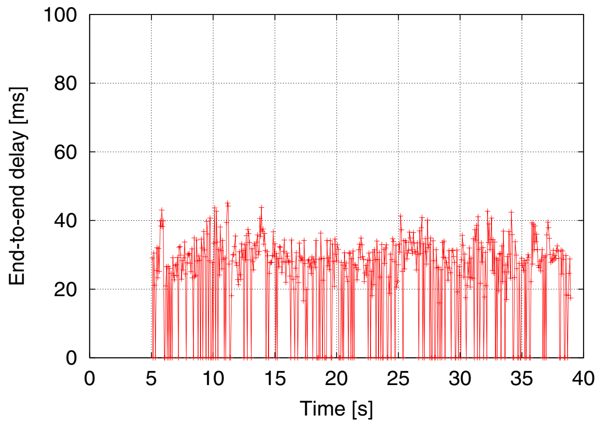
III. Simulative Experimente

Mit dem in Abschnitt II kurz skizzierten Simulationsmodell wurde eine Reihe von Experimenten durchgeführt zur Evaluierung der Eignung von DiffServ für die Unterstützung von IP-Telefonie.

Um einen Vergleich der mit DiffServ erzielten Dienstgüte gegenüber derjenigen ohne DiffServ zu haben, wurden zunächst Experimente ohne spezielle Dienstgüteunterstützung im Netz durchgeführt. Der durch IP-Telefonie verursachte Verkehr wird somit einfach als Best-Effort behandelt. Die Warteschlangen der Router arbeiten gemäß dem FIFO-Prinzip bzw. werden zusätzlich durch RED unterstützt. Die Warteschlangen sind dabei auf eine maximale Länge von 50 Paketen konfiguriert. Exempla-

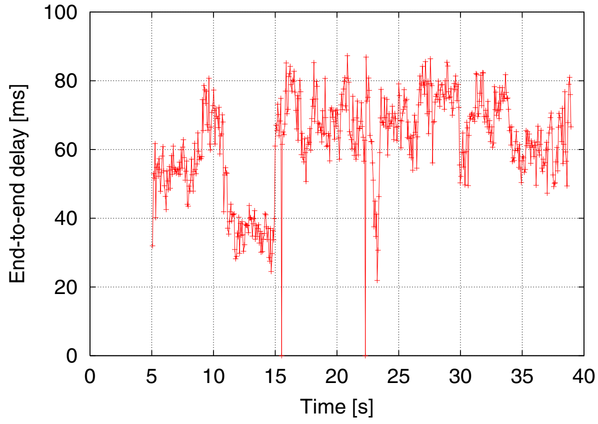


(a) Best Effort mit DropTail/FIFO

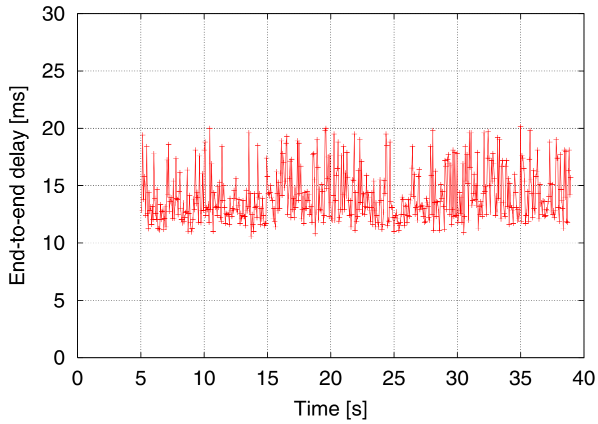


(b) Best Effort mit RED

Fig. 6. Verzögerung mit Best Effort für LiveLan-Verkehr



(a) Assured Service



(b) Premium Service

Fig. 7. Verzögerung mit DiffServ für LiveLan-Verkehr

risch wird in den folgenden Untersuchungen ein dedizierter IP-Telefonie-Datenstrom herausgegriffen.

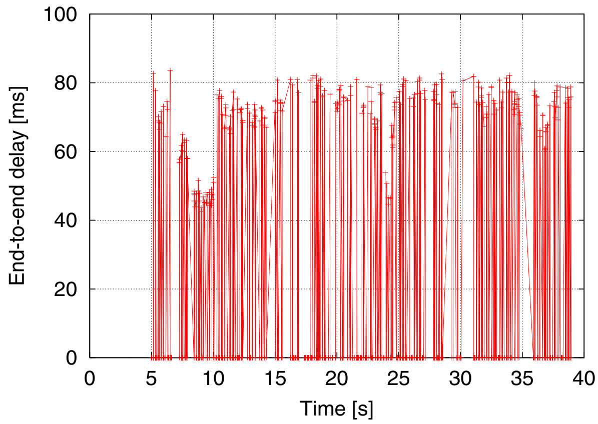
A. *CBR-basierter Telefonverkehr*

Der CBR-Datenstrom ist jeweils dreimal für eine Zeitdauer von jeweils 5 Sekunden aktiv, wobei je 471 Pakete erzeugt werden. In den Diagrammen ist jeweils die Ende-zu-Ende Verzögerung dargestellt. Paketverluste sind durch eine Verzögerung von 0 repräsentiert.

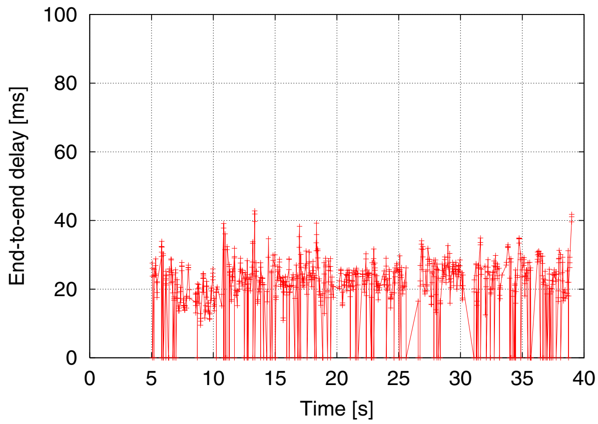
Bei der Verwendung von *Best-Effort* sind hohe Paketverluste von 59% zu verzeichnen. Die Verzögerungen liegen im Interval $[42.5\text{ms}, 89.2\text{ms}]$. Die mittlere Verzögerung beträgt 75.6 ms. Die hohen Verzögerungen und Verluste kommen zustande, da die Quellen für den Hintergrundverkehr mehr Verkehr erzeugen, als das Netz transportieren kann. Dadurch wachsen die Warteschlangen in den Routern bis zur ihrer maximalen Länge. Ab diesem Zeitpunkt werden ankommende Pakete verworfen. Telefondaten werden hier genauso behandelt wie der Hintergrundverkehr auch.

Bereits die Verwendung von *RED* führt zu einer sichtbaren Verbesserung der Qualität. Durch den RED-Algorithmus wird die mittlere Länge der Warteschlange durch frühzeitiges Verwerfen von Paketen gering gehalten, so daß sich die Verzögerung in der Warteschlange reduziert. Burstartiger Verkehr kann die Länge der Warteschlange und damit die Verzögerung jedoch für kurze Zeitintervalle erhöhen. Die Verzögerung reduziert sich durch RED auf das Interval $[12.9\text{ms}, 44.0\text{ms}]$ mit einem Mittelwert von 25.9 ms. Die Paketverluste verbessern sich ebenfalls erheblich, sind aber mit 18% nach wie vor sehr hoch. Die Dienstqualität ist damit für den Benutzer zwar besser aber noch nicht akzeptabel.

Mit DiffServ-Mechanismen läßt sich die Qualität der Telefonie nochmals deutlich verbessern. Bei Verwendung von *Assured Service* verringert sich allerdings die

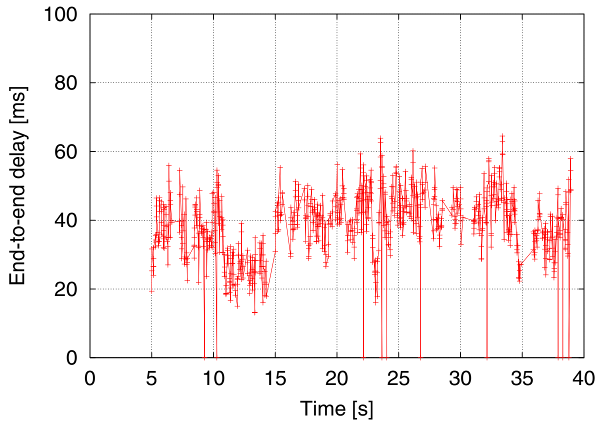


(a) Best Effort mit DropTail/FIFO

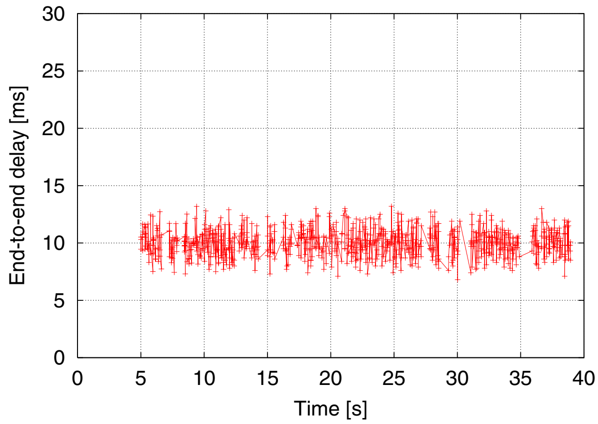


(b) Best Effort mit RED

Fig. 8. Verzögerung mit Best Effort für NetMeeting-Verkehr



(a) Assured Service



(b) Premium Service

Fig. 9. Verzögerung mit DiffServ für NetMeeting-Verkehr

Dienstgüte für den Telefonieverkehr, wenn nicht die Länge der Warteschlangen begrenzt wird. Die Pakete der IP-Telefonie-Datenströme werden hier zusammen mit einem Teil des Hintergrundverkehrs durch den Assured Forwarding PHB (AF) weitergeleitet. Der Rest des Hintergrundverkehrs ist Best Effort. Die Telefoniepakete sind alle als in-profile markiert. Die hohe Verzögerung gegenüber einfachem RED ist darin begründet, daß der AF1-Klasse nur 40% der Link-Bandbreite zur Verfügung stehen und die RIO-Warteschlange für AF1 dadurch langsamer abgearbeitet wird. Als Out-of-profile markierte Pakete werden zwar mit höherer Wahrscheinlichkeit verworfen, dennoch bewegt sich die Länge der Warteschlange an der oberen Grenze, da sie schon mit den in-profile Paketen (3.2 Mbit/s Hintergrundverkehr und 1.1 Mbit/s Telefonieverkehr) leicht überlastet ist. Die Paketverlustrate ist aber gegenüber RED merkbar auf 4% verbessert. In einem überlasteten Netz, in dem der Assured Service keine ungenutzte Bandbreite von anderen Klassen borgen kann, kann eine Übertragung ohne Paketverluste nur erreicht werden, wenn die für Assured Service reservierte Bandbreite ausreichend dimensioniert ist.

Durch Verkürzung der Warteschlangenlänge für jede Warteschlange innerhalb der CBQ-Klassen lassen sich ähnliche Ergenisse wie mit RED erreichen.

Die Verzögerungen der Telefoniepakete in den Warteschlangen kommen einerseits durch Pakete derselben AF-Klasse, die in der Warteschlange weiter vorne stehen, zustande, andererseits aber auch durch Pakete in den Warteschlangen anderer PHBs, also auch Best-Effort. Dies liegt daran, daß der AF PHB zwar bestimmte Ressourcen besitzt, die nicht von Best-Effort-Verkehr genutzt werden können, solange AF-Verkehr vorhanden ist, aber AF besitzt keine höhere Priorität gegenüber Best-Effort. AF garantiert also eine bestimmte Bandbreite mit geringer Verlustwahrscheinlichkeit aber keine geringe Verzögerung.

	Verlustrate	min. Verzögerung	max. Verzögerung	mittl. Verzögerung
Best Effort	59%	42.5	89.2	75.6
Best Effort mit RED	18%	12.9	44.0	25.9
Assured Service	4%	13.0	83.9	47.8
Premium Service	0%	8.2	13.4	10.6

TABLE I
UNTERSCHIEDLICHE DIENSTE FÜR CBR-VERKEHR IM VERGLEICH

	Verlustrate	min. Verzögerung	max. Verzögerung	mittl. Verzögerung
Best Effort	60%	52.8	90.7	82.0
Best Effort mit RED	22%	16.0	45.1	29.7
Assured Service	0.4%	21.9	87.3	62.3
Premium Service	0%	10.6	20.2	14.1

TABLE II
UNTERSCHIEDLICHE DIENSTE BEIM EINSATZ VON LIVELAN-VERKEHR

	Verlustrate	min. Verzögerung	max. Verzögerung	mittl. Verzögerung
Best Effort	65%	42.5	83.6	70.3
Best Effort mit RED	14%	9.5	42.8	23.5
Assured Service	1%	13.1	64.5	38.9
Premium Service	0%	6.8	13.2	10.0

TABLE III
UNTERSCHIEDLICHE DIENSTE BEIM EINSATZ VON NETMEETING-VERKEHR

Über den Assured Service hinaus bietet DiffServ den *Premium Service* an. Die Pakete werden durch den Expedited Forwarding PHB (EF) mit höherer Priorität bearbeitet als alle Pakete der anderen Klassen. Es wird von den Ingress-Routern nur soviel Premium-Service-Verkehr in das Netz gelassen, daß an jedem Router die Summe der Bandbreiten der aggregierten Premium-Service-Ströme kleiner gleich der Bandbreite des ausgehenden Links für diesen Verkehr ist. Dadurch kann die Warteschlange für Premium Service nie mehr Pakete enthalten als die Anzahl der eingehenden Links mit Premium-Service-Verkehr. Zusammen mit der höchsten Priorität ist dadurch minimale Verzögerung gewährleistet.

B. LiveLan-Verkehrsmuster

Die Ergebnisse der Simulationen mit LiveLan-Verkehr sind in den Abbildungen 6 und 7 zu sehen. Die Abbildungen zeigen wieder jeweils die Ende-zu-Ende-Verzögerung für die Verbindung zwischen den Knoten E1 und E2.

Wie beim CBR-Verkehr treten ohne DiffServ-Mechanismen hohe Verzögerungen, Verzögerungsschwankungen und Paketverluste auf. Abbildung 6(a) zeigt Verzögerungen zwischen 52.8 ms und 90.7 ms. Die Paketverlustrate beträgt 60%. Durch das RED-Warteschlangenverfahren kann auch hier die Qualität verbessert werden. Die mittlere Verzögerung sinkt von 82.0 ms auf nur noch 29.7 ms mit Schwankungen im Bereich von $[16.0\mathrm{ms}, 45.1\mathrm{ms}]$. Die Paketverlustrate ist mit 22% aber immer noch sehr hoch. Mit Assured Service kann die Paketverlustrate erheblich verringert werden, wobei allerdings die Verzögerung wie bei CBR steigt. Mit Premium Service gehen dagegen keine Pakete mehr verloren. Die mittlere Verzögerung kann durch Premium Service auf 14.1 ms verbessert werden. Abbildung 7 zeigt, daß mit Premium Service auch die Verzögerungsschwankungen gegenüber Assured Service deutlich verbessert werden.

C. NetMeeting-Verkehrsmuster

In den Abbildungen 8 und 9 sind die Ergebnisse derselben Simulationen mit NetMeeting-Verkehr statt LiveLan dargestellt.

Der Vergleich sowohl der Abbildungen 6 bis 9 als auch der Tabellen II und III zeigt ein sehr ähnliches Verhalten von LiveLan-Verkehr und NetMeeting-Verkehr für alle Dienstklassen.

Der Vergleich von LiveLan- und NetMeeting-Verkehr zeigt, daß die Ende-zu-Ende-Verzögerung von NetMeeting-Verkehr fast durchgehend niedriger liegt als die des LiveLan-Verkehrs. Dies ist auf die geringere Paketgröße von nur 78 Bytes gegenüber 566 Bytes zurückzuführen. Die reine Übertragungs- und Ausbreitungszeit zwischen den Endsystemen E1 und E2 beträgt 6.7 ms für 78 Bytes große Pakete und 10.3 ms für 566 Bytes große Pakete. Dieser Vorteil der kleineren Pakete wird aber teilweise wieder durch den stärker burstartigen Charakter des NetMeeting-Verkehrs zunichte gemacht.

IV. Zusammenfassung und Ausblick

Im Rahmen dieser Arbeit wurde ein ns-2-Simulationsmodell für IP-Telefonie über Differentiated Services implementiert. Mit diesem Modell wurde die erreichbare Dienstgüte für IP-Telefonie bei hoher Hintergrundlast mit verschiedenenen DiffServ-Service-Klassen untersucht. Die Ergebnisse der Simulationen zeigen, daß sich bei der gewählten Hintergrundlast IP-Telefonie mit Best Effort Service nicht in akzeptabler Qualität implementieren läßt. Mit RED ohne Differenzierung der Pakete verschiedener Verkehrsquellen können Pakteverluste und Verzögerung zwar reduziert werden, bleiben jedoch in den durchgeführten Simulationen bei Werten, die nicht für eine gute Sprachqualität ausreichen. Die durchgeführten Simulationen zeigen, daß durch Differentiated Services wie zu erwarten eine erhebliche Verbesserung sowohl der Paketverluste als auch der Ende-zu-Ende-Verzögerung möglich ist, wenn Assured Service oder Premium Service für IP-Telefonie verwendet wird. Dabei zeigt sich erwartungsgemäß, daß sich Premium Service gegenüber Assured Service neben der geringeren mittleren Verzögerung vor allem durch niedrigere Verzögerungsschwankungen auszeichnet.

Weitere Arbeiten werden sich mit der Untersuchung anderer Warteschlangen-Algorithmen als CBQ beschäftigen. Außerdem werden die Simulationen mit burstartigeren Verkehrsmodellen, wie sie z.B. für Videoquellen typisch sind, durchgeführt werden. Zur Zeit wird außerdem an der Implementierung eines Testbetts zur Demonstration der praktischen Umsetzbarkeit der Ergebnisse gearbeitet.

V. Danksagungen

Die Untersuchungen entstanden im Rahmen eines von der Siemens AG, ICN geförderten Projektes.

Für die Bereitstellung der aufgezeichneten Verkehrsströme für LiveLan und NetMeeting bedanken wir uns herzlich bei der Gruppe von Prof. Dr. J. Eberspächer an der TU München.

References

[1] International Telecommunication Union, “Visual telephone systems and equipment for local area networks which provide a non-guaranteed quality of service,” Recommendation H.323, Telecommunication Standardization Sector of ITU, Geneva, Switzerland, May 1996.

[2] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss, “An architecture for differentiated service,” Request for Comments (Informational) 2475, Internet Engineering Task Force, Dec. 1998.

[3] J. Heinanen, F. Baker, W. Weiss, and J. Wroclawski, “Assured forwarding PHB group,” Request for Comments (Proposed Standard) 2597, Internet Engineering Task Force, June 1999.

[4] V. Jacobson, K. Nichols, and K. Poduri, “An expedited forwarding PHB,” Request for Comments (Proposed Standard) 2598, Internet Engineering Task Force, June 1999, URL: ftp://ftp.ee.lbl.gov/papers/ef_phb.pdf.

[5] Sally Floyd and Van Jacobson, “Link-sharing and resource management models for packet networks,” *IEEE/ACM Transactions on Networking*, vol. 3, no. 4, Aug. 1995.

[6] K. Nichols, V. Jacobson, and L. Zhang, “A two-bit differentiated services architecture for the internet,” Request for Comments (Informational) 2638, Internet Engineering Task Force, July 1999.

[7] Mark. E. Crovella and Bestavros Azer, “Self-similarity in world wide web traffic: Evidence and possible causes,” *IEEE/ACM Transactions on Networking*, vol. 5, no. 6, pp. 835–846, Dec. 1997.

IPTEL2000 Workshop

IP-Telefonie über Differentiated Services

U.Thürmann, M.Zitterbart

Institut für Betriebssysteme und Rechnerverbund

TU Braunschweig

{thuerman,zit}@ibr.cs.tu-bs.de

IP-Telefonie über Differentiated Services

Szenario

- Endgeräte für IP-Telefonie in LANs
- Verbindung der LANs durch IP-Backbone (Internet, Intranet)

Problem

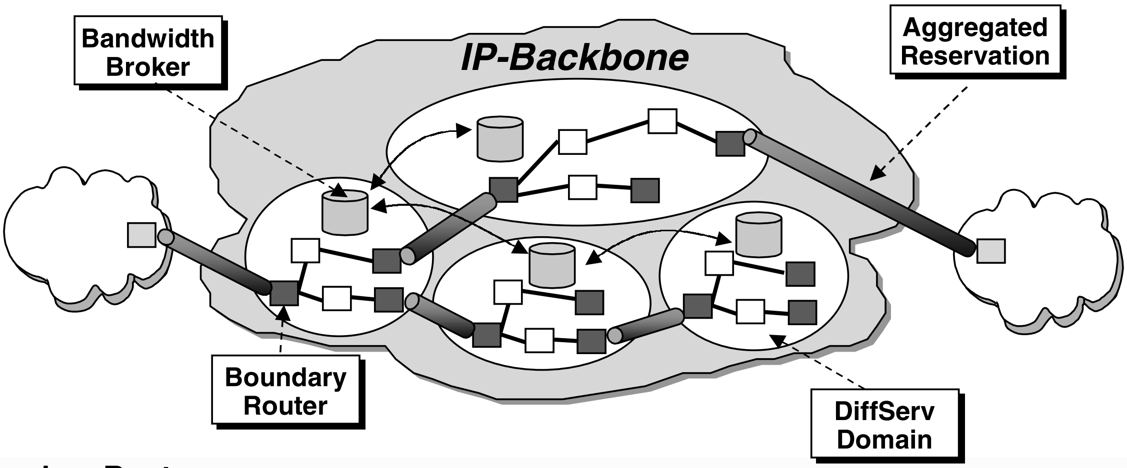
- Hohe Last im Backbone (Paketverluste, Verzögerung)

Ansatz

- Differentiated Services im IP-Backbone
- Ende-zu-Ende Dienstgüte für Telefonieverkehr

⇨ Analyse von Scheduling im DiffServ-Netz

Basisarchitektur Differentiated Services



Boundary Router

- Zugangskontrolle, Klassifikation und Markierung mit DiffServ Codepoint (DSCP)

Innere Router

- Auswahl eines Per-Hop-Behaviors (PHB) zur Weiterleitung anhand des DSCP

Bandwidth Broker

- Aushandlung von Reservierungen

Analyse von Differentiated Services für IP-Telefonie

Betrachtung von

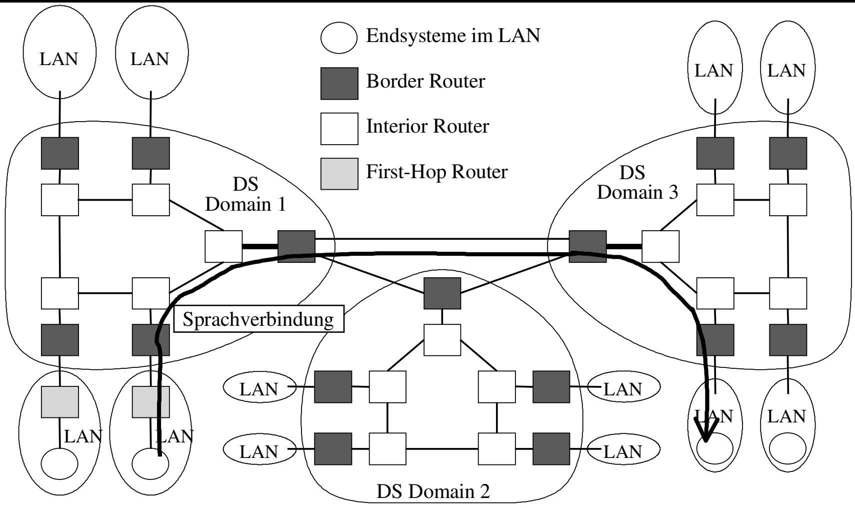
- Verschiedene Verkehrsmustern (Sprache, Hintergrundverkehr)
- Derzeit diskutierten DiffServ Services (Best Effort, Assured Service, Premium Service)
- Warteschlangenalgorithmen (RED, RIO, CBQ, WFQ, ...)

⇨ Simulative Untersuchungen mit ns-2

Ziel

- Bewertung verschiedener DiffServ-Mechanismen
  für den Transport von Sprachverkehr über IP-Netze

Simulationsmodell

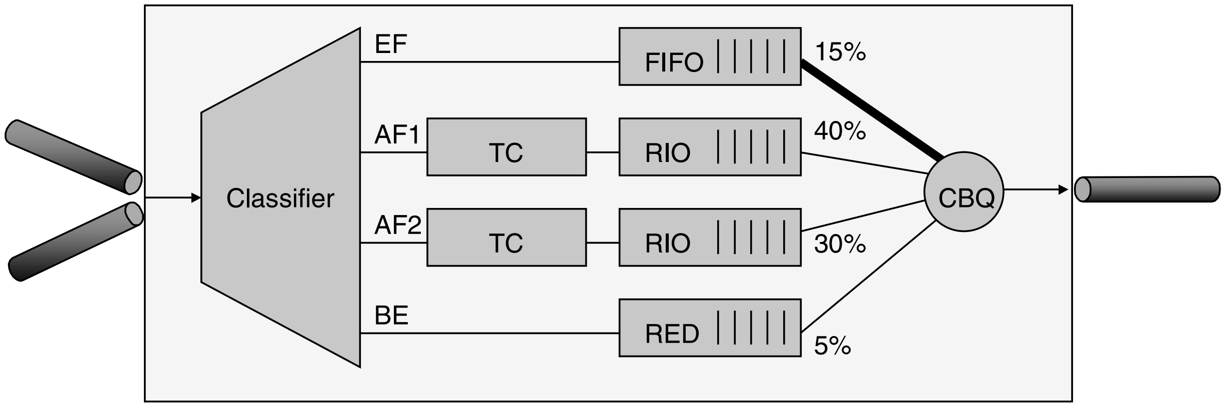


- Simulative Untersuchung von IP-Telefonie über DiffServ
  - Klassifikation, Zugangskontrolle,und Markierung im First-hop Routers
  - Verschiedene DiffServ Per-Hop-Behaviors (PHBs) für Sprachverkehr
  - Mehrere Warteschlangenverfahren zur Implementierung von Differentiated Services
  - CBR-, LiveLan- und NetMeeting-Verkehr

Verkehrscharakteristika

- LiveLan-Verkehr (Messungen an der TU München)
  - 48 Verbindungen, jede passiert 2 DiffServ Domänen
  - 512 Bytes Paketgröße, 50-80 ms/Paket, 70 kbit/s mittlere Datenrate
- Hintergrundverkehr
  - 480 Pareto-verteilte Quellen: 240 Best-Effort, 120 AF1, 120 AF2
  - 1000 Bytes Paketgröße, 100 kbit/s mittlere Datenrate
  - je 800 kbit/s AF1- bzw. AF2-in-profile-Hintergrundverkehr pro LAN, 200 kbit/s out-of-profile
- Höchste Verkehrskonzentration zwischen innerem Router und Boundary Router
  - 16 Sprachverbindungen, 1.1 Mbit/s: EF, AF1, AF2
  - 120 Hintergrundquellen: 3.2 Mbits/s in-profie, 0.8 Mbit/s out-of-profile für AF1 und AF2

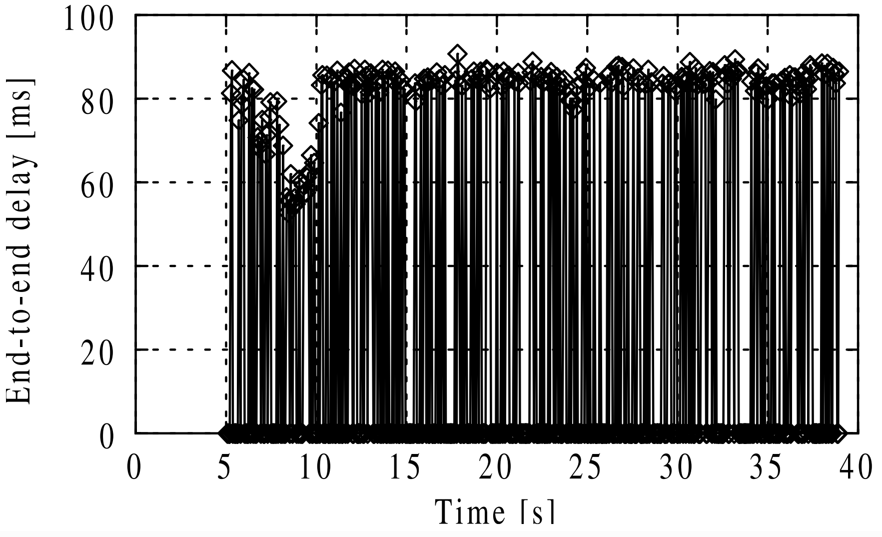
Aufbau der DiffServ-Router



Komponenten der DiffServ Router

- Classifier: Behavior Aggregate (BA) und Multifield (MF)
- Traffic Conditioner (TC): Meter, Marker & Shaper
- Warteschlangen, Scheduling-Algorithmen

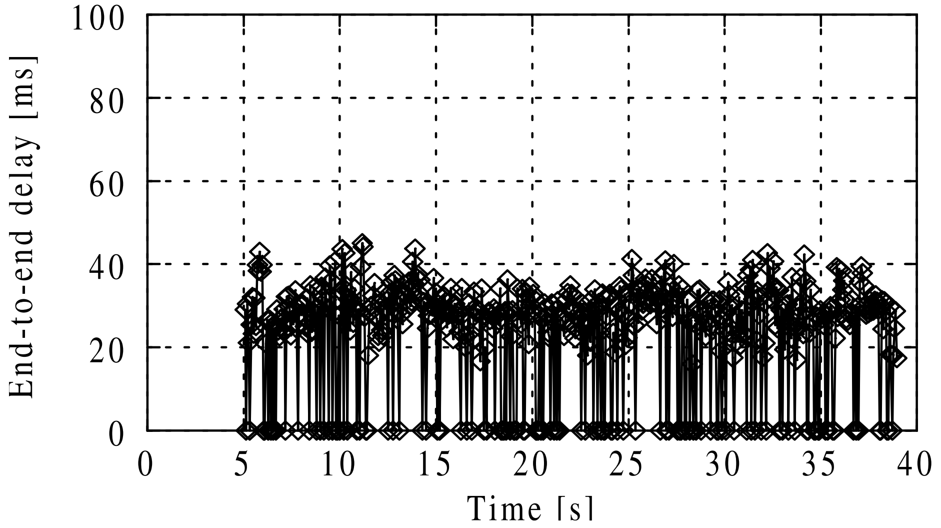
LiveLan-Verkehr mit Best Effort



- Best Effort
  - hohe Paketverlustrate
  - hohe Verzögerung und Jitter
- Ergebnis: ungeeignet für Sprachverkehr über IP

Paketverluste	60%
mittl. Verzögerung	82.0
min. Verzögerung	52.8
max. Verzögerung	90.7

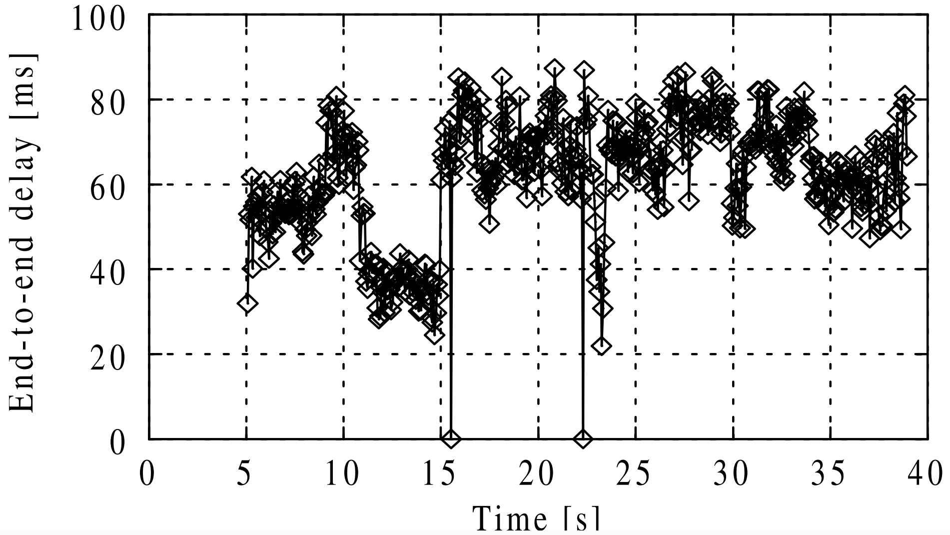
LiveLan-Verkehr mit RED



- Best Effort mit RED-Warteschlangen
  - Verbesserung von Verlustrate, Verzögerung und Jitter
- Ergebnis: ungeeignet für Sprachverkehr über IP

Paketverluste	22%
mittl. Verzögerung	29.7
min. Verzögerung	16.0
max. Verzögerung	45.1

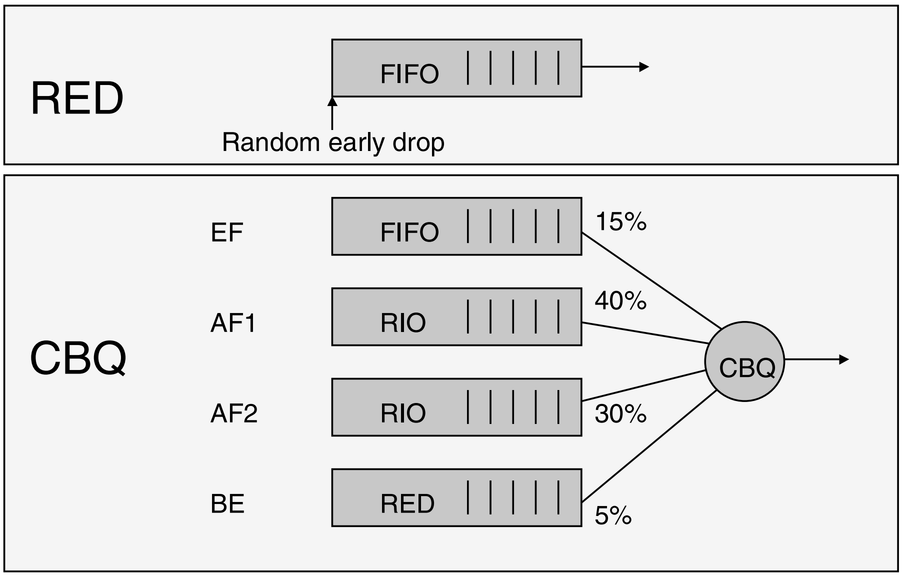
LiveLan-Verkehr über Assured Service



- Assured Service mit CBQ
  - Verbesserte Paketverlustrate
  - Aber große Verzögerung und Jitter
- Ergebnis: nur bedingt geeignet für Telefonie über IP

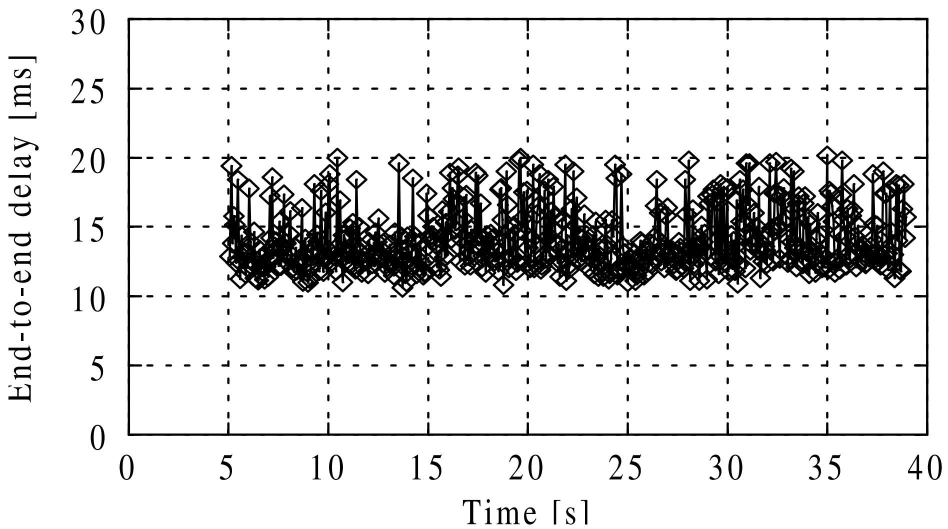
Paketverluste	0.4%
mittl. Verzögerung	62.3
min. Verzögerung	21.9
max. Verzögerung	87.3

RED und CBQ



- Bei CBQ besitzt jede Klasse eine eigene, unabhängige Warteschlange
- In der Summe mehr Puffer für Pakete im Router

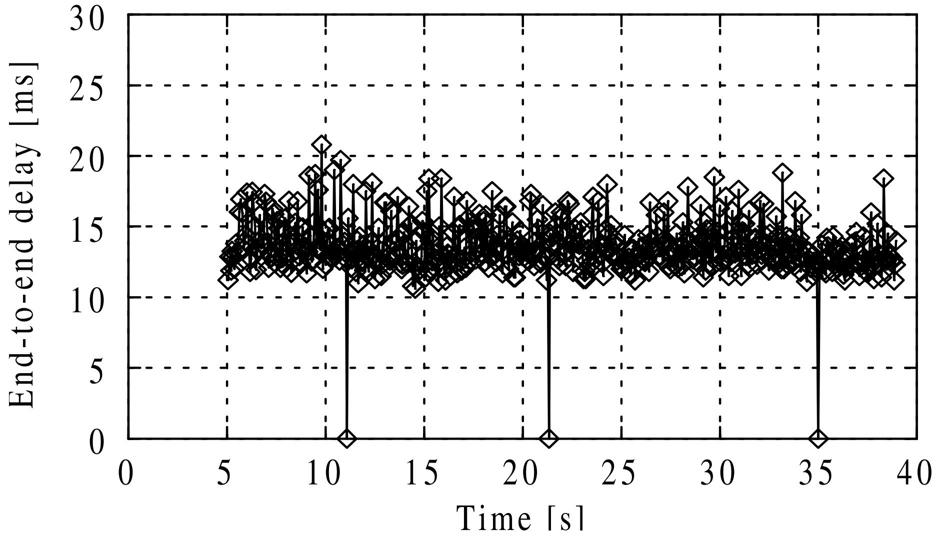
LiveLan-Verkehr über Premium Service



- Premium Service mit CBQ
  - Niedrige Verzögerung und Jitter durch hohe Priorität
  - keine Paketverluste
- Gut geeignet für voice over IP

Paketverluste	0%
mittl. Verzögerung	14.1
min. Verzögerung	10.6
max. Verzögerung	20.2

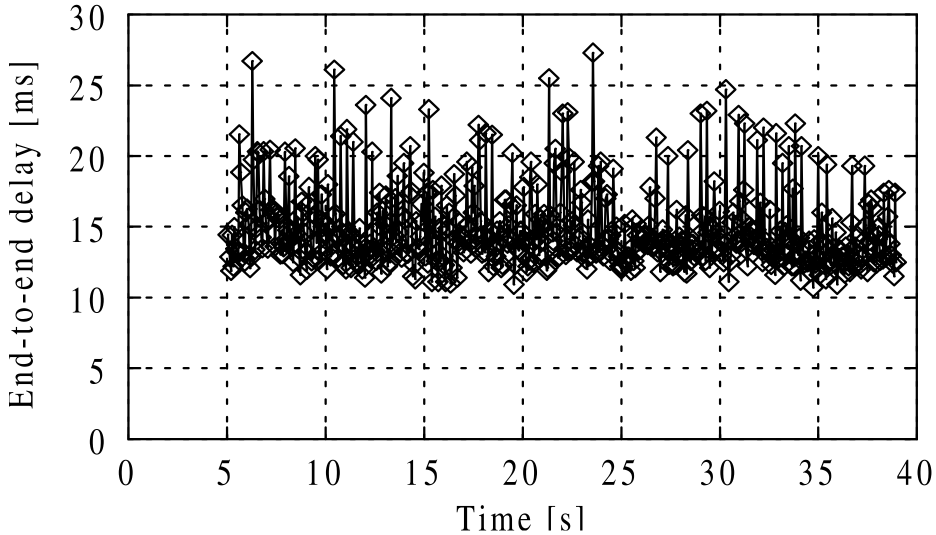
LiveLan-Verkehr mit WFQ



- Assured Service mit WFQ
  - Out-of-profile-Verkehr wird als Best-Effort remarkiert
  - Dadurch Reihenfolgevertauschung möglich

Paketverluste	0.6%
mittl. Verzögerung	13.6
min. Verzögerung	10.6
max. Verzögerung	20.8

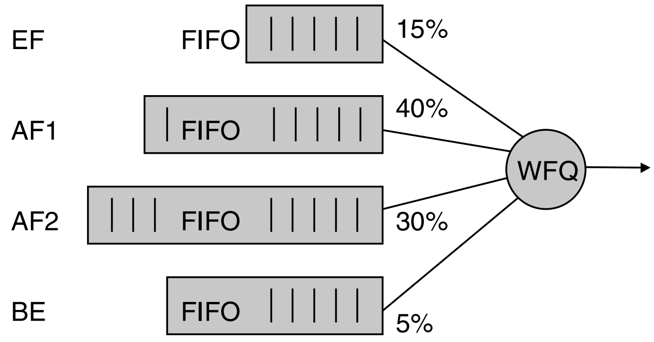
LiveLan-Verkehr mit WFQ



- Premium Service mit WFQ Warteschlangenverfahren
  - keine Priorität für Premium Service
  - höherer Jitter als mit Assured Service

Paketverluste	0%
mittl. Verzögerung	14.8
min. Verzögerung	10.7
max. Verzögerung	27.3

Verzögerung mit WFQ



- keine Prioritäten
- keine inneren Warteschlangen mit eigenem Algorithmus (z.B. RED, RIO)
- gemeinsamer Puffer für alle flows, schlechtere Trennung von Ressourcen
- Pakete in Klassen mit niedriger Bandbreite erfahren höhere Verzögerung

  finish-time = size / (link-bandwidth * weight)
- Ohne Modifikationen nicht gut geeignet für DiffServ

Zusammenfassung und Ausblick

Zusammenfassung

- Deutliche Verbesserung der Dienstgüte für IP-Telefonie durch Einsatz von Differentiated Services Mechanismen gegenüber Best Effort
- Assured Service nur bedingt geeignet
  - gute Paketverlustrate aber große Verzögerungen
- WFQ allein schlecht geeignet
  - keine Prioritäten für Premium Service
  - keine getrennten Warteschlangen (abhängig von der Implementierung)
- Gute Ergebnisse mit CBQ

Weitere Arbeiten

- Implementierung eines Demonstrators auf Basis von Linux
- Design eines DiffServ Bandwidth Brokers

Effiziente Dienstqualitätsunterstützung für IP Telefonie durch selektive Paketmarkierung

Henning Sanneck, Nguyen Tuong Long Le, Georg Carle
{sanneck,le,carle}@fokus.gmd.de
GMD Fokus, Kaiserin-Augusta-Allee 31, D-10589 Berlin

Zusammenfassung— **Mit der wachsenden Bedeutung des Internets wurde Dienstqualitätsunterstützung für multimediale Echtzeitdienste über paket-vermittelnde Netze zu einem wichtigen Forschungsgebiet. Dienstqualitätsunterstützung umfasst Ressourcenreservierung gemäß dem "Internet Integrated Services" Modell, womit sich feste Grenzen für Paketverluste und Verzögerung garantieren lassen, hierzu jedoch Statusinformationen bezüglich jedes Datenstroms an jedem Netzwerkelement benötigt werden. Hieraus ergeben sich Skalierbarkeitsprobleme, die alternative Ansätze zur Dienstqualitätsunterstützung erforderlich machen. Eine weitere Forschungsrichtung ("Differentiated Services") stellt Mechanismen zur Markierung und differenzierten Behandlung von Paketen zur Verfügung. Da hierbei Statusinformationen und die Durchsetzung bestimmter Dienstqualitäten nur für aggregierten Datenströme erforderlich sind, wird eine bessere Skalierbarkeit gewährleistet.**

Zusätzlich zur Möglichkeit der Behandlung von aggregierten Datenströmen hat eine Dienstqualitätsunterstützung per Paket auch die wünschenswerte Eigenschaft, dass eine Anwendung die eigene Dienstqualität per Paket (und somit per ADU - Application Data Unit) kontrollieren kann. Wir untersuchen das Potenzial dieser Eigenschaft für Sprachübertragung, da neuere Arbeiten gezeigt haben, dass bestimmte Segmente eines Sprachsignals essentiell für die Sprachqualität sind, während andere, im Fall eines Paketverlusts, relativ gut am Empfänger aus schon angekommenen Daten extrapoliert werden können. Das trifft insbesondere für moderne segment-basierte ("frame-based") Sprachkodierer (ITU-T G.729 und G.723.1) zu, die einen internen Fehlerverschleierungsmechanismus ("loss concealment") besitzen. Somit können Sender weniger konservativ in ihren Anforderungen bezüglich der Qualität des Netzwerkdienstes sein, bzw. die Anzahl der gleichzeitig akzeptierbaren Verbindungen kann gesteigert werden.

In dieser Arbeit analysieren wir zunächst den Fehlerverschleierungsmechanismus des G.729 Dekoders. Darauf aufbauend entwickeln wir Mechanismen, die selektiv Pakete auf eine höhere Netzwerkpriorität markieren und zwar abhängig von den Sprachsignaleigenschaften und der somit zu erwartenden Verschleierungsqualität. Mit objektiven Methoden zur Bestimmung der Sprachqualität (ITU-T P.861A und EMBSD) lässt sich zeigen, dass bei Markierung bzw. Bevorzugung von ca. der Hälfte der Pakete eines Datenstroms nahezu dieselbe Sprachqualität erreicht werden kann wie bei vollständiger Markierung dieses Datenstroms. Die verschiedenen Paketmarkierungen werden innerhalb des Netzwerkes z.B. mit der "Differentiated Services" Architektur in unterschiedliche Behandlung der Pakete umgesetzt.

I. Einführung

In den letzten Jahren ist das Interesse in der Öffentlichkeit sowie in der Forschungswelt an Internet-Sprachübertragung ("Voice over IP", "Internet Telephony") sprunghaft gestiegen. IP Telefonie hat das Potenzial, zusammen mit anderen Internet Anwendungen interaktive multimediale Dienste zu ermöglichen, die nicht (oder nur sehr schwer) im traditionellen Telefonnetz zu realisieren wären. Ausserdem kann mittlerweile die rechenaufwendige, hochkomprimierende Sprachkodierung mit allgemein verfügbarer Hardware in den Endsystemen beim Benutzer durchgeführt werden. Beispiele für solche Sprachkodierer sind die segment-orientierten ("frame-based") Kodierer G.723.1 ([1]) und G.729 ([2]), die sehr attraktiv für den Einsatz in IP-Netzen sind, da sie die übliche Telefonqualität bei wesentlich niedrigeren Datenraten (5.3/6.3 kBit/s bzw. 8 kBit/s) als mit konventionelle PCM-Kodierung (64 kBit/s) ermöglichen. Somit können die Anforderungen an die Netzwerkressourcen stark reduziert werden.

Paket-vermittelnde Netze ohne Mechanismen zur Ressourcenreservierung wie das heutige Internet basieren auf dem "Best Effort"-Prinzip, das keine Garantien über zu erwartenden Paketverluste, bzw. Paketverzögerungen zulässt. Sprachdatenpakete können verworfen werden, wenn Router überlastet sind oder wenn sie zu spät den Empfänger erreichen (d.h. der Abspielzeitpunkt der Audiodaten ist schon vorüber). Zusätzlich (unter Berücksichtigung der rückwärtsadaptiven Sprachkodierungsalgorithmen der G.723.1 und G.729 Quellenkodierer) resultieren Paketverluste in einer Störung der Synchronisierung zwischen Koder und Dekoder. Somit treten Signalstörungen nicht nur in dem Zeitabschnitt auf, der durch das verlorengegangene Paket repräsentiert wird, sondern pflanzen sich in die folgenden Segmente solange fort, bis der Dekoder schliesslich wieder synchron zum Koder läuft. Um dieses

Problem abzumildern verfügen die beiden genannten Kodierer über einen internen (d.h. für das Kodierverfahren spezifischen) Fehlerverschleierungsalgorithmus.

In [3] haben wir ein Verfahren zur Spracheigenschafts-basierten Vorwärtsfehlerkorrektur (Speech Property-Based Forward Error Correction, SPB-FEC) vorgestellt. Dort werden essentielle Teile des Sprachsignals durch Vorwärtsfehlerkorrektur (FEC) geschützt, wohingegen Verluste, die andere Teile des Signals betreffen, durch die interne Fehlerverschleierung behandelt werden. Dieses Verfahren ermöglicht bei gleichbleibender Sprachqualität die notwendigen zusätzlichen Daten zur Vorwärtsfehlerkorrektur zu reduzieren. Jedoch kommen auch hier die allgemeinen Probleme von FEC-Verfahren zum Tragen: die Übertragung redundanter Daten erhöht die allgemeine Netzwerklast und fördert somit das Auftreten von Stausituationen. Darüber hinaus können solche Verfahren die Wahrscheinlichkeit sehr wichtige Pakete zu verlieren nur reduzieren; sie können sie aber nicht nahezu ausschliessen. Ausserdem sind, wenn innerhalb eines Zeitintervalls keine Pakete verlorengehen, sämtliche redundanten Daten nutzlos, die innerhalb dieses Intervalls übertragen wurden.

Deshalb wenden wir in dieser Arbeit das Konzept des selektiven Schutzes einiger Pakete aufgrund von Anwendungsbedürfnissen auf die Bevorzugung von Paketen (durch Priorisierung) innerhalb des Netzwerkes an.

II. G.729 Fehlerverschleierung

Der G.729 Sprachkodierer arbeitet nach dem CS-ACELP (Conjugate Structure Algebraic Code Excited Linear Prediction) Prinzip bei einer Datenrate von 8 kbit/s. Das Eingabeformat ist lineare PCM (16 bit) abgetastet mit 8 kHz. Das Kodierverfahren beruht auf einer Modellierung der menschlichen Spracherzeugung. In diesem Modell werden der Hals und Mund durch einen linearen Filter (Synthesefilter) dargestellt. Sprachsignale werden produziert, indem dieses Filter durch einen Anregungsvektor angesteuert wird. Beim G.729 Koder dauert ein Sprachsegment[1] 10ms korrespondierend zu 80 PCM Abtastwerten. Für jedes Segment analysiert der G.729 Koder die Eingabedaten und extrahiert die CELP (Code Excited Linear Prediction) Modellparameter wie Filterkoeffizienten der linearen Prädiktion und Anregungsvektoren. Der Ansatz zur Bestimmung der Filterkoeffizienten wird als "Analyse durch Synthese" bezeichnet. Der Koder durchsucht den Bereich möglicher Parameter; bei jeder Suche wird eine Dekodierung durchgeführt und der dekodierte Signalabschnitt mit dem originalen Signalabschnitt verglichen. Der Parametersatz, bei dem sich originales und synthetisiertes Signal am ähnlichsten sind, wird verwendet, kodiert und zum Empfänger übertragen. Beim Empfänger wird dieser Parametersatz verwendet, um das ursprüngliche Sprachsignal wieder herzustellen.

[1] Wir verwenden den Begriff "Segment" ("frame") für die Einheit des Kodier-/Dekodierprozesses und "Paket" für die Einheit der Übertragung. Ein Paket beinhaltet üblicherweise mehrere Segmente.

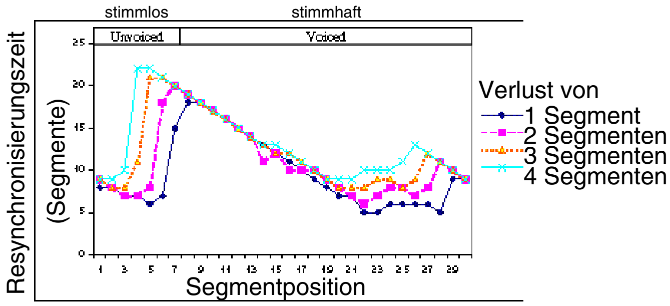


Abbildung 1. Resynchronisierungszeit (in Segmenten) des G.729 Dekoders nach Verlust von k aufeinanderfolgenden Segmenten ($k \in [1, 4]$) in Abhängigkeit der Segmentposition.

In [4] untersuchte Rosenberg das Verhalten des G.729 Dekoders in Bezug auf Segmentverluste mit Hilfe der benötigten Resynchronisierungszeit zwischen Koder und Dekoder. Als Ergebnis wurde festgehalten, dass die Energie des Fehlersignals bei einem Verlust von zwei aufeinanderfolgenden Segmenten stark ansteigt und die subjektive Sprachqualität stark fällt. Bei Verlust von weiteren Segmenten findet eine weitere, aber nicht so drastische, Verschlechterung der Sprachqualität statt. Daraus wurde gefolgert, dass nur isolierte Segmentverluste gut durch den G.729 Dekoder verschleiert werden können.

Wir führen nun folgendes Experiment durch: die Resynchronisierungszeit zwischen Dekoder und Koder nach Verlust von k aufeinanderfolgenden Segmenten wird gemessen. Eine Resynchronisierung wird angenommen, wenn die Energie des Fehlersignals unterhalb ein Prozent der Energie des dekodierten Signals ohne Segmentverluste fällt (das ist äquivalent zu einer Schwelle des Signal-Rausch-Verhältnisses (SNR) von $20dB$). Die Fehlersignalenergie (und somit das SNR) wird pro Segment berechnet. Abbildung 1 zeigt die Resynchronisierungszeit (ausgedrückt in der Anzahl benötigter Segmente zur Überschreitung der Schwelle) in Abhängigkeit der Position des Verlustes für verschiedene Werte von k. Das verwendete Sprachbeispiel stammt von einem männlichen Sprecher, wobei innerhalb des achten Segments ein Übergang von stimmloser zu stimmhafter Sprache (unvoiced/voiced: uv) stattfindet.

Aus Abbildung 1 ersieht man, dass die Position eines Segmentverlustes einen großen Einfluss auf die Verschlechterung des Signals hat[2], wohingegen die Verschlechterung nicht sehr von der Anzahl der an dieser Stelle verlorengegangen Segmente k abhängig ist. Der Verlust von stimmlosen Abschnitten scheint nur geringe Auswirkungen zu haben. Verluste von stimmhaften Segmenten haben einen grösseren Einfluss auf die Signalqualität und der Dekoder braucht somit mehr Zeit, um sich zu resynchronisieren. Jedoch verursacht der Verlust von Segmenten an einem stimmlos/stimmhaft-Übergang eine extreme Verschlechterung des Signals. Das Experiment wurde mit ähnlichen Ergebnissen für verschiedenes Sprachmaterial (von männlichen und weiblichen Sprechern) durchgeführt.

Unter Berücksichtigung des benutzten Kodierverfahrens kann der obige Effekt wie folgt erklärt werden: Da stimmhafte Laute eine höhere Signalenergie haben als stimmlose führt der Verlust von stimmhaften Lauten entsprechend zu einer grösseren Verschlechterung. Allerdings kann der Dekoder durch den periodischen Signalverlauf bei stimmhaften Lauten den Fehler relativ gut verschleiern, sobald er über genügend Statusinformationen verfügt. Der Dekoder versagt bei der Verschleierung von stimmhaften Lauten unmittelbar nach einem stimmlos / stimmhaft-Übergang, da er diese Verschleierung mit Filterkoeffizienten und Anregungsvektoren für einen stimmlosen Laut durchführt. Zusätzlich ist es ausserdem so, dass der G.729 Koder für die Prädiktion der Line Spectral Pairs (LSP) einen Tiefpassfilter benutzt und nur die Differenz zwischen den echten und den Prädiktor-Werten überträgt. Das führt dazu, dass eine sehr lange Zeit benötigt wird den Dekoder zu resychronisieren, sobald er einmal nicht in der Lage war einen angemessenen Filter der linearen Prädiktion zu berechnen.

III. Spracheigenschaftsbasierte selektive Paketmarkierung

Das Ergebnis über die Fähigkeit des G.729 Dekoders Paketverluste zu verschleiern wird nun verwendet um einen Algorithmus zur selektiven Paketmarkierung zu entwickeln: Speech Property-Based Selective Packet Marking (SPB-MARK, [5]). Das SPB-MARK-Verfahren konzentriert die Pakete mit höherer Priorität auf die Sprachsegmente, die essentiell für die Sprachsignalqualität sind und verlässt sich auf die Fehlerverschleierung des Dekoders für andere Signalabschnitte.

Abbildung 2 zeigt den einfachen (Pseudo Code) Algorithmus, der verwendet wird, um einen stimmlos / stimmhaft-Übergang ("uv transition") zu erkennen und die stimmhaften Segmente nach diesem Übergang besonders zu schützen. Die Prozedur *analysis()* wird eingesetzt, um einen Block aus k Segmenten als stimmhaft, stimmlos oder uv zu klassifizieren. Die Prozedur *send()* sendet einen Block von k Segmenten als ein Paket mit der entsprechenden Netzwerkpriorität (entweder "+1" oder "0"). N ist ein vordefinierter Wert, der festlegt, wieviele Abschnitte bei Beginn eines stimmhaften Signals geschützt werden. Unsere Simulationen haben gezeigt, dass der Bereich von 10 bis 20 für N angemessen ist. In der Simulation, die in Abschnitt IV vorgestellt wird, haben wir $k = 2$ als einen typischen Wert für interaktive Sprachübertragung im Internet ($20ms$ Audiodaten pro Paket) angenommen. Ein höherer k-Wert würde den relativen Paket-Headeroverhead reduzieren, gleichzeitig aber auch die Verzögerung durch die Paketisierung erhöhen sowie die Klassifizierung am Sender und die Verschleierung am Empfänger im Falle eines Paketverlustes schwieriger machen (da die Länge der Lücke im Signal grösser wird).

```
protect = 0

foreach (k frames)

    classify = analysis(k frames)
    if (protect > 0)
        if (classify == unvoiced)
            protect = 0
            send(k frames, "0")
        else
            send(k frames, "+1")
            protect = protect-k
        endif
    else
        if (classify == uv_transition)
            send(k frames, "+1")
            protect = N - k
        else
            send(k frames, "0")
        endif
    endif

endfor
```

Abbildung 2. SPB-MARK Pseudo Code

IV. Evaluierung des spracheigenschaftsbasierten Markierungsverfahrens

Wir verwenden ein einfaches Markov-Modell mit einem Zustand (Bernouilli-Modell), um das Verhalten des Netzwerks in Bezug auf jede Prioritätsklasse zu beschrei-

[2]Obwohl SNR-Werte oft nicht gut mit subjektiver Sprachqualität korrelieren weisen die starken Unterschiede der SNR-basierten Resynchronisierungszeitmessung auf einen signifikanten Einfluss auf die Qualität hin.

ben. "Best effort" Pakete (gekennzeichnet durch "0" in Abbildung 3) werden mit einer Wahrscheinlichkeit p ("NO MARK" in Abbildung 3) verworfen. Ist ein Datenstrom vollständig geschützt (alle Pakete sind als "+1" markiert) dann ist die Wahrscheinlichkeit des Verwerfens gliech Null. Pakete eines Datenstromes, der das SPB-MARK-Verfahren benutzt, sehen entweder keinen Verlust ("+1", Abb. 2) oder die Verlustwahrscheinlichkeit p ("0", Abb. 2). Zum Vergleich führen wir einen Markierungsalgorithmus "ALT-MARK" ein, der Pakete alternierend als "0" bzw. "+1" markiert. Das resultierende Gesamtsystem könnte nun auch mit einem Markov-Modell beschrieben werden (z.B. ein Gilbert-Modell mit zwei Zuständen). Da jedoch die verschieden Markierungsmechanismen einige Parameter dieses Modells verändern[3], wird nur der interne Systemparameter p zum Vergleich der verschiedenen Ansätze verwendet. Schliesslich werden auch Resultate für Mechanismen, die wir als "differentielles" Markieren bezeichnen, vorgestellt. Differentiell bedeutet hier, dass jedes Paket, welches mit einer höheren Priorität versendet wird, durch ein Paket mit einer Priorität ("-1") niedriger als "best effort" ("0") kompensiert werden muss. Wir vergleichen zwei Varianten dieses Ansatzes: bei der ersten Methode ("ALT-DIFFMARK") werden Pakete alternierend als "-1", bzw. "+1" markiert. Die zweite Variante ("SPB-DIFFMARK") verwendet auch den SPB Markierungsalgorithmus, jedoch wird nach einer Folge von "+1"-Paketen unmittelbar eine entsprechend lange Folge von "-1"-Paketen gesendet[4]. Die Wahrscheinlichkeit des Verwerfens eines "-1"-Paketes ist $2p$. Somit wird die Verlustwahrscheinlichkeit gesehen über Zeitintervalle, die lang im Vergleich zu der Dauer einer "+1"-Folge sind, sich nicht von der des "best effort"-Datenverkehres unterscheiden. Warteschlangenkontrollalgorithmen die solch ein Netzwerkverhalten realisieren, werden in [6] beschrieben.

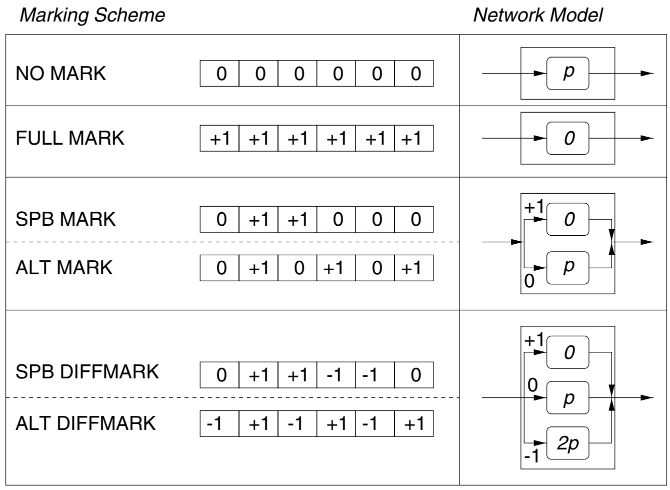


Abbildung 3. Markierungsmethoden / Netzwerkmodelle.

Es war sinnvoll eine segment-orientierte SNR-Berechnung ([7]) in Abschnitt II zu verwenden, da dort ein System (G.729 ohne zusätzlichen Schutz) unter verschiedenen Verlustsituationen untersucht wurde. Nun jedoch werden wir mehrere Systeme (G.729 mit permanenten und verschiedenen temporären Schutzmechanismen) unter gleichen Verlustsituationen untersuchen. Das System mit permanenter Priorisierung wird das Signal ohne jede Verschlechterung reproduzieren (unter der Annahme, dass kein Verlust eines Paketes mit hoher Priorität auftritt) wohingegen die anderen Systeme teilweise den internen Fehlerverschleierungsmechanismus des G.729 Dekoders einsetzen. Dieser ist in der Lage eine geringe Verschlechterung des Signals unter den in Abschnitt II genannten Vorausssetzungen zu gewährleisten. Allerdings kann die Relation der resultierenden Sprachqualitäten nicht angemessen durch einen SNR-Wert dargestellt werden. Zum Beispiel wird die Sprachqualität durch das graduelle Absenken der Verstärkungskoeffizienten des letzten korrekt empfangenen Segmentes bei der Fehlerverschleierung verbessert. Dieses bringt aber eine starke mathematische Abweichung des wiederhergestellten Signals vom Originalsignal mit sich.

Im Gegensatz zu SNR-Ansätzen versuchen neuartige Methoden der objektiven Sprachqualitätsmessung durch Modellierung des menschlichen Gehörs die subjektive Qualität zu schätzen. In unserer Evaluierung verwenden wir zwei objektive Qualitätsmessmethoden: "Enhanced Modified Bark Spectral Distortion" (EMBSD, [8]) und "Measuring Normalizing Blocks" (MNB, [9]), welches im Appendix II des ITU-T Standards P.861 ([10]) beschrieben wird. In der Literatur wird von einer sehr hohen Korrelation dieser objektiven Qualitätsmaße mit subjektiven Tests gesprochen. Die objektiven Messwerte können mittels einfacher Transformationen in mit den Ergebniswerten solcher Tests (MOS - Mean Opinion Score) vergleichbare Werte überführt werden. Die Messmethoden gelten als für den Einsatz in Bezug auf durch Bitfehler und Segmentverluste gestörte Sprache geeignet ([8], [10]).

Bei MNB wird der Unterschied in der Wahrnehmung zwischen dem Test- und dem Referenzsignal in verschiedenen Zeit- und Frequenzabschnitten gemessen. Der Wahr-

[3] Bei der "ALT-MARK"-Methode ist z.B. die bedingte Verlustwahrscheinlichkeit gleich Null.

[4] Eine Variable, die die Anzahl der noch zu sendenden "-1"-Pakete angibt, wird verwendet für den Fall, dass der SPB-Algorithmus die nächste "+1"-Folge bestimmt, bevor alle zur Kompensation nötigen "-1"-Pakete gesendet wurden.

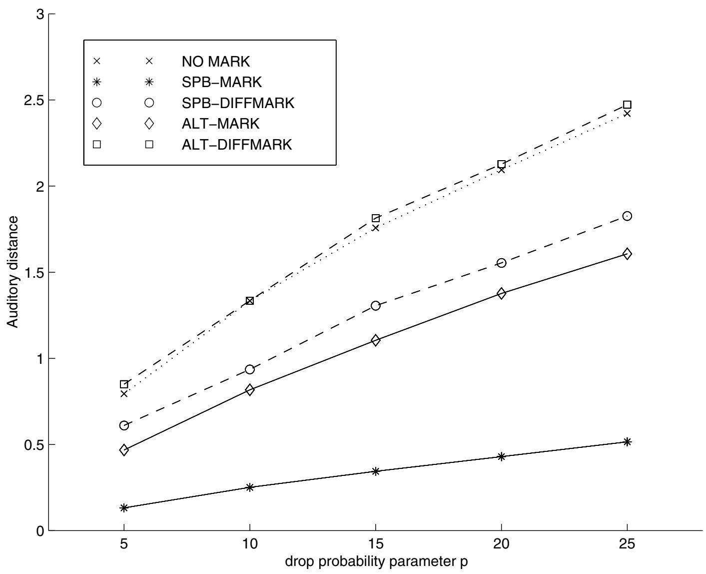


Abbildung 4. Wahrnehmungsunterschied ("Auditory Distance") der Markierungsmethoden evaluiert durch MNB.

nehmungsunterschied, als "Auditory Distance (AD)" bezeichnet, zwischen den genannten Signalen ist eine Linearkombination dieser Messergebnisse, wobei die Gewichtungsfaktoren die Eigenschaften des Gehörs modellieren. Je höher der AD-Wert, desto unterschiedlicher ist die Wahrnehmung der zwei Signale und umso schlechter ist die Sprachqualität des Testsignals. Abbildung 4 zeigt die "Auditory Distance"[5].

Die Ergebnisse für die ungeschützten Datenströme ("NO MARK") zeigen, dass mit steigendem p im Netzwerkmodell (und somit steigender Paketverlustrate und Verlustkorrelation) der AD-Wert monoton ansteigt, d.h. die Sprachqualität des dekodierten Signals sinkt. Wenn die "NO MARK"-Resultate mit den Kurven der Markierungsmethoden verglichen werden, sieht man, dass das dekodierte Sprachsignal ohne Markierung den höchsten "Auditory Distance"-Wert und somit die schlechteste Sprachqualität hat. Die "ALT-MARK"-Methode (50% der Pakete sind markiert) verbessert die Qualität. Der AD-Wert des SPB-MARK Verfahrens (mit $40,4\%$ markierten Paketen[6]) ist erheblich geringer und liegt sogar nahe der Qualität des verlustfreien dekodierten Signals ($AD = 0$). Dies zeigt auch, dass durch den Schutz des gesamten Datenstroms also nur eine geringe Verbesserung erzielt wird. Das differentielle Markierungsverfahren "SPB-DIFFMARK" liefert eine verbesserte Sprachqualität, obwohl nur ein Netzwerkdienst verwendet wird, der über längere Zeiträume gesehen mit einem "best effort"-Dienst gleichzusetzen ist. Die "ALT-DIFFMARK"-Methode unterscheidet sich hingegen nicht von dem "best effort"-Fall. Diese Ergebnisse bestätigen den Ansatz der spracheigenschaftsbasierten Markierung, der nicht alle Pakete mit einer höheren Priorität versieht, sondern eine kleinere Anzahl von Segmenten, die essentiell für die Sprachqualität sind, schützt.

[5] Für das EMBSD Verfahren wurden ähnliche Ergebnisse erzielt, die in der erweiterten Version dieser Arbeit ([11]) enthalten sind.

[6] Wir haben mit anderem Sprachmaterial ähnliche Ergebnisse erzielt.

V. Schlussbemerkungen

In dieser Arbeit wurden die Auswirkungen von Segmentverlusten in verschiedenen Bereichen eines Sprachsignals auf die Sprachqualität untersucht: der Verlust von stimmhaften Segmenten nach einem stimmlos / stimmhaft-Übergang führt zu einer erheblichen Beeinträchtigung der Sprachqualität, wohingegen der Verlust von anderen Segmenten relativ gut durch den Fehlerverschleierungsmechanismus des G.729 Dekoders behandelt werden kann. Daraufhin wurde ein spracheigenschaftsbasiertes Paketmarkierungsverfahren (SPB-MARK) entwickelt, das einerseits stimmhafte Segmente, die essentiell für die Sprachqualität sind, durch Markierung mit einer höheren Priorität schützt und andererseits die Fehlerverschleierung des Dekoders für nicht markierte Segmente einsetzt. Die Evaluierung von Simulationen mit einem einfachen Netzwerkmodell unter Verwendung von objektiven Sprachqualitätsmessmethoden zeigte, dass das SPB-MARK-Verfahren nahezu das gleiche Ergebnis wie bei einem vollständigen Schutz des Datenstroms liefert, bei jedoch deutlich reduzierter Anzahl von Paketen mit höherer Priorität. Die "differentielle" Markierungsmethode SPB-DIFFMARK, bei der nur eine Kontrolle über die Verlustmuster anstatt der Verlustraten stattfindet (d.h. der Netzwerkdienst ist über längere Zeiträume gesehen einem "best effort"-Dienst gleichzusetzen), erzielt ein deutlich besseres Ergebnis als der konventionelle "best effort"-Dienst. Alle vorgestellten Markierungsverfahren können innerhalb der IETF "Differentiated Services"-Architektur ([12]) realisiert werden.

Literaturverzeichnis

[1] International Telecommunications Union, "Dual rate speech coder for multimedia communications transmitting at 5.3 and 6.3 kbit/s," Recommendation G.723.1, ITU-T, March 1996.

[2] International Telecommunications Union, "Coding of speech at 8 kbit/s using conjugate-structure algebraic-code-excited linear-prediction (CS-ACELP)," Recommendation G.729, ITU-T, March 1996.

[3] H. Sanneck and N. Le, "Speech property-based FEC for Internet Telephony applications," in *Proceedings of the SPIE/ACM SIGMM Multimedia Computing and Networking Conference 2000 (MMCN 2000)*, San Jose, CA, January 2000, pp. 38–

51, ftp://ftp.fokus.gmd.de/pub/glone/papers/Sann0001:Speech-FEC.ps.gz.

[4] J. Rosenberg, "G. 729 error recovery for Internet Telephony," Project report, Columbia University, 1997.

[5] H. Sanneck, N. Le, and A. Wolisz, "Efficient QoS support for Voice-over-IP applications using selective packet marking," in *Special Session on Error Control Techniques for Real-time Delivery of Multimedia data, First International Workshop on Intelligent Multimedia Computing (IMMCN 2000)*, Atlantic City, NJ, February 2000, pp. 553–556, ftp://ftp.fokus.gmd.de/pub/glone/papers/Sann0002:VoIP-marking.ps.gz.

[6] H. Sanneck and M. Zander, "A comparison of queue management algorithms for intra-flow loss control," in *Proceedings ICC 2000*, New Orleans, LA, June 2000, ftp://ftp.fokus.gmd.de/pub/glone/papers/Sann0006:Intra-Flow-Comparison.ps.gz.

[7] J.R. Deller, *Discrete-Time Processing of Speech Signals*, Prentice Hall, Englewood Cliffs 1993.

[8] W. Yang, K. Krishnamachari, and R. Yantorno, "Improvement of the MBSD objective speech quality measure using TDMA data," in *submitted to IEEE Speech Coding Workshop*, 1999.

[9] S. Voran, "Estimation of perceived speech quality using measuring normalizing blocks," in *Proceedings IEEE Speech Coding Workshop 1997*, Pocono Manor, 1997, pp. 83–84.

[10] International Telecommunications Union, "Objective quality measurement of telephone-band (300-3400 Hz) speech codecs," Recommendation P.861, ITU-T, February 1998.

[11] H. Sanneck, N. Le, G. Carle, and A. Wolisz, "Efficient QoS support for Voice-over-IP applications using selective packet marking," Technical Report, GMD Fokus, Berlin, Germany, February 2000.

[12] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss, "An Architecture for Differentiated Services," RFC 2475, IETF, December 1998, ftp://ftp.ietf.org/rfc/rfc2475.txt.

Efficient QoS Support for Voice-over-IP Applications Using Selective Packet Marking

Henning Sanneck, Long Le and Georg Carle
GMD FOKUS, Berlin

{sanneck,le,carle}@fokus.gmd.de

First IP Telephony Workshop, Berlin
April 13th, 2000

Overview

- Voice over IP (VoIP)
  - Improved quality for VoIP
- Approach
  - Performance of the G.729 loss concealment
  - Speech Property-Based Selective Packet Marking
- Evaluation
  - Reference packet marking/prioritization schemes
  - Simple network models
  - Objective speech quality measurement
- Conclusions

1

Voice over IP

- Main drivers:
  - current economical incentives (Internet flat rate pricing) → *Internet Telephony*
  - service integration, unified packet-switching infrastructure
- One of the main problems:
  - satisfaction of real-time QoS demands in a packet-switched network (fundamental tradeoff: statistical multiplexing vs. reliability → *packet loss*)

QoS for Voice over IP flows

– low bitrate → high per-flow overhead for reservations (scalability, need for multiplexing)

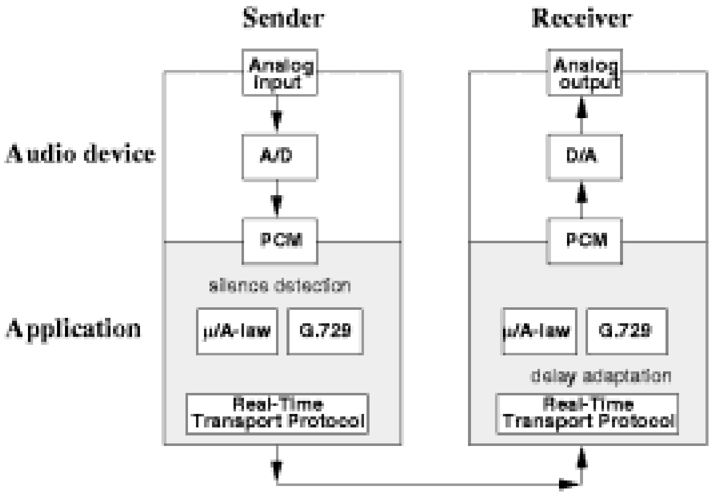
– high compression (backward adaptive coding: ITU-T G.729, G.723.1)

- no further sender adaptation / network adaptation (transcoding) possible
- amplifies high perceptual impact of burst losses (error propagation)

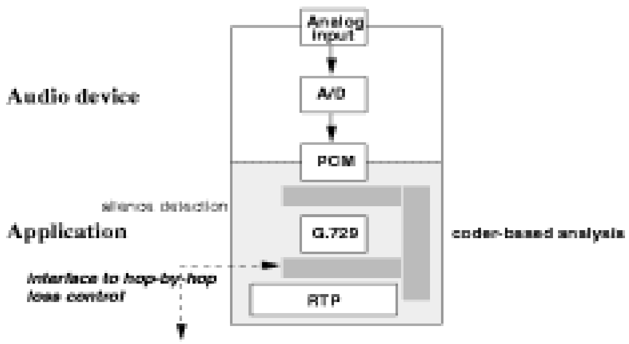
+ tolerance to isolated losses (speech stationarity → extrapolation of coder state → loss concealment)
→ extend the loss resiliency of high-compressing codecs using prioritization (priority enforcement on flow aggregates: Differentiated Services)

2

Structure of an Internet Audio Tool



Additional components: Sender

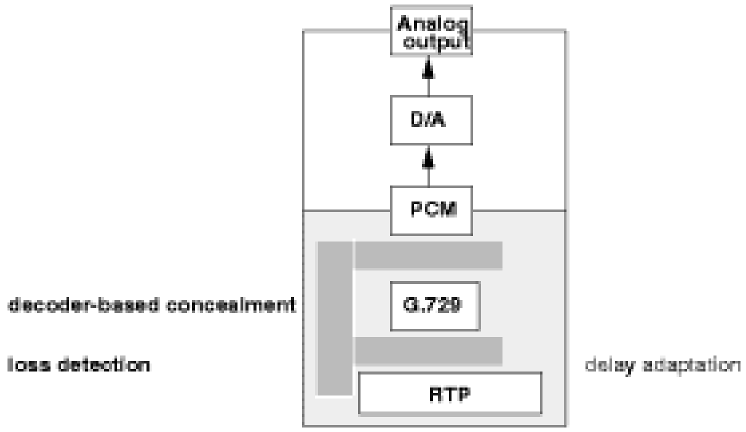


- Side information available at the encoder is used
- Decoder concealment process is taken into account

→ Selective packet marking

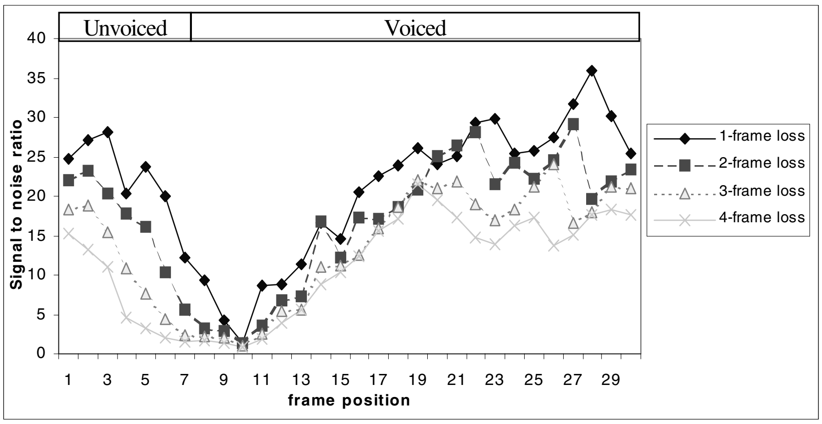
3

Receiver



- No generic (PCM-level) concealment

Performance of G.729 loss concealment



- Decoder fails to conceal losses at unvoiced/ voiced transition due to lack of state (synthesis filter parameters, excitation)

4

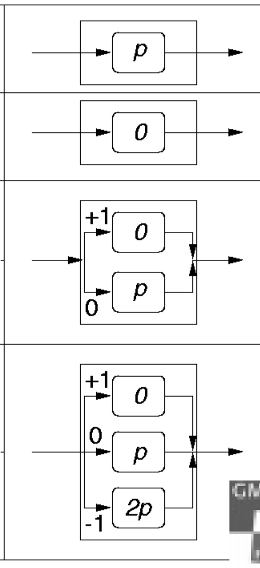
Speech Property-Based Selective Packet Marking

```
protect = 0
    foreach (k frames)
        classify = analysis(k frames)
        if (protect > 0)
            if (classify == unvoiced)
                protect = 0
                send(k frames, "0")
            else
                send(k frames, "+1")
                protect = protect-k
            endif
        else
            if (classify == uv_transition)
                send(k frames, "+1")
                protect = N-k
            else
                send(k frames, "0")
            endif
        endif
endfor
```

- Packet marking („0“, „+1“) is application-controlled (adaptive to expected loss concealment performance): *40-50% of packets marked with higher priority („+1“)*

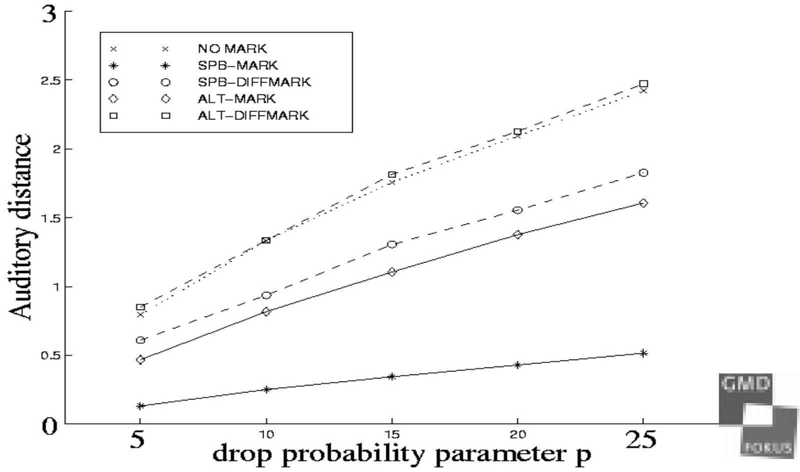
Marking Scheme / Network Model

NO MARK	0	0	0	0	0	0
FULL MARK	+1	+1	+1	+1	+1	+1
SPB MARK	0	+1	+1	0	0	0
ALT MARK	0	+1	0	+1	0	+1
SPB DIFFMARK	0	+1	+1	-1	-1	0
ALT DIFFMARK	-1	+1	-1	+1	-1	+1



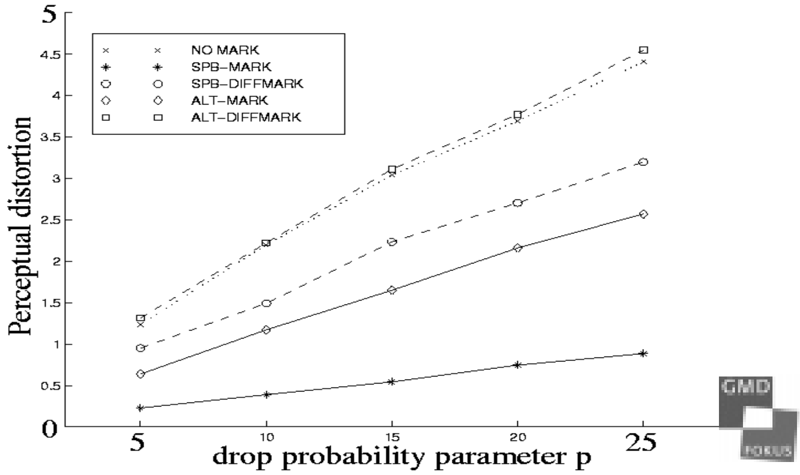
Results: Auditory Distance

- Application of recent advances in objective speech quality measurement: ITU P.861A



Results: Perceptual Distortion

- Enhanced Modified Bark Spectral Distortion (EMBSD; Temple University)



6

Conclusions

- SPB-(DIFF)MARK exploits differences in „concealability“ to adjust the amount of needed network prioritization
- simple network models & objective speech quality measures showed the reduction of number of marked packets while maintaining a good output quality
- selective marking requires appropriate socket interface (or application-level gateway)
- future work: employ speech quality measurement for the marking decision
- `http://www.fokus.gmd.de/glone`

7

Quality of Service and Enhanced Services

ZELLSUBSTITUTION BEI PAKETORIENTIERTER SPRACH- UND AUDIOÜBERTRAGUNG

Lilia Lajmi

Institut für Nachrichtentechnik und Theoretische Elektrotechnik
Technische Universität Berlin, Einsteinufer 25, 10587 Berlin, Germany
lajmi@ft.ee.tu-berlin.de

***Übersicht*- Es werden verschiedene Verfahren zur Substitution fehlender Signalpakete bei paketorientierter Audioübertragung vorgestellt. Einige davon gewährleisten bei Paketverlustraten bis zu 10 Prozent noch gute Klangqualität. Dazu gehören Verfahren, die die statistischen Eigenschaften und die Korrelationen in einem Audiosignal ausnutzen. Sie liefern eine erhebliche Qualitätsverbesserung gegenüber einfachen Verfahren mit Nullsubstitution oder Paketwiederholung.**

I. Einleitung

Bei der PCM-Übertragung von Audiodaten über paketvermittelnde Netze (z.B. ATM-Netze oder Internet) werden die Audiosignale in kleine Blöcke digitaler Information unterteilt. Diese Blöcke, auch Pakete genannt, stellen Ausschnitte eines Audiosignals dar. Sie werden unabhängig voneinander übertragen. Beim Empfänger kann es dazu kommen, dass die gesendeten Audiopakete in falscher Reihenfolge, verspätet oder überhaupt nicht am Empfänger eintreffen. Eine Ursache dieser Paketverluste kann z. B. eine Überlastung im Netz sein, die zu Pufferüberläufen in den Vermittlungsknoten führt.

Wenn Pakete den Empfänger nicht oder zu spät erreichen, so können sie nicht ausgegeben werden und es entstehen Lücken, die die Klangqualität in hohem Maß beeinträchtigen und so für den Hörer sehr unangenehm sein können. Ziel unserer Forschung ist es daher, diese Effekte durch Zellsubstitution mit geeigneten Ersatzsignalen zu reduzieren, so dass der subjektive Höreindruck des Audiosignals nicht darunter leidet.

Auch wenn Sprach- und Audiosignale in komprimierter Form übertragen werden, z.B. im Format MP3, sind unsere Ergebnisse verwendbar, da bei der Decodierung komprimierter Daten wieder eine Rückumsetzung in PCM-Formate stattfindet.

II. Substitution von fehlenden Paketen

Bei einem Paketverlust muß ein Ersatzsignal ausgegeben werden. Die Paketsubstitution kann entweder auf Code- oder auf Signalebene stattfinden. Bei einem PCM-Audioformat sind diese Darstellungen praktisch identisch, da jedem Codewort ein Abtastwert zugeordnet wird. Bei effizienteren Verfahren der Quellencodierung des Audiosignals sind die Darstellungen sehr unterschiedlich und vom jeweiligen Codierverfahren abhängig.

Im folgenden werden einige Substitutionsverfahren vorgestellt, die bei PCM-Formaten angewendet werden:

Nullsubstitution: Der einfachste Ansatz zur Substitution eines verlorenen Paketes ist die „*Nullsubstitution*“. Hier werden im Empfänger statt des verlorenen Paketes Nullen ausgegeben. Bei diesem Verfahren sind keine Schätzungen der Signaleigenschaften erforderlich.

Paketwiederholung: Eine weitere Möglichkeit der Substitution fehlender Signalpakete besteht darin, ein fehlendes Signalstück durch ein dem Signal entnommenes vorhergehendes Segment zu ersetzen. Die einfachste Möglichkeit dabei ist die „*Wiederholung*“ des letzten empfangenen Paketes.

Links/Rechts-Ersatz: Mit diesem Substitutionsverfahren werden bei Stereoübertragung die Stereo-Eigenschaften eines Audiosignals ausgenutzt.
Ausgehend von der Korrelation zwischen beiden Kanäle eines Stereosystems (linker und rechter Kanal) werden bei Paketverlusten in einem Kanal die fehlenden Abtastwerte durch zeitgleiche Abtastwerte des Nachbarkanals ersetzt.
Dieses Verfahren kann für Signale bei großer Ähnlichkeit (Korrelation) zwischen dem linken und dem rechten Kanal sehr gute Ergebnisse liefern.

Packet Merging: Um die aufgrund von Phasensprüngen nach der Substitution verbleibenden Knacksgeräusche zu vermindern, können die Übergänge zwischen dem Ersatzpaket und den angrenzenden Paketen durch Überlappung einiger Abtastwerte geglättet werden.
In [Goo86] wird dafür ein „*Packet Merging*“-Verfahren vorgeschlagen. Bei der Substitution eines verlorenen Paketes der Länge N wird ein Paket der Länge $N' = N+2\Delta N$ verwendet. Die ersten und letzten ΔN Abtastwerte dieses Ersatzpaketes werden den ΔN Abtastwerten des letzten bzw.

folgenden Paketes mit einer entsprechenden Gewichtung überlagert.

Im folgenden wird gezeigt, wie diese oben genannten Substitutionstechniken auch auf der Ebene des LPC-Differenzsignals (LPC: **L**inear **P**redictive **C**oding) angewendet werden können [Clü96].

II.1 Substitution auf LPC-Ebene: Mit der LPC-Analyse werden optimale Prädiktionskoeffizienten a_i aus dem zuletzt korrekt empfangenen Paket berechnet (s. Bild 1). Die berechneten Prädiktionskoeffizienten werden zur Adaption von zwei zueinander inversen Filtern herangezogen (das Prädiktionsfehlerfilter und das Synthesefilter).

Das Audiosignal x(n) durchläuft zunächst das Prädiktionsfehlerfilter, welches das LPC-Differenzsignal d(n) erzeugt. Im Fall eines gültigen Paketes ist d´(n) identisch mit d(n), so dass am Ausgang des Synthesefilters wieder das Eingangssignal rekonstruiert wird (y(n) = x(n)).

Bei einem Paketverlust werden keine neuen Filterkoeffizienten berechnet, die Koeffizienten des Synthesefilters werden beibehalten. Am Eingang dieses Filters muß in diesem Fall eine Innovation erzeugt werden.

Von [Clü98] wurden für den Fall einer digitalen Sprachübertragung mehrere Verfahren zur Erzeugung des Ersatzsignals untersucht. Das Differenzsignal wurde substituiert durch:

- Einfügen von Null-Abtastwerten: Am Eingang des Synthesefilters liegt das Anregungssignal d´(n) mit:
  $$d'(n) = 0 \quad n \in \{0..N-1\} \tag{1}$$
  wobei N der Paketlänge entspricht.
- Wiederholung des Differenzsignals des zuvor empfangenen Paketes.
  $$d'(n) = d(n-N) \quad n \in \{0...N-1\} \tag{2}$$
- Einfügen eines weißen Rauschens:
  Für das Anregungssignal d´(n) gilt dann:
  $$d'(n) = (random\ () - 0.5) \cdot \sqrt{12} \cdot \sigma_d, \tag{3}$$

  wobei random() eine gleichverteilte Zufallsfunktion ist, die Zufallswerte zwischen 0 und 1 erzeugt. σ_d ist die Standardabweichung des letzten korrekt empfangenen Paketes.
  Die Innovation d´(n) wird dem Synthesefilter zugeführt. Dieses Rauschsignal wird durch das Synthesefilter, das noch die Filterkoeffizienten des letzten Paketes enthält, spektral eingefärbt.

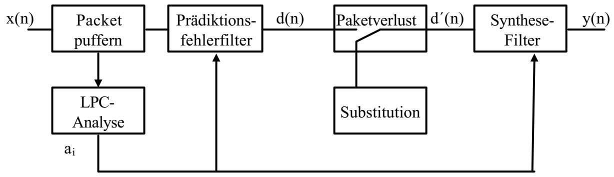


Bild 1: Paketsubstitution auf der Ebene des LPC-Differenzsignals

- Bei Stereoübertragung kann auch eine "Links-/Rechts"-Substitution auf LPC-Ebene durchgeführt werden.
  Für jedem Kanal wird eine LPC-Analyse und eine Monoprädiktion durchgeführt. Es werden zwei Differenzsignale erzeugt, die jeweils aus dem rechten und dem linken Kanal gebildet werden. Fehlt beispielsweise ein Paket aus dem linken Kanal, so wird das fehlende Differenzpaket im linken Kanal durch das zeitgleiche Differenzpaket aus dem rechten Kanal ersetzt. In diesem Fall behält das Synthesefilter seine alten Koeffizienten nicht, für die Synthese des Differenzsignals werden nun die entsprechenden Koeffizienten aus dem rechten Kanal genommen.

II.2 Substitution mit Stereoprädiktion: Die Stereoprädiktion stellt eine Methode zur Ausnutzung der Inter-Kanal-Redundanz von Stereosignalen dar. Eine Stereoprädiktion wird beispielsweise bei der Datenkompression von Audiosignalen verwendet (MPEG, *Joint stereo Audio Coding* [Fuc93]).

Bei Stereosignalen sind beide Kanäle (linker und rechter Kanal) oft miteinander korreliert. Die Stereoprädiktion betrachtet die statistischen Abhängigkeiten (Korrelation) zwischen zwei zeitgleichen Ausschnitten (Paketen) aus dem rechten und linken Kanal. Der bei der Stereoprädiktion erreichbare Prädiktionsgewinn wurde mit dem Gewinn bei konventioneller linearer Prädiktion (Monoprädiktion) verglichen. Die Untersuchungsergebnisse haben gezeigt, dass mit der Stereoprädiktion ein höherer Prädiktionsgewinn als mit der Monoprädiktion erreicht werden kann [Cam93].

1. Berechnung der optimalen Filterkoeffizienten

Analog zu der konventionellen LPC wird bei der Stereoprädiktion für jeden Wert $L(n)$ des linken Kanals ein Schätzwert $\hat{L}(n)$ aus vorangegangenen Signalwerten $L(n-k)$ des selben Kanals und aus $R(n-k)$ des Nachbarkanals berechnet (s. Gl. 4).

$$\hat{L}(n) = \sum_{k=1}^{k_a} a_k L(n-k) + \sum_{k=1}^{k_b} b_k R(n-k) \tag{4}$$

Diese Überlegung gilt natürlich auch für den rechten Kanal (s. Gl. 5).

$$\hat{R}(n) = \sum_{k=1}^{k_c} c_k R(n-k) + \sum_{k=0}^{k_d} d_k L(n-k) \qquad (5)$$

Jeder Schätzwert lässt sich als Summe aus einem "Autoprädiktions"- und einem "Kreuzprädiktions"-Teil darstellen. So liefert z. B. der Autoprädiktor des linken Kanals eine Summe aus den vorangegangenen k_a Koeffizienten desselben Kanals, gewichtet mit den adaptiv berechneten Koeffizienten a_k. Der Kreuzprädiktor des linken Kanals liefert im hingegen eine Summe aus den vorangegangenen k_b Koeffizienten des Nachbarkanals, gewichtet mit den adaptiv berechneten Koeffizienten b_k (s. Bild 2).

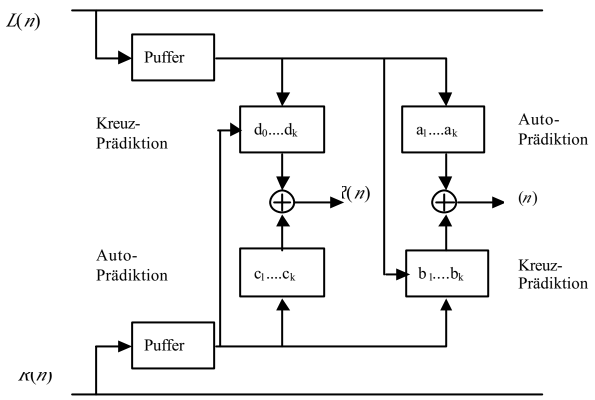


Bild 2: Das Prinzip der Stereoprädiktion SLPC

Ziel der adaptiven Prädiktion ist es, eine möglichst gute Signalschätzung zu erhalten. Das wird dann erreicht, wenn die Differenz zwischen Original- und Schätzsignal minimiert wird. Für eine optimale Vorhersage der L(n) (bei fehlenden R(n)) bzw. der R(n) (bei fehlenden L(n)) müssen die Prädiktoren optimiert werden. Dazu wird der mittlere quadratische Fehler zwischen Original- und Schätzsignal und damit die Varianz des Differenzsignals minimiert.
Eine Minimierung des Differenzsignals bzw. dessen Varianz bezüglich der Koeffizienten a_k bzw. b_k liefert das folgende Gleichungssytem:

$$\begin{pmatrix} R_{LL}(1,0) \\ \vdots \\ R_{LL}(k_a,0) \\ \hline R_{LR}(0,1) \\ \vdots \\ R_{LR}(0,k_b) \end{pmatrix} = \left(\begin{array}{ccc|ccc} R_{LL}(1,1) & \cdots & R_{LL}(1,k_a) & R_{LR}(1,1) & \cdots & R_{LR}(1,k_b) \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ R_{LL}(k_a,1) & \cdots & R_{LL}(k_a,k_a) & R_{LR}(k_a,1) & \cdots & R_{LR}(k_a,k_b) \\ \hline R_{LR}(1,1) & \cdots & R_{LR}(k_a,1) & R_{RR}(1,1) & \cdots & R_{RR}(k_b,1) \\ \vdots & \cdots & \vdots & \vdots & \vdots & \vdots \\ R_{LR}(1,k_b) & \cdots & R_{LR}(k_a,k_b) & R_{RR}(1,k_b) & \cdots & R_{RR}(k_b,k_b) \end{array}\right) \cdot \begin{pmatrix} a_1 \\ \vdots \\ a_{k_a} \\ \hline b_1 \\ \vdots \\ b_{k_b} \end{pmatrix}$$

In diesem Gleichungssystem sind $R_{LL}(i,j)$ und $R_{RR}(i,j)$ Autokorrelationswerte und $R_{LR}(i,j)$ Kreuzkorrelationswerte.

Für die Berechnung der Koeffizienten c_k und d_k wird die Varianz des Differenzsignals des rechten Kanals minimiert. Es ergibt sich das folgende Gleichungssystem:

$$\begin{pmatrix} R_{RR}(1,0) \\ \vdots \\ R_{RR}(k_c,0) \\ \hline R_{RL}(0,0) \\ \vdots \\ R_{RL}(0,k_d) \end{pmatrix} = \left(\begin{array}{ccc|ccc} R_{RR}(1,1) & \cdots & R_{RR}(1,k_c) & R_{RL}(1,0) & \cdots & R_{RL}(1,k_d) \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ R_{RR}(k_c,1) & \cdots & R_{RR}(k_c,k_c) & R_{RL}(k_c,0) & \cdots & R_{RL}(k_c,k_d) \\ \hline R_{RL}(1,0) & \cdots & R_{RL}(k_c,0) & R_{LL}(0,0) & \cdots & R_{LL}(k_d,0) \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ R_{RL}(1,k_d) & \cdots & R_{RL}(k_c,k_d) & R_{LL}(0,k_d) & \cdots & R_{LL}(k_d,k_d) \end{array}\right) \cdot \begin{pmatrix} c_1 \\ \vdots \\ c_{k_c} \\ \hline d_0 \\ \vdots \\ d_{k_d} \end{pmatrix}$$

Mit dieser Analyse werden, wie oben beschrieben, die optimalen Filterkoeffizienten a_i, b_i, c_i und d_i aus den richtig empfangenen Paketen des linken und rechten Kanals berechnet (s. Bild3).

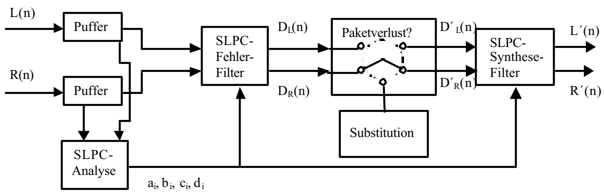


Bild 3: Paketsubstitution mit Stereoprädiktion

2. Synthese eines Ersatzsignals aus Innovationssignalen

Die berechneten Koeffizienten werden zur Adaption von zwei zueinander inversen Filtern herangezogen, jeweils für den linken und rechten Kanal.

Im Fall eines Paketverlustes in einem der beiden Kanäle werden keine neuen Koeffizienten berechnet, die Koeffizienten des SLPC-Synthesefilters werden beibehalten. Am Eingang des SLPC-Synthesefilters liegen jetzt die Anregungssignale $D'_L(n)$ und $D'_R(n)$ an.
Analog zu der Substitution auf der Ebene des LPC-Differenzsignals können hier auch die Anregungssignale durch Nullwerte, gleichverteiltes Rauschen oder durch die Wiederholung der letzten Differenzsignale aus den zuletzt korrekt empfangenen Paketen ersetzt werden. Eine weitere Möglichkeit besteht in dem Links-/Rechts-Ersatz der Differenzsignale: Im Fall eines Paketverlustes im linken Kanal wird das Differenzsignal aus dem rechten Kanal genommen. Damit würden nun am Eingang des SLPC-Synthesefilters die Anregungssignale $D'_R(n)$ (als Ersatz für $D'_L(n)$) und $D'_R(n)$ liegen.

III. Ergebnisse

Folgende Verfahren wurden miteinander verglichen:

- Nullsubstitution (Signalebene)
- Überlappende Blockwiederholung (Signalebene)
- Links-/Rechts- Ersatz mit Merging (Signalebene)
- Nullsubstitution auf LPC-Ebene
- Blockwiederholung auf LPC-Ebene
- Substitution mit Rauschen auf LPC-Ebene
- Links-/Rechts-Ersatz auf LPC-Ebene (mit zwei Monoprädiktionen).
- Stereoprädiktion (Nullsubstitution).

Die oben erwähnten Substitutionsverfahren wurden für Paketlängen N von 512, 256, 128, 64 und 24 Abtastwerten untersucht. 24 Abtastwerte entsprechen bei einem 16 Bit PCM-Signal einer ATM-Zelle (48 Byte). Die verwendeten Audiodateien sind mit 16 Bit quantisiert und haben eine Abtastfrequenz von 44,1 kHz.

Für die Untersuchung wurden nur Signale mit Einzelpaketverlusten und Paketverlustraten von 1% und 10% verglichen. Eine Bewertung der verwendeten Substitutionsverfahren wurde mit Hilfe eines im Rahmen eines Telekom-Forschungsprojektes im Fachgebiet Fernmeldetechnik der TU Berlin entwickelten Programms („*DIXI*") durchgeführt.
DIXI stellt eine Benutzeroberfläche für das Meßverfahren PEAQ zur Verfügung. PEAQ ist eine von der ITU-T standarisierte Meßmethode zur *objektiven* Messung der empfundenen Audioqualität, sie verwendet zu wesentlichen Teilen an der TU-Berlin erarbeitete Modellierungsansätze.

Als Maß für die objektive Qualitätsbewertung von Audiosignalen liefert die DIXI-Software einen ODG-Wert (***O****bjective* ***D****ifference* ***G****rade*). ODG-Werte liegen zwischen -4 und 0 und haben die folgende Skala:

ODG	Audio-Qualität
0	Störung nicht wahrnehmbar
-1	Wahrnehmbar aber nicht störend
-2	kaum störend
-3	störend
-4	sehr störend

Tabelle 1: ODG-Skala [Thi98]

Um mit den für die *subjektive* Qualitätsbewertung üblicherweise verwendeten MOS-Werten (*Mean Opinion Score*) [Jay84] vergleichen zu können läßt sich die objektiven Bewertungsgröße ODG in die Größe PODG (*positiv ODG*) = 5 + ODG umrechnen, die dann Werte zwischen 1 und 5 liefert

In den Bildern 4 und 5 sind die Simulationsergebnisse für Paketverlustraten von Pv = 1% und Pv = 10% graphisch dargestellt.

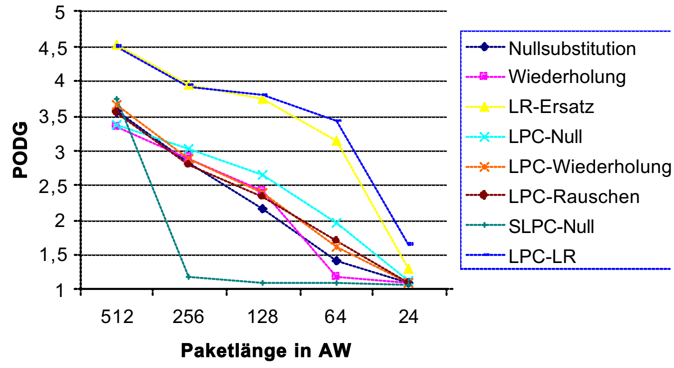


Bild 4: Vergleich der Substitutionsverfahren für Pv = 1%.

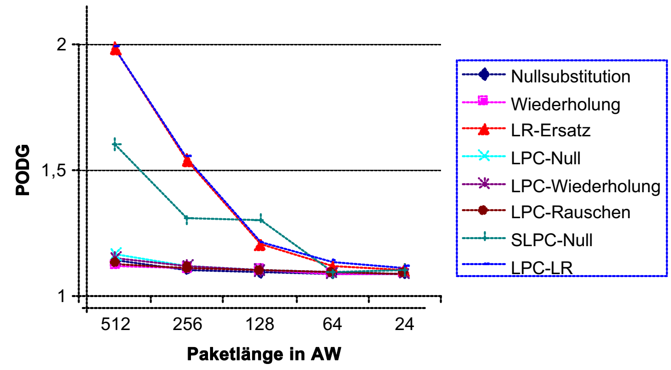


Bild 5: Vergleich der Substitutionsverfahren für Pv = 10 %.

Aus den Bildern 4 und 5 ist ersichtlich, dass mit der Links-/Rechts-Substitution auf LPC-Ebene und auf PCM-Ebene die besten Ergebnisse erreicht wurden. Die überlappende Blockwiederholung zeigt hier keine wesentliche Qualitätsverbesserung gegenüber der Nullsubstitution.
Die Qualitätsverschlechterung bei 24 AW liegt an der hohen Anzahl der insgesamt im Signal verlorengegangenen Pakete.
Die wesentliche Qualitätsverbesserung, welche sowohl mit der LPC-Analyse als auch mit L/R-Ersatz erreicht wurde, deutet darauf hin, dass eine Kombination beider Verfahren in Form einer Stereoprädiktion für die Substitution von Paketverlusten von Vorteil sein könnte.
Eine Untersuchung der Stereoprädiktion hat jedoch gezeigt, dass die mit diesem Verfahren erreichte Audioqualität sehr stark schwankt. Dies liegt daran, dass das SLPC-Synthese-Filter instabil werden kann und es aus diesem Grund zu großen Schwankungen der Amplitude und

damit auch der wahrgenommenen Audioqualität kommen kann.

Mit zwei Monoprädiktionen, welche unabhängig voneinander auf den linken und auf den rechten Kanal angewendet wurden, konnten bessere Ergebnisse als mit einer Stereoprädiktion erreicht werden.

IV. Literatur

[Cam93] P. Cambridge
Audio Data Compression Techniques. AES 94th Convention. March 16-19, 1993 Berlin. Preprint 3584 (K1-9)

[Clü96] K. Clüver
Ein Verfahren zur Rekonstruktion fehlender Sprachsignalrahmen mit linearer Prädiktion und Teilbandanregung.
Frequenz 50 (1996) S. 9-10.

[Clü98] K. Clüver
Rekonstruktion fehlender Signalblöcke bei blockorientierter Sprachübertragung. Dissertation TU Berlin 1998

[Fuc93] Fuchs Hendrik
Improving Joint Stereo Audio Coding by Adaptive Inter-Channel Prediction.
Proc. Of the 1993 IEEE ASSP Workshop on Application of Signal Processing to Audio and Acoustics. Oct. 17-20 1993, Mohomk Mountain, New Paltz, New York

[Goo86] D. J. Goodman, G. B. Lockhart, O. J. Wasem, W.-C. Wong
Waveform Substitution Techniques for Recovering Missing Speech Segments in Packet Voice Communications, IEEE Trans. on Acoustics, Speech and Signalprocessing, Vol. ASSP-34, no. 6. Dezember 1986, S. 1440-1448

[Jay84] N. S. Jayant, P. Noll
Digital Coding of Waveforms. Prentice Hall. Englewood Cliffs 1984.

[Thi98] T. Thiede
Perceptual Audio Quality Assessment Using a Non-Linear Filter Bank.
Dissertation TU Berlin 1998.

Evaluating and Improving Firewalls for IP-Telephony Environments

Utz Roedig[1], Ralf Ackermann[1], Ralf Steinmetz[1,2]
1 - Darmstadt University of Technology - Industrial Process and System Communications (KOM)
Merckstr. 25 - 64283 Darmstadt, Germany
2 - German National Research Center for Information Technology - GMD IPSI
Dolivo-Str. 15 - 64293 Darmstadt, Germany
{Utz.Roedig, Ralf.Ackermann, Ralf.Steinmetz}@KOM.tu-darmstadt.de

***Abstract* -- Firewalls are a well established security mechanism for providing access control and auditing at the borders between different administrative network domains. Their basic architecture, techniques and operation modes did not change fundamentally during the last years.**

On the other side new challenges emerge rapidly when new innovative application domains have to be supported. IP-Telephony applications are considered to have a huge economic potential in the near future. For their widespread acceptance and thereby their economic success they must cope with established security policies. Existing firewalls face immense problems here, if they - as it still happens quite often - try to handle the new challenges in a way they did with "traditional applications". As we will show in this paper, IP-Telephony applications differ from those in many aspects, which makes such an approach quite inadequate.

After identifying and characterizing the problems we therefore describe and evaluate a more appropriate approach. The feasibility of our architecture will be shown. It forms the basis of a prototype implementation, that we are currently working on.

I. Introduction

A. IP-Telephony

IP-Telephony is used to establish a conversation comparable to a classic telephone call using an IP infrastructure. Typical applications and scenarios are currently based on different protocol suites. At the moment there are two main approaches - the H.323 [1] protocol family and the Session Initiation Protocol SIP [2] with a changing distribution and relevance. Though today, a high percentage of applications and scenarios is still H.323 based (and we will therefore initially focus on it), it is supposed that in the near future the use of the SIP protocol may increase [3]. Both protocol types will even be usable together with appropriate gateways.

B. Firewalls

Within a global networked environment, security aspects have become more and more important and access control at network borders is considered essential. Therefore, most organizations replaced their simple internet routers by firewalls.

These firewalls consist of packet filters, "stateful filters", proxies or a combination of all these. A firewall examines all network traffic between the connected networks. Only packets that are explicitly allowed to (as specified by a security policy) are able to pass through [4],[5]. In addition to the inspection of data flows, some firewalls also hide the internal network structure of an organization. From the Internet the only visible and therefore attackable network system is the firewall. This is achieved by the use of proxy functionality or a Network Address Translation (NAT) mechanism.

To perform its observation tasks the firewall components (filters, stateful filters, proxies) need to interoperate with a special component for the services (e.g. IP-Telephony) they want to support. We refer to this component as a parser. Based on the analysis of the traffic, the firewall decides whether packets may be passed through. A parser may also interact with NAT or proxy components since it extracts the information that can be modified or used.

II. Problem Domain

A. Multimedia Applications

The type of applications considered here are multimedia applications which use continuous (e.g. audio, video) and discrete media (control, text, meta data) data [6]. Multimedia applications significantly differ from traditional applications.

Especially
- multiple flows for one logical session,
- complex protocols and dynamic protocol behavior,
- high data rate and other QoS constraints,
- the usage of multicast mechanisms

are common features and may cause problems in a network environment which is protected by firewalls.

A comprehensive description and general approaches to deal with these characteristics can be found in [7],[8], [9],[10] and [11]. In this paper we intentionally focus on

IP-Telephony related topics.

B. *Specific IP-Telephony related characteristics*

Figure 1 describes a scenario in which H.323 components and firewalls are used together. It is considered to be representative for common operational areas and may slightly be adapted to individual other configurations.

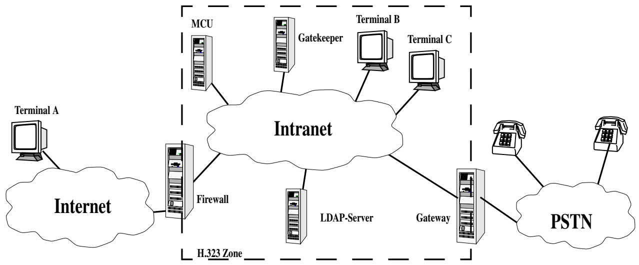


Figure 1: IP-Telephony scenario including Firewalls

The figure shows a private intranet of an organization, protected against the public Internet by a firewall. Within the intranet one or even more H.323 zones may exist. A H.323 zone consists of a gatekeeper and several optional devices such as a Multi Conference Unit (MCU), gateways and terminals.

1) Variety and complexity of communication mechanisms

The communication mechanisms used in the scenario depend on the involved components and may differ for different use cases. If only two terminals (Terminal A and C) establish a H.323 connection, the following basic order of events proceeds:

- **Q.931 (TCP) signaling:**
  Terminal A contacts Terminal C via TCP. The TCP connection is used by Q.931 to set up the call and to negotiate the parameters (e.g. ports) for the following H.245 connection.
- **H.245 (TCP) signaling:**
  Terminal A contacts Terminal C via TCP using the negotiated port. The H.245 connection is used to determine the characteristics of the following media streams (e.g. audio or video).
- **RTP/RTCP (UDP) media and control traffic:**
  Several streams may be used between the two terminals. At least 4 UDP streams are necessary to transmit audio (1 RTP and the corresponding RTCP stream in each direction). Additional streams could be used if also video has to be transmitted.

If, in the same scenario, a gatekeeper is used, the communication mechanisms differ. In this case we observe:

- **RAS registration (TCP):**
  At system start up the terminals use a TCP connection to register themselves at the gatekeeper using the Registration, Admission and Status (RAS) protocol.
- **RAS Admission Control (TCP):**
  Before the communication can be set up between both terminals, the calling terminal (Terminal A) requests a permission at the gatekeeper using the RAS protocol. If this permission is granted, the communication setup proceeds incorporating the steps (Q.931, H.245, RTP/RTCP) described above.

The communication mechanisms also change, if other devices like MCU or gateways are used.

2) Vendor specific implementations / features

Not only the use of other components within the scenario has major implications. Our experiments show, that different vendors also use different (and sometimes not interoperable) implementations, though they claim to be fully H.323 compliant.

In case the Terminal A is not a “pure” H.323 terminal but implements Microsoft Netmeeting the following extension will be used (and some firewall solutions rely on it):

- **ILS/LDAP (TCP) name / address resolution:**
  Before the communication is set up, Terminal A tries to inquire at a Internet Location Service (ILS) or Lightweight Directory Access Protocol (LDAP) server, to perform a name lookup. That way it can use a symbolic alias name to address the client (phonebook functionality). After the client has determined the destination address, it proceeds using the basic H.323 communication mechanisms. Within Microsoft NetMeeting scenarios the name lookup process is usually based on an ILS request.

The selected examples show, that the communication behavior may change significantly each time the scenario changes. As soon as the resulting control or media traffic crosses network borders, firewalls have to deal with that dynamic variety, which is not a trivial task.

3) Network Address Translation (NAT)

Another problem arises when Network Address Translation (NAT) has to be performed by the firewall. In this case the internal terminals (Terminal B and C) can not be called directly from the “outside” networks, because their address is not visible for an external terminal. This is a desired firewall function - it hides internal details and prevents internal systems from being attacked directly. It conflicts with the usual H.323 protocol flow though.

If, in our scenario (Figure 1), Terminal A wants to connect to Terminal B this could not be done directly. Terminal A has to connect to the firewall first, then Terminal A has to tell the firewall to whom it wants to talk. The firewall then has to contact Terminal B and must proxy the control / audio streams between both terminals. There exist different methods to achieve this goal.

If no gatekeeper is present in the scenario, the following method, described in [12] can be used:

- The external terminal has to be modified. There must be a configuration entryy in which the user can specify a firewall which will proxy the call.
- The calling party must connect to the remote proxy, and tell that proxy whom it wants to talk with. The H.323 setup message supports this operation mode. The destCallSignalingAddress and/or the destinationAddress (alias list) must contain the address of the proxy. The remoteExtensionAlias field should contain the information about the actual target user. The proxy must resolve the name into an IP address. This could be done by using DNS, LDAP or different protocols.
- Then, the proxy connects the target and relays the control/audio streams between both terminals.

If a Gatekeeper is present, the following method is proposed in [13]:

- The gatekeeper in the internal network has to be installed in parallel to the firewall. It has to be configured with a valid address.
- The external Terminal A has to be configured to use the gatekeeper.
- If Terminal A wants to initiate a call to Terminal B, it asks the gatekeeper for permission to call Terminal B.
- The gatekeeper responds with the address of the firewall to Terminal A.
- Terminal A calls the firewall (or the proxy within the firewall).
- The proxy consults the gatekeeper for the true destination which is Terminal B.
- The proxy then complements the call setup and relays the control/audio streams between both terminals.

In the case, that an internal terminal wants to initiate a call, the same methods can be used. In addition the firewall can try to handle the call "transparently". The internal terminal places the call to the external terminal directly, because this one has a valid address. The firewall has to monitor and remember the communication state and has to map all internal IP addresses (for internal terminals) to addresses that are valid externally (as e.g. the address of the firewall itself).

We expect both parties (calling party and called party) to be behind their own company firewall in most practical scenarios. Therefore the incoming call problem is a general and very important one. As shown above, all available solutions, to handle incoming calls in NAT environments require an interaction between the firewall/proxy and the components that perform address resolution. The name resolution could be performed by H.323 components (e.g. a gatekeeper) or by other services (e.g. DNS, LDAP,...). Therefore a parser component within a firewall must be able to interoperate with these services.

C. *Parser related problems*

The task of traffic observation within the firewall is performed by a parser. Commonly used firewalls use static and integrated protocol parsers. These parsers are often written in a firewall specific language (e.g. INSPECT in a FIREWALL-1 [14]). Usually they are compiled in advance and then statically loaded into the firewall.

They may interact with the firewall, request data streams for analysis and reconfigure the overall system based on their inspection results. A system of this type is shown in Figure 2.

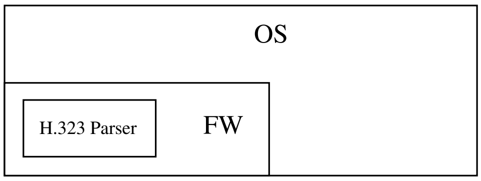


Figure 2: Integrated protocol parser

The figure shows a H.323 parser which is directly embedded (as other parsers for other protocols may be too) in the firewall (FW) itself. The firewall relies on and uses functions of the Operating System (OS) of the firewall host. IP-Telephony data streams are passed to the firewall components (e.g. by configuring OS specific sockets / packet filters) and the parser within the firewall is responsible for analyzing them. In this paper we will generally use this kind of basic schematics for explaining and comparing the differences between various architectures.

A "parser as integral part of the firewall" approach works very well with common applications, but with IP-Telephony applications it does not. The following reasons cause this fact:

1 **Different communication mechanisms:**
Obviously, different parsers are necessary for every type of H.323 scenario. If the scenario is changed only slightly, the parser can often not be adjusted to the new requirements and a new parser becomes necessary. Our practical evaluation shows, that static and embedded parsers are not able to adapt to the described complex scenarios.

2 **Network Address Translation (NAT):**
The parser can only communicate with the firewall, but not with other components. As shown, a connection to other components is necessary to successfully

enable the use of NAT.

III. Evaluation of Current Solutions

A "conventional" firewall/parser architecture, as shown in Figure 2, is obviously not sufficient to support IP-Telephony scenarios. This fact has been recognized by various firewall vendors and has lead to implementations cope with the problems. The first example describes the H.323 solution of the firewall market leader (80% of the market). The two other examples show dedicated solutions, which explicitly address the described problems.

A. Firewall-1

The architecture of the Firewall-1 [14] product basically corresponds to the architecture shown in Figure 2. Therefore all problems described above occur in a Firewall-1 protected network. Because the parser is static, a dedicated parser is necessary for each communication scenario. Currently two parsers are available, one for Microsoft Netmeeting and another generic one for H.323 traffic. We tested these parsers with the following results:

- The Netmeeting parser supports the direct connection between two Netmeeting (version 2 and version 3) terminals only. If one of the terminals is replaced by another product (in our experiment a H.323 compliant Innovaphone IP400 [16]), the parser does not work correctly and the intended connection setup is blocked by the firewall.
- The generic parser did not work at all. Almost no documentation is available for this parser, so the reasons for its failing could not be inspected in detail, nor could it be reconfigured correctly.
- NAT scenarios are not fully supported. There is no mechanism to handle incoming external calls in a NAT network configuration mode.

Summary:
The parser components are very static. Only some basic scenarios with standard applications could be run successfully in our experiments. Because of the missing interaction between the firewall and the parser components, inherently not all address translation scenarios can be supported.

B. Cisco MCM

The Cisco Multimedia Conference Manager (MCM) [13] provides both gatekeeper and proxy functionality. It forms an additional system which can be used to extend existing firewalls with IP-Telephony functionality. The MCM can be installed on a Cisco System (e.g. router using Cisco IOS), in parallel to or behind a firewall. Its architecture is shown in Figure 3.

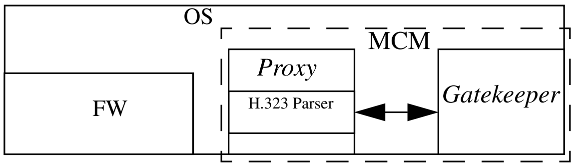


Figure 3: MCM Architecture

All IP-Telephony traffic is handled by the MCN and thereby "bypasses" the original firewall. An interaction between the firewall and the MCM is not intended. If the MCM is used parallel to the firewall, NAT scenarios can be supported. This is possible, because the MCM consists of a gatekeeper and a proxy which are able to interact.

Summary:
The approach basically addresses the NAT problem. All possible NAT scenarios could be supported. The parser within the proxy part of the MCM is also static. The parser component can not be adapted to dedicated scenarios and applications. Interaction with a gatekeeper is possible, interaction with other components like ILS or DNS is not used.

C. Phonepatch

The PhonePatch [15] component focuses on a NetMeeting scenario and works like a proxy with some additional functionality (PBX like functions, e.g. callback). PhonePatch is used in parallel to an existing firewall and is explicitly responsible for handling the IP-Telephony traffic. An interaction between the firewall and PhonePatch is not implemented.

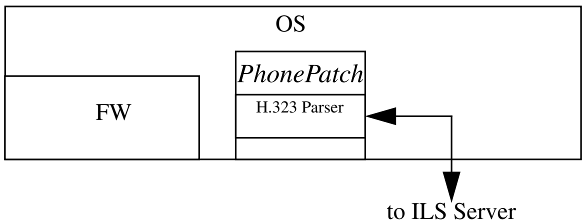


Figure 4: PhonePatch Architecture

All Internet Location Service (ILS) requests are passed through PhonePatch. This allows to examine the IP addresses transfered as part of the ILS protocol and adjust them to redirect the call from its original destination to the PhonePatch host. When data streams then arrive, the PhonePatch component directs them to the host that was mentioned in the original ILS request. This transparently fools NetMeeting applications into making a "proxy call", even though the application configuration does not have to support proxies explicitly (Netmeeting

Version 3 supports using proxies for outgoing calls now).

Summary:

This approach basically addresses scenarios using NetMeeting terminals using ILS in NAT environments.

Varying protocol scenarios and generic H.323 applications are not targeted and could not be supported in our experiments using other H.323 systems (e.g. InnovaPhone).

IV. Our new extended Approach

As we have shown, a commonly used internal firewall architecture as shown in Figure 2 is not very useful for IP-Telephony scenarios. Various vendors recognized this and implemented / proposed other architectures. These - up to now - may handle parts of the described problem domain. A general solution for all of the problems is not available yet. That is why we introduce a new parser architecture (Figure 5) which is explicitly targeted to be more general.

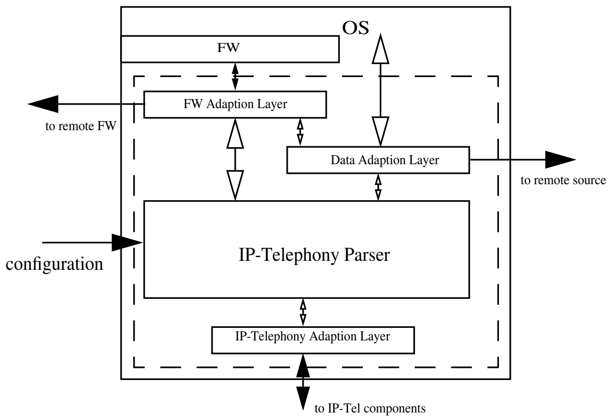


Figure 5: Proposed Alternative Architecture

We decide to place the parser outside the conventional firewall core.

- This allows the parser component to communicate with other (e.g. IP-Telephony) components. As a consequence all relevant NAT scenarios can be supported.
- Additionally the parser component can be loaded dynamically and configured separately (e.g. with an optimized / dedicated configuration language) from the firewall. This enables a general and still lightweight support for dedicated and even changing or emerging scenarios and components.

These design considerations directly influence our architectural and implementation strategy.

To be able to move the parser out of the firewall core, an interface is necessary. It allows the parser to interact with the firewall system as it did before when it was an integral part of it.

- An adaption layer is used, to allow the reuse of the parser when the firewall type/vendor is changed. The so called “Firewall Adaption Layer” is responsible for mapping the generic firewall commands generated by a specific (e.g. IP-Telephony) parser to commands that are understandable for a specific (and thereby enhanced) firewall. As an example, generic commands are used to inform the firewall, which connections are negotiated and should be passed through or redirected to a specific filter.
- We use a so called “Data Adaption Layer” which is responsible for redirecting the data streams to the parser. This layer allows to modify the internal source of the observable and modifiable data.
- We use a dedicated adaption layer for communicating with external components. In our scenario it is called “IP-Telephony Adaption Layer”. The parser can generate generic requests and the adaption layer is able to map these request to the protocol language of a special component. This for example allows to map a parser request like “determine the destination address for a call to user steinmetz” to a specific DNS, LDAP and/or Gatekeeper request.

A variety of additional benefits directly results from this architecture:

- Not only can the parser be easily adapted to dedicated H.323 scenarios. It may also be changed for scenarios which use a different IP-Telephony signaling protocol. Support for SIP scenarios or heterogenous scenarios can be implemented by just modifying the IP-Telephony parser.
- As our current implementation shows, by extending the FW Adaption Layer, the parser can support different firewalls and firewall systems. The parser must not be rewritten from scratch, if ported to another system.

V. Summary

In this paper we have shown that and why the usage of firewalls leads to problems within IP-Telephony scenarios. We analyzed available firewall products and showed that they do not fully support all relevant IP-Telephony needs. To allow the unrestricted use of IP-Telephony applications within firewall environments we propose a new architecture. This one is currently evaluated as part of an experimental prototype implementation.

VI. References

[1] ITU: ITU-T Recommendation H.323, Packet-Based Multimedia Communication Systems, 1998

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

[3] Douskalis, B.: IP Telephony - The Integration of Robust VoIP Services, Prentice Hall, 2000

[4] Chapman, D. B.: Building Internet Firewalls, O'Reilly, Cambridge, 1995

[5] Cheswick, W. R.; Bellovin S. M.: Firewalls and Internet Security, Addison Wesley, 1994

[6] Steinmetz, R.; Nahrstedt, C.: Multimedia: Computing, Communications & Applications, Prentice-Hall, 1995

[7] Finlayson, R.: IP Multicast and Firewalls, Internet Draft, draft-ietf-mboned-mcast-firewall-02.txt, 1998

[8] Christoph Rensing, Utz Roedig, Ralf Ackermann, Lars Wolf, Ralf Steinmetz: VDMFA, eine verteilte dynamische Firewallarchitektur für Multimedia-Dienste, In Informatik aktuell, Kommunikation in Verteilten Systemen (KiVS), Springer, 2.-5. März 1999

[9] Utz Roedig, Ralf Ackermann und Christoph Rensing: DDFA Concept, Technical Report KOM-TR-1999-04, KOM, Dezember 1999

[10] Utz Roedig / Ralf Ackermann: Firewalls and their Impact on Multimedia Systems, Multimedia Computing and Networking 2000, January 2000, Panel Discussion "Security Firewalls and their Impact on Multimedia Systems"

[11] Ellermann, U., Benecke, C.: Parallele Firewalls - skalierbare Lösungen für Hochgeschwindigkeitsnetze, veröffentlicht in: DFN-CERT Workshop Sicherheit in vernetzten Systemen, Hamburg, 1998

[12] Intel: http://support.intel.com/support/videophone/trial21/H323_WPR.HTM

[13] Cisco: MCM, http://www.cisco.com/univercd/cc/td/doc/product/software/ios113ed/113na/1137na/mcm_cfg.htm

[14] Marcus Goncalves Steven Brown: Checkpoint Firewall 1, Administration Guide, McGraw-Hill, 1999

[15] PhonePatch: http://www.phonepatch.com

[16] InnovaPhone: http://www.innovaphone.com

New Tools for Programming IP Telephony Services

Inmaculada Espigares, Jose M. Costa, and Raimo Kantola
Department of Telecommunications Technology
Helsinki University of Technology, P.O. Box 3000, FIN-02015 HUT, Finland

***Abstract*-- Our project shows an innovative method to design new IP Telephony services. The most recent tools such as JAVA and XML are used for programming the services in a Session Initiation Protocol (SIP) environment. We have implemented the Internet Call Waiting (ICW) service, but many other advanced services could have been defined using the same languages and protocols. This paper describes our project and summarises our experience in implementing services in the IP Telephony environment.**

***Index terms*— XML, script, JAVA, C, parser, service, CPL**

A. INTRODUCTION

IP Telephony has brought new signalling protocols where signalling information is encoded using ascii characters. The whole landscape for service implementation is different from circuit telephony including the Intelligent Networks (IN). The differences include intelligent terminals instead of dumb ones, the emergence of the Call Processing Language (CPL) [11] and direct processing of call state in the service logic instead of mediation of the call state to the service through a complex Basic Call State Model typical of IN.
Our project called "An implementation of the Internet Call Waiting service using SIP" [1], studies the Session Initiation Protocol [2] for the purpose of evaluating it and making an implementation of the Internet Call Waiting (ICW) service [4]. It is useful for the calls that otherwise would be lost when the line is busy and also for rejecting undesirable incoming calls. It is also a way of not wasting network resources and contributing to call completion.
For service implementation, we have used the XML language [13]. XML is considered one of the best languages for describing complex data relationships [15]. XML is easily extended, flexible and has a text-based syntax.
The project has two parts: a JAVA [16] program that implements a UAS/UAC running in a PC, and an extension of the SIP server borrowed from the Columbia University. The SIP server extension consists of embedding an XML parser [14] written in C to handle the scripts written in XML defining the service required by the users.

B. SIP OVERVIEW

Nowadays three standards compete for IP Telephony signalling. The older standard is the ITU-T recommendation H.323, which defines a multimedia communications system over packet-switched networks, including IP networks [8]. The second standard, the Session Initiation Protocol (SIP) [2] comes from the IETF MMUSIC working group. The newest entrant is the MGC or the Megaco Protocol. In case of residential media gateways Megaco becomes the user to network signalling system.
We have used SIP because of its attractive features like simplicity, lower call setup time, caller preferences, ability to return different media types, and forking incoming calls to several extensions where the callee could be reached. Furthermore, the same SIP protocol is used between services and call control entities [9].
There are two components in a SIP system: a User Agent (UA) and a network server. A UA is an end system that acts on behalf of a user. Usually it consists of two parts, a client (UAC) and a server (UAS), as the user probably is wishing to both be able to call and to be called. The UAC is used to initiate a SIP request while the UAS receives requests and returns responses on behalf of the user.
There are two kinds of network servers, namely, the proxy and the redirect servers.

1. A SIP proxy server forwards requests to the next server after deciding which one it should be. This next server could be any kind of SIP server and the proxy does not need to know the type of the next server. Before the request has reached the UAS it may have traversed several servers. Those will be traversed in reverse order by the response.
2. A redirect server (Figure1) does not forward requests to the next server. Instead of that, it sends a redirect response back to the client containing the address of the next server.
3. There is also a server that accepts REGISTER requests, which is called the registrar. A registrar is typically co-located with a proxy or redirect server and may also offer location services.

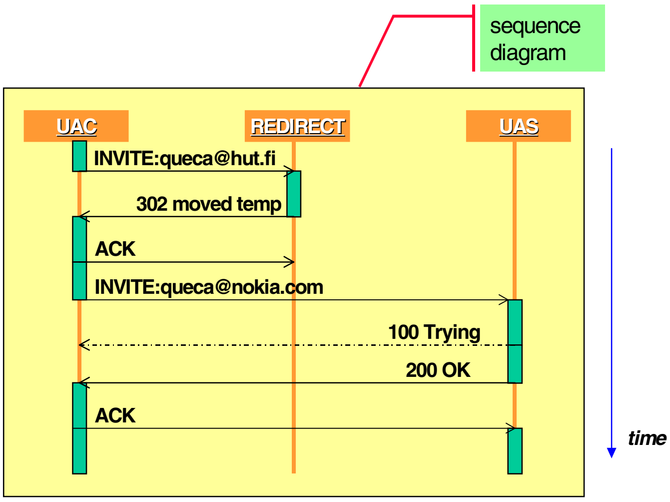


Figure 1: SIP operation in redirect mode

A SIP message is either a *request* from a client to a server or a *response* from a server to a client [5]. There are six possible *methods* that can be used in a *request* message. Those methods are *invite, ack, bye, cancel, option* and *register*.

In SIP there are six types of responses depending on how the request proceeds. They are *informational, success, redirection, client error, server error* and *global failure*.

A successful SIP invitation consists of two requests: an *INVITE* followed by an *ACK*. Figure 2 presents the basic transactions to setup a call [7].

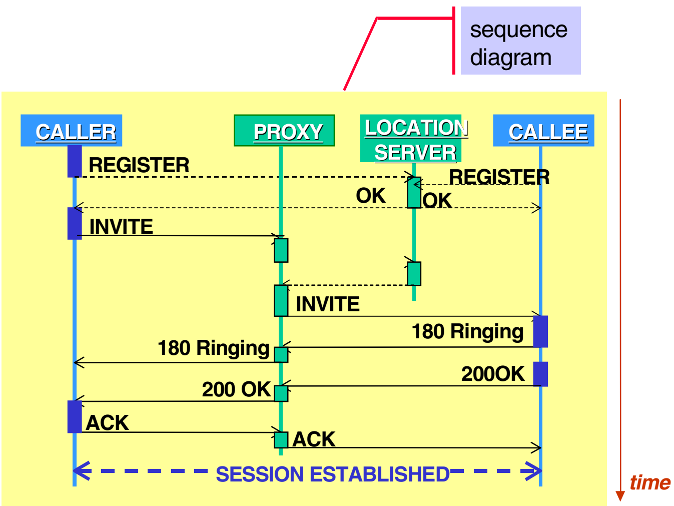


Figure 2: Basic transactions of Call Setup in SIP

The *INVITE* request asks the callee to join a particular conference or to establish a two-party conversation. After the callee has agreed to participate in the call, the caller confirms that it has received that response by sending an *ACK* request. If the caller no longer wants to participate in the call, it sends a *BYE* request.

SIP users are identified through a hierarchical URL defined by elements such as a user's phone number or host name, similar to the e-mail address.

When user A decides to call user B, the caller sends an *invite* request to the callee to initiate the call. The *invite* request contains enough information (caller media type, format, preferences,…) for the called party to join the session.

The *invite* request is sent to the SIP server that can be a proxy or redirect one. The request can go through several servers until it arrives to the final party. Afterwards, the callee receives the *invite* request and depending on his/her decision the connection can be established or the call can be directly forwarded to a voicemail or to another address or user.

C. THE INTERNET CALL WAITING PROJECT

1st. Objectives

In this project, the main objective was to demonstrate how new services can be defined from end points attending to the end-user wishes. The Session Initiation Protocol (SIP) was used for signalling features and the Call Processing Language (CPL) [12] for describing the service. The mechanism briefly is as follows [6]. A user decides that he/she wants a certain service, so he/she sends to the SIP server a *register* message, where his/her information and the definition of the service required are included [3]. To write the service definition we have used XML. When the server receives the *register* message, it reads the user preferences from the script included in the message. Next the server stores the user's service information that will be consulted when a new call for the user arrives to the server. Users can define as many services as they desire, storing them in the server and enjoying them.

In this way, services in IP Telephony can be defined like a new concept of services *residing* in their own level independent of the signalling protocol.

2nd. Project structure

Figure 3 shows the structure of the project. Empty boxes represent the modules developed in Helsinki University.

The main SIP page was the initial point to start with before trying to develop anything. Understanding the behaviour of the protocol was essential to be able to work on it. We also studied the Session Description Protocol for media requirements [10].

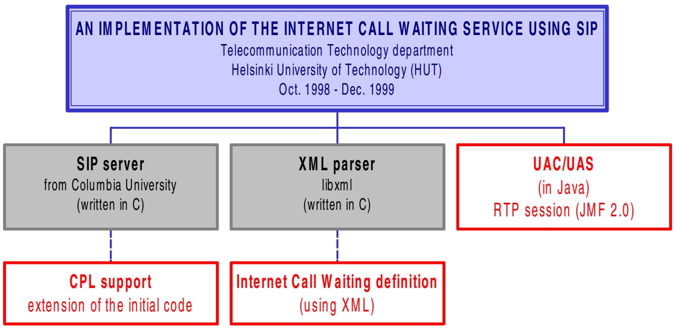


Figure 3 Project Structure

After installing the Columbia SIP server with the external libraries that it requires (clc, gdbm), the next step was writing a JAVA program implementing the User Agent Client (UAC) and the User Agent Server (UAS). The final JAVA code (called *IPtele* program) has around 2200 lines.
At this point it was also necessary to study and analyse the Java Media Framework Module (JMF 2.0) to insert an RTP session within the code [16].
Jumping to the service we decided to use the Call Processing Language (CPL) method to implement the service [11]. For that purpose we had to find a suitable parser (libxml) for embedding in the SIP server to manage the service. This part of the project also required familiarising with XML to write the document describing the Internet Call Waiting service. Installing and adding the parser to the server (SIP server extension) was done very carefully to have all the libraries well fixed.

3rd. Helsinki University of Technology extension

The extension of the Columbia University (CU) SIP server made by Helsinki University of Technology (HUT) can be understood comparing figures 4 and 5.

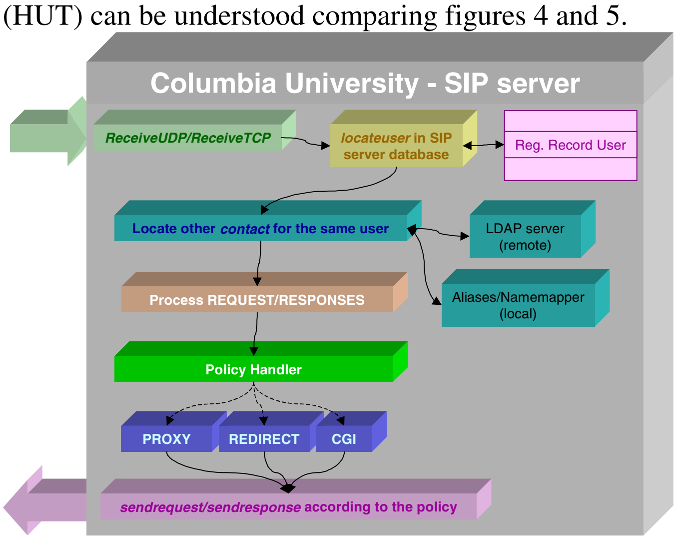


Figure 4: Columbia University SIP server

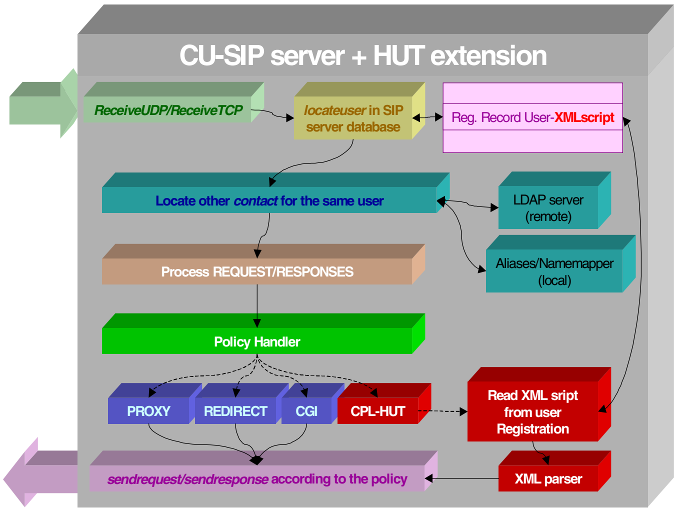


Figure 5: CU-SIP server and HUT extension

As the SIP server code is totally written in C, it was better to choose an XML parser also written in C. Studying all the possibilities (Expat, LT XML, etc.) we finally decided to work with the XML Library called *libxml* built by Daniel Veillard [14].
Libxml understands the structure of XML Document Type Definitions (DTDs) and can validate documents against them. While developed and mainly used by its author under Linux, it is a fairly portable library under different platforms. We have used it under Sun Solaris system.
Embedding the XML parser (libxml) in the SIP server code, we provide the server XML document management and somehow *service understanding*.

4th. IPtele program

The main JAVA program features are: handling windows and managing user actions, creating a window to act as an interface with the user, creating the SIP User Agent (SIP_UA) and finally giving the control to both modules.
The SIP_UA created by the main program presents four modules:

1. **User Agent Client (UAC) module**.

This class is invoked when the SIP_UA is created. The UAC constructor initialises all the variables for incoming/outgoing messages and the sockets needed for the communication. This class also implements different methods to send requests and responses on behalf user actions and upon variable values or status.
Three modules are created inside the UAC: the UAS (*FormPack*), the *send()* function and the *CaptureSender()* class.

The *FormPack* function is initialised to create the messages according to the SIP syntax. It requires a well-formed SIP packet and the SDP declaration message based on user resources [10].

The *send()* function is created in the UAC constructor and is used to establish a link with the SIP server for sending the packets.

After the message transactions a call is established and thereafter the UAC initialises the *CaptureSender()* class.

2. **User Agent Server (UAS) module.**

This class waits for incoming packets from the SIP server and also invokes the parser to manage the scripts.

3. **SIP *parse* module.**

This class is created at the UAC initialisation. Its structure includes SIP messages syntax to be able to strip out the incoming messages sent by the UAS. It also sets the environment variables properly for posterior UA action.

4. **Real Time Protocol (RTP) module.**

This module is implemented using JAVA JMF 2.0. To establish the RTP session, it captures the participant IP addresses sent by the UAC and the media type required according to the SDP declaration included in the messages.

After the initial SIP message transactions, this module creates a media session between the two addresses previously captured. The media requirements are read from the environment variables filled from the SDP packets received.

Finally, when the call is established this module is re-called. Afterwards, the voice is taken from the microphone and sent through this session to the remote user. This function also processes the incoming RTP packets carrying voice.

5th. Service definition

The Internet Call Waiting is defined by the following structure, which will be explored by the XML parser [1].

```
<?xml version="1.0"?>
<call Type="ICW">
        <proxy>
          <icw>
            <forward>
              <link ref="voicemail"/>
            </forward>
            <success>
              <location url="queca@pc2.tct.hut.fi"/>
            </success>
            <reject>The user is Busy now</reject>
          </icw>
          <busy/>
          <noanswer/>
          <failure/>
        </proxy>
        <response status="busy"/>
</call>
```

When the script is received in the SIP server, the XML parser translates all the information about the user requirements to data structures. Those will be used to perform the right decision when another request is coming.

6th. Graphic Implementation

Our ICW implementation uses a pop-up window to notify the user about a new call [1] (Figure 6).

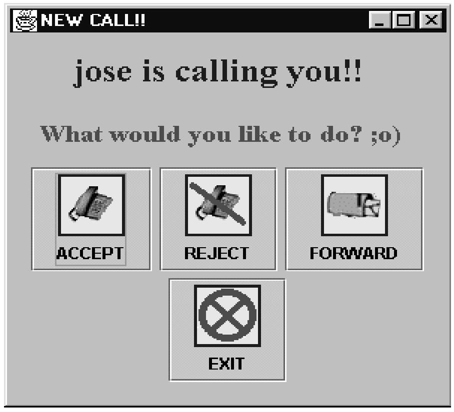


Figure 6: New incoming call window

As it is shown, the callee can accept, reject and forward the call to a voicemail server.

D. CONCLUSIONS

An innovative method to design new services has been proposed. New IP Telephony services could be implemented using these new tools. The Internet Call Waiting was defined as an example in order to present all the advantages that these tools will give to the advanced IP Telephony services. In our context the main part of the service resides in the user agent client and the user agent server written in JAVA. The user agent also registers service instructions written in XML in the SIP server of the user. These intructions are called the Call Processing Language programs. Consequently, service intelligence is fully controlled by the user but its implementation is distributed in the terminal and in the network. As a minimum the network server has to fulfil the gaps in continuous service left when the terminal or the user is either occupied or not available.

E. REFERENCES

[1] Espigares I., "An Implementation of the Internet Call Waiting service using SIP," *Master's thesis at Helsinki University of Technology, Finland, and Polytechnic University of Valencia, Spain*, <URL:http://keskus.hut.fi/tutkimus/ipana/paperit/> <URL:http://www.cs.columbia.edu/~hgs/sip/papers.html>, Dec 1999.

[2] M.Handley, H.Shulzrinne, E.Schooler and J.Rosenberg, "SIP: Session Initiation Protocol," *Request for Comments 2543, Internet Engineering Task Force*, Mar.1999.

[3] J. Lennox and H. Schulzrinne, "Transporting User Control Information in SIP REGISTER Payloads," *Internet Draft, Internet Engineering Task Force,* Feb. 1999. *Work in progress*.

[4] J. Rosenberg, J. Lennox and H. Schulzrinne, "Programming Internet Telephony Services," *IEEE Internet Computing Magazine*, May/Jun. 1999.

[5] J. Rosenberg and H. Schulzrinne, "The Session Initiation Protocol: Providing Advanced Telephony Services across the Internet," *Bell Labs Technical Journal*, Vol. 3, No. 4, Oct/Dec. 1998, pp. 144-160.

[6] A. Johnston, S. Donovan, R. Sparks, C. Cunningham and K. Summers, "SIP Telephony Call Flow Examples," *Internet draft, Internet Engineering Task Force*, Oct.1999. *Work in progress*.

[7] R. Sparks, C. Cunningham, A. Johnston, S. Donovan and K. Summers, "SIP Telephony Service Examples With Call Flows," *Internet draft, Internet Engineering Task Force*, Oct. 1999. *Work in progress*.

[8] I. Dalgic and H. Fang, "Comparison of H.323 and SIP for IP Telephony Signalling," *Photonics East*, *Proceeding of SPIE'99,* Boston, Massachusetts, Sep.1999.

[9] H. Schulzrinne and J. Rosenberg, "A Comparison of SIP and H.323 for Internet Telephony," *proceedings of the 1998 Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV'98)*, Jul. 1998, Cambridge, England.

[10] M. Handley and V. Jacobson, "SDP: Session Description Protocol," *Request for Comments 2327, Internet Engineering Task Force*, Apr. 1998.

[11] J. Lennox and H. Schulzrinne, "Call Processing Language Framework and Requirements," *Internet Draft*, *Internet Engineering Task Force*, Oct. 1999. *Work in progress.*

[12] J. Lennox and H. Schulzrinne, "CPL: a language for user control of Internet telephony services," *Internet Draft, Internet Engineering Task Force*, Mar. 1999. *Work in progress*.

[13] "Extensible Markup Language (XML) 1.0," online <URL: http://www.w3.org/TR/ REC-xml>, W3C Recommendation 10 Feb.1998.

[14] XML Resource Guide - XML Parsers, online <URL: http://www.xml.com/xml/pub/Guide/XML_Parsers>

[15] N. Walsh, "A Technical Introduction to XML," online <URL: http://xml.com/xml/pub/98/10/guide0.html>

[16] "JAVA tutorial," <URL:http://java.sun.com/docs/books/tutorial/>.

TEKNILLINEN KORKEAKOULU
HELSINKI UNIVERSITY OF
TECHNOLOGY

New Tools for Programming Intelligent IP Telephony Services

Inmaculada Espigares
Jose M. Costa
Raimo Kantola
{queca, jose}@tct.hut.fi
Raimo.Kantola@hut.fi

The First IP Telephony Workshop
Berlin, April 2000

TEKNILLINEN KORKEAKOULU
HELSINKI UNIVERSITY OF
TECHNOLOGY

Contents

- SIP overview
- The Internet Call Waiting Service
  - Objectives
  - Project structure
  - Helsinki University of Tech. extension
  - JAVA SIP client: *IPtele* program
  - Registration
  - Service Definition using XML
  - Graphic implementation
- Conclusions

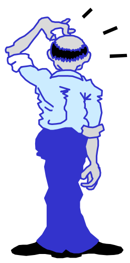
TEKNILLINEN KORKEAKOULU
HELSINKI UNIVERSITY OF
TECHNOLOGY

SIP overview

- **SIP attractive features**
  - Simplicity
  - Lower call set up than H.323
  - Proxy and redirect servers
  - Caller preferences
  - Ability to return different media types
  - Forking
  - Used between both,services and call control entities
  - Advanced services: SIP personal mobility...

TEKNILLINEN KORKEAKOULU
HELSINKI UNIVERSITY OF
TECHNOLOGY

Objectives



How can new services be defined

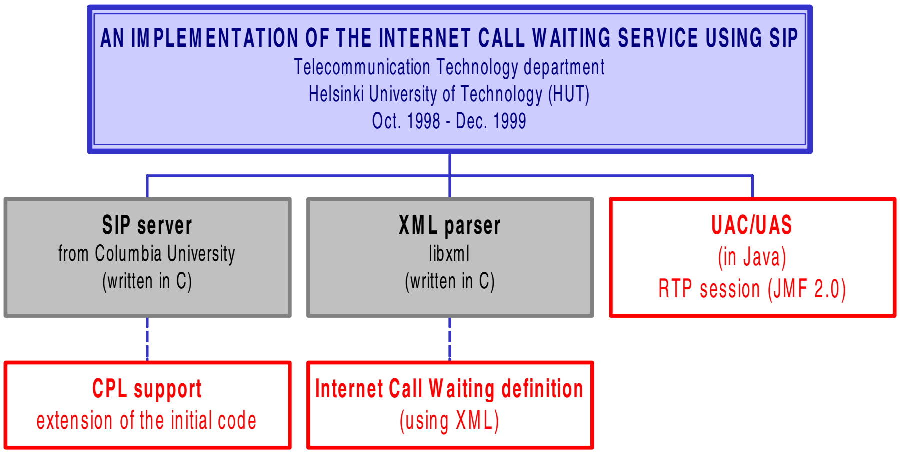
from end points???????

Definitely... using **NEW TOOLS !!**

- <u>SIP</u> for signalling features
- <u>JAVA</u> for building the SIP client
- <u>CPL</u> for describing the services
- <u>XML</u> for defining the service files
- <u>XML parser</u> for interpreting the scripts

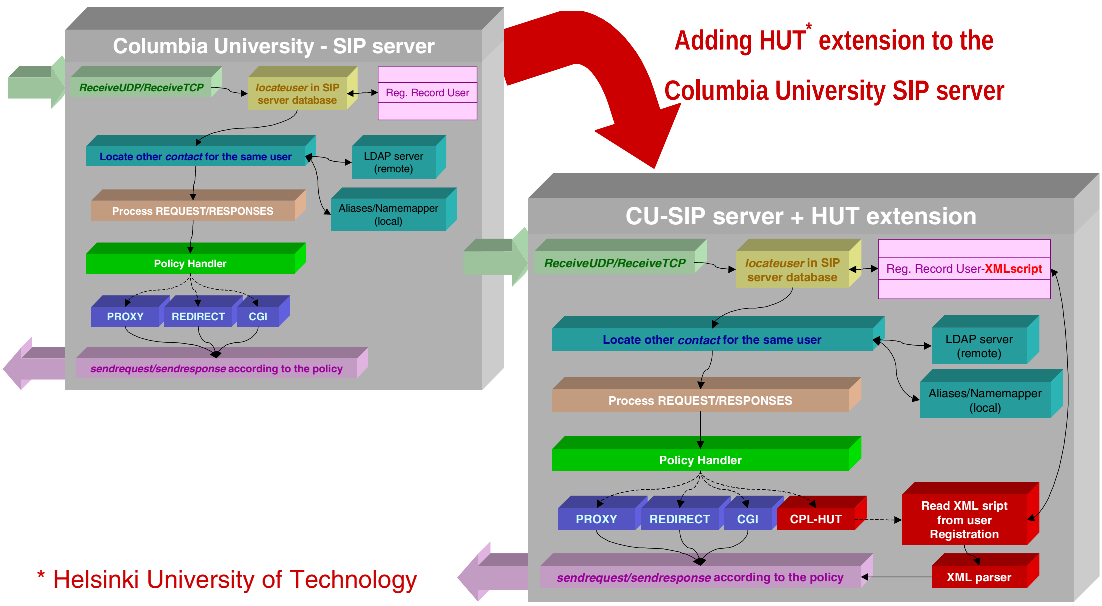
TEKNILLINEN KORKEAKOULU
HELSINKI UNIVERSITY OF
TECHNOLOGY

Project structure



TEKNILLINEN KORKEAKOULU
HELSINKI UNIVERSITY OF
TECHNOLOGY

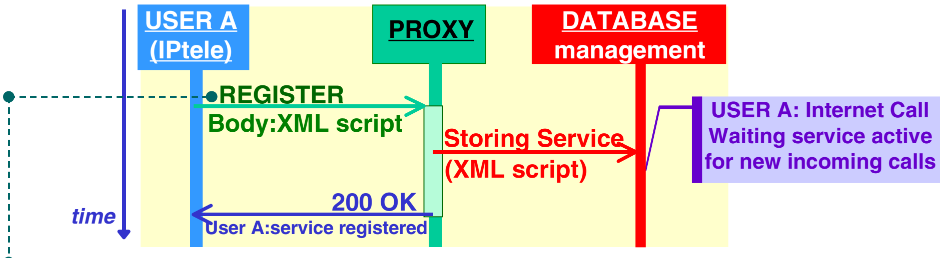
HUT extension



TEKNILLINEN KORKEAKOULU
HELSINKI UNIVERSITY OF TECHNOLOGY

Registration

User A registers the Internet Call Waiting service sending the REGISTER message to the server (with the script defining the service in the body).

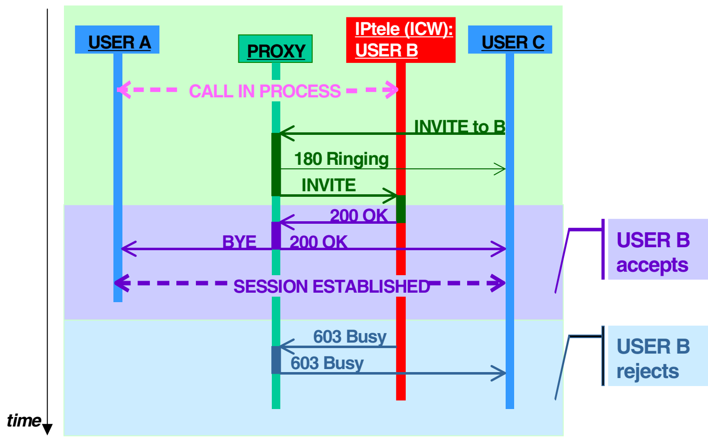



```
REGISTER sip:ericy.com SIP/2.0
Via: SIP/2.0/UDP 63.86.21.120:5060
From: sip:queca@helsinkiu.ericy.com
To: sip:queca@helsinkiu.ericy.com
Call-ID: 6088@helsinkiu.ericy.com
Cseq: 1 REGISTER
Contact:<sip:queca@helsinkiu.ericy.com:5060;transport=udp>
Expires: 7200
Content-Length: 350
Content-Type: application/hut-cpl
<?xml version="1.0"?>
<call Type="ICW">
.....
</call>
```

TEKNILLINEN KORKEAKOULU
HELSINKI UNIVERSITY OF TECHNOLOGY

IPtele program

Messages transferred between the IPtele client and the SIP server



TEKNILLINEN KORKEAKOULU
HELSINKI UNIVERSITY OF
TECHNOLOGY

Service definition

- **XML advantages**
  - Simplicity: basic structures
  - Extensibility
  - Interoperability
    - On a wide variety of platforms
    - Interpreted with a wide variety of tools
  - Openness: anyone can do it!
  - Applications:
    - Data exchange: Machine-Machine
    - Data interchange: Human-Machine

```
<?xml version="1.0"?>
<call Type="ICW">
  <proxy>
    <icw>
      <forward>
        <link ref="voicemail"/>
      </forward>
      <success>
        <location url="queca@pc2.tct.hut.fi"/>
      </success>
      <reject>The user is Busy now</reject>
    </icw>
    <busy/>
    <noanswer/>
    <failure/>
  </proxy>
  <response status="busy"/>
</call>
```

TEKNILLINEN KORKEAKOULU
HELSINKI UNIVERSITY OF
TECHNOLOGY

Graphic implementation

Internet Call Waiting service

Callee notification

New incoming call !!

Attending to the user wishes:

Accepting the call?

Rejecting the call?

Forwarding the call?

TEKNILLINEN KORKEAKOULU
HELSINKI UNIVERSITY OF
TECHNOLOGY

Conclusions

Internet Call Waiting service is a useful solution for the calls that otherwise would be LOST and also for rejecting undesirable incoming calls !!

Internet Call Waiting service helps not to waste network resources and contributes to call completion !!

Pop-up dialogue boxes make it simpler and easier the notification to the user !!

We have developed a new mechanism to create advanced IP Telephony services using innovate TOOLS !!

Keynote 3: From Telephony to IP Communications

Henry Sinnreich
MCI Worldcom, Richardson, Texas, USA

The 1.st IP Telephony Workshop

From Telephony to IP Communications

Henry Sinnreich
MCI WorldCom*

Berlin, April 12-13, 2000

* Disclaimer: The opinions presented here may or may not be those of MCI WorldCom

"Free" IP Telephony Enables Premium IP Communication Services

Transport Level

- Best effort w. bandwidth
- Premium
  - QoS Assured
  - QoS Enabled
- Authentication
- Encryption

Web Integration*

- Unified IP messaging
- Internet multimedia
- Games

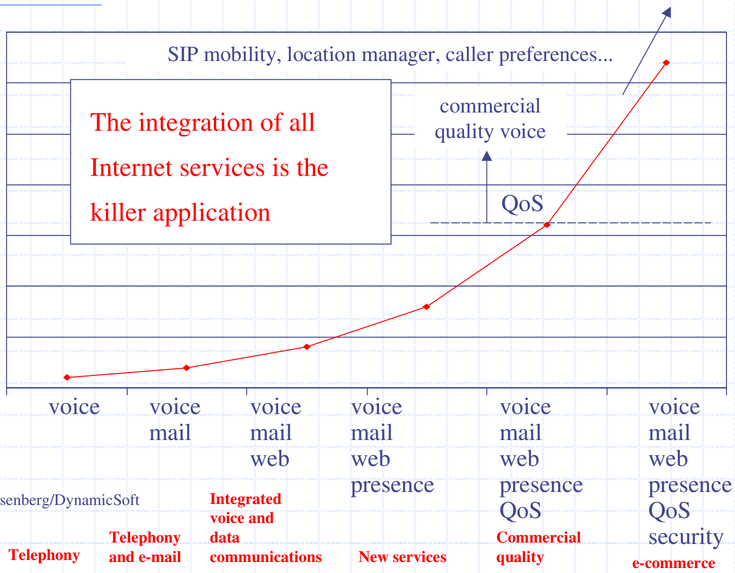
* Most yet to be invented

Application Level (short list)

- **Gateway services to PSTN**
- **PSTN and Internet interaction**
- **AIN-like: CS-1 and CS-2**
- **Caller/called user preferences**
- **IP Centrex and PC integration**
- **Presence and instant chat**
- **Conferencing portfolio**
- **Local number portability**
- **Mobility**
  - **Global**
  - **Network**
  - **Device**
  - **Application**
- **Location manager**
- **PIM integration**

References at http://www.cs.columbia.edu/sip/papers.html

Value of IP Communications



Reference:
Jonathan Rosenberg/DynamicSoft

04.04.00 From VoIP to IP Communications 3

Relevant IETF Working Groups

http://ietf.org/html.charters/wg-dir.html

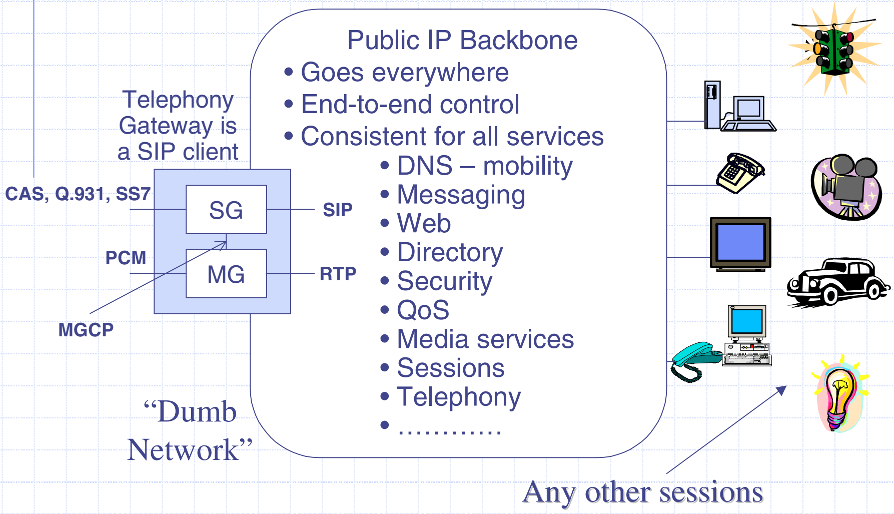
- **Audio/Video Transport (avt) - RTP**
- **Differentiated Services (diffserv) – QoS in backbone**
- **IP Telephony (iptel) – CPL, GW location, TRIP**
- **Integrated Services (intserv) – end-to-end QoS**
- **Media Gateway Control (megaco) – IP telephony gateways**
- **Multiparty Multimedia Session Control (mmusic) – SIP, SDP**
- **PSTN and Internet Internetworking (pint) – mixt services**
- **Resource Reservation Setup Protocol (rsvp)**
- **Service in the PSTN/IN Requesting InTernet Service (spirits)**
- **Session Initiation Protocol (sip) – signaling for call setup**
- **Signaling Transport (sigtran) – PSTN signaling over IP**
- **Telephone Number Mapping (enum) – surprises !**
- **Instant Messaging and Presence Protocol (impp)**

04.04.00 From VoIP to IP Communications 4

Telephony on the Internet

may not be a stand-alone business, but part of all Web/IP services

SIP/RTP Media Architecture



Carrier IP Telephony Using SIP

- *Cablelabs/AT&T:*

The Distributed Control Services Architecture

- *MCI WorldCom:*

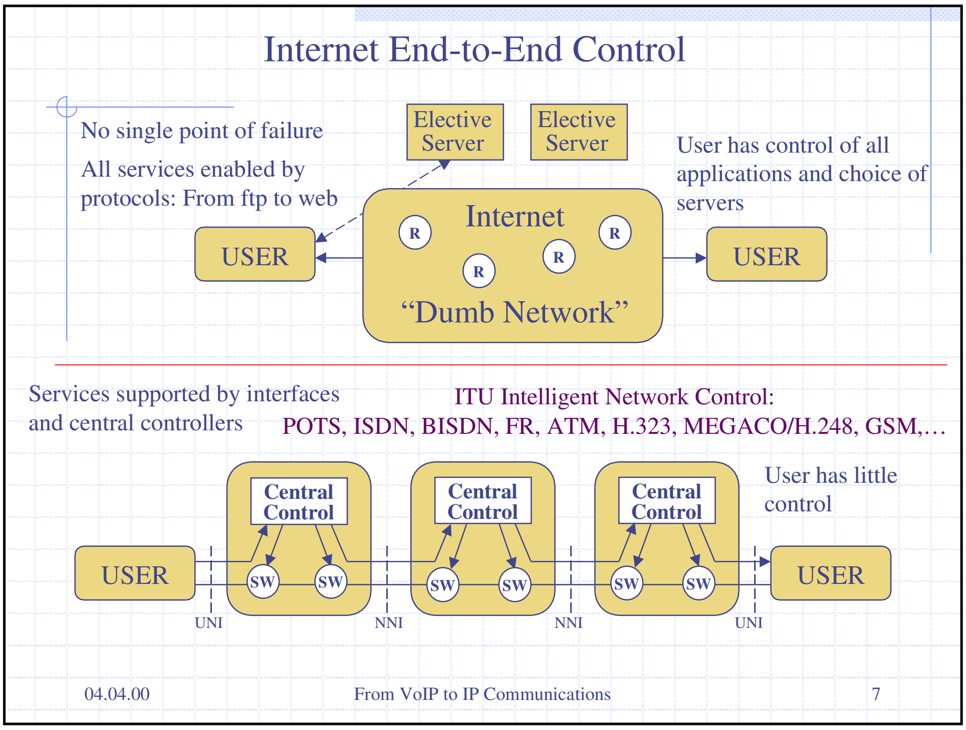
IP Communications Trial, IETF SIP call flows

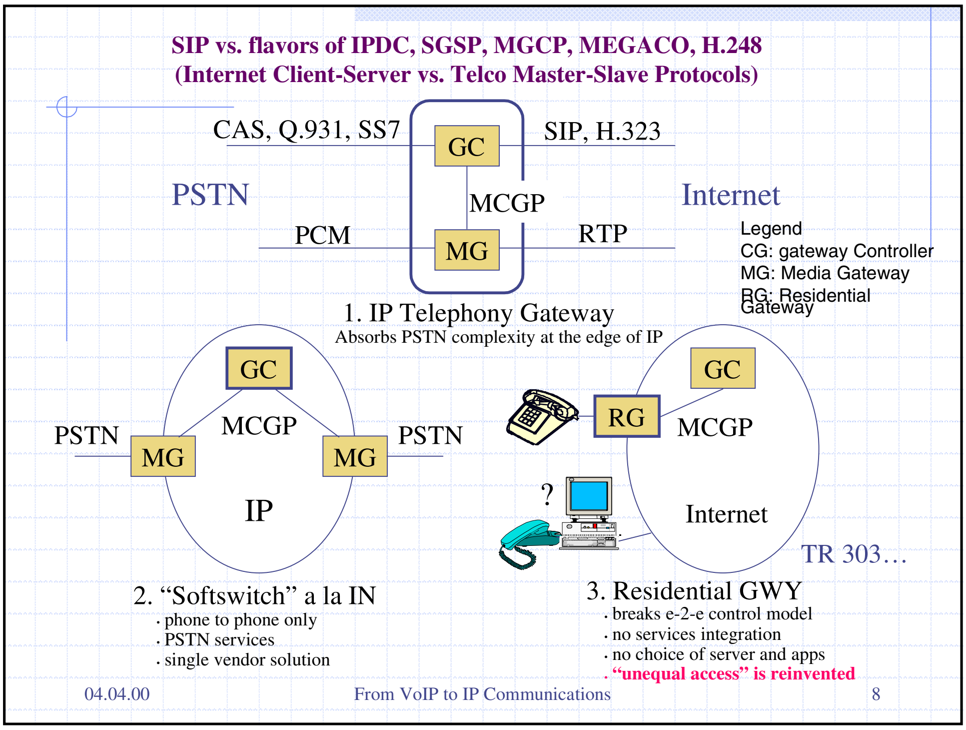
- *Level 3:*

Inter-gateway signaling (SIP BCP-T)

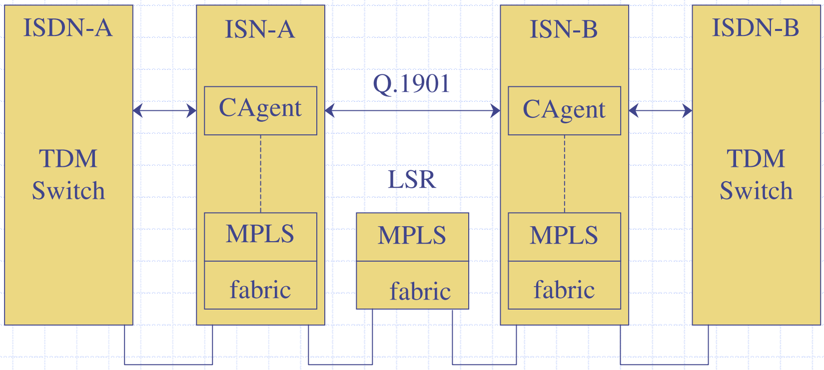
- *Emerging interest and contributions by European carriers and industry*

References
http://ietf.org/ID.html - search for DCSGROUP
http://www.cs.columbia.edu/sip/papers.html


Internet End-to-End Control
No single point of failure
All services enabled by protocols: From ftp to web
Elective Server
Elective Server
User has control of all applications and choice of servers
USER
Internet
R
R
R
R
"Dumb Network"
USER
Services supported by interfaces and central controllers
ITU Intelligent Network Control:
POTS, ISDN, BISDN, FR, ATM, H.323, MEGACO/H.248, GSM,…
Central Control
Central Control
Central Control
User has little control
USER
SW
SW
SW
SW
SW
SW
USER
UNI
NNI
NNI
UNI
04.04.00
From VoIP to IP Communications
7


SIP vs. flavors of IPDC, SGSP, MGCP, MEGACO, H.248
(Internet Client-Server vs. Telco Master-Slave Protocols)
CAS, Q.931, SS7
GC
SIP, H.323
PSTN
MCGP
Internet
PCM
MG
RTP
Legend
CG: gateway Controller
MG: Media Gateway
RG: Residential Gateway
1. IP Telephony Gateway
Absorbs PSTN complexity at the edge of IP
GC
MCGP
PSTN
MG
MG
PSTN
IP
RG
GC
MCGP
?
Internet
TR 303…
2. "Softswitch" a la IN
. phone to phone only
. PSTN services
. single vendor solution
3. Residential GWY
. breaks e-2-e control model
. no services integration
. no choice of server and apps
. "unequal access" is reinvented
04.04.00
From VoIP to IP Communications
8

Q.1901 Call Control for BICC over MPLS



Combining problems of QoS of packet networks with central control and signaling of circuit switched networks.

IP SIP Phones and Adaptors

Are Internet hosts
- Choice of application
- Choice of server
- IP appliance

Implementations
- 3Com (2)
- Cisco
- Columbia University
- Mediatrix (1)
- Nortel (3)
- Pingtel
- Siemens

1

2

3

Quality of Service and Payments

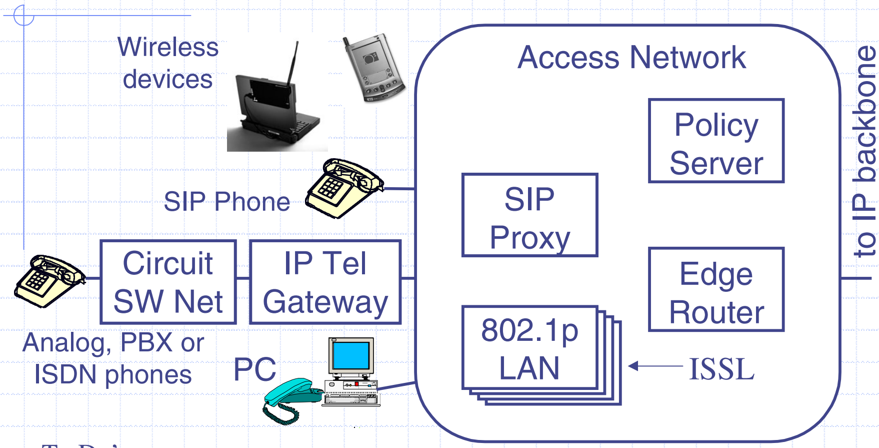
- Commercial IP telephony requires QoS
- QoS requires payments and settlements
- Service providers need business case for QoS

Circular dependency…

Conclusion: IP telephony may be the trigger for global QoS deployment and interoperable payments

Reference: **H. Sinnreich, S. Donovan, D. Rawlins, S. Thomas: Interdomain IP Communications with QoS, Authorization and Usage Reporting. Work in progress, IETF, October 1999. http://www.ietf.org/internet-drafts/draft-sinnreich-interdomain-sip-qos-osp-01.txt**

QoS Network Model for IP Telephony



To Do's:

- New COPS clients for SIP-RSVP-OSP coordination
- RSVP aggregation, traffic classes, Differentiated Services
- Policy and metering in access and transit networks

QoS Models for IP Telephony

- QoS Assured: Emulates the PSTN
  - No QoS – no call
- QoS Enabled: Internet style
  - QoS most of the time
  - Give user the choice to proceed without QoS
- Best Effort: No commercial grade service

Local Policy Models

- Push model, pros/cons
- Pull model, pros/cons

http://www.ietf.org/internet-drafts/draft-sinnreich-interdomain-sip-qos-osp-01.txt

Call Flows for Basic Telephony

SIP proxy server controls calls in each domain

- SIP Registration Services
- Call Setup
  - Client to Client
  - Client to Gateway
  - Gateway to Client
  - Gateway to Gateway

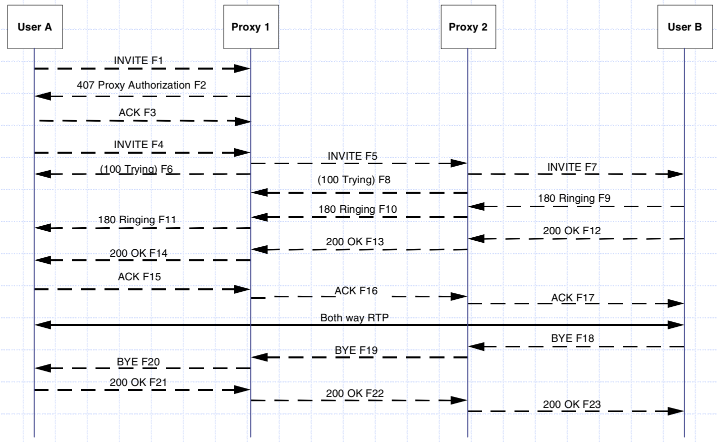
• Success scenario

• Failure scenarios

Reference: SIP Telephony Call Flow Examples by A. Johnston, S. Donovan, R. Sparks, C. Cunningham, K. Summers, D. Willis, J. Rosenberg, H. Schulzrinne. February 2000.

http://ietf.org/internet-drafts/draft-ietf-sip-call-flows-00.txt

Basic SIP Call Flow Example



IP Centrex - Service Examples

Across Internet Domains *

Service Features:

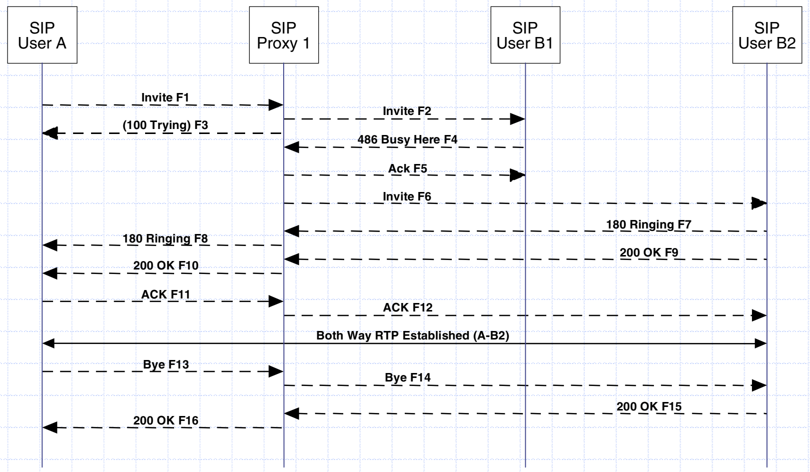
- Call Hold
- Consultation Hold
- Unattended Transfer
- Call Fwd Unconditional
- Call Fwd on Busy
- Call Fwd on No Answer
- 3-Way Conference
- Single Line Extension
- Find-Me
- Incoming Call Screening
- Outgoing Call Screening
- Secondary Number – In
- Secondary Number – Out
- Do Not Disturb
- Call Waiting

* Think of QSIG…

Reference: SIP Telephony Call Flow Examples by A. Johnston, S. Donovan, R. Sparks, C. Cunningham, K. Summers, D. Willis, J. Rosenberg, H. Schulzrinne. February 2000.

http://ietf.org/internet-drafts/draft-ietf-sip-call-flows-00.txt

Service Example: Forward on Busy



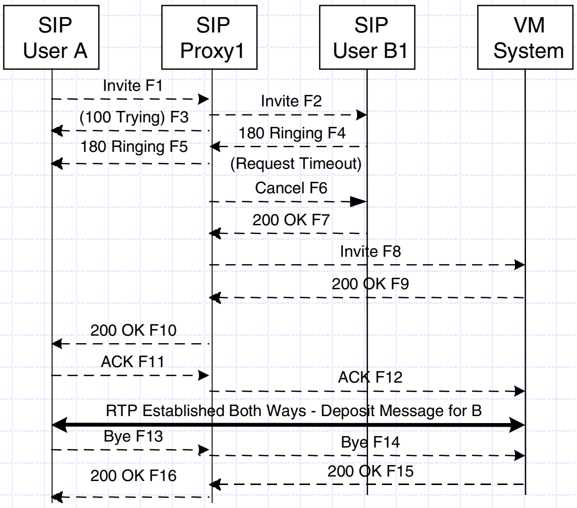
IP Voice Mail

- Direct call to voice mail system (for top level menu)
- Message deposit
  - Call to known subscriber
    - Forwarded on no answer
    - Forwarded on busy
    - Direct call to mailbox
- Message retrieval
  - Call from known client
  - Call from authenticated user

Reference: Control of Service Context using SIP Request-URI by Ben Campbell and Robert Sparks, I-D, January 2000

http://ietf.org/internet-drafts/draft-campbell-sip-service-control-00.txt

Voice Mail Example: Forwarded on No Answer



Presence, Instant Messaging and Voice

Presence in proprietary commercial systems got the interest of the IETF IMMP WG

http://www.ietf.org/internet-drafts/draft-ietf-impp-model-03.txt

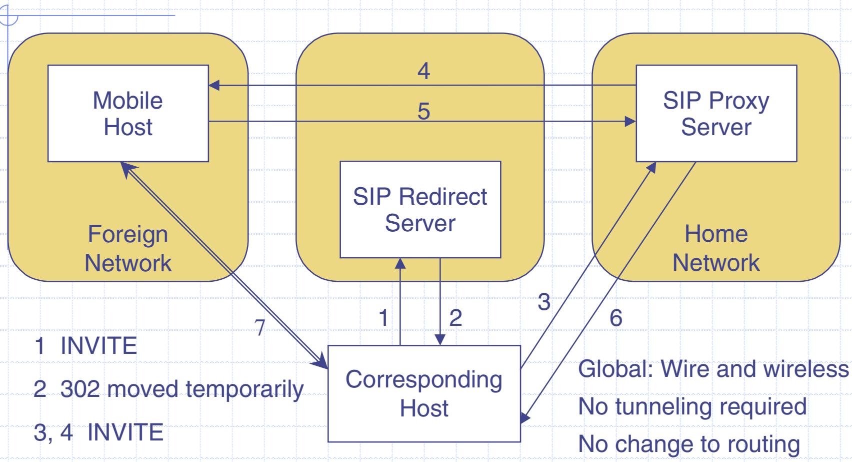
SIP User Location Example

SIP supports mobility across networks and devices

Q=*quality* gives preference

```
SIP/2.0  310 Moved temporarily
Location: sip:henry@wcom.com
          ;service=IP,voice mail
          ;media=audio ;duplex=full ;q=0.7
Location: phone: +1-972-555-1212; service=ISDN
          ;mobility=fixed; language=en,es, ;q=0.5
Location: phone: +1-214-555-1212; service=pager
          ;mobility=mobile
          ;duplex=send-only ;media=text; q=0.1; priority=urgent
          ;description="For emergency only"
Location: mailto: henry@wcom.com
```


SIP Mobility Support



1 INVITE
2 302 moved temporarily
3, 4 INVITE
5, 6 OK
7 Data

Global: Wire and wireless
No tunneling required
No change to routing
For fast hand-offs use:

- Use Cellular IP or
- Use DRCP

SIP Mobility

Pre-call mobility

- MH can find SIP server via multicast REGISTER
- MH acquires IP address via DHCP
- MH updates home SIP server

Mid-call mobility

- MH->CH: New INVITE with Contact and updated SDP
- Re-registers with home registrar

Need not bother home registrar: Use multi-stage registration

Recovery from disconnects

http://www.cs.columbia.edu/sip/talks/mobility.pdf

Interoperability Testing
Who are the players?

Proprietary networks come and go* but the Internet stays and grows. Still, let's see…

Teams at the 3rd SIP bakeoff on December 6-8, '99:

Nortel-1, Pingtel, Nortel-2, Mediatrix, Netspeak, Telogy, Nuera, Catapult, Ericsson-2, Mitel, Agilent, Dynamicsoft, Broadsoft, VTEL, Delta, Radcom, E*Club, 8x8, Helsinki Tech U., Columbia U., HP Labs, Indigo, OZ.com, Vovida, Cisco, IPCell, FacetCorp, MCIW-3, 3Com-1, MCIW-1, Ericsson-1, 3Com-2, MCIW-2

Separate list of companies supporting the SIP based Cablelabs/AT&T Distributed Call Signaling Architecture

* Remember SNA and DECnet dominance in the '80s…

Summary: IP Communications Services

Principal reason to invest: New services

- ⇧ The potential of complete integration with Web based services has not yet been fully explored
- ⇩ IN/AIN, etc. central control models have run the course

Innovation

- ⇧ Innovation will be the main enabler for new IP communication services – just like innovation drove e-commerce
- ⇧ Flourishes in distributed control environment as in all other Internet and Web based services
- ⇩ Hard to impossible under central control.

Additional References

Book on “Internetworking Multimedia” by Jon Crowcroft, Mark Handley, Ian Wakeman, UCL Press, 1999 by Morgan Kaufman (USA) and Taylor Francis (UK)

RFC 2543: “SIP: Session Initiation Protocol”, version bis
http://www.cs.columbia.edu/~hgs/sip/drafts/draft-ietf-sip-2543bis-00.pdf

The IETF SIP Working Group home page
http://www.ietf.org/html.charters/sip-charter.html and see the supplemental page at http://www.softarmor.com/sipwg/

SIP Home Page
http://www.cs.columbia.edu/sip/

Papers on IP Telephony
http://www.cs.columbia.edu/sip/papers.html

Architectures and Implementations

IPTalk: Bringing DSS1-like Services to IP Telephony

Marcel Dasen, Jonas Greutert, Bernhard Plattner
ETH Zürich, Computer Engineering and Networks Laboratory
March 31, 2000
dasen@tik.ee.ethz.ch, greutert@tik.ee.ethz.ch

Abstract -- **IPTalk is a experimentation platform for IP telephony. The IPTalk terminal is a light weight IP voice terminal which connects to Ethernet. High quality voice, low cost, plug and play operation were its main design objectives. We believe that telephony over IP based networks (VoIP) faces two main challenges service quality and service signaling. The typical best effort IP service offered by IP networks is not suited for voice. Delay and throughput guarantees are needed in order to provide a high quality voice service. Today the main source of delay are end-systems and delay jitter in the network. Reduction of this delay to levels of the public switched telephony networks (PSTN) is a requirement for the success of IP based telephony. The second challenge is service signaling. PSTN offers a wealth of services such as call deflection, multiparty conferences and other IN type services. Unlike the PSTN, in the IP network services are implemented as end-system applications and not in the network nodes. In this work we have defined the services provided by DSS1 in the Internet style user-to-user (end-to-end) signaling by defining appropriate SIP procedures. QoS signaling and charging has been integrated by extending SIP. The proposed network and terminal architecture, allows for end-to-end telephony service with high service quality. In our architecture service quality is reached through synchronous terminal design and through admission control at ingress routers. We have addressed charging by including a simple reservation and feedback mechanism for guaranteed network service.**

I. Introduction

Telephony is one of the most challenging applications for IP based networks. The best effort service of today's Internet is well suited for most data transfer applications, but not for the service quality requirements (QoS) of a voice applications. IPTalk is an experimental voice over IP terminal for the validation and design of new protocols for a quality of service enabled Internet. High quality voice, "plug and play" configuration and distributed intelligence approach were the major design objectives. A VoIP terminal has to offer all the services of today's public switched telephony network (PSTN) but in addition VoIP has the potential for efficient integration with data services. Claiming to replace an existing, well functioning technology, good reasons have to be offered. We believe in the following three arguments supporting VoIP:

The tremendous growth of IP traffic will eventually turn a separate network for the voice application uneconomical. Interpolating the current growth rates of IP traffic (~30% for IP, 3% for wired voice) within the next five years voice will consume only a few percent of the total bandwidth of backbone networks. Therefore, it is beneficial to integrate voice into the IP networks.

The liberalization of telecommunication markets have foresterd the deployment to new access network technologies, such as cable-modem, wireless local loop (WLL) and xDSL. Voice service could become an interesting business for providers using these new access technologies.

Another trend is the use of IP technology in corporate networks. In an Intranet setup, Voice over IP (VoIP) offers simple and efficient integration with the data services. Furthermore, wiring of buildings and network management can be simplified using IP technology for both data and voice.

Voice is especially sensitive to delay and delay jitter. It has been shown that a significant source of delay, in many of today's PC based VoIP solutions, is the end-system itself [7]. Due to its synchronous design, IPTalk drastically reduces the delay introduced by the end-system. To address delay and jitter in the network our system architecture assumes the differentiated services (diffserv) approach. It can be shown that diffserv is well suited for provisioning voice service in the Internet [3].

Besides the basic transportation of voice with adequate QoS, a VoIP solution has to offer the signaling services of modern PSTN. For IPTalk we have implemented most of the services offered in ISDN through the DSS1 signaling standard. We have sought to implement these services user-user (end-to-end) as opposed to the "centralized" user-network approach of DSS1. Thus, the only enhancement we had to assume from the network is a (diffserv-) transportation service with guaranteed upper delay bounds. Our objective is end-to-end delay not exceeding 100ms. Since no established standard for call signaling exists in the Internet, we have defined SIP procedures to define DSS1 services. SIP has various advantages over existing standards such as H.323. Schulzrinne and Rosenberg [17] have compared H.323 and SIP. H.323 experiences major difficulties inter working with increasingly popular entities in the Internet, such as firewalls and address translation boxes (NATs). Additional deficiencies are that H.323 is not that easily extendable, is heavy weight and does not really scale.

We believe the emerging Internet signaling protocol SIP is much better adapted to the dynamics of the Internet, but is lacking services definition and important functionality. Besides the definition of DSS1 services, we have included admission control and user service signaling into our SIP extension. However, many aspects of charging and guaranteeing service level agreements in a multiprovider network remain open in this context [5][14].

II. Related Work

In Voice over IP, the voice quality depends mainly on the delay and the packet loss of the IP network. In a LAN, where we can assume a highly over provisioned network, best effort service is sufficient. However, as soon as we leave the LAN, voice packets have to compete for resources with other type of network traffic. Ideally, voice packets are transported in a virtual leased line, where they encounter zero delay and packet loss. There are several concepts and implementations available that support such a type of service in an IP network. The question that remains is how the reservation of resources can be coupled with the call signaling to enable high quality voice communication through an IP network.

An approach to integrate the call signaling with the resource management has been presented by Pawan Goyal et al. [8]. They propose a new architecture, consisting of customer premises equipment, edge routers and gate controllers. Edge routers represent the gate to a managed IP network and implement the policy enforcement to the access of that network. The gate controller controls the gates and provides telephony specific services, such as authentication and authorization, number translation and call routing, admission control, and signaling. The setup of a call includes the admission to access the managed IP network and the reservation of the necessary bandwidth through the network. Further, the resource management consists of a separate reserve and commit phase. Only after the called party has answered the call, the reservation is committed and the resources are made available to the call. Our approach follows a much simpler scheme but shares the basic architectural features.

Schulzrinne, Rosenberg and Lennox [18] present five different approaches to how the resource management can interact with the call signaling based on SIP. Our approach differs in the assumption of the network architecture and in that our approach does not require extensions to the Session Initiation Protocol (SIP).

The question how to add services to an Internet telephony is one of major concern. The success and the long-term viability of Internet or IP telephony depends on the set of services it can support and how fast and flexible new services can be added. First and as an initial minimum, the services of today's PSTN should be available. But IP telephony should go beyond the traditional telephony services. The capability is there in the form of the Internet protocol, but the question remains how to exploit it.

Rosenberg et al. [16] suggest two approaches to programming new SIP [10] services. The first approach uses SIP CGI executed on a SIP server. Second, they present the call processing language (CPL) [12], which allows untrusted users to define services. The CPL scripts are uploaded to a SIP server and, after verification, are activated. Both approaches allow executions triggered on any SIP message. The implementation of the DSS1 services using SIP CGI would be an overkill. We look at the DSS1 services as the basic feature set every SIP/IP terminal has to support. However, for advanced services the proposed approach looks promising.

Rizzeto and Catania [15] describe a voice over IP service architecture based on the H.323 [9] protocol. The architecture consists of a home servers and service platforms. The home server contains service objects assigned to a user. When a service is required, the service object is downloaded to the service platform and executed. The service platform is contained in the gate-keeper platform. Thus, the executing service object can access and modify all required information about the call. We think that DSS1 services would better be implemented extending H.450. For advanced services, the above architecture looks promising for extending the H.323.

An architecture for residential Internet telephony service has been described by Huitema et al. [11]. The architecture is based on MGCP [1], and consists of residential gateways, trunk gateways and call agents. The basic idea behind is to separate the media transformation functions from the gateway control function. The call agent has the responsibility for the call establishment. They describe shortly how services can be mapped on the call agent through providing an interface to SCPs to allow similar mechanisms as in intelligent networks (IN) [6]. DSS1 services could be implemented very similar. This approach does consider only the integration of traditional telephone services and does not strive to provide a platform for new services made possible by the Internet.

Lennox, Schulzrinne and La Porta [13] show how services of the traditional Intelligent Network protocols, as well as additional services can be implemented on top of SIP. However, they do not describe the provisioning of DSS1 services to SIP terminals.

The diffserv approach for enhancing the Internet with voice service seems most promising. Bandwidth broker as proposed in the diffserv framework build on service level agreements (SLA). SLA are contracts between the different diffserv domains in the network or between Internet service providers (ISP) and end-users. For end-to-end QoS guarantees SLA should be traded to make efficient use of the available resources. Such schemes have been propose by Neilson et al. in [14] and by Fankauser and Plattner in [5]. Interaction of bandwidth brokers and call signaling needs further studies.

III. The IPTalk Architecture

A. The Network Architecture

IPTalk assumes an network architecture, which can deliver better than best-effort service for the voice channel. Furthermore, we assume that there will be usage based charges for this enhanced network service. Different economic models for charging are possible, such as time based as in traditional telephony, traffic based or flat rate. We have focused our work on a differentiated services type architecture (diffserv). Diffserv can be implemented with relatively little technical complexity and needs less signaling than the integrated service model. However, our model is also suited for VPN type networks or the integrated services model.

We have included *resource reservation* and *charging information (advice of charge, AOC)* into the signaling Protocol, similar to DSS1. For these functions SIP had to be extended with a new method (INFO). Especially, in diffserv type networks full blown resource reservation protocols are an over kill, since the reservation and admission control is Traffic of the same class is aggregated in the network. Thus, end-to-end reservation messages are not needed. For the first hop alone a much simpler scheme suffices and consumes less resources on the terminal.

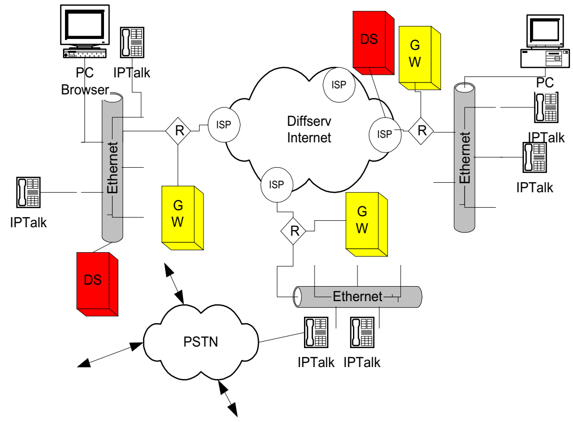


Figure 1: The network model. We assume that IPTalk terminals are placed on Ethernet type access networks. Directory servers (DS) are needed to resolve SIP addresses outside of the broadcast domain. The resource reservation gateway (GW) is located at the ingress node of an internet service provider (ISP). This is also the entrance into the diffserv domain. The gateways responsibility is to enable prioritized traffic from a terminal and interacts with the charging and billing system of the ISP.

If the resolved SIP address lies outside the local domain, resource reservation is needed. Before sending the INVITE message to the called party the terminal tries to acquire the necessary service from the network by sending a reservation request to the gateway. The gateway responds with a “200 OK” message to indicate the that the resources are reserved. In turn the terminal sends the invitation message to the called party. This terminal will now try to reserve a voice service to the inviting terminal. If this does not fail and the user picks up the phone the terminal signals successful completion of the call. During the call both terminals signal INFO messages to the gateway to indicate that the call is still ongoing. In the response they receive the advice of charge messages. If these information messages to not appear for some time the gateway assumes the terminal dead, stops the reservation and charging for it. A call terminating normally will cancel the reservation on the reception of a BYE tear down message. Figure 6 shows the signaling diagram including resource reservation, keep alive and charging messages.

An “anomaly” exists in this description. It assumes that each party is being charged for the traffic *from* their respective terminal. These “split-charges” are the price to be paid for simplicity: “Caller only” charges, as in PSTN, require direct contracts between both party’s ISPs. Not so the “split-charge” model. Only local customers (which can be assumed to have a contract) have to be billed. In the “split-charges” model service level agreements between the involved diffserv domains can be traded independent of what is being charged to the user. Even different time scales and different economic models can be applied[2].

B. The Terminal Architecture

The IPTalk terminal is light weight IP voice terminal with an Ethernet type network access. There were three motivations for the design of a stand-alone voice terminal:

We believe that specialized devices are more cost effective than a single highly flexible device for multiple purposes, such as a traditional PC. We also believe that such specialized devices are simpler to operate and maintain.

Investigations of our own and other [7] show that a major source of delay in to-days IP telephony are the end-systems themselves. This is due to inadequate operating systems concepts and, to a lesser extend, hardware architectures.

For “IP appliance” to become a reality, their cost must become extremely low. We were curious to see which components add to the cost and which integrated solutions are offered by the market.

The terminal has been built out of an off-the-shelf, highly integrated communications controller with integrated Ethernet MAC and a PC-style multichannel audiocodec (Figure 2). Furthermore, there is an optional board implementing active and passive PSTN interfaces. This allows for connecting the terminal to a PSTN network and use regular analog phones as headsets. The terminal can be controlled by DTMFs generated by a analog phone or through the integrated Web-server.

The IPTalk terminal can concurrently serve as a terminal and as gateway to PSTN. The gateway functionality simplifies integration of a VoIP solution with the existing PSTN infrastructure. The signaling software on the terminal implements in a distributed fashion functionality traditionally offered by PBXs or by intelligent network services (IN).

The software structure is such, that it is independent of the hardware, choice of operating system (OS) and the communication middle-ware. There are three layers of abstraction: The hardware abstraction layer (HAL) abstracting from the underlying hardware (CPU, Codec, etc.). The OS Topsy[1], a light weight message passing kernel, implements a generic send/receive message API, which can be ported to other OS offering inter-process-communication (IPC), such as UNIX.

[1] http://www.tik.ee.ethz.ch/~topsy

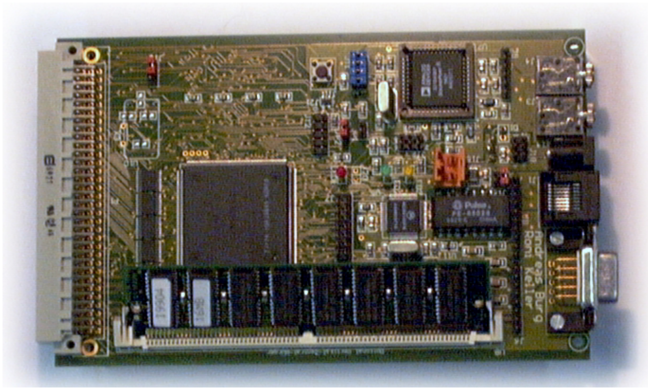


Figure 2: The IPTalk terminal. The board is built around a integrated communication controller. In the upper left corner the 16 Bit, 48 kHz, Stereo Codec is placed. In the middle left the Ethernet Phy can be identified. Ram is provided through a single SIM. 2MB flash Rom are placed on the downside of the board. There are various extension connectors for a display, an analog linecard and an ISDN interface.

The third abstraction is commonly called the socket layer and abstracts communication end-points. However, we are not using sockets, but have chosen to build the software on more generic message passing. Despite the many abstraction layers the software is small in code size (<128KB) and data size (< 128KB, depending on the configuration, e.g. the amount of buffers used and the maximal MTU supported). Figure 3 depicts the software layout.

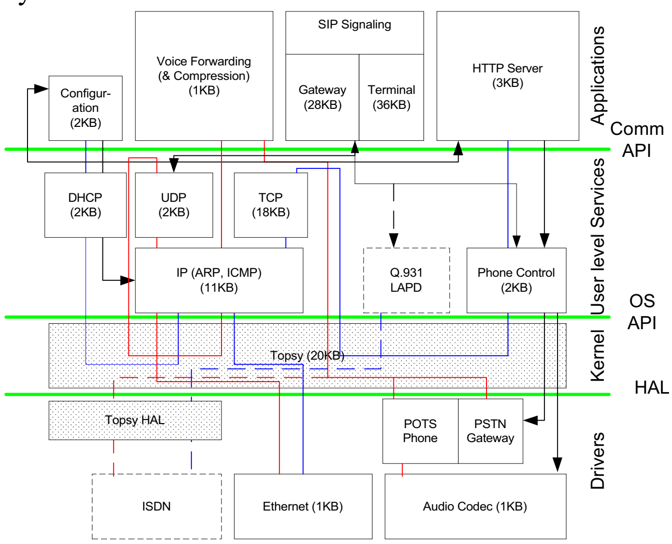


Figure 3: Terminal Software Structure. There are three abstraction layers. First, the hardware abstraction layer (HAL), abstracts from the hardware. Second, the operating system API (OS API) abstracts from the underlying OS. Third, the communication API abstracts from the networking software. In the figure the size of the code for each functional block is indicated in parenthesis.

To make IP terminals economically successful, their design has to be simple and highly integrated to make up for the added software size and complexity of terminal based services. This limits size and functionality of the software. An active up-loadable services could be a solution for this problem.

DHCP allows for automatic terminal configuration. Integrated neighbor and gateway discovery and least cost routing enables cost effective plug and play operation. The terminal will be in-field remotely serviceable and upgradable.

The embedded web-server is used to display the state of the terminal, user-to-user signaling messages and configuration information. Furthermore, it can be used to simplify the operation of advanced services such as call forwarding, call waiting and multiparty conferencing. Today many users are not capable of using these features, because of the limited user interface capabilities (keyboard, display) of the terminals. Through the embedded web server these services can be controlled through a click on the mouse.

IV. Signaling DSS1 services in SIP

A. *Digital Subscriber Signaling System No. 1 (DSS1)*

The DSS1 is the ISDN user-network layer 3 interface. It provides the means to establish, maintain and terminate network connections across an ISDN. In addition, it provides generic procedures, which can be used to provide additional supplementary services. The supplementary services of DSS1 are defined in Q.950 to Q.957. Figure 5 shows how the DSS1 services are embedded in the call control. The *Supplementary Services* of DSS1 can be classified into the following categories:

· Number Identification
· Call Offering
· Call Completion
· Multiparty
· Community of Interest
· Charging
· Additional Information Transfer

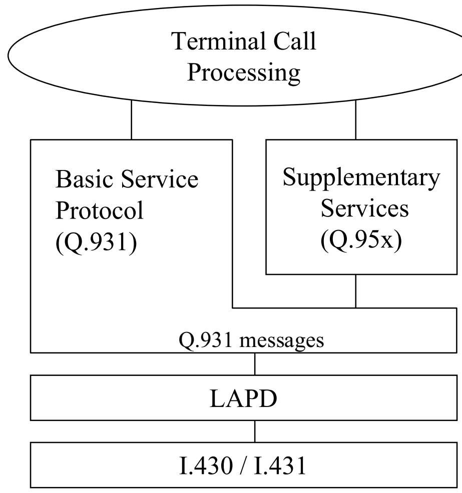


Figure 4: Modelling of Basic and Supplementary Services in DSS1

B. *Implementation of DSS1 Services in SIP*

To implement the DSS1 services in SIP we modified the SIP in several ways. First, we added two new SIP methods: INFO and INQUIRE. The INFO method is used to carry mid-signaling information as proposed by Steve Donovan [4]. The INFO method does not change the state of any SIP entity. The INQUIRE method is used to inquire cost information from a gateway. The answer to an INQUIRE message is a 250 Cost_Info answer. The idea behind the INQUIRE method is that there are several gateways reachable. Each gateway interfaces to a different voice network provider. The INQUIRE method returns the cost informations for a call to the specified party. Thus, a least cost routing functionality can be easily implemented in a SIP terminal.

Second, we added new header fields to two existing SIP methods. One header field has been added to the INVITE method that is used for conference signaling, the other field has been added to the 200 OK message, such that it can carry charging information.

Further, our IPTalk SIP terminal takes temporary the role of a SIP redirection server and issues SIP redirection messages.

1) Number Identification Supplementary Services

Some of the supplementary services of DSS1 are available in SIP by default. Calling Line Identification (CLIP) and Connected Line Identification Presentation (COLP) are available in SIP through the “from”-field in SIP messages. Calling Line Identification Restriction (CLIR) and Connected Line Identification Restriction (COLR) are difficult to implement in a SIP terminal. Obviously, it is not difficult to write the “from”-field in SIP messages with an anonymous tag, but the IP address of both parties is always known. An identification restriction service could be easily provided using an anonymous proxy server, but this is out of scope.

Malicious Call Identification (MCID) is a service that records details of each call to the user of that service. To implement this service in SIP, every SIP entity has to have the ability to record SIP messages.

2) Call Offering Supplementary Services

To implement the services Call Forwarding Busy (CFB), Call Forwarding No Reply (CFNR) and Call Forwarding Unconditional (CFU), our IPTalk SIP terminal takes temporary the role of a SIP redirect server and answers with a 302-redirect message. In CFB, the redirect message is sent only when the user is busy. In CFNR, the redirect message is sent after a configured time has elapsed without that the user has answered the call. In CFU, the redirect message is sent immediately.

Call Deflection (CD) is forwarding the call on a call-by-call base. The user has the possibility to decide whether the call shall be forwarded while the phone is ringing. Again, our IPTalk SIP terminal will take the role of a redirect server and send a 302-redirect message.

Explicit Call Transfer (ECT) is forwarding a call from an established connection. User A has an established connection with user B. Now, user A wants to transfer the call to a user C. In our implementation, A sends a BYE message with a "Contact"-header that contains the URI or phone number of C. With that, B can send an INVITE message to C and the connection between B and C can be established.

3) Call Completion Supplementary Services

While a user has an established connection, he has four possibilities to answer an incoming call: He can ignore the call, he can explicitly reject the call, he can terminate the first (established) connection and answer the call, or he can put the first connection on "HOLD" and answer the second call. Call Waiting (CW) provides the means in the end-system to alert the user of a second incoming call. In IPTalk this is solved through a beeping tone.

Call Hold (CH) can be implemented as described in the SIP Internet Draft [10]. In IPTalk CH is solved in hardware. If user puts a connection to hold the terminal sends a hold music instead of forwarding voice from the microphone and incoming voice packets are dropped.

When the callee is busy and does not want to answer an incoming call immediately, the Completion of Calls to Busy Subscribers (CCBS) service will establishes a connection between the parties as soon as the callee is idle. The first approach to implement that service in SIP is analog to its implementation in mobile communication. The terminal would do a retry to establish the connection in regular intervals. In IPTalk, we followed another approach. The caller will automatically receive an INFO message with a "NotifyIdle" header field from the callee as soon as he is idle.

The Terminal Portability (TP) service allows for a user to unplug/plug a phone from the network, while an established connection will remain. In SIP, this comes for free.

4) Multiparty Supplementary Services

On IPTalk, Three Party (3PTY) service is implemented as a special case of Conference Calling (CONF). 3PTY allows only three parties in a conference, while CONF allows larger conferences, limited only by resources. In DSS1 a centralized scheme using a bridge device that mixes the voice streams is used.

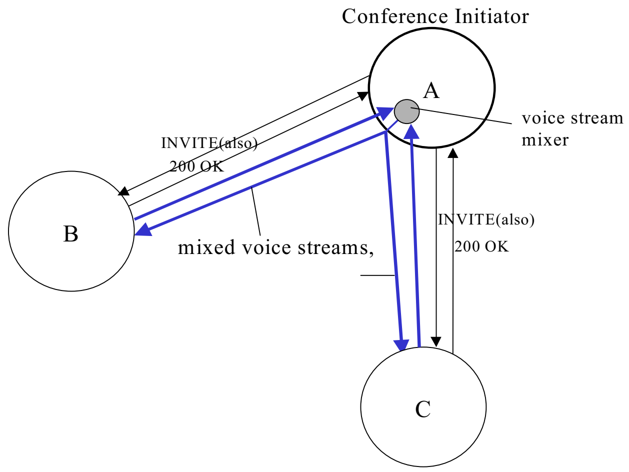


Figure 5: Three party conference

The IPTalk platform is capable of mixing voice streams. The number of voice streams that can be mixed is limited by the CPU resource. Figure 5 shows a three party conference. A user that wants to setup a conference has to send an INVITE message to all parties of the conference with an "also" header field that contains a list of all participating parties. Each invited party can response with either 200 OK to accept the invitation or 603 DECLINE to reject the invitation. Conference parties that already had an established connection will also receive the INVITE message and thus be able to see who is going to be at the conference. The idea is, that the conference parties can be displayed on a SIP terminal. When the conference initiator receives a BYE message of one of the conference parties, or if an invited party declines to be part of the conference, he sends an INVITE message with the "also" header field to all the conference parties, again. Thus, the conference parties always know who is joining and leaving the conference.

The mixed voice streams could be transmitted to the conference parties.

5) Community of Interest Supplementary Services

The Closed User Group (CUG) service is implemented through filter lists, which allow or disallow calls. Calls that originate from a user that is not in the allow list, are ignored.

Another possibility to implement the CUG service is to have special SIP proxy server for a CUG, which restricts calls to inside a CUG.

6) Charging Supplementary Services

The implementation of the Advice of Charge (AOC) service is explained in section III.A. Figure 6 shows how on request charging information is supplied to the user.

7) Additional Information Transfer Supplementary Services

User-to-User Signaling (UUS) is implemented in IPTalk using the INFO message.

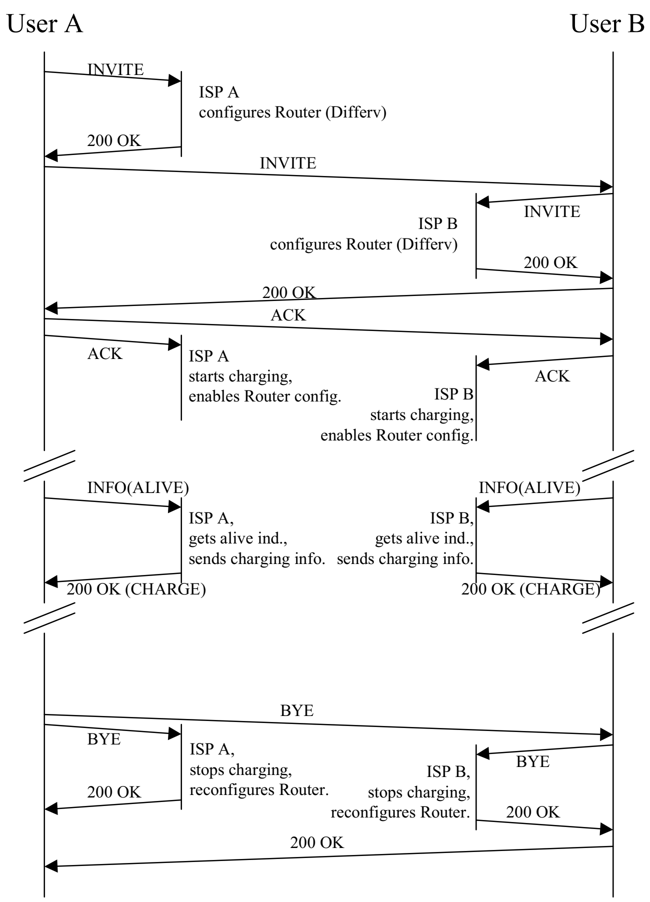


Figure 6: IPTalk SIP call setup and tear-down including resource reservations and advice of charge messages

V. Conclusions

We have shown that it is possible to implement DSS1 services in SIP. Most of the services have been implemented without extending SIP. One SIP message type was added to implement user to user messages, much alike the D-channel user-to-user signaling of ISDN. The same informational message type has also been used to inform a gateway of an ongoing reservation and to get the AOC messages.

The IPTalk SIP terminal reduces the end-system delay to less than 1ms plus packetization delay. This allows for high quality voice traffic on local area networks.

We have proposed a network architecture, which allows for end-to-end telephony service with high service quality. In our architecture QoS can be guaranteed through admission control at ingress routers. We have addressed charging by including a simple reservation mechanism for these guaranteed services. We see a main potential of IP base voice terminals in local area and (virtual-) private networks. Charging for services might not be necessary in such environments, however, reservation and admission control to guarantee acceptable service levels on bandwidth limited links are required. An interesting case of such local area networks are residential networks using new access technologies such as cable-modems and xDSL, which today do not offer voice service directly.

Future work will focus on integrating all the components on the terminal. DSS1 signaling and web-configuration have not been fully tested on the terminal.

A new field of research we see in active service deployment for such thin-clients. This would allow for a wealth of new services to be offered despite the limited resources to be expected from such terminals.

Many aspects of the signaling and service provisioning within the multiprovider network are still open. Especially, the traditional charging model (caller paying) needs a higher level of cooperation between service provider.

VI. Implementation Status

DSS1 signaling has been implemented an tested on a UNIX operating system. It has been compiled an linked for the IPTalk terminal but not tested. The software on the terminal does not yet support all the features described in this paper, such as hold music and voice mixing.

Topsy has been ported to many platforms such as MIPS, i386 M68K and UNIX user space. TCP has been tested on the UNIX version of Topsy only.

VII. Acknowledgements

We would like to thank the following people for their contribution to IPTalk: Andi Burg and Roni Keller for hardware design of the terminal. Gregor Battilana, Heidi Rogenmoser and Patrick Kinsch for their efforts in implementing the signaling protocol. We would like to thank our colleagues, especially George Fankhauser, for their critical input.

VIII. References

[1] M. Arango et al., “Media Gateway Control Protocol (MGCP)”, Internet draft, IETF, Oct. 1998, work in progress.

[2] B. Briscoe, “The Direction of Value Flow in Multi-Service Connectionless Networks.” BT Labs Technical Report. September 1999.

[3] D. Clark. and W. Fang, “Explicit Allocation of Best-Effort Packet Delivery Service.” IEEE/ ACM Transactions on Networking Vol. 6, No. 4, pp. 362– 373. August 1998.

[4] Steve Donovan, “The SIP INFO Method”, Internet Draft, October 1999.

[5] G. Fankhauser, B.Plattner, “Bandwidth Brokers as Mini-Markets” MIT workshop on Internet Service Quality Economics, December 1999

[6] I. Faynberg et al., The Intelligent Network Standards, McGraw-Hill, NewYork, 1997

[7] B. Goodman, “Internet Telephony and Modem Delay“, IEEE Network, May 1999

[8] Pawan Goyal et al., “Integration of Call Signaling and Resource Management for IP Telephony”, IEEE Internet Computing, IEEE May/June 1999.

[9] H.323, “Packet-Based Multimedia Communication Systems”, ITU-T recommendation, Feb. 1998

[10] Handley, Schulzrinne, Schooler, Rosenberg “SIP: Session Initiation Protocol” IETF Internet Draft, August 1999.

[11] Christian Huitema et al., “An Architecture for Residential Internet Telephony Service”, IEEE Internet Computing, IEEE May/June 1999.

[12] J. Lennox and H. Schulzrinne, “CPL: A Language for User Control of Internet Telephony Services”, Inter-net draft, IETF Mar. 1999, work in progress.

[13] J. Lennox, H.Schulzrinne, T. F. La Porta, “Implementing Intelligent Network Services with the Session Initiation Protocol”, Technical Report CUCS-002-99, January 1999.

[14] R. Neilson, J. Weehler, F.Reichmeyer, S Hares “A Discussion of Bandwidth Broker Requirements for Internet2 Qbone Deployment” Internet2 Qbone Bandwidth Broker Advisory Council, http://www.merit.edu/working.groups/i2-qbone-bb, work in progress

[15] Daniel Rizzeto and Claudio Catania, “A Voice over IP Service Architecture for Integrated Communications”, IEEE Internet Computing, IEEE May/June 1999.

[16] Jonathan Rosenberg, Jonathan Lennox, and Henning Schulzrinne, “Programming Internet Telephony Services”, IEEE Internet Computing, IEEE May/ June 1999.

[17] H. Schulzrinne and J. Rosenberg, “A Comparison of SIP and H.323 for Internet Telephony”, Proceedings of. 1998 Workshop Network and Operating System Support for Digital Audio and Video (NOSSDAV'98), July 1998, Cambridge, England

[18] H. Schulzrinne, J. Rosenberg, J. Lennox, “Interaction of Call Setup and Resource Reservation Protocols in Internet Telephony”, Technical Report, June 1999.

An Architecture for an SCN/IP Telephony Routing Testbed

Raimo Kantola, Jose M. Costa Requena, and Nicklas Beijar
Laboratory of Telecommunications Technology
Helsinki University of Technology, P.O. Box 3000, FIN-02015 HUT, Finland

Abstract-- **Our project shows an approach for achieving interoperability between the Circuit Switched Networks and Voice over IP networks in the areas of numbering and Telephony routing. This architecture proposal provides an efficient solution for the location of the most suitable gateway for calls across the technology boundary while supporting number portability and number translations for 800- and cellular numbers. In the project we have already implemented the SCSP protocol for data synchronisation between Location Servers. We are now building a comprehensive telephony routing testbed using MySQL as the routing database and Python for data manipulation. Routing applications under development include the Telephony Routing over IP –protocol and its counterpart for the SCN of our own design. We call the SCN counterpart Circuit Telephony Routing Information Protocol (CTRIP).**

Index terms— **SCN, IP, TRIP, CTRIP, SCSP, Python, MySQL.**

A. Introduction

The aim of this paper is to describe the project under way to create a testbed for SCN-IP numbering and routing interoperability. The main objective of this project is testing the functionality and scalability of the proposed solutions. These solutions become more and more necessary as the size of the IP Telephony Networks grows and as they become peer networks to the SCN.

It seems logical that before we can achieve the popular goal of all IP networking, we must make the IP Telephony Network a full peer to the SCN. This means that the two technologies are connected using a public Network to Network Interface such as ISUP over IP and that the IP Telephony Network has more than one connection to the SCN.

On the IP side the problem of gateway location for calls across the IP/SCN technology boundary was recognised a couple of years ago. Two Internet drafts have been issued by the IPTEL group to establish clear requirements for the problem [5] and to provide a protocol based solution [6]. The proposed protocol, called TRIP, is used for both interoperator and intra-operator routing data distribution and synchronisation between Location Servers (LS). The most recent version of TRIP is modelled on the BGP-4 protocol. For data synchronisation between multiple Location Servers in an administrative domain it uses the SCSP.

The Signalling Servers can make queries to LSs on a call-by call basis to locate the most suitable gateway for a hybrid IP originated call. The current version of TRIP assigns the responsibility for number translations e.g. for 800-numbers to Redirect Servers which are a specific type of signalling servers present for example in the SIP architecture [7]. Alternatively, such redirection operations could be integrated into TRIP and the Location servers.

The current TRIP specification leaves open the question of origination of the reachability data. This data essentially describes the SCN numbering space and maps it to gateways suitable and willing to complete calls to the respective SCN number prefixes. We are proposing to design a counterpart to TRIP which would distribute numbering information in the SCN e.g. between the IN Service Data Points and across the technology boundary through Numbering and Routing Information Gateways (NRIGWs). This way the information distributed by TRIP need not be manually entered into multiple LSs but is originated at its real source. This way the management of the numbering space can be largely policy based.

The motivation for such an elaborate architecture is first the need to avoid unnecessary conversions on the media path between IP and SCN. The conversion causes delay and jitter and easily degrades voice quality to an unacceptable level. Second, automation of service management in a multioperator environment based on IP and circuit switched technologies is an economic necessity. Number portability and roaming may move numbers inside an operator's network based on a single or based on the two technologies or even move numbers from one operator's network to another provided an arrangement exists such that billing can be reliably performed. Third, IP Telephone numbers can migrate in the IP telephony network topology and a different gateway may need to be selected by the SCN for an SCN originated call to complete a quality call to that number. The problem of gateway location is equally important for SCN originated calls as it is for IP originated calls.

The rest of this paper is organised as follows. Section B discusses the purpose of the project. Section C describes the project itself including its detailed objectives, structure and technical content.

B. Purpose of the Project

In this project we aim to design a comprehensive telephony routing information testbed for the IP/SCN hybrid using modern software tools such that the system can easily incorporate new features. We have chosen MySQL as the routing information database in order to bring the power of relational database tools to our environment. At the moment, for flooding of the routing information between SCN and IP we use the SCSP protocol, which was implemented in C++ in our project [4, 8]. The algorithms and data management specified by the TRIP protocol and its circuit counterpart has been developed using Python (see Figure 1).
Information about the IP-telephones themselves may end up being stored in the DNS. Possible integration with the DNS is one potential future development, which we may have to undertake later on.

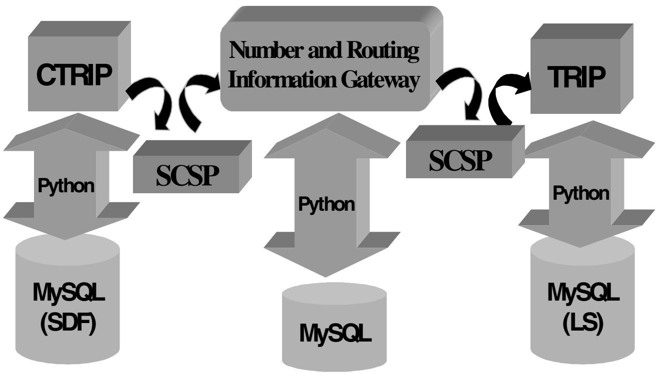


Figure 1: The Architecture of the NRI Testbed

C. The Project

The first phase of the project was the implementation of the SCSP protocol for loose data synchronisation in a server group of an arbitrary topology [4,8]. The SCSP is based on the proven link state interior packet routing protocol for the Internet called the Open Shortest Path First, OSPF –protocol. The SCSP has three subprotocols. First the servers need to establish neighbour relationships, second they need to accomplish initial data synchronisation by exchanging complete copies of their databases. In normal operation the SCSP servers will flood the changes in their data to their neighbours in the server group as the changes appear in the originating server. The SCSP has limited means for implementing elaborate data distribution policies, therefore TRIP uses both the SCSP and the BGP-4 models.

1.. Objective

Figure 2 shows the components of our target architecture. It depicts the Routing Information Interoperability layer and the call control and signalling layer which can use the services of the former.

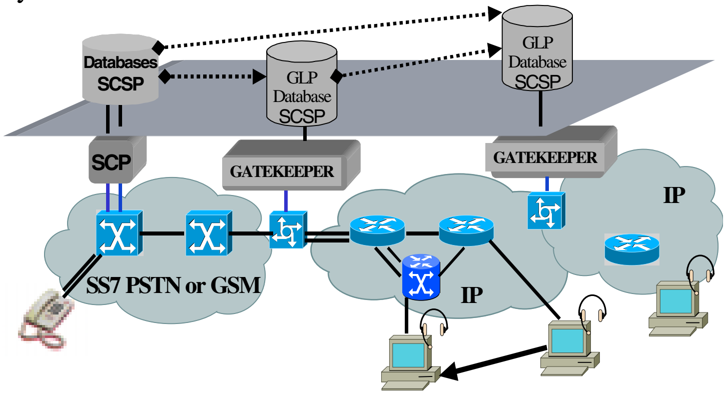


Figure 2 NRI layer and Call Control

2.. Project structure

Initially, we developed a stand-alone implementation of the SCSP protocol. Next, in the IP side we implement the TRIP to store and retrieve the routing information in the location servers. Finally, in the SCN side we develop the CTRIP protocol with similar behaviour. The CTRIP performs the same function as TRIP but with the information required by the SCN architecture i.e. the information about the gateways to choose to complete calls to IP phones. The last major component in this project is the design and implementation of the module that will make the information translation between IP and SCN. All these modules are depicted in Figure 1.

Both application protocols (TRIP, CTRIP) and the gateway logic are implemented in Python scripting in order to be able to easily add and modify routing attributes and modify the policy of passing around the information.

3.. Project modules

In this section we describe each module and how they have been implemented.

1. Information Flooding module (SCSP).

The SCSP algorithm establishes the data synchronisation among a set of server entities. After that it maintains an actively mirroring state of any change in every database information. This module implemented in C++ in our project facilitates an easy translation to new communication architectures. One of the implementation goals was minimising the platform-dependence by using C++. In Figure 3 we have depicted the modules in the TRIP implementation and how they use the SCSP for data replication.

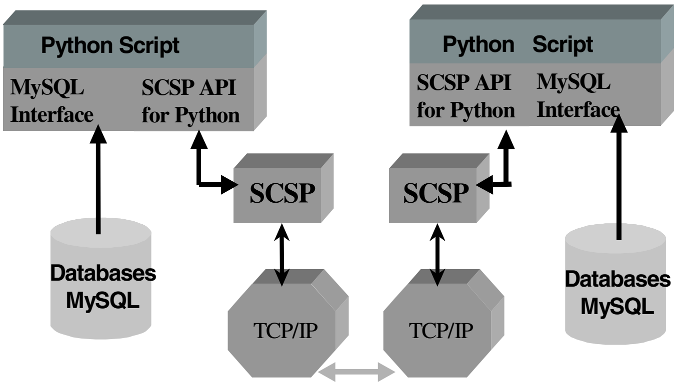


Figure 3:The architecture of TRIP.

For the purpose of our testbed implementation we have added a new API to the SCSP especially for interfacing the protocol to Python.

2. Data allocation and management module.
The data is stored in the MySQL database and it is managed by the TRIP. The MySQL package [9] has been chosen to store the data, which the protocol has to synchronise. This software is based on typical SQL primitives. The basic functions necessary to move the data to and from the database can be embedded either in a C program or a Python program. In our case we will use the Python interface to access the information in the MySQL database. Furthermore, this module provides the basic functions to exchange, remove and insert data to/from the MySQL. These functions are embedded in the program written in Python. The Python functions will read the data from the MySQL database and will insert all the information inside the SCSP data structures, which are based on the protocol specifications. This module also has to receive all the newest data from remote servers and based on the TRIP policies it will drop or accept that data replacing the old one.

3. TRIP algorithms module (Python).
The scripts are defined following the structure depicted in Figure 4.

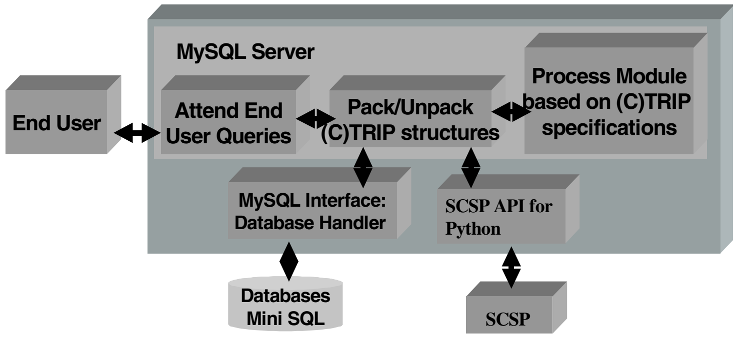


Figure 4: The Python Scripting structure

Figure 4 also shows the interaction between the different modules of our implementation. Next we will give a brief description of each of the Python modules.

- Process module: This is the part of the Python program that performs the TRIP policies to select the information according to the specifications. To give a feeling of Python programming let us look at an example that describes a small piece of data processing. First, it selects the Call Routing Object (CRO) that is going to be processed and then it starts analysing the data and takes the decision to append, update or erase the old entry with the new one.

```
# Then we try to find from SignalingProtocols all
# available BaseprotocolOBJECTS.
text = "select SigObjType1,SigObjVal1,SigObjType2,
... pe15,SigObjVal15 from SignalingProtocols where ("
   for x in range(len(st)):
     text = text + "a=%s" %st[x][1]
     text = text[0:(len(text)-1)]
     text = text + " or "
   text = text[0:(len(text)-3)]
   text = text + ")"
   print text
   stnew = DB[text];
   print stnew
   k=0
   taul=[]
   for x in range(len(stnew)):
     for y in range(15):
       if (stnew[x][y*2] == '0'):
         taul.append(ord(stnew[x][y*2+1])-48)
         k=k+1
   print taul
 . . . . . . . . . . . .
```

- MySQL Interface: In this module we define the routing tables in the MySQL database and the functions to extract and reinsert the new data into it.

- SCSP API for Python: This module takes care of inserting and extracting the data to/from SCSP and Python structures. This script makes the SCSP structures (written in C++) visible from Python.

- Attend End User Queries: This part will receive the End-User queries and selects the most suitable signalling servers.

- Pack/Unpack TRIP structures: This module receives the data from the SCSP or End-User and changes its format to the registers stored in the MySQL.

4. The CTRIP protocol

The purpose of the Circuit Telephony Routing Information Protocol is to distribute reachability information about the E.164 numbers assigned to IP phones. It also informs about the location of the gateways to be used to complete calls to those IP phones and also gives the location of different servers that may be used to gain more detailed routing information about certain special E.164 prefixes such as 800-numbers. In addition, CTRIP can carry information about the location of genuine CSN routing targets e.g. for the purpose of Number Portability. This last feature is complementary to the IN functionality used today for the purpose of Number Portability in the CSN. CTRIP is meant to be deployed between any CSN nodes that store the kind of routing information carried by CTRIP. It would be up to the operators to choose an appropriate mode of deployment. Possible CTRIP nodes include Service Data Points, Service Control Points that include a Routing Information Base and also ISDN exchanges. The Numbering and Routing Information Gateways are like any other routing information nodes in the SCN. In a real network environment CTRIP could run over an Intranet or alternatively over the Signalling Connection Control Part of CSS7. In our testbed it always runs over IP.
CTRIP divides the routing domain into smaller subdomains consisting of geographical areas of a single operator's network. Aggregation is possible between subdomains. Routing is performed in a hierarchical manner: first the destination network is located, then the destination area is located and finally the routing number is used to locate the terminal.

The CTRIP is implemented in a similar fashion as the TRIP protocol. SCSP is used for replication of routing entries within subdomains and special border nodes transfer information between subdomains.

D. CONCLUSIONS

A novel approach for numbering interoperability between SCN and IP has been proposed. The project for implementing a comprehensive routing information testbed is under way for the purpose of evaluating new telephony routing features and for the purpose of studying the scalability of proposed solutions. We propose to extend the model of distributing routing information among Location Servers adopted on the IP Telephony side by the IETF also to the CSN side. We believe this is the way to achieve seamless interoperability between the two networking technologies. We believe that this is a necessary step for the purpose of automating service management while we are moving towards all IP Networking where all communications services run over the ubiquitous IP network.

E. REFERENCES

[1] J. Luciani, G. Armitage, J. Hlapern N. Doraswamy. "*RFC 2334*. Server Cache Synchronization Protocol" . Network Working Group. Standards Track, April 1998.

[2] A. Watters, G. van Rossum, J. Ahlstrom. *"Internet Programming with Python"*. M&T Books 96.

[3] J. Costa Requena**,** "*An Implementation of the Server Cache Synchronization Protocol***,** M.Sc thesis, Laboratory of Telecommunications Technology, Helsinki University of Technology, 1999.

[4] J. Rosenberg, H. Schulzrinne, A Framework for a Gateway Location Protocol, IETF Internet Draft, draft-ietf-iptel-gwloc-framework-03.txt, work in progress, June 1999.

[5] J. Rosenberg, H.Salama, M.Squire Telephony Routing over IP (TRIP), IETF Internet Draft, draft-ietf-iptel-trip-01.txt, Work in Progress, January 2000.

[6] M. Handlay, H. Schulzrinne, E. Schooler and J. Rosenberg, SIP: Session Initiation Protocol," IETF Internet Draft, draft-ietf-mmusic-sip-12.txt, Work in Progress, January 1999.

[7] M. Handlay, H. Schulzrinne, E. Schooler and J. Rosenberg, SIP: Session Initiation Protocol," IETF Internet Draft, draft-ietf-mmusic-sip-12.txt, Work in Progress, January 1999.

[8] Jose Costa Requena, Raimo Kantola, "Server Cache Synchronization Protocol (SCSP) : component for directory enabled networks", SPIE Photonics, Boston September 1999

[9] http://www.hughes.com.au/products/msql/

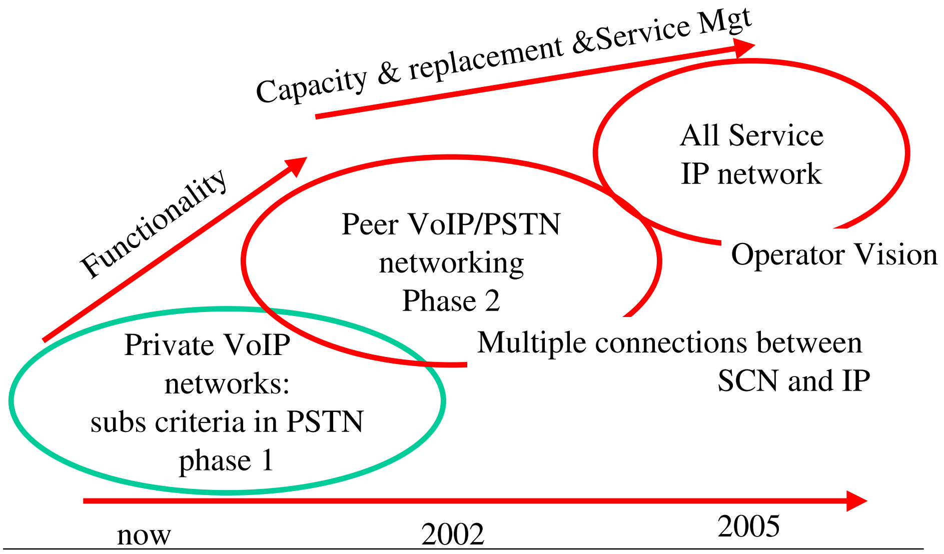
An Architecture for an SCN/IP Telephony Routing Testbed

Prof. Raimo Kantola, Jose M. Costa Requena, Nicklas Beijar
Helsinki University of Technology
Laboratory of Telecommunications Technology
raimo.kantola@hut.fi, jose@tct.hut.fi, nbeijar@tct.hut.fi
http://www.tct.hut.fi/tutkimus/ipana

Outline

- Roadmap
- Motivation
- Routing information problem
  - Requirements
  - Locating GWs from the IP Telephony network
  - Locating a SG from the ISDN network angle.
  - Number portability across the technology boundary.
- The Solution to Telephony Routing over SCN/IP - hybrid network.

Roadmap to the Future



Interoperability Issues

- Signaling and Call control
- Quality of Service

Phase 1
--->

- Telephony Routing and addressing
  - Input Information gathering
  - Alternative routing over IP

Phase 2
-->

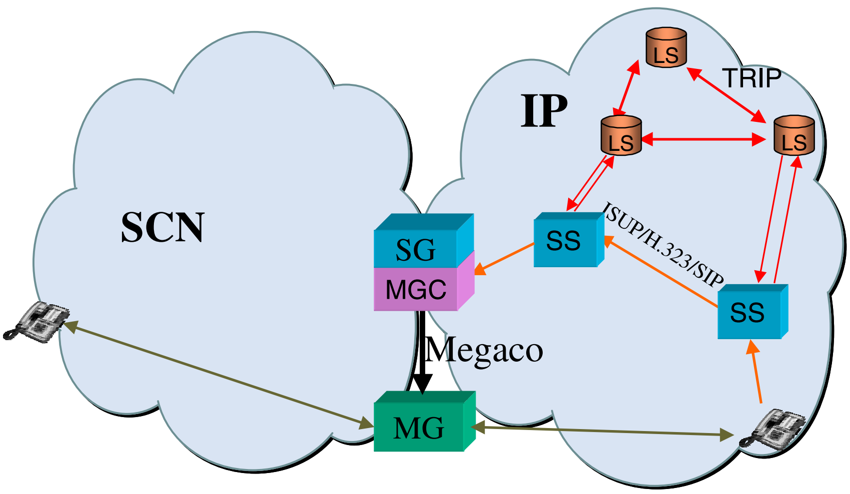
- Service Management in the hybrid network

Phase 3

Phase 2 Requirements

- Efficient routing and numbering infrastructure across the emerging hybrid network is a necessity
  - Delay and jitter highly depend on call path
- In all call scenarios at all cost we must avoid unnecessary conversion between IP and PSTN.
  - Call Forwarding, Number Portability, Roaming, 800-numbers ...

Current Architecture



TRIP = Telephony Routing over IP, SG - Signalling Gateway, MGC - Media Gateway Controller
MG - Media Gateway, SS = Signaling Server, LS = Location Server

For Telephony routing we must choose optimal Gateway

- The IP Telephony view:
  - LS provides info about Next hop Signaling server e.g. a Signaling Server or an MGC in the same domain
  - TRIP keeps information in LSs updated across IP Telephony systems
  - MGCs are registered e.g in LS (this information may be local to an Admin Domain)
  - SS can use LS to locate MGC and MG

How does the SCN choose a GW?

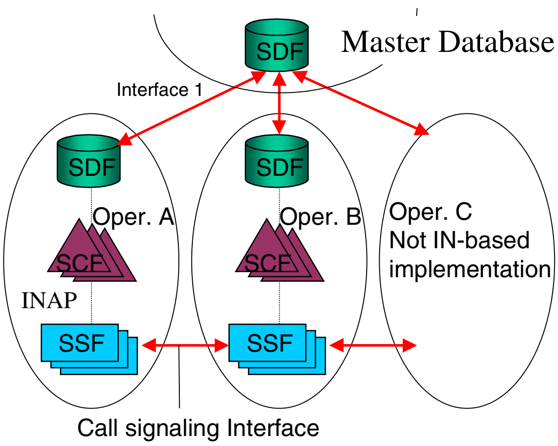
- ISDN, GSM, PSTN view
  - Good news: SGs are large - easy to locate
  - Bad news: I do not hear Any Body working on the problem of Gateway location from the ISDN point of view
    - From the SCN it is equally important to select the most suitable Gateway for SCN to IP calls
    - Current solutions are based on static routing tables which are managed manually

Numbering Issues

- What if an IP Telephony Number is ported to another ITSP operator?
  - ISDN side may need to choose another SG for calls to that number
- What if an ISDN number is ported to another ISDN operator?
  - IP side may need to choose another set of SG, MGC, MG
  - LSs need to know about the change
- What if a number is ported SCN to IP or vice versa

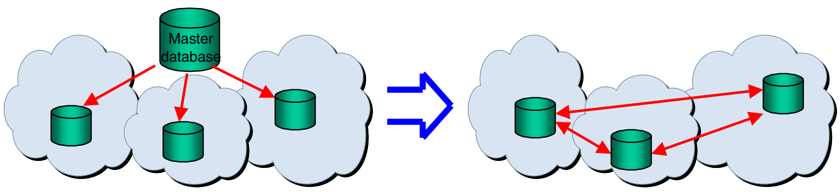
Current situation at the ISDN side

Number Portability is mandated by regulators in Europe and the US
Typical solution is based on IN



We need a Distributed Architecture

- ✓ No single point of failure.
- ✓ Scalability.
- ✓ Database updates made directly by the operators. Support for subscriber-initiated updates possible.
- ✓ Flexibility in routing attributes opens the door to easy introduction of new services. Services can roam.



Requirements for 800- and GSM numbers

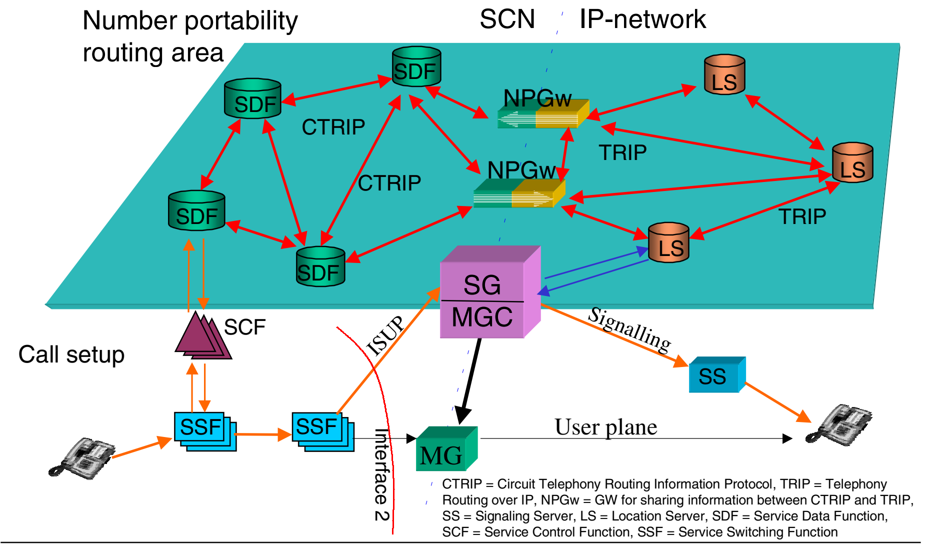
- IP Telephony view
  - an 800-number and a Cellular Mobile Number may be located anywhere in the ISDN/PSTN cloud or the Cellular cloud respectively
  - additional round of indirection for choosing the GW is needed to ensure adequate quality voice
  - LS needs to cascade the request to an SDF or to an HLR or return the address of an SDF or HLR so SS can make a subsequent query

6

Requirements for 800- and GSM numbers

- SCN view
  - an 800-number (and a Cellular Mobile Number - only a matter of time!) may be located anywhere in the IP cloud
  - additional round of indirection for choosing the GW is needed to ensure adequate quality voice
  - SDF needs to cascade the request to an LS
  - It is not efficient to flood Mobile numbers among LSs when a mobile number is in an IP cloud - a solution scalable to frequent location changes is needed

The solution is CTRIP + Numbering gateway



TRIP vs CTRIP

TRIP Information

- Withdrawn Routes
- Reachable Routes
- Next Hop Server
- Advertisement Path
- Routed Path
- Atomic Aggregate
- Local Preference
- Multi Exit Disc
- ITAD Topology
- Authentication

CTRIP Information

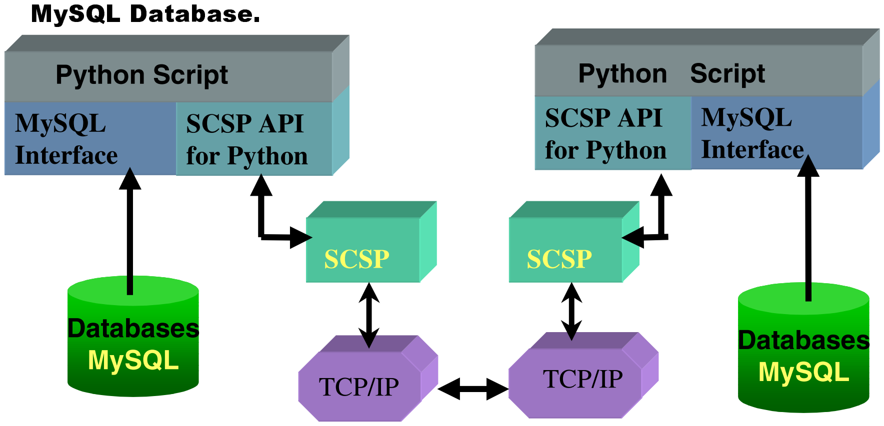
- Directory number (key)
- Destination subdomain id
- Signalling capability id
- Routing number
- Area
- Advertisement path
- State

POLICY controls distribution, mapping and aggregation

TEKNILLINEN KORKEAKOULU
HELSINKI UNIVERSITY OF TECHNOLOGY

TRIP and CTRIP Modules

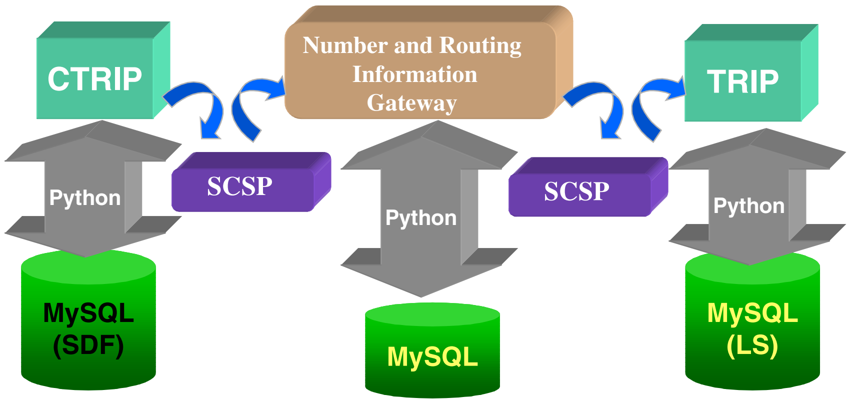
We use Python as Interface to work with SCSP primitives through Python objects and access to the information (CRO) stored in the MySQL Database.



TEKNILLINEN KORKEAKOULU
HELSINKI UNIVERSITY OF TECHNOLOGY

Overall view of the Testbed

The common tool for interoperability between TRIP and CTRIP over SCSP is Python. It is the Interface to work with SCSP and access to the information stored in the MySQL Database.



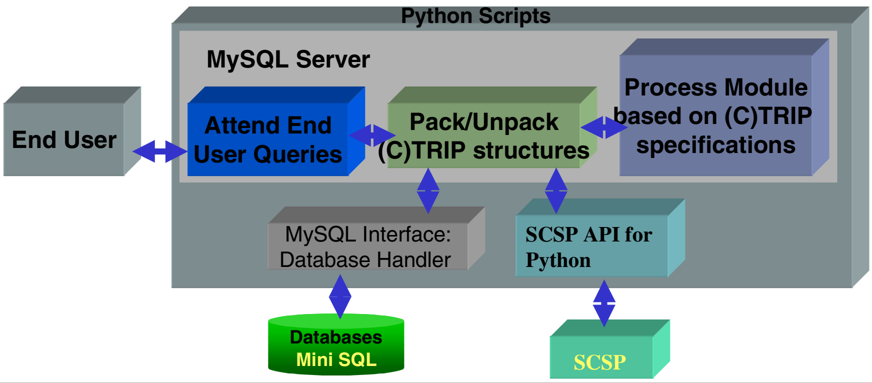
SCN/IP Telephony Routing Testbed / The 1st IP Telephony Workshop/ Berlin 4/2000 Slide 17

TEKNILLINEN KORKEAKOULU
HELSINKI UNIVERSITY OF TECHNOLOGY

Modules Interaction

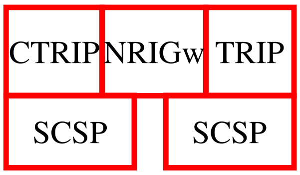
The Python scripts are split into modules:

- **MySQL Interface,provides the handler to the MySQL database.**
- **SCSP API,receives and stores data from the SCSP in (C)TRIP structures.**
- **MySQL server,pack-unpack the MySQL CRO and attend the User Queries.**
- **Process module,implements the TRIP/CTRIP specifications.**



SCN/IP Telephony Routing Testbed / The 1st IP Telephony Workshop/ Berlin 4/2000 Slide 18

Conclusions



- Gateway model needs to be complemen-ted by Numbering&Routing Information gateways
- SCSP can be the common Numbering infrastructure component for both SCN and IP Telephony networks

- TRIP, CTRIP and possibly ENUM are used to distribute Routing information among Location Servers and SDFs
- We are building a Telephony Routing Information Testbed in which new routing attributes can be easily added and distributed.

OSIP: An Open Source SIP Architecture

Stefan Foeckel, Matthias Kranz, Jiri Kuthan, Dorgham Sisalem
GMD-Fokus
Kaiserin-Augusta-Allee 31
10589 Berlin, Germany
{foeckel,kranz,kuthan,sisalem}@fokus.gmd.de

Abstract—**In this paper, we present an overview of the architecture of OSIP. OSIP is an open source implementation of the session initiation protocol (SIP) currently discussed for IP Telephony and group communication. To allow for easy portability and wider usage this implementation was developed in JAVA. The message parsing part of OSIP was designed to be automatically generated based on the protocol specification. Finally, in order to support various communication scenarios OSIP was designed in a modular manner with open interfaces that allow the development of applications of various complexity using the same core protocol implementation.**

I. Introduction

The rapid growth, increasing bandwidth and the availability of low-cost multimedia end systems has made it possible to use the Internet for multimedia applications ranging from telephony to conferencing, distance learning, media-on-demand and broadcast applications [1]. Due to the prospects of tremendous financial gains IP-based telephony is ever more attracting the attention of service providers and consumers as well. However, due to the entirely different nature of IP-based networks compared to traditional circuit-switched networks new paradigms for signaling a communication request between two end systems are needed.

Due to its simplicity, the session initiation protocol [2] designed within the IETF is ever more gaining in acceptance as the standard protocol for IP Telephony.

To provide for a portable and expandable open source implementation of SIP and hence support and simplify the efforts of other institutions and companies in their investigations and testing of SIP we started some time ago designing and implementing the protocol stack of SIP in JAVA. In addition to providing the protocol functionalities we considered in our design the aspects of portability and possible extension of the language itself as well as the applications using the protocol stack.

In Sec. II of this paper we first present a short overview of the session initiation protocol and related work. Sec. III describes the general architecture of OSIP, the current implementation status as well as the interfaces provided for realizing different communication scenarios and applications. We conclude the paper in Sec. IV with a look at our further plans and required enhancements to OSIP.

II. Background and Related Work

While the session initiation protocol was favored for IP Telephony by the IETF, H.323 [3] is being propagated by the ITU standardization group and is being used in various conferencing tools such as *netmeeting*[1] and some of the currently popular IP Telephony applications such as *iphone*[2]. The ITU H.323 series of recommendations includes H.245 for control, H.225.0 for connection establishment, H.332 for large conferences, H.450 for supplementary services, H.325 for security and H.246 interoperability with circuit switched services. Already the number of defined protocols indicates the complexity and size of an H.323 compliant implementation. Besides the size, Schulzrinne and Rosenberg describe in [4] various other disadvantages of H.323 compared to SIP such as extensibility, security, scalability and supported services.

The most important SIP operation is that of inviting new participants to a call. A user first obtains an address where the user is to be called and translates this address into an IP address where a server may be found. Once the server's address is found the client can send an invitation message to the server. However, as the server which receives the message is not likely to be the host where the user to be invited is actually located we need to distinguish between different server types that a complete SIP implementation should fulfill [5]:

Proxy: A proxy server receives a request and then forwards it towards the current location of the callee -either directly to the callee or to another server, that might be better informed about the actual location of the callee.

Redirect: A redirect server receives a request and informs the caller of the next hop server. The caller then contacts the next hop server directly.

User agent server: This server resides on the host where the callee is actually located. It is capable of querying the

[1] Available from: http://www.microsoft.com/windows/netmeeting
[2] Applications available under: http://www.vocaltec.com/products/iphone5

user about what to do with the incoming call, i.e,. accept, reject or forward, and sends the response back to the caller.
Register: To assist the end systems in locating their requested communication partner, SIP supports a further server type called register server. The register server is mainly thought to be a data base containing locations as well as user preferences as indicated by the user agents.

There are currently already different publicly available SIP implementations[3].

However, most of the available implementations use hard-wired parsers, are not publicly freely available or are programmed in C or C++. To our knowledge there are no publicly available Java implementations that are shipped with source code, use a parser generator, or are completely free.

III. General Architecture of OSIP

Based on our previous experience with earlier versions of SIP [6] we identified the following requirements:
Portability: To ease its usage over different platforms JAVA was chosen for implementing OSIP. This is especially helpful for users who execute their signaling services at heterogeneous end-devices.
Extensibility: Due to the continuing changes and developments in the area of IP Telephony, the specifications of SIP is facing continuous changes and enhancements. To reduce the implementation overhead required for taking such developments into account, the message parsing part in OSIP was generated using a compiler-compiler approach based on the abstract specifications of the protocol.
Flexibility: As described in Sec. II a SIP entity can be used to fulfill different tasks such as acting as an end user agent or as a proxy server. While the actual functionalities of those entities differ, the actions of parsing the protocol messages and handling their transport is generic to all of them. Hence, to allow the development of SIP agents with different functionalities, the parsing and networking parts of the SIP stack were separated by a clear interface from the message handling and application parts. This allows for more flexible upgrading of either the generic parts or the message handling instances.

A major goal of the OSIP implementation was to simplify the task of building different kinds of SIP servers such as user agents, proxies and redirect servers. This is to be realized by using a BNF generated parser and combining reusable building blocks.

[3]Listings of SIP implementations being available or under construction may be found at: http://www.cs.columbia.edu/~hgs/sip/implementations.html http://www.fokus.gmd.de/research/cc/glone/projects/ipt

As Fig. 1 depicts, the OSIP stack consists of three layers. The first layer constitutes the "core" SIP stack. It contains a number of service-primitives common to all SIP servers including network handling, parsing of SIP messages, management of SIP transactions, retransmission of UDP packets and locating SIP servers. The second layer consists of SIP server modules implementing different specific functions based on the server personality using SIP requests and responses. Finally, the third layer represents the application which is responsible for processing the SIP message body and handling or initiating the actual telephony communication.

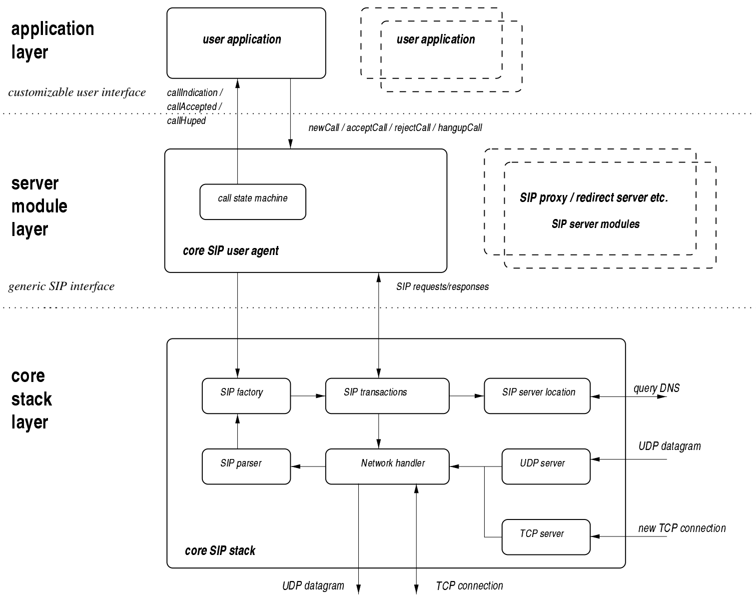


Fig. 1. OSIP stack overview

A. Extensible SIP Parser

The aim of the parser is to process the abstract incoming data stream, perform a lexical analysis and a semantic analysis according to the grammar specified in [2]. The output is an intermediate data structure, which from then on will be the only object for manipulation through a well defined interface.

The goal was to develop a rule based one-pass parser, which would be easy to maintain particularly with regard to new header elements or more changed grammar. Therefore we decided to use a tool called *mparse*[4], a Java-based compiler-compiler that generates JAVA code out of BNF specifications. Its input language is an extension of Java, offering capabilities such as inheritance, method overloading, encapsulation and more. The parser production spec-

[4]*mprase* is a successor of Sun Microsystems' JavaCC and is available under www.metamata.com

ifications are written within methods and the definition of tokens looks similar to static initializers providing a very clear representation.

After receiving an incoming request or response the parser processes the provided stream and generates an instance of a class called SipMessage. The Sip Message object provides methods for accessing all header fields, the message body or parts of them if necessary. The parser also detects all unknown header fields and puts them in a list, which can be evaluated by the application. While the message body is recognized, it is not processed as this should be done by the application layer. Currently, all header fields defined in [2] are recognized. If any malicious headers or header field contents are detected, the parser will tolerate them as far as possible, but set a failure flag for each so that subsequent processing entities can evaluate them appropriately.

B. The Core Stack Layer

The core stack layer consists of the following main modules:

UDP server: In case UDP was used for transporting SIP messages, the UDP server receives incoming SIP messages on a pre-defined port and forwards them to the network handler module

TCP server: In case TCP was used for transporting SIP messages, the TCP server is responsible for accepting new TCP connections on a pre-defined port, creating a handler object for each new connection and passing them to the network handling module.

Network handler: The handler dispatches incoming UDP packets and manages TCP connections. It creates abstract streams and passes their content to the SIP parser. Additionally, it sends outgoing SIP messages, performs retransmission when using UDP, alerts the assigned SIP transaction objects on timeouts and handles network errors.

SIP factory: The SIP factory provides static methods to create and initialize SIP stack objects and to check their validity. Additionally, in the case of local generated requests it creates new SIP transaction objects.

SIP transactions: This module validates message headers and assigns SIP request/response objects to a SIP transaction. Further, it administers SIP transaction objects and calls the registered server module for further processing of SIP messages. Finally, it passes outgoing messages to the network handler according to the requested transport method. In the case of UDP communication it additionally configures timers for retransmitting lost packets.

SIP server location: The SIP server location module is responsible for locating SIP servers on outgoing requests. Currently, this is only realized by simple DNS address record queries. However, we are in the process of adding more advanced methods such as DNS SVR record search.

C. Server Module Layer

This layer performs the actual handling of SIP requests and responses, maintains the call state and informs the application about the arrival and processing status of SIP requests.

In contrast to the generic nature of the core stack layer, the server module layer is customized to handle server specific functionalities like proxying or redirection.

The communication with the core SIP layer is achieved by passing SIP requests and responses objects. The server module has the ability to access all parts of a SIP message through these objects. Depending on its main function and internal state the server module decides on the actions to take. For example it can alert a user application, create response-objects to answer a SIP request or create new requests by using the factory methods provided by the core stack.

At this stage only a user agent module is available. This agent is invoked by the core stack in order to process SIP request/response objects. It maintains a state machine, that creates provisional and final responses and calls the registered user application to indicate invitations or call-changes. It provides methods to the user application to create client requests.

D. Application Layer

The application layer is finally responsible for processing the body of the SIP messages, initializing the appropriate media agents for handling the actual multimedia communication and possible providing an interface to the user. Currently, a compliant SDP parser class [7] is provided for processing the media description in the SIP body message.

To initiate the communication between a user application and the SIP server modules, the application has to register with the SIP server modules using a common interface. Additionally, in order for the server modules to inform the application about the arrival and processing status of SIP requests, the application layer needs to provide for an interface that can be used by the server modules.

IV. Summary and Future Work

OSIP is an extensible implementation of the session initiation protocol based on a modular design, rule-based parser and implemented in the JAVA programming language. The current version already supports user agent functionalities and the recognition of all fields supported in the standard [2].

We are currently in the process of enhancing the server module layer with proxy and register functionalities and are considering the aspects of security and authentication required for IP Telephony. Further, the interfaces provided between the layers are being refined to allow for easy binding with call services. Additional features like DNS server record queries will be added.

To allow for high quality communication, we are also studying the aspects and approaches for integrating QoS provisioning concepts with IP Telephony signaling.

To test the efficiency of a JAVA-based implementation compared to C or C++ implementations various tests need to be conducted and evaluated.

REFERENCES

[1] Henning Schulzrinne, "Re-engineering the telephone system," in *Proc. of IEEE Singapore International Conference on Networks (SICON)*, Singapore, Apr. 1997.

[2] M. Handley, H. Schulzrinne, E. Schooler, and J. Rosenberg, "SIP: session initiation protocol," Request for Comments (Proposed Standard) 2543, Internet Engineering Task Force, Mar. 1999.

[3] International Telecommunication Union, "Packet based multimedia communication systems," Recommendation H.323, Telecommunication Standardization Sector of ITU, Geneva, Switzerland, Feb. 1998.

[4] Jonathan Rosenberg and Henning Schulzrinne, "A comparison of SIP and H.323 for internet telephony," in *Proc. International Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV)*, Cambridge, England, July 1998.

[5] Henning Schulzrinne and Jonathan Rosenberg, "Signaling for internet telephony," Technical Report CUCS-005-98, Columbia University, New York, New York, Feb. 1998.

[6] Dorgham Sisalem and Henning Schulzrinne, "The multimedia internet terminal," *Journal of Telecommunication Systems*, vol. 9, no. 3, pp. 423–444, Sept. 1998.

[7] M. Handley and V. Jacobson, "SDP: session description protocol," Request for Comments (Proposed Standard) 2327, Internet Engineering Task Force, Apr. 1998.

OSIP – An Open Source SIP Architecture

Stefan Foeckel, Matthias Kranz,
Jiri Kuthan, Dorgham Sisalem
GMD-Fokus

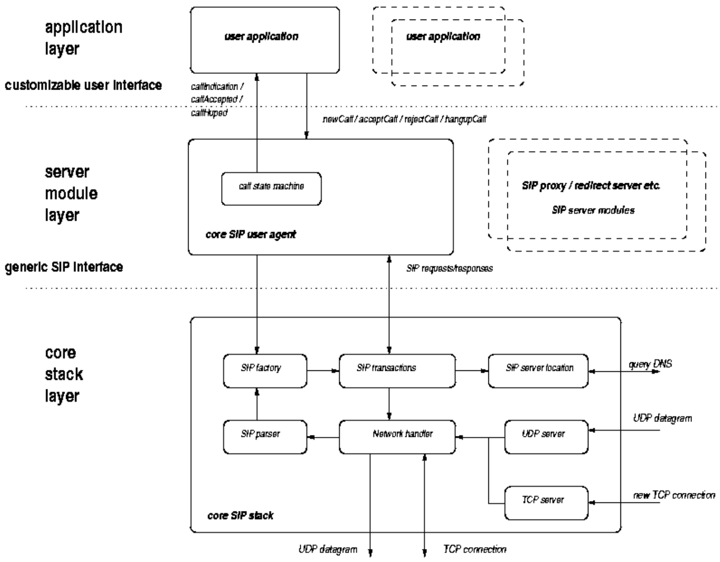
OSIP

- Open Source SIP: layered, Java-written, extensible, portable, SIP stack
- Multiple APIs
  - Low-level API for detailed control of stack‘s signaling logic
  - High-level, easy-to-use API to accommodate common applications

OSIP Benefits at a Glance

- Portability
  - OSIP written completely in Java
- Extensibility
  - Parser compiled from an abstract protocol definition
- Flexibility
  - 3-layer architecture allows for various deployments of OSIP stack
- Availability: low-cost

OSIP Architecture



OSIP Layers -1

- Core Stack Layer
  - Makes transport transparent to application
  - Maintains SIP transactions
  - Performs Server Location
- Extensible SIP Parser
  - Verifies and translates SIP messages to parsed binary structures
  - Is based on an abstract syntax definitions

OSIP Layers -2

- Server Module
  - Defines Signaling Logic (UA, proxy, ...)
  - Maintains Call State
- Application Layer
  - Passes Signaling Control to Users
  - Processes bodies of SIP messages

Next Steps

- Implementing Security Features
- Implementing SIP Add-Ons (INFO, reliable provisional messages, call control services)
- Integration with QoS-enabled applications

Demonstrations

Siemens AG

SIEMENS

iptel2000 SIP Phone Prototype Presentation

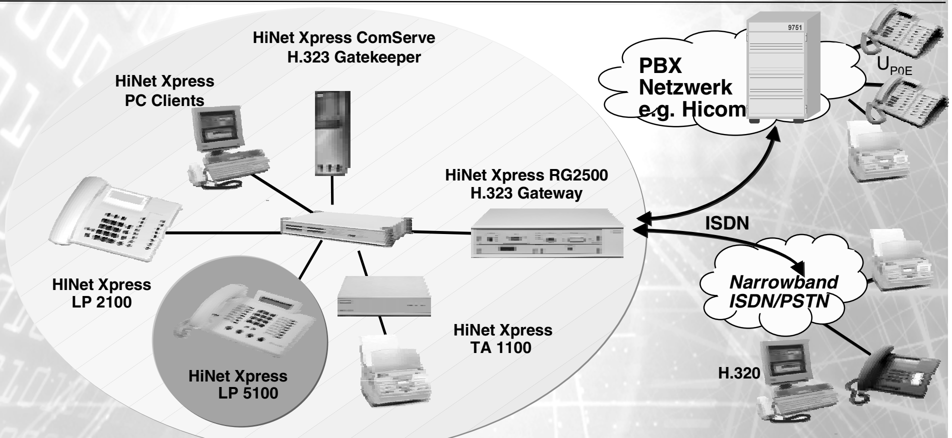
John Moran, Mediatrix
Juergen Brieskorn, Siemens

MediaTrix

Information and Communications

SIEMENS

HiNet Xpress IP Phones in the H.323 environment



MediaTrix

Information and Communications

SIEMENS

HiNet Xpress LP 5100: Overview

General availability: 11/99

- Comfort telephone
- Standard based
  - H.323, *H.450 in 12/00*
  - G.711, G.723
  - SNMP, HTTP, DHCP, FTP
- Handsfree speaker phone
- Local control of tones
- OptiGuide context sensitive user Interface
- Supplementary Services:
  Call forwarding, consultation, transfer
  Call waiting,…
- Remote administration

MediaTrix

Information and Communications

SIEMENS

HiNet Xpress LP 5100 Implementation - SW Concept

User Interface/Local Feature Presentation

DL Handler | User Data Handler | Phone Feature Control

API

TFTP | SNMP | CTI Daemon | G.711 G.723 / RTP RTCP | HTTP Server | H.225.0 RAS | H.450.x / H.450.1 / H.225.0 CC | H.245

UDP | TCP

IP (ICMP) (ARP) (DHCP)

MAC/LLC

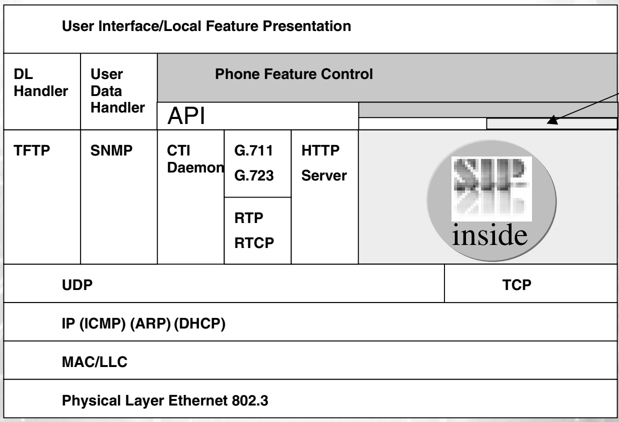
Physical Layer Ethernet 802.3

MediaTrix

Information and Communications

SIEMENS

HiNet Xpress LP 5100-SIP Implementation - SW Concept



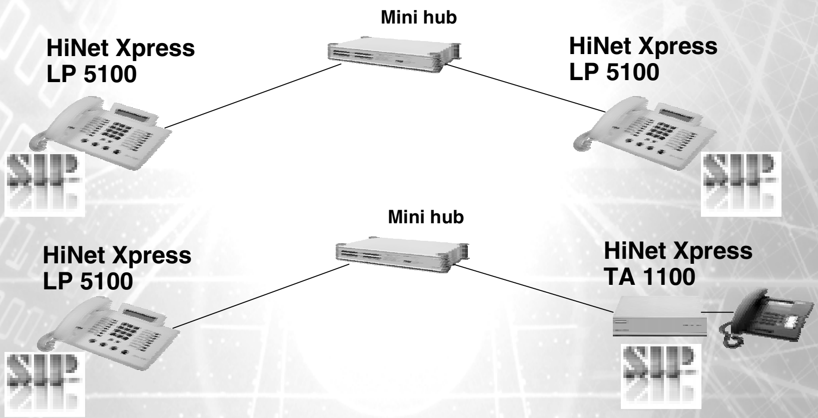
Mediatrix SIP Stack: Basic call is implemented at the API level for the demo

MediaTrix

Information and Communications

SIEMENS

HiNet Xpress LP 5100-SIP Demo scenarios - direct call model

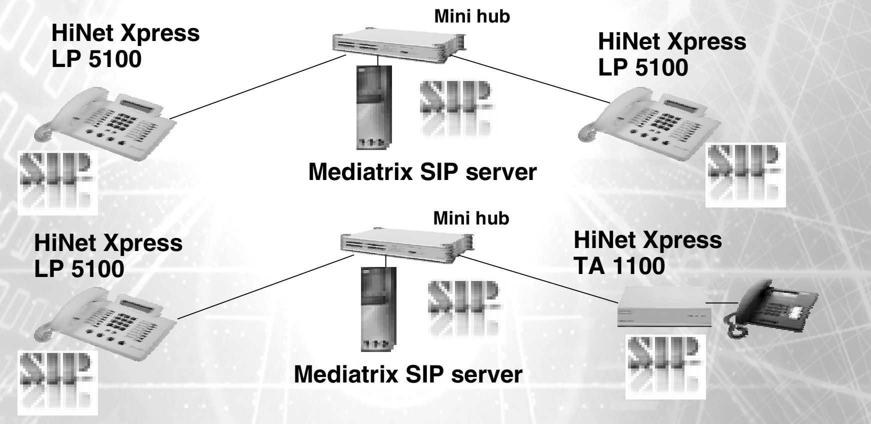


MediaTrix

Information and Communications

SIEMENS

HiNet Xpress LP5100-SIP
Demo scenarios: SIP server based call model



SIEMENS

HiNet Xpress LP 5100-SIP:
Findings and planned next steps

- SIP stack easily fits into H.323 API environment of LP5100
  - started integration task in 2/00
  - demo version ready in 3/00
- SIP call performance vs. H.323 improved ~ 50%
- Participate in 4th SIP bake off 4/00
- Beta Version w/ supplementary service support available in mid 2000
- Participate in IMTC SuperOp 7/00
- GA in 4Q00