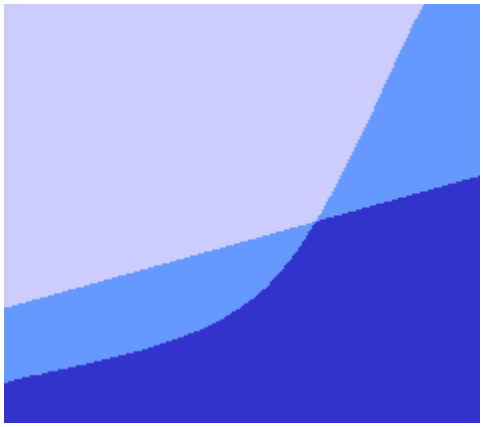

WIPTEL

Piloting an Infrastructure for IP Telephony
in the German Research Network

Deliverable D6:

Infrastructure and deployment report

Deliverable D6: Infrastructure and deployment report

11.12.2000

Universität Bremen
Technologie-Zentrum Informatik (TZI)
Bereich Digitale Medien und Netze (DMN)
Bibliothekstr. 1
D-28359 Bremen

TZI/DMN	**Dipl.-Inf. Stefan Prelle**
	Dipl.-Inf. Niels Pollem
	Dr.-Ing. Jörg Ott
	Dr.-Ing. Carsten Bormann
RRZN	**Dipl.-Ing. Eduard Siemens**
GMD Fokus	**Dipl.-Ing. Jiri Kuthan**
	Dipl.-Ing. Dorgham Sisalem
	Stefan Foeckel

Kontakt	**Dr.-Ing. Jörg Ott**
	jo@tzi.uni-bremen.de
	+49 421 201-7028
	+49 421 218-7000 (fax)

0 Management-Zusammenfassung

Das Projekt WIPTEL hat eine Architektur für den Aufbau von IP-Telefonie an wissenschaftlichen Institutionen entwickelt und aufgezeigt, wie sich diese IP-Telefonie-Inseln innerhalb des Wissenschaftsnetzes zu einer integrierten IP-Telefonie-Infrastruktur zusammenschließen lassen. Hierzu wurden am Markt verfügbare Produkte evaluiert, teilweise beschafft und getestet, um eigene Systemkomponenten ergänzt und in eine gemeinsame IP-Telefonie-Umgebung integriert. Als Ergebnis ist eine Referenzinstallation im Bereich Digitale Medien und Netze des TZI der Universität Bremen entstanden, die heute im Regelbetrieb alle Mitarbeiter untereinander und mit dem öffentlichen Telefonnetz verbindet und dabei unterschiedliche Technologien integriert. Spezielle Infrastrukturkomponenten und ein Namens- und Nummernplan gestatten das einfache Einbinden beliebiger weiter Standorte in diese offene IP-Telefonie-Umgebung.

Bei den Entwicklungen im WIPTEL-Projekt wurden zwei wesentliche Schwerpunkte gesetzt: zum einen die Orientierung an internationalen Standards und zum anderen die Konzentration auf die Bereitstellung einer *Infrastruktur* (dies im Gegensatz zu einem möglichen Schwerpunkt auf Endgeräten).

Mit der Zielsetzung einer offenen Kommunikationsplattform wurde der Einsatz proprietärer Systeme (wie sie im Bereich traditioneller TK-Anlagen dominieren und auch für IP-Telefonie von verschiedenen Herstellern angeboten werden) frühzeitig ausgeschlossen. Derart geschlossene Lösungen sind gerade im Wissenschaftsnetz mit seinen vielen unabhängigen Entscheidungsträgern in den angeschlossenen Institutionen nicht anwendbar. Aus dem gleichen Grund wurden auch die verfügbaren standardisierten Alternativen — H.323 und SIP — gleichermaßen berücksichtigt und ein Protokoll-agnostisches Infrastrukturkonzept entwickelt.

Die Konzentration auf Infrastrukturkonzepte und -komponenten von Beginn an und vor allem mit der Entscheidung, eigene Entwicklungen hier einzubringen, anzupassen und (weiter) zu entwickeln haben entscheidend zum Erfolg von WIPTEL bei der Systemintegration sowie bei der Anpassung an das akademische Umfeld beigetragen. Der Einsatz eigener Verwaltungssysteme (H.323-Gatekeeper, SIP-Server) und Gateways ist die Basis für die Integration andernfalls inkompatibler Endgeräte, für Protokoll-unabhängige, den institutionellen Verhältnissen angepaßte Benutzerverwaltung und für standortübergreifende Namens- und Adreßauflösung.

Die pragmatische Herangehensweise zum Erzielen kurzfristiger Ergebnisse kombiniert mit einer zukunftssicheren Architektur haben so zu einer funktionsfähigen Umgebung geführt, in die noch in der Entwicklung befindliche Technologie schrittweise zum Einsatz gebracht werden kann: in Form von Produkten (sobald verfügbar) wie auch in Form von eigenentwickelten Prototypen.

Trotz aller inhaltlichen Erfolge ist kritisch festzustellen, daß der Projektfortschritt in WIPTEL eine spürbare Verzögerung erfahren hat: der standortübergreifende Pilotbetrieb begann erst kurz vor offiziellem Projektabschluß und bezieht auch nur zwei (statt der geplanten drei) externe Partner ein. Als primäre Ursache hierfür sind die großen Schwierigkeiten bei der Beschaffung standardkonformer Komponenten zu nennen: Nichtverfügbarkeit von Produkten, Lieferengpässe bei den Herstellern und so gut wie keine Teststellungen haben zu deutlichen Verzögerungen und nicht unerheblichen Kosten geführt. So konnte die Referenzinstallation erst spät im Projekt mit einer Vielzahl verschiedener Produkte ausgestattet werden, eine wichtige Voraussetzung für die Anbindung der externen Standorte mit ihren heterogenen Infrastrukturen. Als Resultat sind die standortübergreifenden Tests im Wissenschaftsnetz noch nicht abgeschlossen, so daß hierzu ein separater Erfahrungsbericht erst im Jahr 2001 vorgelegt werden kann.

Zu IP-Telefonie allgemein läßt sich als Resümee festhalten: IP-Telefonie ist im Prinzip verfügbar, aber ihre Entwicklung ist weitem noch nicht abgeschlossen: Heutige IP-Telefonie-Produkte sind noch weit von einer ubiquitär einsetzbaren *Plug&Play*-Technologie entfernt. Der potentielle Nutzer muß sein Anwendungsgebiet und die zur Verfügung stehende Technologie genau kennen, will er IP-Telefonie in einem zukunftsweisenden Sinne einsetzen und nicht nur seine (bestehende) TK-Anlage auf der Basis von Paketvermittlung ersetzen. Darüber hinaus sind Standardkomponenten (Telefone,

Gatekeeper/Proxies, Gateways) meist nur in homogenen Umgebungen nutzbar, da die Hersteller zum einen keine wohldefinierten Schnittstellen zur systemunabhängigen Kopplung verschiedener Standorte anbieten und zum anderen die Systeme selbst nicht immer vollkommen standardkonform implementiert und deshalb nur begrenzt interoperabel sind. Schließlich weisen alle betrachteten IP-Telefonie-Systeme einen wesentlichen Mangel auf: Sie verfügen nicht über die Sicherheitstechnik, die notwendig wäre, um eine sichere IP-Telefonie-Umgebung aufzubauen. Statt etablierter kryptographischer Verfahren verlassen sich die Hersteller in den ersten Versionen ihrer Produkte mehr auf *Security by Obscurity*. Auf dieser Basis lassen sich (zumindest im universitären Umfeld) keine Telefoniedienste mit verursachungsgerechter Kostenzuordnung betreiben.

Der Einsatz eigenentwickelter Infrastrukturkomponenten mit (herstellerspezifischen) Erweiterungen konnte die Inkompatibilitäten von Systemen verschiedener Hersteller weitgehend ausgleichen. Auf diese Weise ließ sich der Kern für eine standortübergreifende Infrastruktur im Wissenschaftsnetz aufbauen, die auch nach dem Abschluß von WIPTEL zunächst als Serviceleistung von TZI DMN für Institutionen des WiN weiterbetrieben wird.

1 Introduction

The WIPTEL project headed by TZI/DMN was aimed at putting the high-performance IP connectivity provided by the German research network (WiN) to use as a test bed for introducing IP telephony and, in a subsequent step, to provide ubiquitous IP telephony connectivity (and the adjunct services) in the German research environment.

This came at a time when many research institutions were (and still are) facing investment decisions with respect to replacing or expanding their existing telephony infrastructure, which currently mainly consists of large PBXs with proprietary phones and interfaces. It was important for someone to take the lead in exploring possible solutions for the specific requirements of academia. Obviously, the premier provider of IP connectivity to the scientific community in Germany, the DFN, was in a rather good position for this role.

Therefore, the WIPTEL project was set up at the TZI to investigate the particular requirements of academic institutions and devise a prototype solution for introducing IP telephony as a regular service within and across such institutions that are part of the German Research Network (WiN).

This deliverable, the *"Infrastructure and Deployment Report"*, concludes the second phase of the WIPTEL project, which mainly included designing, implementing and refining the WiN and site infrastructure proposed in the preceding deliverables.

While the remainder of this chapter is dedicated to an overall wrap-up of the project, the subsequent chapters are structured as follows: Chapter 2 provides an update on standards and protocols. Chapter 3 illustrates the (possible) integration of the current POTS numbering space with those proposed for IP-telephony. Chapter 4 depicts the proposed WiN infrastructure, whereas Chapter 5 depicts the proposed site infrastucture. Chapter 6 lists the IP-telephony software developed by the TZI relevant to WIPTEL. Chapter 7 shows the current state of the testbeds (Berlin, Hannover, and Bremen) within WIPTEL. Finally, Chapter 8 concludes this deliverable. Appendix A and B contain more information from resp. on our testbed partners: RRZN at the University of Hannover and GMD Fokus in Berlin.

1.1 Fields of Activity

The many different activities within the WIPTEL project can be grouped into four main fields. These are briefly summarized in this section. All of them originate (or were already proposed) in the original project proposal that set the direction for this project.

1.1.1 Evaluation of Vendors and Products

As stated, the WIPTEL project was aimed at developing a reference configuration for an IP telephony infrastructure that is heterogeneous in two respects:

- both H.323 and SIP were (and still are) considered for use within the local infrastructure and were to be integrated to form a coherent environment, and
- hardware as well as software solutions for endpoints, gateways, and management components from different vendors were to be integrated to avoid dependencies on a particular supplier from the very beginning.

The architecture developed in WIPTEL was to support both dimensions of heterogeneity. Therefore, an evaluation of the available products (and thereby, also their vendors and their VoIP strategies) was an important part of the project. While only deliverable 1 was fully devoted to this topic, it has been updated in all other deliverables and carried on throughout the project, eventually leading to the TZI IP-telephony test lab depicted in chapter 7 of this deliverable.

Evaluating the vendor strategies included a series of vendor presentations at the TZI in the first months of the project.

While testing showed that the heterogeneity wished for can be implemented, it should be mentioned again (as in deliverable 4/5), that obtaining components fitting our needs has been a tedious process

that has only recently become slightly easier. For example, regarding the somewhat limited protocol heterogeneity in our test lab, we have only now, finally, been able to buy additional SIP phones.

Direct support for the WIPTEL project from the vendors has not increased in general, though. We still could not obtain loaner equipment. Then, there have been quite a number of issues with their protocol conformance. (At least, the bug reports generated by us brought some of the vendors' attention to us.)

Finally, the implementation of security standards and protocols is generally not well-progressed. This currently limits the production use of IP-telephony in all cost-incurring areas, e.g. (an available-to-all access to) gateways to the POTS.

1.1.2 Design of WiN and Site Infrastructure

While IP-telephony solutions that only service a single site or even zone (e.g. a working group) have been available (at least in theory) for quite some time, expanding this to a wide-area approach has been the technically most challenging part of the project. The WiN and site architecture proposed by us has been introduced in deliverable 3 and refined in deliverable 4/5 (*"Site and WiN Infrastructure Specification"*) and again in sections 3 through 5 of this deliverable. Their most salient characteristics can be summarized as follows:

- The architecture is protocol agnostic.While industry has converged on a single standard (RTP) for the transport of voice information, there have been and will be two families of standards in use for control (H.323 and SIP) and probably an even larger set for administration. As the DFN member institutions are autonomous in their choice of local infrastructure, it seems unlikely that a single standard can be forced onto them by DFN.
- The architecture affords full autonomy to the institutions. DFN provides basic rules for bilateral and/or multilateral interoperability, with the objective to minimize the amount of rules required. Until appropriate administration protocols (such as TRIP for call routing, see section 4.1.3) are available, a central DFN database is maintained and automatically distributed to participating member institutions. During the course of the WIPTEL project, central components for control and administration are set up in the TZI; later, these components can be distributed in the WiN.

  Note: This has been put into practice within the WIPTEL testbed. Also, the TRIP implementation by the TZI is continuously maturing.
- Introducing billed services is exceedingly difficult, both for technical reasons (e.g., the difficulty of setting up secure AAA protocols) and in particular because of the contractual implications. Since it was deemed unlikely that results could be achieved during the short time of the WIPTEL project in the first place, the architecture instead focuses on services that do not require accounting and billing between institutions. However, there still isn't anything that would prevent institutions to offer billable services either internally or on the basis of bilateral agreements with other institutions; this possibility has been taken into account in the WIPTEL architecture.

For the latest additions to the architecture resp. numbering plans summarized here, please refer to section 3 of this deliverable.

1.1.3 Setup of Testbed and Support Website

Creating a WIPTEL VoIP testbed based on the aforementioned local and wide-area architecture(s) has been (and still is) the vital next step on the way to put into practice the theoretical considerations we have made, or, at best, tested in the local setting of our university's network so far. We are relieved that this third phase has now started with partners in Hanover and Berlin, as depicted in section 7.

Alongside with this, we intend to build an information base at wiptel.org that will eventually serve as a hub for all institutions interested in IP-telephony connectivity and services offered by the DFN. This was originally intended for an earlier point in time, too, but was postponed due to the testbed's delay.

After the initial testing phase had to be delayed due to the severe problems encountered while trying to get and then evaluate an adequate range of VoIP equipment from the more prominent vendors, it

has now been possible to provide initial recommendations, tools and funding to the institutions that will take part in that testing phase at last. Amongst this phase's goals are:

- Verify and possibly further revise the overall management infrastructure proposed by the TZI on the site and well as the WiN level. This might eventually yield input e.g. to the ongoing TRIP development, as the TZI is working on one of the first TRIP implementations on the market.
- Verify and possibly revise the inter-gatekeeper enhancements (as well as some vendor-specific ad-hoc hacks) made to the TZI gatekeeper modified for WIPTEL.
- Together with the participating institutions, get a first rough hands-on impression of end-user VoIP over the existing WiN network. Thereby, identify existing/remaining problems and propose solutions to them. These might require the cooperation of/steps to be taken by the DFN-NOC.
- Raise awareness for VoIP (and thereby the WIPTEL project) within the initial peer institutions as well as the WIN-connected organizations as a whole. This implicates a thorough, but also prompt documentation and presentation of the testbed phase and its results.

It should be noted, though, that it wasn't part of the original project proposal to implement/provide own components on the present scale. This, together with the inclusion of corresponding testing into the initial rollout, became necessary after we were unable to aquire more (that is, the initially expected amount of) test equipment from the relevant vendors, as already noted above.

Candidates as peer institutions were especially those that had already shown an interest in VoIP at the research level and had participated in the relevant standardization organisation working groups, conferences and the like. This will, for example, ensure quality feedback that will enable the inclusion of user-level peers in the subsequent phase.

1.1.4 Work on Standards and Protocols

The WIPTEL project has been characterized by the requirement to coordinate the project activities with the contributions to ITU-T and IETF standardization being developed in parallel by the TZI. Future-proofing the investments soon to be made in IP-based telephony in the research environment requires understanding the rapidly progressing standardization (both at a formal level and in the form of industry trends) and converting this understanding into practical interoperability. In this regard, TZI staff has actively participated in ITU-T as well as IETF activities throughout the whole of the project.

Based on our experiences in the field of standards and standardization, we have proposed a set of rules early in the project (see deliverable 3) that we have stuck to. We recommend this be continued in the further deployment of IP-telephony within the DFN resp. the German research network.

- The IP telephony infrastructure must be based upon international standards where possible and should not consider solutions using proprietary protocols in their place. Main stream standards should be followed rather than particular profiles developed by consortia for specific user communities.
- Functionality and services in the WIPTEL project shall be driven from the edges; that is, functionality shall be provided and enriched from the institutions with the WiN acting only as a transport platform and facilitator (not dictator) of services provided. Therefore, WIPTEL is to follow the Internet end-to-end architecture rather than the network-centric approach of the traditional telephone environment.
- This leaves H.323 and SIP as protocols to be considered. It is important to keep the WIPTEL infrastructure protocol-agnostic with respect to the call signaling protocol (SIP vs. H.323). H.323 and SIP infrastructures should use common data bases for administration, call routing, service provision, etc. as far as possible.
- The local infrastructure components should be kept independent from the protocols and infrastructures for inter-domain communication and back end services, to facilitate supporting rapid evolution in a multiplicity of inter-domain and back end protocols. In the short term, a fully

standardized inter-domain communication platform cannot be implemented based upon readily available commercial products.

- A heterogeneous environment — within the WIPTEL infrastructure as well as at the institutions — needs to be supported. This applies to vendors and products as well as to protocols. In the face of system heterogeneity, transparent call completion from any endpoint to any other endpoint is required: this necessitates running signaling gateways (SIP to H.323) as part of the WIPTEL infrastructure. For communication with the traditional telephone network, IP telephony gateways to PSTN/ISDN need to be provided as well.

TZI staff will continue to be highly visible in the field of ITU-T and IETF standardization, not only regarding IP-telephony.

1.2 Past Deliverables

So far, we have presented five WIPTEL deliverables, albeit in different ways. D1, D3 and D4/5 were handed in as full reports, whereas D2 has been a status report given as a presentation at DFN in Berlin. The following briefly summarizes the content of these deliverables, as laid out in the preceding section.

- The initial deliverable 1 presented an overview over the existing as well as available IP-telephony components. Deliverable 1 is dated 16.9.1999.
- Deliverable 2 was presented as an interim review of the WIPTEL architecture and progress report at the DFN in Berlin in late 1999.
- Deliverable 3 defined the system architecture of the WIPTEL project, starting from a discussion of IP telephony services, an overview over the (then) current state of and future outlook on pertinent standards and protocols, an outline for the potential uses of IP telephony in the WiN, going into more detail on the system architecture for local IP telephony structures as well as global WiN structures. Deliverable 3 is dated 1.12.1999.
- Deliverable 4/5 documented the first version of the WIPTEL site reference configuration made up of commercial of the shelf (COTS) IP telephony components for endpoints (stand-alone IP telephones) and gateways as well as various implementations developed by the TZI (prior to WIPTEL but adapted to the WIPTEL needs as part of the project) as well as software from the public domain. The management components for a site and/or institution (Gatekeepers) were (and still are) entirely made up from software developed at TZI; several commercial products have been tested as well. Furthermore, deliverable 4/5 elaborated on the implementation of the first step of the concept for the WiN-wide distribution of call signaling information that was outlined in deliverable 3. This included providing a server infrastructure at TZI for collecting IP telephony routing information from all institutions, processing the input to create an IP telephony routing database and disseminating this result. Deliverable 4.5 is dated 20.7.2000.

This deliverable is dated 11.12.2000. There will be an additional deliverable 7 covering the results of the testbed phase.

2 Update on Standards and Protocols

This chapter provides a brief update on progress in ITU-T standardization, assuming that the reader is familiar with the basic technology. It was taken from packetizer.com ,maintained by the editor of the H.323 standard. For an introduction to the current issues regarding SIP, please refer to appendix B.2.

Many new enhancements have been introduced into the market-leading VoIP protocol H.323. Version 4 will be approved November 17, 2000 and contains enhancements in a number of important areas, including reliability, mobility, and flexibility. New features will help facilitate more scalable Gateway and MCU solutions to meet the growing market requirements. H.323 has been the undisputed leader in voice, video, and data conferencing on packet networks, and Version 4 makes strides to keep H.323 ahead of the competition. Gateway Decomposition

Recognizing the need to build larger, more scalable gateway solutions for carrier solutions, the ITU-T SG16 worked jointly with the IETFto produce the new Recommendation H.248, which describes the protocol between the Media Gateway Controller (MGC) and the Media Gateway (MG). To support this "decomposition" of the Gateway, H.323 contains a new section that describes some of the various architectural designs that may be achieved by decomposing the Gateway into the separate MGC and MG.

In addition, text exists to explain how the other protocols utilized within the H.323 system may be utilized in order to produce a complete system. Considering the various needs of enterprises, service providers, and equipment suppliers, H.323 discusses Access and Trunking Gateways as used in both the services provider and enterprise markets and suggests possible ways of dealing with FAS and CAS signaling.

Multiplexed Stream Transmission

One weakness with the current usage of RTP is difficulty in synchronizing the separate audio and video streams. Version 4 now includes an optional procedure which allows both video and audio to be multiplexed in a single stream. This will assist endpoints in syncronizing video and audio so that presentation to the user looks more natural.

Supplementary Services

One of the most important features of a VoIP protocol is its ability to provide services to the service provider and end users. H.323 has a rich set of mechanisms to provide supplementary services. Version 4 introduces a few more which strengthen the protocol in this regard. In addition to the new Annexes K and L and the new H.450.x documents (described below), there is a new section in the main body that attempts to "tie it all together" so that the reader can better understand when and where to apply specific service models.

Annex K/H.323

The new Annex K describes a means of providing HTTP-based control for H.323 devices. With this Annex, service providers have the ability to display web pages to the user with meaningful content that ties into the H.323 systems. In essence, it is a third party call control mechanism that utilizes a separate HTTP connection for control. This should not be confused with the simple ability to redirect a user to a web page-- something that H.323 has been able to do since Version 2. Rather, this is a new "service creation" environment, which is unlike anything described to date. In addition, because the procedures for HTTP-based do not need standardization, new features may be introduced without the delay introduced by any formal standardization process.

Annex L/H.323

Annex L provides a new "stimulus-based" control mechanism for H.323. With Annex L, an H.323 device may communicate with a feature server to provide the user with various services. The H.323

endpoint may possess some intelligence, but some intelligence may reside only in the feature server or multiple feature servers. Annex L builds on the strengths of the "package" concept introduced in H.248, so the feature possibilities are numerous. More importantly, because anyone may define and publish package specifications, new features may be introduced without the delay introduced by any formal standardization process.

Annex M.1/H.323

The purpose of this annex is to give guidance how the generic tunnelling mechanism described in section 10.4 of H.323 can be used to tunnel QSIG over H.323 networks.

Annex M.2/H.323

Tunnelling of signalling protocols (ISUP) in H.323.

H.450.8 - Name Identification Service

H.450.8 builds upon H.323 "caller identification" procedures by providing a standard means of conveying user identification data to the remote endpoint.

H.450.9 - Call Completion

This new supplementary service definition provides a standard means of allowing calls to complete when the user is either busy or there is no answer.

H.450.10 - Call Offer

Call Offer (SS-CO) is a supplementary service which, on request from the calling user (or on that user's behalf) enables a call to be offered to a busy called user and to wait for that called user to accept this call.

H.450.11 - Call Intrusion

The Call Intrusion supplementary service (SS-CI) enables a calling user A, encountering a busy destination user B, to establish communication with user B by breaking into an established call between user B and a third user C.

Additive Registrations

One weakness that previous versions of H.323 had was the inability of a large device, such a Gateway or MCU, which possessed hundreds or thousands of alias addresses, to register those addresses with the Gatekeeper. The problem was quite simple: the size limitation of a UDP packet just prevented that from happening. Version 4 gets around this problem with a new concept called "Additive Registrations". In essence, an endpoint may register with a Gatekeeper and provide an initial list of aliases, but then may follow the RRQ with additional RRQs in order to provide the Gatekeeper with a complete list of alias addresses.

Alternate Gatekeepers

One of the most important aspects of any telephony system is "uptime". Customers do not want to be without phone service and service providers do not want a loss in revenue. Gatekeeper failure often results in missed calls, lost revenue, or both. Fields were introduced into H.323v2 to provide for Gatekeeper redundancy, but the usage of those fields was never fully explained. Version 4 introduces a new section that details the procedure that endpoints may follow in order to provide some robustness to the system.

In addition to procedural text, a new field was added to allow an endpoint to indicate whether it supports the Alternate Gatekeeper procedures. This allows the Gatekeeper to make intelligent

decisions about redirecting an endpoint to provide for some level of load balancing across Gatekeepers.

Usage Information Reporting

To help provide accurate billing information, the Gatekeeper may request the endpoint to provide usage information reporting to the Gatekeeper at various times during the call, including at beginning of the call, during the call, and at the end of the call. This new capability works well with Annex G/H.225.0, where usage information reporting may be necessary and when call signaling is not routed through the Gatekeeper. This feature may also be used by alternate Gatekeepers, which handle the call termination for a call that did not originate with that Gatekeeper. Usage information reporting includes the start and end times, the call termination cause, and any non-standard data the endpoint wishes to provide.

Endpoint Capacity

One very frustrating aspect of many IP telephony services is that calls are often directed to Gateways or other devices that do not have available capacity to handle new calls. H.323 has had an indicator to indicate that a Gateway is "almost out of resources", but this cannot be used by other devices, such as high-capacity conference servers. In addition, indicating that capacity is low says nothing about the true state of the device. For example, are resources low because one of the two ports on a low capacity Gateway is in use or that only 20 ports on a 10,000 port Gateway are available?

With Version 4, the endpoints have the ability to provide precise information about resource availability to the Gatekeeper in a number of messages. The Gatekeeper can use this information to intelligently route traffic to a device it knows can handle the call. This increases the call success rate, and, in turn, increases revenue to the service provider.

Caller Identification Service

New fields were added to H.323 Version 3 to describe provide the means of caller identification, but no description existed for the proper usage of those fields. Version 4 now contains complete text to explain how to provide caller identification services with H.323.

Tones and Announcements

Version 4 details the procedure for indicating the presence of in-band tones and announcements. Such tones and announcements are often heard when the destination number is incorrect or unreachable.

In addition to in-band tones and announcements, the Gatekeeper may signal an endpoint to play specific announcements at various times: pre-call, mid-call, or end-call. This mechanism facilitates two-stage dialing, for example, where the Gatekeeper may request the Gateway to prompt the user for additional information. In that case, the Gateway will play an additional prompt to collect a PIN, for example, and then attempt once again to place the call.

Mapping Aliases

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

Indicating Desired Protocols

When placing a call prior to Version 4, the Gatekeeper had no way of knowing whether or not the calling party needed special services, such as fax support in a Gateway. With Version 4, however, an

endpoint may request in the ARQ that the Gatekeeper resolve the address so that the "desired" protocols are met by the destination endpoint. This will allow a caller, for example, to indicate that it wants to place a fax call and that only Gateways that support fax should be returned-- there is no point placing a fax call to a voice-only gateway, after all!

Bandwidth Management

Prior to H.323 Version 4, and endpoint could request much more bandwidth than it actually needed and, thus, cause network resources to go unutilized. With Version 4, it is now mandatory that an endpoint make bandwidth requests with a lower value if, indeed, the endpoint is using less bandwidth than it had initially indicated in the ARQ.

In addition, managing bandwidth for multicast sessions has been nearly impossible since, unless the Gatekeeper routed the H.245 signalling and carefully monitored the media channels that were opened, it could not determine whether two endpoints that request bandwidth are actually requesting bandwidth for a multicast session or unicast session. This becomes a much bigger issue when many people are participating in a multipoint multicast conference. With Version 4, specific details about the media channels are conveyed to the Gatekeeper in IRR messages (if the Gatekeeper requests them), so that the Gatekeeper can better control bandwidth utilization.

Reporting Call Status

Like the issue with large registrations, large endpoints, such as MCUs and Gateways, have trouble reporting call details to a Gatekeeper due to limitations in the size of a UDP packet. For this reason, Version 4 now provides a mechanism through which an IRR containing information for multiple calls may be broken into several distinct messages. This allows the endpoint to convey all of the call details to the Gatekeeper.

Enhancements to Annex D (Real-Time Fax)

A very useful feature of fax devices is the ability to initiate a voice call and then switch to fax at some point. Version 4 of H.323 extends Annex D to allow an endpoint to do just that. Along with the obvious benefit of allowing an IP-based fax device to operate in a similar manner as today's PSTN fax devices, the media switch is performed in such a way that DSP resources is conserved, which reduces the overall cost of equipment.

Annex D was also enhanced to utilize TCP for carrying fax data. Previously, UDP was the only real option for carrying fax data.

Call Linkage

H.323 Version 2 introduced the concept of the Call Identifier, which is a unique identifier that can be used to identify a call with multiple segments from end to end. However, there is also a need to associate that call to the original party when a call transfer or other service is invoked, in which the original participants are no longer present in the call. Version 4 introduces several new fields that allows equipment to "link" call legs together for this purpose. This provides for, among other things, more accurate billing for a call.

Tunneling

People understand that when signals are translated from one system to another and then back to the original signaling, certain information is lost. In H.323 systems used in both public and private networks, H.323 is often used to interwork between two circuit networks. To provide better interworking, Version 4 now provides a mechanism whereby QSIG and ISUP may be tunneled without translation-- essentially, H.323 may act as a transparent tunnel for those non-H.323 signaling protocols.

QoS

Quality of Service is very important in any VoIP network. As a first step in improving QoS in H.323 systems, new procedures are defined in H.323 to allow for RSVP when not using Fast Connect. Obviously, work is continuing in this area in both the ITU and the IETF.

Mobility

Version 4 introduces new procedures for user, terminal, and service mobility. (Details are not yet available for publication.)

H.245 in Parallel with Fast Connect

H.323 now allows H.245 to be started in parallel to Fast Connect by including H.245 messages in the Setup message. This allows an endpoint to exchange capabilities in order to determine whether certain features are supported, such as DTMF support in the UserInputIndication message. In addition, by starting H.245 early, two endpoints can more quickly establish an H.245 session in the event that Fast Connect cannot be accepted by the called endpoint.

Generic Extensibility Framework

One of the issues with H.323 as it matures is simply the number of parameters that exist in the base protocol specification. To prevent continued and unbounded growth of the ASN.1 that defines the H.225.0 protocol, a generic extensibility framework has been added to version 4. This framework actually serves two purposes. First, it allows one to send opaque data between H.323 entities without adding new fields to H.225.0, as just mentioned. Second, it introduces a new means of performing feature negotiation. The latter is definitely the most powerful use of this new framework.

An H.323 entity may use the generic extensibility framework in order to indicate its supported features, desired features, and needed features. Entities may exchange this feature information and may then take advantage of mutually supported features. When routing call signaling, entities in the middle of the call signaling path may add to or subtract from the specified feature set if those entities can perform the feature on behalf of one of the endpoints in the call. This means that more intelligence may be built into the network without having to add such intelligence to the endpoints in a call.

H.323 URL

The URL scheme "h323" is introduced in Version 4 of the protocol. The H.323 URL will allow entities to access users and services in a consistent manner, much like other defined URLs allow for other IP-based services. The form of the H.323 URL is "h323:user@host", where "user" is a user or service and "host" might be the Gatekeeper that can translate the URL into a call signaling address.

Call Credit-Related Capabilities

An extremely popular service which utilizes IP telephony today is to allow users to dial a Gateway to place a call (with the anticipation that the call will be much lower than a traditional PSTN call), which is then charged against a pre-paid calling card or to a user's account. Until now, there has been no standard means of communicating available funds or for the Gateway to control early call termination based on available funds. H.323v4 adds these features to the RAS protocol.

DTMF Relay via RTP

H.323 version 4 now allows an endpoint to utilize RFC 2833 to send and receive DTMF digits. This is important, for example, in order to convey precise timing of DTMF information. Also, it is a logical choice when the call is routed through the Gatekeeper and the Gatekeeper is not interested in that information.

3 Using the WiN for telephony

This chapter is intended to inform about the possibility to use the German Research Network (WiN) as a telephony provider. The focus of the considerations lies at the migration of PSTN and IP telephony and the integration of the IP numbering space into the usual numbering space.

3.1 Address Resolution

Besides providing basic IP connectivity at sufficient quality, address resolution for IP telephony calls constitutes the core requirement for the WiN infrastructure. This task can obviously be split between the individual institutions and the WiN core. The individual institutions decide for which addresses (e-mail-style UPIs and fully qualified international telephone numbers) they are able to terminate calls and export the corresponding information. The policies on how much information to reveal about services, persons, gateways, etc. as well as to whom to export the information and from whom to accept calls are determined independently by each institution.

Address resolution is also key to the first value added services to be provided by the WiN as these are primarily mapping functions. For these services as well as for inter-domain communication, trust relationships between institutions are needed so that message exchanges for inter-domain communication can be secured - with the WiN infrastructure providing additional support from the beginning.

3.2 Numbering / Naming Plan

For the development of the individual institutions' numbering plans, this section provides common guidelines to make the address structure more transparent to users and prevent address clashes between the different institutions. Obviously, the address structure should exploit that, with IP telephony, people can be reached by email style UPIs as well as traditional telephone numbers.

The following aspects are recommended to be considered for the telephony numbering plan:

(a) The usual numbering space has to be re-used; i.e. (country and) area code of the institution.

(b) Prefixes for service selection in the internal telephone network should stay the same. The simplest case is the leading '0' or '9' to dial in order to obtain an outside line. Public institutions in Germany are additionally connected through a private network ("Behördennetz") paralleling the public telephone network that can be accessed through another dedicated prefix. Finally, per-call carrier selection is done through a particular prefix as well.

    Independently from the originating network (telephone or IP) of the call dialing prefixes should have the same meaning. An dialing prefix to place external calls to the PSTN should be the same in both networks and therefore should allow an IP phone user to place calls to the PSTN.

    Such a preservation of the calling prefixes is highly desirable, particularly with respect to later migration to avoid (repeated) changes in telephone numbers.

(c) A new dialing prefix to place externals calls using the WiN should be added. This prefix should not be limited to IP phones but must be available for conventional phones too. This allows all users to take advantage of services provided by the WiN for example to use the WiN as a more cost-effective route to reach the destination.

(d) Besides calling prefixes, also (or only) the extension numbers could be maintained identical for individuals (or rooms) at an institution. In the ideal case, a caller from a conventional telephone would dial the same phone number as a user at an IP telephone - and a device (possibly the same one!) on the same desk / belonging to the same person would ring.

    Today, however, at least in academic research institutions, two obstacles immediately spring to mind:

- Typically, not every individual has her own telephone number; rather, several employees share a common telephone set (and thus number). The numbering spaces available may not even be sufficient to give each individual their own number.
- Numbering plans currently found in institutions (this holds true for Bremen as well as a number of other universities, enterprises, etc.) have grown over time and do not / no longer reflect any kind of organizational structure. Rather, telephone numbers seem to have been assigned (moved around) randomly. If existing numbers are kept in their entirety, this may prevent the development of a more structured numbering plan (which would also be more convenient for distributed administration in the IP telephony environment).

A pragmatic and recommendable approach is to keep the old extensions and only assign new number prefixes (matching the prefixes of newly installed gateways) for IP telephony. This preserves most of the already memorized telephone numbers (all of the internal ones which are likely to be predominant anyway) but avoids clashes with external numbers and allows adding new gateways independently.

To integrate the unstructured numbering plan of the conventional/legacy address space into the structure of the IP telephony address space the later should contain numbers that map to numbers in the legacy address space. An institution deciding to provide every single member his own number would give every user a new number which maps to the number of the users PSTN phone (if there is one).

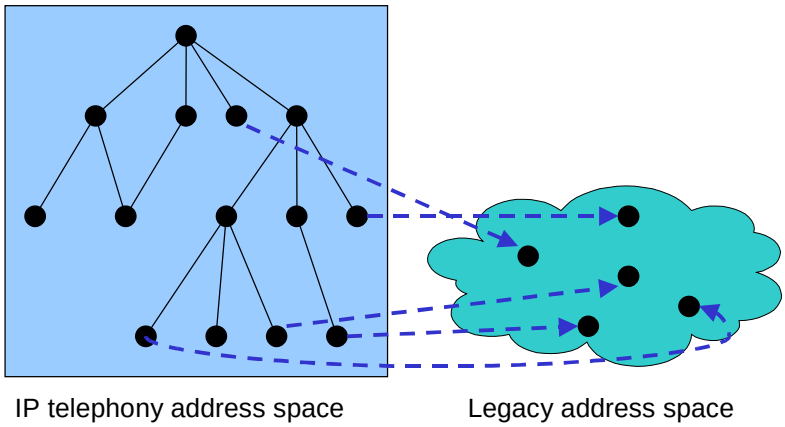

Figure 1 - Mapping from the IP telephony address to the legacy address space

In addition to telephone numbers, UPI-based addressing for users' PC/workstation-based clients is needed as well. Again, conventions for naming and naming hierarchies may be helpful. Depending on the internal conventions of an institution, email addresses (which could nicely be re-used as UPIs) may have one of the following formats:

- first_name.last_name@domain (but see RFC 1439!),
- user-id@domain, or
- function@domain.

For easier mapping of telephone numbers, an additional format may be considered for local phone numbers only: phone-number@domain could map to the respective user's or function's email-style UPI.

Figure 2 shows mapping paths for an integrated name and numbering concept: UPIs are the central identifiers for persons as well as for groups. A group can be expanded (recursively) to a number of persons. Both person and group UPIs can be translated into addresses (possibly represented in URL-style). For destinations in the IP-based network, an address refers to an IP-based endpoint's transport address plus a signaling protocol, for destinations in the telephone network, an address identifies a

gateway's transport address to be used plus signaling protocol plus an extension to be called on the telephone network. Traditional telephone numbers are simply modeled as alias names for groups or persons.

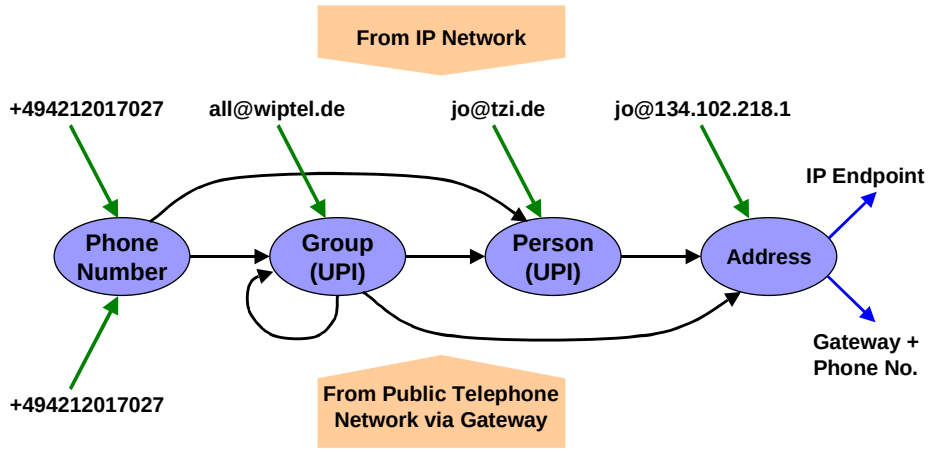

Figure 2 - Name and number mapping process

In addition to such a localized DNS based address resolution scheme for telephone numbers, a scheme on a broader scale may be developed over time - to which the current scheme will then have to migrate. This implies for the numbering that all IP-based endpoints need also be included into a globally mapped telephone numbering scheme - another reason to avoid clashes in the numbering space from the very beginning.

3.3 Intra site infrastructure

To support a seamless migration from PSTN to IP telephony there has to be an interim solution (which might last for several years or even decades) to mix both techniques. A requirement for such a solution would be a connection between the PSTN and the IP PBX to route calls from one network to the other. Most PBXes support prefixes to specify that a call leaves the internal network which technically results in a mapping of a prefix to a trunk. To connect an IP PBX a recommended way is to add a prefix in the PSTN PBX that leads to IP PBX.

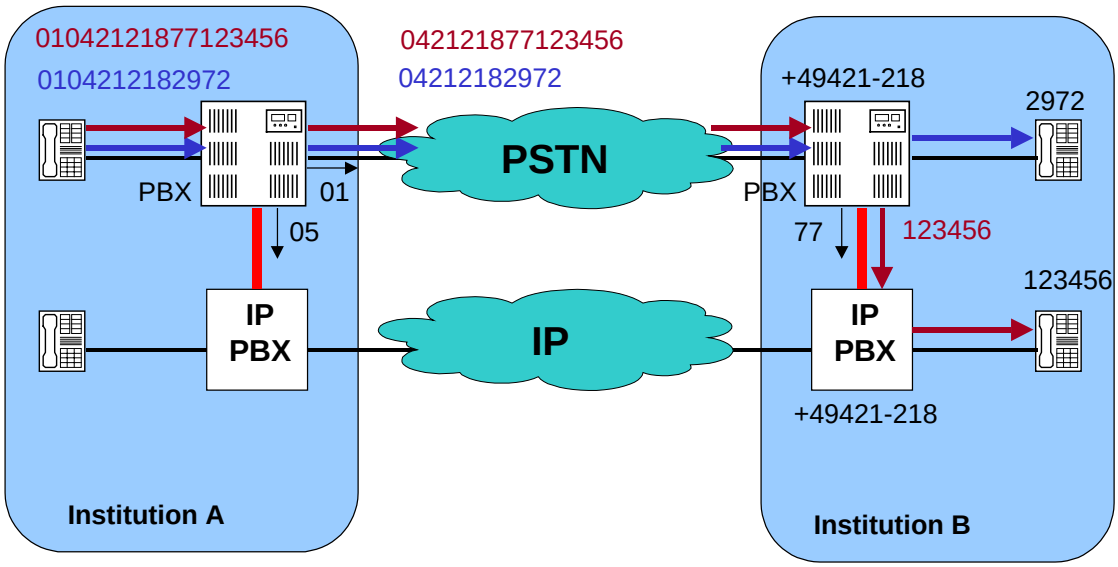

Figure 3 - Shared address space when using PSTN phones

In the example (Figure 3) the PBXes of both institutions are configured to use the prefix 01 to leave the local zone into the PSTN and the new created prefix 05 to reach the IP PBX. These prefixes are only valid for calls that originate in Institution A. To enable an incoming PSTN call to reach an IP phone the prefix 77 is configured to reach the IP PBX (see below).

To reach a destination in institution B a user at an PSTN phone at A can simply dial the well known prefix 01 to reach B via PSTN. The PBX must decide if the number is part of the legacy PSTN address space or the new IP address space. This might be done by using a mapping which maps every number to either PSTN or IP. For that the new IP address space might be very large it seems more likely that all IP destinations are combined below a new prefix. Therefore the PBX strips the external part of the number (in Figure 3 it is 0421218) and decides using the rest number whether to call an PSTN phone or route the call to the IP PBX.

If the called party is currently unavailable the IP PBX may of course decide to redirect the call to another IP phone or even back to the PSTN PBX. The redirection may be done from the IP PBX (e.g. when processing user definable CPL scripts) or from the called IP phone itself.

In the above example no advantage was drawn from the fact that both institutions are connected to the IP network. With the connection between both PBXes a PSTN user may choose to use the IP network to route the call to the destination.

A prerequisite is that the necessary information of which IP to use to reach a telephone number are distributed throughout the IP network. As stated in section 4.1.3 this should be done using the upcoming Telephony Routing Information Protocol (TRIP). With TRIP each IP PBX (to be precise: the associated border element) advertises the reachable IP phone numbers as well as the reachable PSTN numbers. By exchanging those information each IP PBX knows which IP address is the next hop to the target.

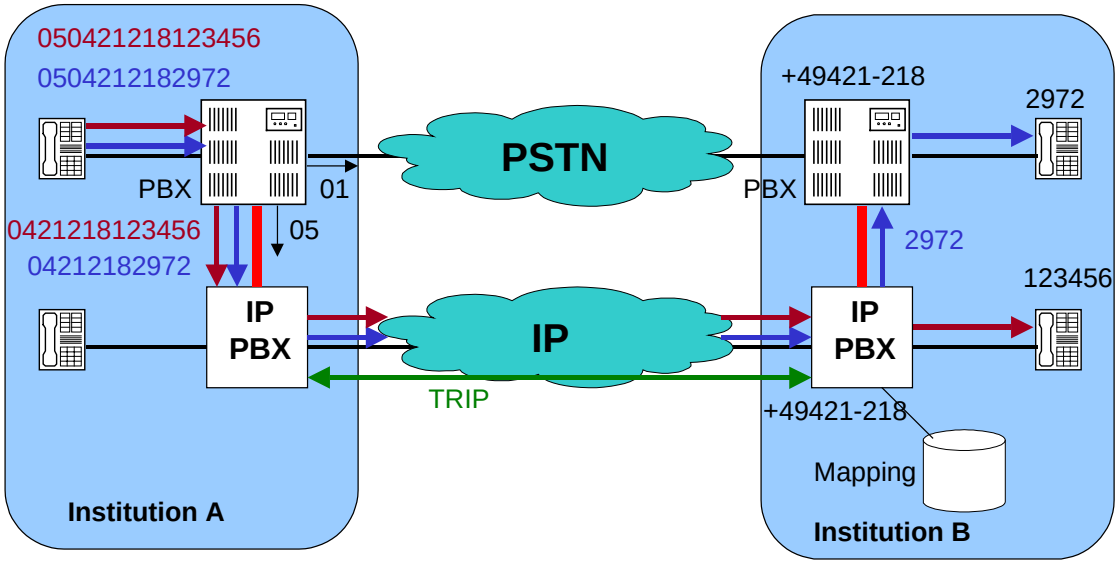

Figure 4 - Routing calls via the Internet

On the caller side this scenario is similar to the scenario explained before. The PSTN user simply dials another prefix which causes the call to be routed via IP. However the side of Institution B differs from the other scenario: There is no dedicated prefix to reach PSTN phones as it was in Figure 3. The IP PBX uses a mapping to decide which destinations are located behind the PBX and should be routed there.

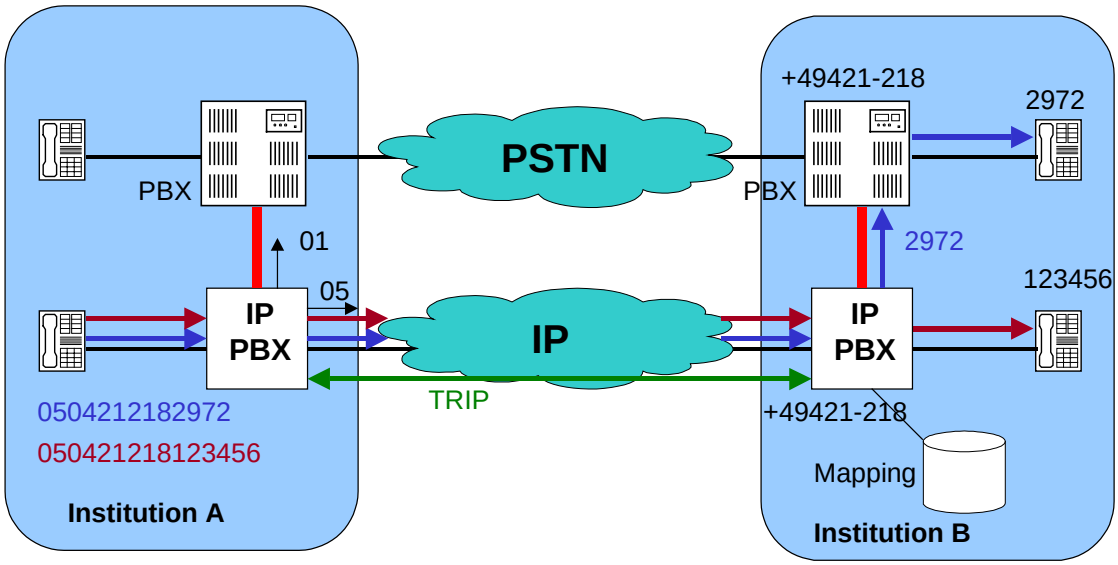

Figure 5 - Routing IP calls

Not very different from routing calls from PSTN via IP is the routing of calls from IP phones via IP. In the first scenario as well as in this scenario the user on the phone chooses the transport method by selecting the prefix for the call. If the PSTN prefix was dialed the PBX is involved in addition to the IP PBX. The side of institution B is the same as in the scenario before.

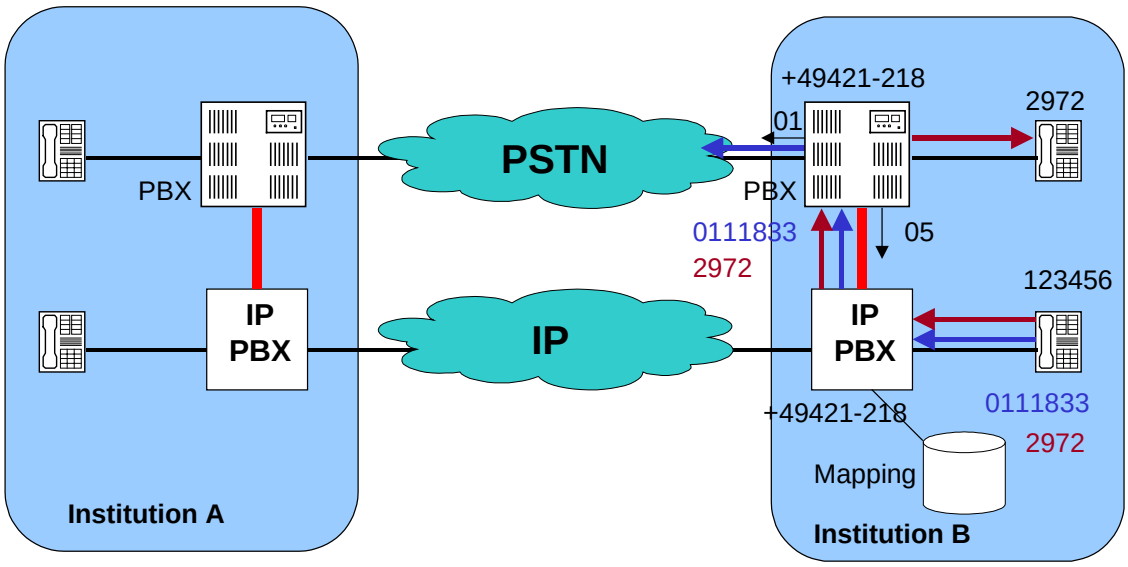

Figure 6 - Calling PSTN targets from an IP phone

While users on PSTN phones have to dial a prefix to reach an IP phone, even an internal IP phone users at IP phones just need a prefix for external calls. The mapping data of the IP PBX (see above) allows the IP PBX to decide if an internal call must be routed to an IP phone or to the PBX.

When using the prefix for external PSTN calls a user at an IP phone is still able to use services of a telecom provider or call PSTN targets.

3.4 Least cost routing

Along with this architecture comes the possibility to offer least cost routing services. If all IP PBXs in additions to the destinations of the their institution advertise routes to all destinations in their area and add pricing information this data may be used for least coast routing.

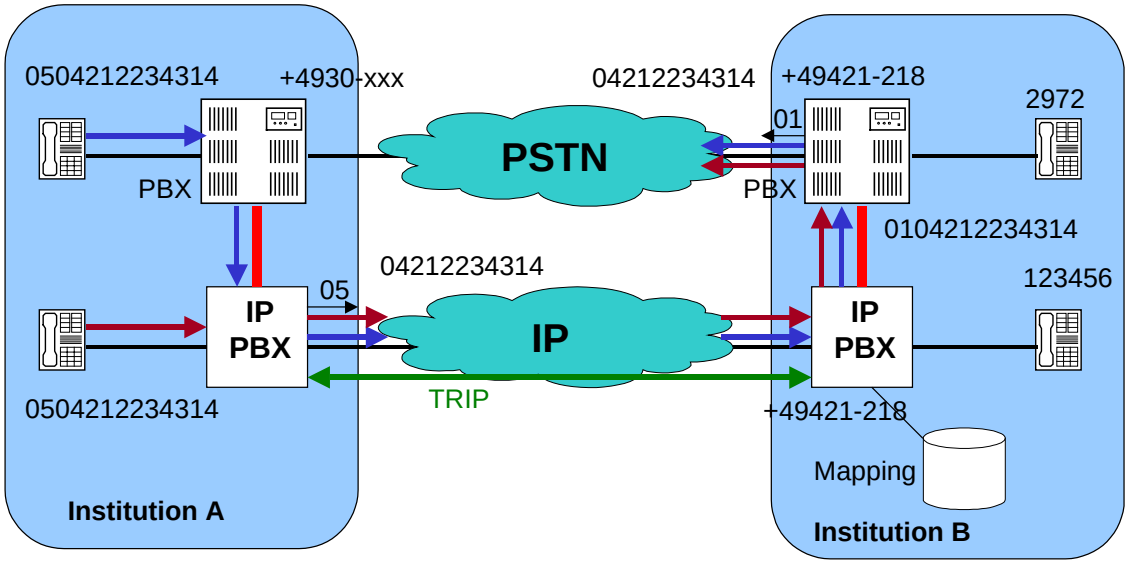

Figure 7 - Least cost routing

Two important features are necessary to provide such a least coast routing service:

- All TRIP location servers involved must be aware of the pricing information in the advertisements of the institution's border elements. Although TRIP requires that a location servers must keep and advertise those information along with a route even when it does not understand or support it, least cost routing would be more efficient of all location server choose the best route regarding to the pricing information.
- There must be a reliable way of billing the originating institution. An site would only allow other sites to use their gateway for calls in the PSTN if it can be sure to get refunded for the cost caused by the calls. The matter of how to proof that calls took place (maybe the call logs of both sites differ regarding routed calls) is an important question for such scenarios.

Both points need further study to make the WiN a least cost routing platform. Without clarification of those points least cost routing is only possible basing on a trust relationship between to sites.

4 WiN Infrastructure

This chapter focuses on the address resolution mechanisms for WiN-wide calls. Such a mechanism is a prereqisite for providing basic call services

When connecting sites in the German Research Network, two major aspects need to be looked at. The first is the way address resolution is done: How does a border element know where a target outside its local scope is? The second is of a rather administrative type - how to organize the allocation of names and numbers that work as "area codes" for institutions in the WiN?

4.1 Inter-domain address resolution

The first step to route a call through the WiN is determining the administrative domain and, there, finding the entity to be contacted for call completion. Depending on the addressing scheme used to specify the callee, roughly two alternatives can be distinguished:

1. If that callee is given using an identifier that contains a domain name (such as `jo@tzi.de`) then the Domain Name System (DNS) can be used to obtain a further point of contact; this is further simplified if URL notation is used, which also specifies the desired protocol to contact the user (`h323:jo@tzi.de`). Conventions (including well-known DNS prefixes and the use of DNS service records) are defined for both H.323 and SIP to determine the address of an instance that is able to complete a call — whatever further steps this may involve.
2. If, however, as in the conventional telephone system, the destination is specified by means of a telephone number as defined in ITU-T E.164 (such as +494212181), no suitable, globally available mechanism is readily available. DNS is not aware of country and city/region codes, extensions, etc. Hence a new mechanism has to be developed that enables call routing in IP telephony networks based upon E.164 numbers.

For reaching the callee, a further distinction has to be made with respect to the called endpoint's location (and this is orthogonal to whether the callee / the endpoint is identified through a URL or an E.164 number):[1]

(a) The called endpoint is an IP-based device with continuous IP connectivity between the calling and the called endpoint.

(b) The called endpoint is located within the PSTN/ISDN/GSM or is an IP-based endpoint without continuous IP connectivity between the calling and the called endpoint.

This yields a total of four cases: 1(a) and 1(b) are simplified because a natural way is to locate an entity in the called endpoint's administrative domain as described above and have this one determine where the callee is currently located and whether or not a gateway (and, if so, which one) needs to be involved to reach the callee. Given this, no further investigations are required for dealing with these cases in the short-term; the future may bring a variety of optimizations, though, and all subsequent considerations can obviously be applied to this case as well. 2(a) requires translating the E.164 number into the called endpoint's IP address. 2(b) means determining the IP address of a gateway suitable for completing the call to the called E.164 number — with additional aspects such as minimizing call cost or maximizing quality optionally to be considered in the choice of the gateway; this will be called the *gateway location problem* in this document.

The following subsections list possible address resolution methods to provide determine the IP location of a called target.

[1] Note that, of course, the calling endpoint as well may be located in an IP network or not. If it is not, the decisions about IP-based call routing are not made by the endpoint but — on its behalf — by some gateway that the caller has explicitly called / the call was routed to so that the same considerations apply as in the former case.

4.1.1 H.323 built-in: LRQ

The most simple (but also least reliable) mechanism comes with H.323 itself. H.323 defines a special message - the Location Request (LRQ) - that shall be used to find an IP address for an alias. When a gatekeeper must resolve an alias address that is neither mentioned in the database nor registered at the gatekeeper, it might send such a request to other gatekeepers. This can be done in two ways:

1. Via multicast - The request should reach several gatekeepers in the WiN. These gatekeepers only need to answer this request when they do know the requested address. If no gatekeeper knows the address the sending gatekeeper will - according to H.323 - repeat sending the request (2 times) because after a timeout of 5 seconds it would look like the delivery of the message failed.
2. Via unicast - A gatekeeper may have peer gatekeepers to ask for resolution of the alias address. The receiving gatekeepers have to answer the request - no matter whether they can resolve it or not.

   Unfortunately this mechanism can't be recommended as a replacement for sites where multicast isn't supported all along the way. A gatekeeper receiving a LRQ only looks up his local database so a gatekeeper A must contact all gatekeepers in the WiN via unicast to resolve any possible address.

Although this mechanism should be supported by every gatekeeper there are problems with existing implementations: Presumably as a matter of security some gatekeepers decide not to answer requests from unregistered endpoints. This means, that a gatekeeper would need to register with its peer gatekeepers - a behavior left unmentioned in the H.323 recommendation.

During our tests we found several gatekeepers "out in the world" and used them all for address resolution. About half of them refused to process the request because our gatekeeper wasn't registered with them.

Another observation was that some gatekeepers even don't use the feature of LRQs. This is regrettable because up to today it is the only mechanism that gatekeepers that are already in use could possibly support.

4.1.2 Nameserver + SRV-Records

Because sending LRQs is not the best solution, a mechanism of finding the gatekeeper associated with the target endpoint appeared in H.225.0, Appendix D "Discovering Using DNS". The appendix defined the usage of DNS service records (SRV) to distribute information which gatekeeper resolves aliases for a given domain.

```
$ORIGIN tzi.test.
@               IN SOA      demoversion.tzi.test. prelle.tzi.de. (
                                        46              ; serial
                                        3H              ; refresh
                                        15M             ; retry
                                        1W              ; expiry
                                        1D )            ; minimum

                IN NS       @
                IN TXT      "ras gatekeeper@informatik.uni-
bremen.de"
localhost       IN A        127.0.0.1
sip             IN CNAME    damn.informatik.uni-bremen.de.
gatekeeper      IN A        134.102.218.71
gatekeeper2     IN A        134.102.218.62

_sip._udp       IN SRV 0 100 5160    sip
_ras._udp       IN SRV 0 0   1719    gatekeeper
_ras._udp       IN SRV 10 5  1719    gatekeeper2
_h225._tcp      IN SRV 0 0   1720    gatekeeper
_tel._udp       IN SRV 0 0   1719    gatekeeper
_tel._tcp       IN SRV 0 0   1720    gatekeeper
```

```
; NO other services are supported
*._tcp           IN SRV  0 0  0       .
*._udp           IN SRV  0 0  0       .
```

Figure 8 - Example bind configuration with SRV records

If the nameserver does not support SRV records, TXT records may be used alternatively. H.225.0 Appendix D specifies an TXT record for a gatekeeper like this:

ras [<**gk id**>@]<**domain name**>[:<**portno**>] [<**priority**>]

Where <**gk id**> is the gatekeeper ID, <**domain name**> either the name of the A-record which contains the gatekeepers IP address or the IP address itself, <**portno**> the gatekeeper's portnumber (default: 1719) and <**priority**> specifies the order in which the gatekeepers should be used if multiple TXT entries are present and where lower numbers indicate a higher priority.

We don't know of any available gatekeeper implementation that already supports address resolution via SRV records. Only the gatekeeper that was modified within the WIPTEL project has implemented support for this feature - although there hadn't been the opportunity to test it extensively.

4.1.3 TRIP

While using LRQs is fine for finding registered endpoints and SRV-Records can be used for resolving H.323-ID alias addresses of registered endpoints both mechanisms fail on distributing reachability information from gateways. To enable the WiN to provide least cost routing services a mechanism is required to distribute such information.

Such a mechanism is provided by the "Telephony Routing Information Protocol" (TRIP). Currently TRIP is only an internet draft, so support is still lacking in current gatekeeper implementations. However a preliminary implementation of this protocol is in progress in the TZI that later on can be integrated in the gatekeeper.

TRIP only supports the distribution of telephone numbers. H.323-IDs can still be resolved via contacting the nameserver of the domain portion of the H.323-ID alias.

Proposed infrastructure

To enable IP Telephony in the German Research Network each site must provide at least one border element that provides an aggregation of all targets that are reachable in its domain. Such a border element is contacted from inside it's domain to resolve addresses outside the domain and exchanges information of its domain with other border elements to enable other border elements to provide address resolution services to their domain.

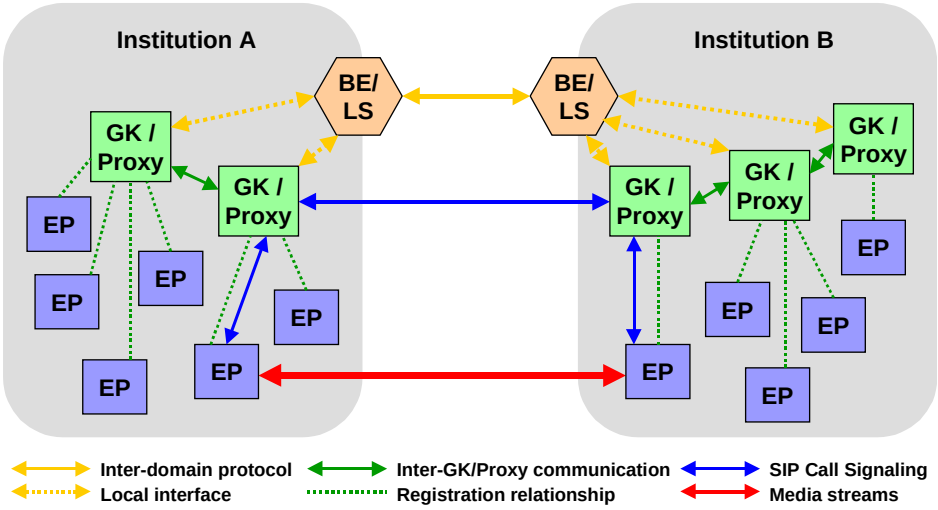

Figure 9 - Peer site information exchange

Figure 9 shows the infrastructure of two sites that have a peer TRIP relationship. Although it is possible to exchange complete routing information for the WiN with at least one peer per site this is not recommendable. Instead the WiN could provide sites with a TRIP routing backbone build from a hierarchical location server infrastructure as shown in Figure 10.

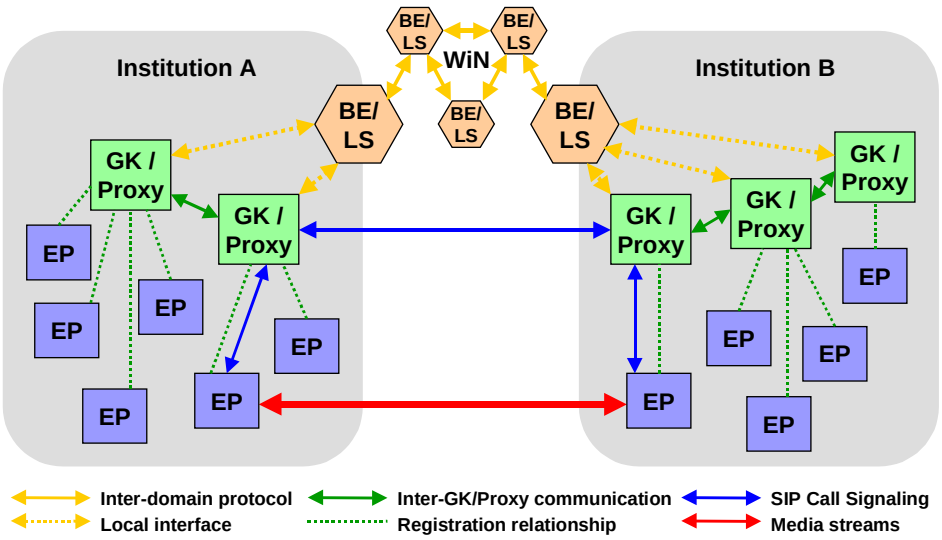

Figure 10 - WiN provided routing service

4.1.4 Shared routing tables

Unless all gatekeepers used in the WiN support TRIP, another mechanism must be used. An interim solution is that the WiN provides a centralized table with routing information to all connected border elements (see Figure 11).

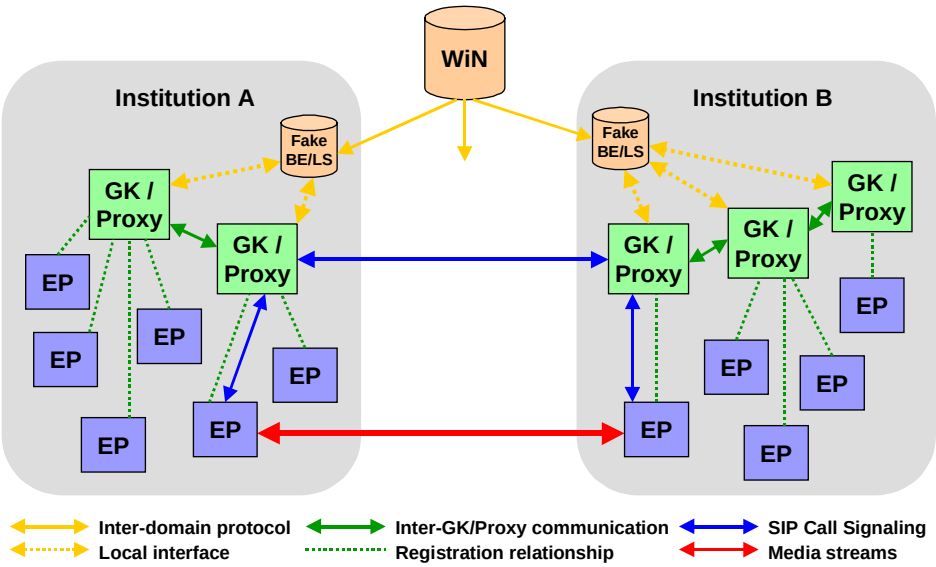

Figure 11 - Centralized routing information database

In the WIPTEL project this is done by a shared routing table that indirectly can be modified by all sites. Every site provides information about reachable destinations. Initially this information is transferred to a central server. This can be done using a web page or via email. The server stores the information in a database from which on a regular basis a complete routing file is generated. To distribute this file it can be actively pulled from all sites in the WiN from an FTP server. Another possibility would be the use of a tool for a remote file distribution. The last option is not finally decided.

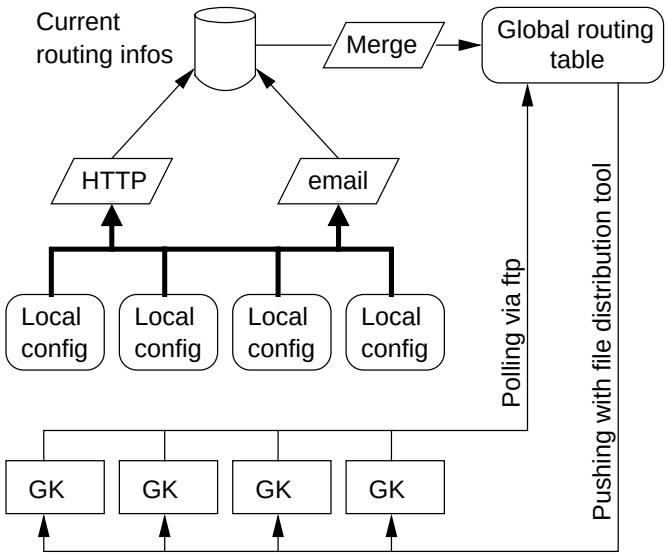

Figure 12 - Distribution of routing information

To ensure that an administrator from site A cannot modify routing information from site B, access to the database is protected with login and password. When accessing the information via WWW, only those records are visible that belong to the user that logged in.

```
# Uni Bremen
tel     +49421          134.102.218.46
h323    tzi.de          dutchman.informatik.uni-bremen.de
h323    tzi.org         134.102.218.46
```

```
h323    uni-bremen.de    whatever.uni-bremen.de        1725
```

Figure 13 - Example entry in routing file

Figure 13 shows an example of a small routing file - telling that telephone numbers starting with +49421 and H.323-Ids ending with tzi.org shall be routed to the IP address 134.102.218.46, H.323-Ids ending with tzi.de shall be routed to the gatekeeper dutchman.informatik.uni-bremen.de and all H.323-Ids ending with uni-bremen.de to the host whatever.uni-bremen.de - at port 1725 (instead of the default 1720).

Modifying routing information

To modify the routing information via WWW a yet to be defined URL - planned is http://e164.wiptel.de - leads to a page where login and password is prompted. After giving that information the user reaches a page displaying the previous information (see Figure 14) where he has the possibility to add, modify or delete routing entries.

There are also plans for an email interface that adds the possibility to make those changes by simply sending an PGP encrypted email containing the most recent version of a institutions routing data.

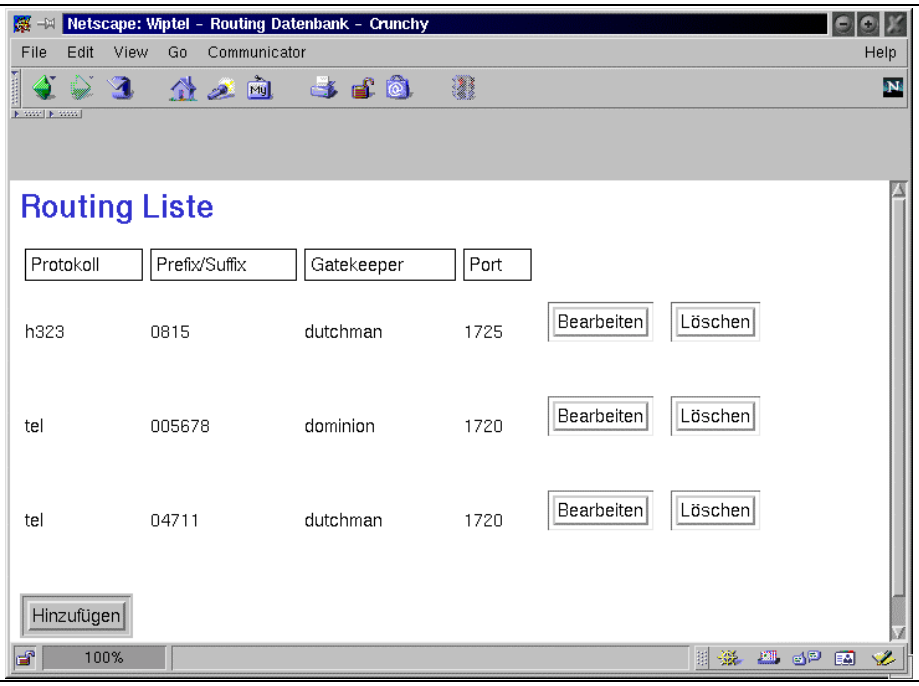

Figure 14 - Modifying routing information via WWW

Distribution of routing information

As stated above the server creates a complete list on a regular basis. This list must be shared with all institutions on the WiN. There are multiple ways to do this which can be split into push- and pull-technologies.

- **Push-technologies** could be active FTP where the server is uploading the new file to each other border element gatekeeper in the WiN. While this is an simple and easy to implement solution it does not scale well. To avoid that the server has to make several hundreds of uploads there would have to be some kind of artificial tree-like hierarchy for the border elements in the WiN.

  There are other considerations about using a multicast file distribution service. This solution would reduce the use of resources and maintenance but presumable is a bit more complex to set up. Currently no intense studies have been done on this subject.

The disadvantage of these technologies is that the receiving site must give a bit of control of itself out of hand by enabling the server to push the data without the possibility to influence the servers behavior (e.g. the location the server uploads the file to).

- **Pull-technologies** include client-side requests via FTP (recent data file is located on a dedicated server) , HTTP or email (an email to a dedicated address produces an automatic reply with the recent version of the complete routing data).

  The advantage of all these methods is the opposite to the disadvantage of the push-technologies: The client side has the total control of when, where and how it receives the requested information.

As mentioned at the beginning of this section the mechanism of shared routing tables is just an interim solution unless all gatekeepers in the WiN support TRIP. Therefore no big efforts will be made to find a perfect scalable solution. The first mechanisms available will surely include the pull technologies like FTP and HTTP. Then perhaps the usage of a multicast file distribution service will be looked into more detailed.

WIPTEL tries to provide a variety of possibilities - the decision which mechanism to use will be up to the institutions that are setting up IP telephony.

4.2 Gatewaying between H.323 and SIP

Another subject that must be addressed is gatewaying between H.323 and SIP. If an institution equipped with SIP phones wants to call a H.323 target somewhere in-between, the call signaling must be translated from SIP to H.323 and vice versa. To achieve this, either a local signaling gateway or one or more public signaling gateways must be available (see Figure 15).

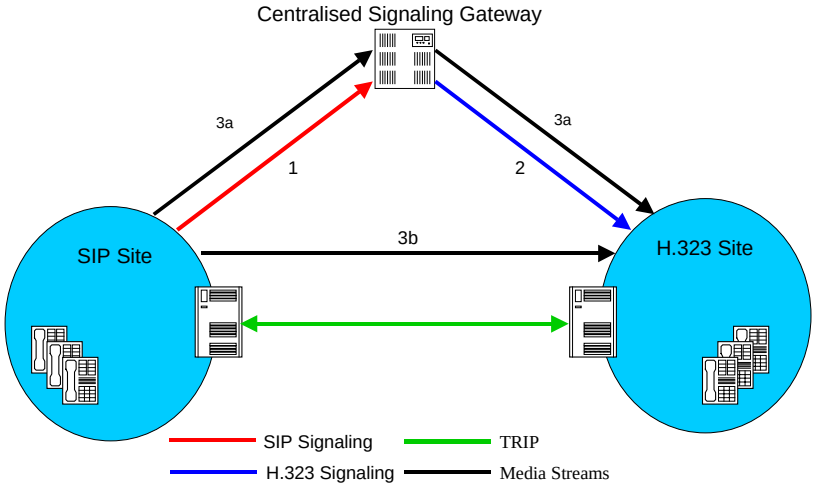

Figure 15 - Centralized signaling infrastructure

Given the existence of a central signaling translation service, the domains would need to exchange routing information to gain knowledge of the location of possible targets and would have to route all their calls to a signaling gateway including information of the location of their target.

Although the central signaling gateway would not need to locate the target anymore, this solution does not scale very well. For networks like the German Research Network, the resources of a single gateway would be exceeded. It is more likely that some sites have their own signaling gateway that offer signaling translation to the site or even as a public service.

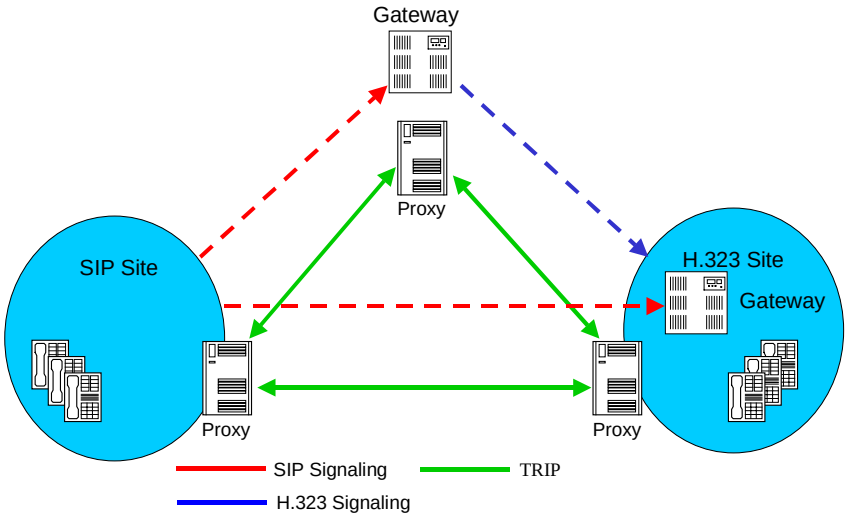

Figure 16 - Distributed gatewaying architecture

Although SIP-H.323 interworking has been a subject of studies for several months, it will still take some time until recommendable solutions will exist. As an somewhat interim solution, TZI/DMN already provides StarGate (see section 6.2) a H.323-SIP-ISDN gateway that will still be maintained after the end of the WIPTEL project.

Each site running such a H.323-SIP(-ISDN) gateway may use its border element to tell other border elements/proxies which H.323 and SIP addresses it supports. This information is distributed using TRIP. TRIP (which is modeled like BGP) also defines that a peer has to select a preferred route if multiple possibilities to reach an endpoint in another domain exist. Therefore a local domain's border element always chooses the "best" route to the target.

This mechanism might be used to provide least coast routing services, if ISDN routing information is provided along with that regarding H.323 and SIP.

5 Site Infrastructure

This section addresses the issue of aspects when a site plans to introduce IP telephony. Section 5.1 outlines the requirements to an IP telephony system for complex institutions while the next sections introduce architectural ideas to met those requirements supported by the TZI/DMN gatekeeper. Section 5.4 lists components of an H.323 system that are considered important for an IP telephony system. The problems of site internal gatekeeper communication are addressed as well as problems resulting from heterogenous gatekeepers. The last two sections give an example of how to set up a numbering plan / naming scheme which is done exemplarily for the University of Bremen.

5.1 Supported site Architecture

When building an IP Telephony system for the needs of institutions in the German Research Network there are two important aspects that need to be considered: scalability and distributed administration.

- Scalability is important because institutions may vary in size from a few tens up to several thousands of people. In the latter case, a single server handling all requests incoming is likely to be inefficient - not mentioning the robustness aspects when there is a single point of failure.
- Besides the technical aspect of scalability there is the need for distributed administration on the organizational side. When using a single (centralized) PBX it is nearly impossible to enable organizational units within an institution to take care of their needs on their own. On the other hand, definition of a numbering plan and allocation of free phone numbers must be controlled in a centralized fashion.

The WIPTEL project has taken a distributed approach at the technical side - as supported by IP telephony products - by defining building blocks that can be used to construct a larger system without prescribing a particular structure. The basic building block (e.g. to administer a site) allows local assignment of responsibilities and local control of resources and can refer to other building blocks in an arbitrary fashion (thereby allowing the creation of more complex structures (such as an institution). This approach supports server distributed administration and scalability at the same time. The individual building blocks can serve as backups for each other thereby addressing the issue of overall system robustness as well. The WIPTEL approach also makes use of standard Internet mechanisms (such as the DNS system and the recently introduced SRV records) where applicable.

The system built in the WIPTEL project is designed to support organizations up to a high degree of organizational complexity. The assumption is that universities and related institutions are the most complex structures that can be found in the German Research Network. A university can usually be subdivided into several departments and a central administration. A department may be a combination of faculties that are again divided into several research groups. Usually departments or faculties share a common administration. A possible structure may look like the one depicted in Figure 17.

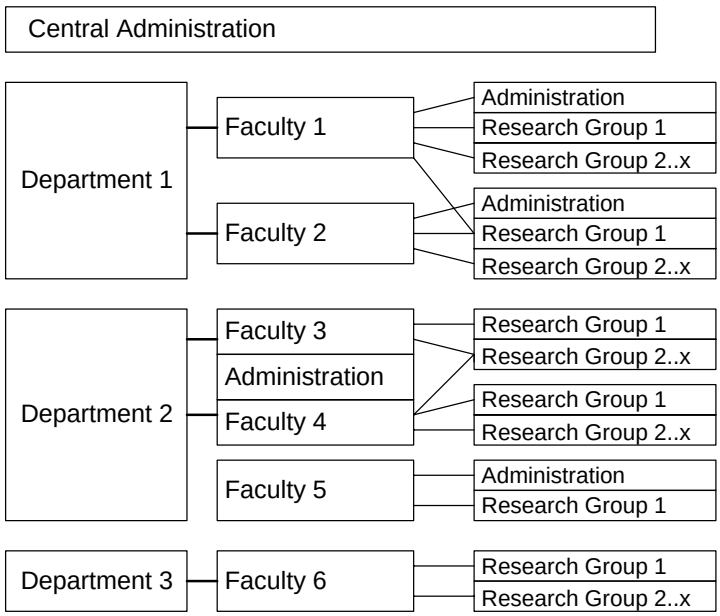

Figure 17 - Example for a supported organizational structure

5.2 Shared address space

WIPTEL does not aim to come up with predefined configurations for all possible organizational structures in the WiN but offers a mechanism to customize the system so it meets their respective requirements. The basic concept is that of a distributed administration with trust relationships.

The entry point to a site is a border element - a gatekeeper (see section Gatekeeper) - that communicates with systems outside the local site. There might be a gateway (see section Gateways) enabling communication with an already existing PBX. In a very flat architecture all endpoints would be registered with that gatekeeper (see Figure 18).

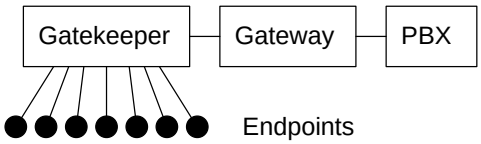

Figure 18 - Simple architecture

A gatekeeper may have slave gatekeepers that are administrated by other persons. This may be useful if subordinate units shall be enabled to change the configuration (e.g. create or modify accounts) on their own thus enhancing the speed of this process bye reducing the number of persons involved. Each slave gatekeeper can be reached by a number prefix and/or naming suffix. It should be noted that this prefixing is used to simplify maintenance of the numbering plan and provide structured phone numbers for the convenience of the users but is by no means technically required. Arbitrary subsets of the address space could be administered by any gatekeeper with reachability information being exchanged upon request (e.g. via LRQ) or in advance (via H.225.0 Annex G or TRIP). Following an approach that resembles the phone world more closely has the additional advantage that it is easier to explain to those who have been managing the PBX system before (which may in turn ease introduction of such prototype systems).

Example 1: *An institution "Example Industries." has two locations in different towns. Both locations shall be reachable via the same prefix "080012". Both locations run a gatekeeper but only one*

location has a PBX connected to the ISDN network. So this location's gatekeeper is defined to be the primary gatekeeper with the number prefix 9 or name suffix "loc2" to reach the other gatekeeper.

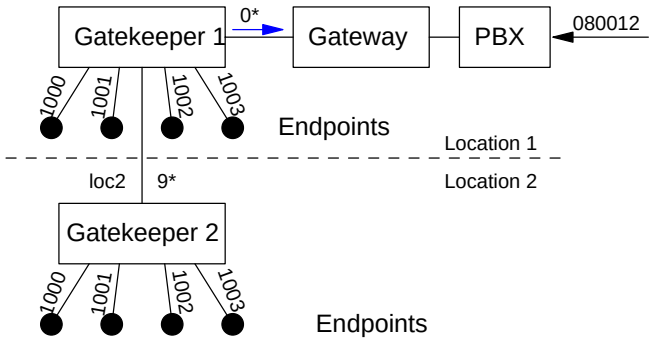

Figure 19 - Example architecture 1

Considering the figure above there are now 8 endpoints that can be reached with the numbers `1000` *...* `1003` *and* `91000` *...* `91003` *and the names* `1000` *..* `1003` and `1000.loc2` *..* `1003.loc2` *from inside the institution. To reach them from outside the prefix number* `080012` *or suffix name* `exampleInc.org` *must be added.*

To place a call outside the institution the prefix `0` *must be dialed to reach the ISDN gateway.*

Address resolution starts at the root of the tree - meaning the primary gatekeeper. If a user of endpoint 1000 at gatekeeper 2 dials 1001 he/she does not reach the endpoint 1001 at gatekeeper 2 but the one at gatekeeper 1. This ensures that the addresses are the same no matter at which gatekeeper one is registered.

Example 2: *Now for some reason the second/slave location decides to experiment with a gatekeeper and a gateway of their own. Endpoints registered with the new gatekeeper (3) shall be reachable via prefix* 4 *or suffix test. The gateway registered at gatekeeper 3 can be reached via the prefix* 8*. This adds the addresses* `941` *and* `942` *or* `1.test.loc2.exampleInc.org` *and* `2.test.loc2.exampleInc.org` *for the new endpoints and* `949` *as a prefix for the new gateway to the address scheme.*

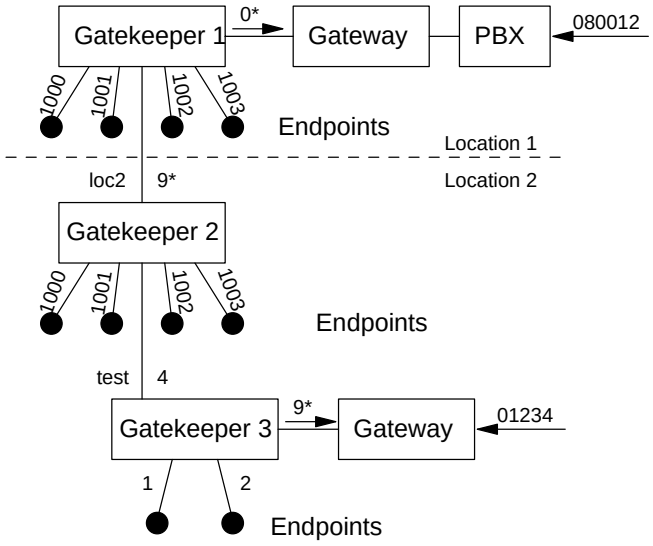

Figure 20 - Example architecture 2

5.3 Distributed administration

For sites with a higher degree of complexity it may be useful to (partly) turn away from the concept of a central administration. For example, the administration of a division of an institution may be done by a local person. In the above example, there may be an administrator for location 1 and one for location 2. One may generalize this to a 1:1 relationship between gatekeeper and administrators.

The relationship between the administrative tasks is the sharing of the responsibility. While the primary administrator is responsible for the whole domain he/she may transfer the responsibility for specific parts of the naming or numbering tree to another administrator. That administrator perhaps delegates a even smaller part to another admin and so on.

Beside the possibility to transfer the administration of naming and numbering subsets to another person it also must be possible for every admin to create accounts that are valid throughout the whole address space. This feature is needed when accounts for users shall be created that can be used everywhere inside the institution. Returning to our example shown in Figure 20 this would enable employees with accounts created at one location 1 to move to location 2 and the other way around while keeping their phone number or name.

The immediate question of how to ensure the misuse of the administrator's ability to modify site-wide valid accounts has to be answered in a non-technical way within each institution. WIPTEL does not come up with a technical solution for this problem although it may be focused again in a later stage of the project.

The system mentioned above highly depends on the support provided by the gatekeepers involved. The gatekeepers that have been tested so far lacked an (documented) mechanism to interconnect different gatekeepers to create a corporate system. Thus the gatekeeper written at TZI will be modified to support the proposed infrastructure. Where possible, interfaces to other gatekeepers shall be provided to allow the possibility of a heterogeneous system. Currently the upcoming protocol TRIP is the only supported interface to distribute routing information that is currently being implemented.

Sites that are small enough (regarding number of users) and do not need the distributed administration may of course use any gatekeeper that supports the standards needed for inter-domain communication.

5.4 Components

Before the details of the institution and site architecture are presented, this section elaborates on the physical and logical components used at the sites.

5.4.1 Terminals

A terminal might be any kind of hardware or software to originate and terminate calls. Usually it would be a phone device on the desktop but it may also be a piece of software running on any kind of operating system. Endpoints differ in the signaling protocols that are used. Possibilities may be ISDN, H.323 or SIP.

Most IP telephony terminals offer the possibility to directly call another endpoint. This enables the user to reach each another under the prerequisite that the caller knows the callee's IP address and that they share the same signaling protocol. To have a more comfortable telephone system where the user doesn't have to care for IP addresses and call signaling protocols the system requires a kind of location server and a signaling protocol gateway.

Terminals that were tested in the WIPTEL project include:

- Siemens HiNet LP 5100 (H.323 and SIP)
- 3com SIP Phone (since July 2000)
- WIPTEL software client (wipone)
- NetMeeting 3.0

- OpenPhone

5.4.2 Gatekeeper

A gatekeeper is a component that manages terminals and other endpoints and offers services like address resolution and resource control. While the services a gatekeeper provides are defined in H.323, the way how this is done (configuration, data storage) differs from vendor to vendor. Amongst gatekeepers from multiple vendors it is unlikely to find two that for example share the same database format.

This is not surprising because apart from H.341 which defines the SNMP MIB for gatekeepers there is no standardized method to configure gatekeeper and describe their facilities. Unfortunately the (recently completed) efforts made to support intercommunication between gatekeepers of different vendors - that would allow to build a heterogeneous gatekeeper infrastructure - have not yet been widely integrated into commercial gatekeeper products.

5.4.3 Databases

In the proposed architecture, databases serve as a storage unit for different kinds of information to be kept about a gatekeeper's zone. The kind of data stored depends on the gatekeeper using it.

It is difficult to provide a common configuration database for a (potentially) heterogeneous environment because each vendor provides his gatekeeper with its own database solution. For example the Gatekeeper HiNet RC 3000 from Siemens requires a Microsoft SQL-Database or an Microsoft Access Database (for less than 11 users) to store the configuration data. Information about the structure of this database is not available.

The gatekeeper of the University of Bremen theoretically works with any kind of SQL-Server but was only tested against the MySQL database.

5.4.4 Gateways

A gateway is a special endpoint that links the IP telephony network with the PSTN or bridges between H.323 signaling and SIP signaling. For interworking between SIP and H.323, at least two gateways are under development (one developed by the TZI in the MECCANO project of the European commission, the other developed at Columbia University in New York) that are only available in early beta stage. Because of this, the WIPTEL approach concentrates on IP/PSTN gateways.

A gateway may be a piece of software running on a standard PC with an ISDN board or a (often rack mountable) piece of hardware with PRI or BRI ports. The proposed approach is indifferent to what make of gateway is used - as long as the gateway uses H.323 and supports the standardized ways of communication with an external gatekeeper.

5.4.5 Media Server

The media server is a special kind of endpoint that operates as an answering machine (or, in more complex scenarios, as a full-featured IVR system). The gatekeeper routes calls for known but unregistered users to a media server which terminates the call presenting the caller a prerecorded voice message. Two ways for interactively retrieving voice messages in a standardized fashion are conceivable:

- Enable the media server to accept standard H.323 calls on one or more dedicated "phone numbers" and use DTMF for gaining access to and manipulating (e.g. deleting) messages as well as recording announcements.
- Make use of the Real-Time Streaming Protocol (RTSP) as defined in RFC 2326 for handling voice messages as well as announcements.

In addition, an email gateway (or complete integrated messaging solutions) could interface to a media server for further processing of incoming voice mail. Besides email, this is beyond the scope of the WIPTEL project.

There can be multiple media servers within a domain. Each gatekeeper is configured with a media server it should use. The issue of how a shared pool of users can use multiple media servers is yet to be defined.

5.5 Intra-institution gatekeeper communication

As mentioned in section 5.3, the gatekeepers of a domain need a protocol to exchange routing and database information.

The requirements for such a protocol include several kinds of synchronization that shall be explained using an example site with 3 gatekeepers (1 primary gatekeeper with 2 slaves) where each gatekeeper has its own database.

- **Routing information**: Each gatekeeper tells its master gatekeeper which prefixes or suffixes it supports. On the other hand a gatekeeper receiving updated routing information (either from a slave or its master gatekeeper) shares this information with its master and slave gatekeepers. The result is that every gatekeeper of the domain knows which gatekeeper to contact to reach a target.

  The protocol needs to support the distribution of such routing information.

- **User data**: Creating, modifying or deleting a user account should be possible from every gatekeeper in the domain. When such changes are made, the gatekeeper of the administrators zone has to make sure that every other gatekeeper in the domain receives these changes. To achieve this, the primary gatekeeper's database acts as an authoritative database. Before a change is valid, a gatekeeper has to inform the primary gatekeeper of the change and wait for a confirmation. Then a slave gatekeeper informs all its peers about the changes. The peers themselves inform their peer gatekeepers until the changes are spread all over the zone.

  It might be useful to have some kind of caching for user data within every gatekeeper to help keeping the gatekeeper local database up to date. This is an item for further study.

  The protocol must offer the possibility to directly communicate with the primary gatekeeper to add, modify or delete user data and a mechanism to inform all gatekeepers of the zone of the changed data.

- **Authentication**: To ensure that a message really comes from a peer or the primary gatekeeper of public and private keys in conjunction with some kind of directory service could be used. Although at the moment authentication isn't used in WIPTEL this is recommended to use it when rolling it out to the WiN.

Until now no such protocol meets this requirements. As an interim solution gatekeepers are expected to exchange their routing information using TRIP to provide at least basic call routing functionality. The TZI already started work at a protocol to provide full functionality and will continue it after WIPTEL already ended.

5.6 Dealing with heterogeneous gatekeepers

In a system where different gatekeepers come into operation the distributed administration mechanism will not work because of the lack of a uniform protocol for site administration. There is a chance to get routing to work if all involved gatekeepers support TRIP or H.225.0 Annex G. But due to the lack of such gatekeepers for evaluation this could not be tested. The only possibility left is sending Location Request PDUs (LRQ) to peer gatekeepers to resolve a single address. Most gatekeepers support answering LRQs but not all also support sending LRQs or allow answering LRQs from gatekeepers outside their zone. A guaranteed solution for finding targets in a heterogeneous system does not exist at the moment.

Because there is no easy way of exchanging user data in a heterogeneous gatekeeper system administrators must either inform each other of changed data and modify their local database manually or simply abandon the option of a shared address space.

As a conclusion one could say that a heterogeneous gatekeeper system has only limited possibilities and cannot be recommended at the moment. From the commercial gatekeeper that were available for tests no gatekeeper supported distributed administration like that proposed in WIPTEL. This does not exclude the possibility that there are solutions that do support this feature and that support intercommunication with gatekeeper of other vendors.

For the moment the gatekeeper of the University of Bremen is the only product known to us that matches the needs of WiN institutions and is designed to operate with gatekeepers from other vendors.

5.7 Naming and Numbering scheme

This chapter illustrates how the proposed system can be customized for an example using the University of Bremen as an example.

5.7.1 Step 1: Determining the infrastructure of the domain

The proposed system offers the possibility to reflect the organizational or topographical structures within the naming scheme or numbering plan. For the University of Bremen all possible targets are classified as functions (organizational), rooms (topographical) and persons.

5.7.2 Step 2: Setting up the basic concepts for naming and numbering

First of all, the entry points of the university must be defined. Finding the naming entry point should be simple for an already existing suffix can be used. When selecting the numbering entry point, eventually national rules and procedures must be considered. For this example, assume that the University of Bremen can be reached via "uni-bremen.de" or "0421218" when coming from the IP-side and via `042121866` from the PSTN-side.

After the classes of destinations are determined, the basics of the naming and numbering scheme can be defined. For the example site, the primary requirement was that every user is assigned a unique identifier as a name and number. A secondary requirement is that addressing users should be as short as possible.

Furthermore each faculty shall have the possibility to administrate their phones on their own, so a prefix for each faculty is desirable. There also shall be a possibility to place calls to a particular room in a particular building and even a possibility to call functions (like Chancellor, Student society, or Mensa) and services that are not subordinated to any faculty.

5.7.3 Step 3: Implementing the naming scheme and numbering plan

Now names and numbers must be associated to the destination classes. It is not necessary that every destination class is reachable via a name and a number, so this might look like this:

Destination	**Numbering (`042121866-x`)**	**Naming (`x.uni-bremen.de`)**
User	`2<user-id>`	`<user-name>`
Faculty “Computer Science”		`<whatever>.informatik`
Faculty “Mathematik”		`<whatever>.mathematik`
Faculty “...”	`...`	`...`
Rooms	`3<building><room#>`	`<building><roomnr>`
Functions/Services	`1<function>`	`<function>`

Table 1 - Example naming an numbering scheme

When a call to a destination originates from outside the University, the prefix `0421866` or the suffix `.uni-bremen.de` must be appended. Internal calls do not need those prefixes or suffixes so a user may simply be called by her name. If function `12` is reserved for an emergency doctor then dialing `112` could be used for emergency (1=functions, 12=emergency).

5.7.4 Step 4: Administration, Responsibility

Now that the addressing scheme is fixed, decisions have to be made who shall have the possibility to administrate a subset of the address space. Usually large faculties or buildings with many people should have their own gatekeeper and thus the possibility to care for their needs themselves.

Apart from technical questions the way of dealing with costs arising from calls to the PSTN must be decided. Similar to the already existing telephone system someone must be authorized to log all call data to write bills and collect money afterwards. This must be done with regards to laws, especially privacy laws.

The obvious idea is that those people/organizations that are already doing this job for the existing telephone system also care for IP telephony calls.

5.8 Configuring the site infrastructure

When using the TZI/DMN gatekeeper the network of the sites gatekeeper defining the site infrastructure is built up using the [PLAN]-section in the gatekeepers configuration file (see section 6.1.3). This section contains entries in the format

```
node.<nr> = desc=<info-text>, [ip=<gatekeeper-ip>,]
            [reachVia=<reachVia-String>,]+ [type=[INTERN|EXTERN_IP
             |EXTERN_ISDN|GRP<x>], [cost=[USER|COMMON]]
```

`desc`	Informative text describing the node.
`ip`	If the node is a subordinate gatekeeper this value describes the IP address of that gatekeeper(optional).
`reachVia`	The way this node can be reached. The value must be in the format `<parent-number>/<parent-name>/branch-number/branch-name` There can be multiple occurrences of this parameter.
`type`	If a call should leave the local domain this describes the way this should happen: EXTERN_IP describes an IP call while EXTERN_ISDN describes an PSTN call. The default value for this type is INTERN. Another possible kind of value starts with GRP and is followed by a number that describes a group that is allowed to use the node. E.g. GRP12 defines a the group number 12.
`cost`	This field declares who has to pay for the cost inflicted by this call. This may be either USER if the caller has to pay or COMMON if the institution bears the cost itself.

When looking at the site infrastructure as a tree where each node may be, but need not to be, a gatekeeper, the entries define a node by the way to reach it. Unlike a simple tree structure each node may have several parent nodes. Each parent knows an own way to reach the node so the name and the number of the node depend on the branch used to reach it.

Example: *Fictional PLAN-section*

```
[PLAN]
node.1= \
  desc     =Services, \
  ip       =134.102.201.1, \
  reachVia =root/root/0/service
node.2= \
  desc     =BE - offiziell - ISDN, \
  reachVia =0/service/1/ , \
  ip       =134.102.218.71, \
  type     =EXTERN_ISDN, \
  cost     =COMMON
node.3= \
  desc     =BE - offiziell - IP , \
  reachVia =0/service/2/ , \
  type     =EXTERN_IP, \
  cost     =COMMON
node.4= \
  desc     =BE - privat - ISDN, \
  ip       =134.102.218.71, \
  reachVia =0/service/3/dialout2 , \
  type     =EXTERN_ISDN, \
  cost     =USER
node.5= \
  desc     =BE - privat - IP, \
  reachVia =0/service/4/dialout2 , \
  type     =EXTERN_IP, \
  cost     =USER
node.6= \
  desc     =Funktionen, \
  reachVia =root/root/1/func, \
  type     =GRP3

node.7= \
  desc     =Räume, \
  reachVia =root/root/2/, \
  type     =GRP2

node.8= \
  desc     =MZH, \
  reachVia =2//01/mzh, \
  ip       =134.102.218.71

node.9= \
  desc     =Personen, \
  reachVia =root/root/3/, \
  type     =GRP1

node.10= \
  desc     =FB 3 Informatik, \
  ip       =134.102.218.71, \
  reachVia =1/uni-bremen.de/03/informatik
```

Table 2 - Numbering plan and naming scheme configuration example

5.8.1 Managing user accounts

The recommended way to manage user accounts is using the gatekeeper client (see Figure 20). The client allows to create, modify or delete a user's account. It does not talk to the database server directly but to its gatekeeper. The gatekeeper decides which database server must be contacted.

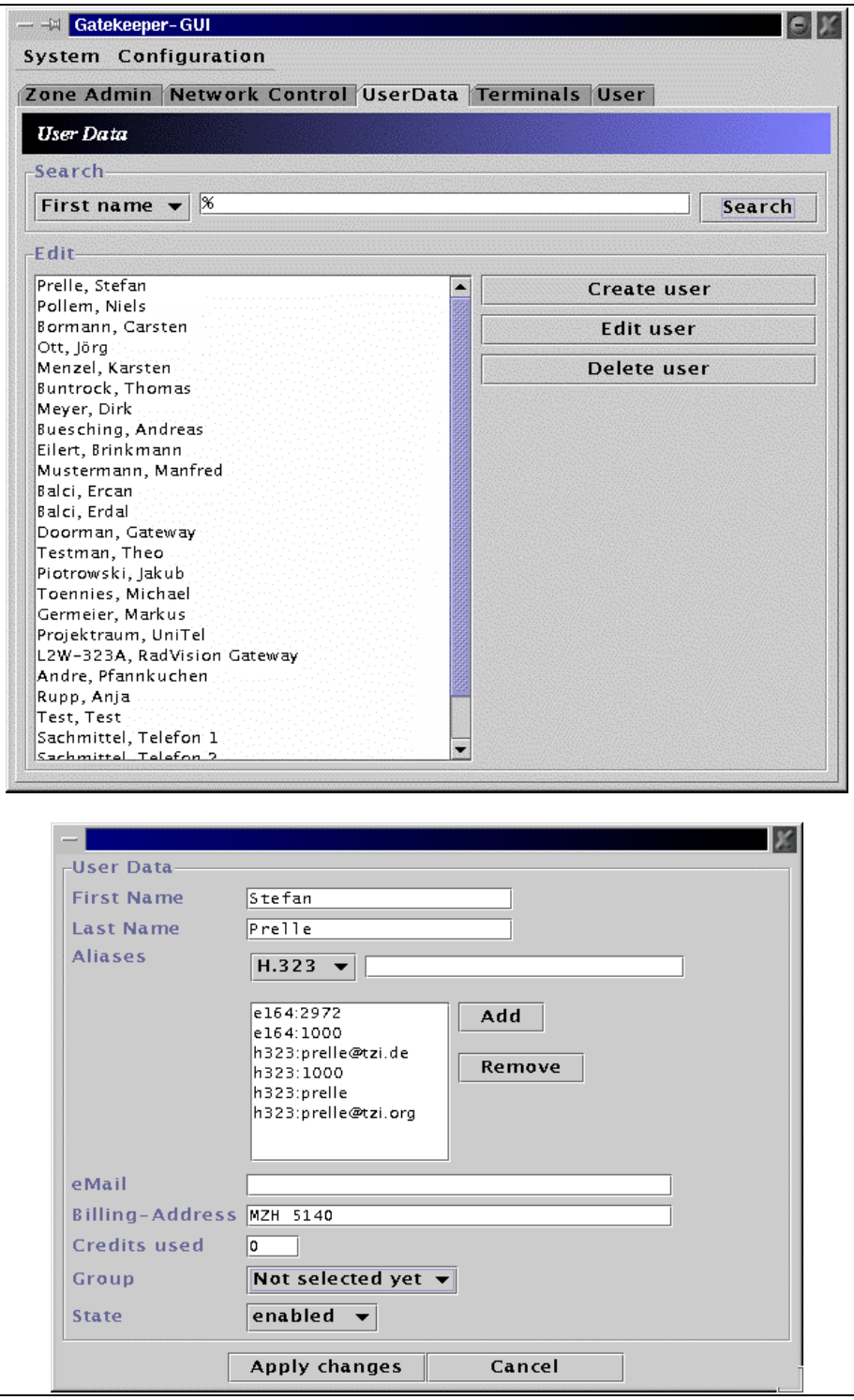

Figure 21 - Administration tool - Managing user accounts

6 IP-Telephony Software developed by TZI/DMN

While it wasn't originally intended at the present scale, the software components enhanced and/or developed in parallel with the WIPTEL project at TZI/DMN have now come to be an important part of the overall WIPTEL VoIP (management) architecture and are therefore presented in this chapter.

It should be noted, though, that work done on the software components listed in this chapter has never been carried out as a part of WIPTEL itself, but instead rather independant of it by TZI/DMN staff.

Amongst the software components that have been provided to the initial peers, and likely also to those to follow, are the TZI/DMN gatekeeper modified for WIPTEL, a likewise modified ISDN-H.323 (and soon also SIP) call signaling and media stream gateway and a H.323 software client for Linux.

Just as intended for the overall architecture, the initial test phase has already and will allow for a further verification and enhancement of the provided software components. While they have been tested within the local test scenario(s) at TZI/DMN in Bremen to our overall satisfaction, issues regarding the distributed nature of the VoIP testbed that has been implemented have already arisen and may continue to arise in the next testing stages.

Development work at TZI/DMN will continue in parallel to ensure/provide the core management (and possibly other) functionality required for a successful as well as timely deployment of VoIP at those WIN locations wishing to participate, as laid out in the original project proposal.

The software packages described in this chapter all make use (at least in part) of the fundamental protocol stacks and mechanisms that have been developed and/or implemented by TZI/DMN: The Message Bus (Mbus) and our H.323 and SIP stacks.

The Mbus has been designed as a basis for interaction between software components with the aim of enabling construction of complex multimedia conferencing systems out of simple (stand-alone) components and simplifying coordination of these various system modules.

The Mbus constitutes a local (intra-system) communication infrastructure that provides transport layer functionality and addressing schemes — as well as semantics that are currently specified following the needs of conferencing applications. General-purpose Mbus messages and procedures provide system bootstrapping, component failure detection and policy/voting mechanisms, among others. Conferencing-specific Mbus message semantics are defined for establishment and teardown of (multimedia) calls, control of media engines (codecs, protocols), component configuration and interaction with the user.

6.1 H.323 Gatekeeper

When the WIPTEL project started, TZI/DMN already had a simple gatekeeper implementation. When it became evident that getting testing equipment would be very difficult, this gatekeeper was extended to suite the needs of WIPTEL (albeit independent from the project itself) . While there is still some work to do to have a release version, a rather surprising, but nevertheless important "feature" of this gatekeeper became evident: being a H.323-to-H.323 gateway. Endpoints from different vendors often have their own interpretation of how H.323 works (or might, that is). The design of the gatekeeper has been changed to enable vendor specific communication by adding new modules to encode or decode messages from certain vendors. This hopefully qualifies this gatekeeper to operate in heterogeneous environments.

The work on this gatekeeper isn't finished yet and doubtfully won't be finished in the near future, because of the still evolving H.323 standard. The TZI is interested in enhancing the gatekeeper – for the own needs of the university, but also for other interested institutions. At this moment, anyone interested can use the gatekeeper, but should be aware that it is clearly still a "work in progress" version with some "loose ends". The next sections are dedicated to the installation and configuration of the current version of the gatekeeper.

6.1.1 Requirements

To run the gatekeeper one needs:

- JAVA 2 / 1.2 or higher,
- a dedicated server with at least 128 MB of RAM; if supported by the used JAVA version, the gatekeeper would benefit from a multiprocessor system, and
- a MySQL-Server; other SQL databases might work too (given a suitable JDBC driver), but the gatekeeper has only been tested against MySQL with the included JDBC driver for now.

There have been no load tests for the gatekeeper yet. A first guess would be that it should work rather well for zones with up to about 100 endpoints.

6.1.2 Installation

The first step is the installation of the MySQL server. For there are detailed instructions along with the MySQL package, this isn't explained in detail here.

The next step is setting up the database. Following the instructions for MySQL, an account must be set up and a database created. Then the ASCII file in the gatekeeper package (`mysql.dump`) can be used to reinstate the necessary structure of the database.

After the installation and configuration of MySQL is finished, the gatekeeper files can be copied to any directory. The start script that comes along with the package must be edited to contain the chosen directory path, though.

6.1.3 Configuration

The gatekeeper uses two configuration files. `modules.conf` and `gatekeeper.conf`. These are explained in this section. The first contains the list of modules that should be loaded and the latter keeps the configuration of those modules. While `modules.conf` usually shouldn't be changed and therefore isn't described here the understanding of the other configuration file is important.

The file `gatekeeper.conf` must be located in the path that is given in the –d option in the gatekeeper startup script. It contains several sections with key/value pairs. These are explained now.

GENERAL section

```
[GENERAL]
ip.1= 134.102.218.71
product.name   =WIPTEL Gatekeeper
product.version=1.1
```

- **ip.<x>** : The gatekeeper can be limited to one or more IP addresses, if running on a host with multiple interfaces.
- **product.id** : The product name for the corresponding field in the H.221 vendor identifier.
- **product.version** : The product version for the corresponding field in the H.221 vendor identifier.

CODER section

```
[CODER]
codingSet.1=org.wiptel.coder.DefaultVersion2CodingSet
codingSet.2=org.wiptel.coder.NetMeeting3CodingSet
codingSet.3=org.wiptel.coder.DefaultVersion1CodingSet
```

This section tells the gatekeeper which coding sets it shall support. Coding sets might be added if special flavors of encoders or decoders are necessary. The example above is taken from a current configuration and tells the gatekeeper to support Microsoft's interpretation of ASN.1.

CALL SIGNALING section

```
[CALL SIGNALING]
port    =1720
interval=60
threads =2
```

This section controls the call signaling module of the gatekeeper. The port 1720 is the default port for H.225 call signaling connections. The keys `interval` and `threads` tell the gatekeeper how many threads should process incoming messages and at which interval (in milliseconds) connections should be checked for new messages.

RAS section

```
[RAS]
port.mc =1718
port.uni=1719
interval=60
threads =1
ttl     =7
```

Similar to the CALL SIGNALING section this section configures the RAS module of the gatekeeper. The `port` keys allow the configuration of the multicast and the unicast socket, `threads` and `interval` control how many thread how often (in ms) check for new data. The `ttl` key tells the gatekeeper which TTL is to be used for multicasted requests.

REGISTRATION section

```
[REGISTRATION]
registration.ttl = 3
registration.interval = 60
registration.ageGW = false
```

In this section, the behavior of how the gatekeeper treats registrations can be controlled. Usually, an endpoint's registration does not last forever. The `registration.ttl` key tells the gatekeeper how many minutes a registration should last, the `registration.interval` describes the granularity (in seconds) in which to check for the expiry of an registration.

To increase compatibility with the gateways currently on the market, it is possible to switch off the expiration of registrations from gateways. This is done using the `registration.ageGW` key.

DATABASE section

```
[DATABASE]
dbase.host=134.102.218.77
dbase.port=3306
dbase.name=gkng
dbase.JDBCProvider=org.gjt.mm.mysql.Driver
```

```
dbase.login=gatekeeper
dbase.passwd=PassWord
```

Modify this section to enable the gatekeeper to connect to your own SQL server. Except for `dbase.JDBCProvider`, which shouldn't be changed when using MySQL, all other values should be derived from your database installation.

RASCTRL section

```
[RASCTRL]
gkID            =WIPTEL Gatekeeper 1.0 (prelle@tzi.de)
irrInterval     =120
discoveryNeeded =false
```

This section controls some configurations of the gatekeepers RAS channel. `gkID` is the identifier the gatekeeper shall use to identify itself to other endpoints. The `irrInterval` tells the gatekeeper in which interval (in seconds) he wants the endpoints that are in a call to send IRR messages. The key `discoveryNeeded` controls if the gatekeeper allows registrations from endpoints that did not send a discovery request before.

RESOVLER section

```
[RESOLVER]
zone.country=49
zone.area   =421
zone.name   =uni-bremen.de
dialout.national=0
dialout.internat=00
```

This is one of the most important configuration sections, for it influences the numbering plan and naming scheme of the gatekeeper's domain. The `dialout` keys tell the gatekeeper which prefixes are used for external calls. The example suits a german configuration (dial 00 for international calls or 0 for national ones). The zone keys indicate the gatekeeper's location within the overall numbering plan/naming scheme.

So, the example above would configure that the gatekeeper is responsible for all numbers starting with `[0049|0]`421 or names ending with `uni-bremen.de`.

PLAN section

This section is already described in this document in 5.8.

TRIP section

```
[TRIP]
int.holdtime    =60
int.peer.1=rasen.informatik.uni-bremen.de
int.peer.2=bse.informatik.uni-bremen.de
ext.peer.1=elmo.rvs.uni-hannover.de
itad = 2
```

This section configures the module for the preliminary TRIP implementation. The `int.holdtime` key defines the interval for the KEEPALIVE messages in seconds. The `itad` key sets the ITAD

identifier for the gatekeeper's domain. With `int.peer`, the domain internal TRIP peers are specified – the keys `ext.peer` are analogue for external peers.

DNS section

```
[DNS]
nameserver.1=134.102.218.46
nameserver.2=134.102.200.14
```

This configures the nameservers the gatekeeper should ask to resolve H.323-IDs via SRV records.

ZONEFILERESOLVER section

```
[ZONEFILERESOLVER]
location=/home/prelle/gatekeeper-dutchman/zonefile
```

This section just tells the gatekeeper where to look for the zonefile that provides an interim solution for WiN routing (see section 4.1.4).

GATEWAYS section

```
[GATEWAYS]
gateway.1= \
  ras = radvision.informatik.uni-bremen.de:1024, \
  cs  = radvision.informatik.uni-bremen.de:1820, \
  prefix = 9999 9999 ,\
  area = 00 0 49 421
gateway.2= \
  ras = damn.informatik.uni-bremen.de:1721, \
  cs  = damn.informatik.uni-bremen.de:1720, \
  prefix = 97   , \
  area = 00 0 49 441
```

In this section, the gateways the gatekeeper should use and accept are configured. Each gateway needs to be configured with its RAS and Call Signaling address, optionally prefixes for the dialed number in the DestinationInfo- and the RemoteExtensionAddress field in the ASN.1 PDUs and the area the gateway is local to (prefix for international calls, prefix for national calls, country code and area code).

ACCOUNTING section

```
[ACCOUNTING]
cdrfile=cdr.log
cdr.def= Call {0} to {1} \nStart   : {2}\nEnd     : {3}\nDuration:\
 {4}h {5}min {6}s
```

For now, this section is just used to configure the file for storing call detail records (`cdrfile`) and the format that entries in this file should adhere to (`cdr.def`).

RESOURCES section

```
[RESOURCES]
resMan.maxCalls=30
resMan.maxBandwidth=2560
resMan.minBandwidth=1280
resMan.maxTotalBand=10000
resMan.maxRegistered=100
```

Within this section are the parameters to control the gatekeepers resource manager. There are options for the maximum amount of simultaneous calls, the maximum or minimum bandwidth for a single call, the maximum total bandwidth for all calls and the maximum number of registered endpoints.

6.2 H.323-SIP-ISDN Gateway: StarGate

STARGATE is the H.323-SIP-ISDN gateway developed by TZI/DMN within the MECCANO project headed by UCL, funded by the EU. It has been used by and enhanced in parallel to the WIPTEL project described in this deliverable. STARGATE attaches itself to one or more ISDN interfaces as well as an H.323 gatekeeper to bridge calls between these two zones. SIP support is near completion and has therefore been included into this deliverable in full. STARGATE runs on any "off-the-shelf" PC running Linux that is equipped with an ISDN BRI board and an ISDN basic rate interface (and connected to your LAN, of course).

In general, this call signaling and media transcoding gateway developed by TZI/DMN is supposed to provide connectivity for audio communication between different kinds of endpoints interconnected through different types of networks (hence the wildcard character "*" in its name). This includes in particular:

- Conversion between the three most important call signaling protocols (H.323, SIP, and ISDN resp. their call signaling components) including media stream conversion if necessary;
- Actively accessing Mbone sessions from H.323 endpoints; and
- Inviting H.323 endpoints into Mbone sessions for audio communications.

STARGATE is made up of several independent components implementing the H.323-, SIP-, ISDN- and core gateway-functionality, respectively. These components communicate with each other via the Mbus, a communication bus developed mainly by TZI/DMN. See `http://www.mbus.org/` for a little more on the Mbus, especially the relevant Internet Drafts and the source-packages (C++ and Java; C version developed at UCL).

STARGATE's ISDN engine is based on version 4.0.5 of RAT, the Robust Audio Tool developed at UCL. See `http://www-mice.cs.ucl.ac.uk/multimedia/software/rat/` for the current release. RAT has been made an integral part of STARGATE to easily allow for some of these interconnectivity options listed above.

This section on STARGATE as been interspersed with excerpts from our Mbus documentation (tables of commands and the like). See this deliverable's bibliography for more on the Mbus technology.

6.2.1 Functional Requirements

On the functional level, the setup of a gateway capable of the above implies a set of requirements towards the actual implementation:

- Full compatibility with the standards to be converted. In each case, the gateway should pose like one of the components defined therein. All mandatory mechanisms need to be implemented;

- Transparent address resolving. The caller ought not have to know which protocol “cloud” the callee or even he himself is in at the time of the call;
- Transparent media negotiation including the conversion of suitable media descriptions;
- Setup of RTP/RTCP streams directly between the endpoints; and
- Transparent conversion of different audio (and later also video) coding schemes.

While all media streams are transmitted using the Real-time Transport Protocol (RTP), the problem remains that different media encoding schemes, not understood by both endpoints, might have to be used. The gateway must therefore provide a mechanism to convert between these in a transparent, but still efficient manner.

Another reason for using a transcoding mechanism are the different resource requirements of the encoding schemes. It might sometimes be feasible to transcode between two (or more) of them dynamically, e.g. to save crucial bandwidth in case of too many calls in progress. It should be noted though, that any form of transcoding implies a rather steep increase on the hardware requirements for the machine hosting the gateway.

The address resolving functionality eventually leads to call routing by the gateway and/or the entities providing the address resolving. It therefore seems to be appropriate to tighten the bounds between the call-control entities within the individual protocol clouds, e.g. the H.323 gatekeeper, and the gateway controller itself. This has been put into practice already, integrating STARGATE closely with the H.323 gatekeeper developed by at TZI/DMN.

Any management functionality has been postponed for now, but is likely to be included as STARGATE evolves. This includes least-cost routing as well as keeping (and later invoicing) call detail records.

Still, there is some rudimentary access control, of course, to prevent any misuse, especially of the rather cost-intensive ISDN lines connected to the gateway. A more detailed access control will require a closer link to the individual call-control entities, as described for the call routing functionality above.

The current architecture of STARGATE allows us to easily extend the number of supported call signaling protocols. In addition, as the Mbus itself is shifted from a call-based to a conference-based model, the call-/conference-control functionality noted to be absent above will be added once that shift has fully taken place. And, if feasible from the standardization point of view (i.e. the necessary specifications are all complete and stable), security aspects will be incorporated into the STARGATE implementation as well.

6.2.2 Mbus Architecture and Functionality

This section presents an architectural overview of the entities forming STARGATE, as well as their respective interactions, whereas the subsequent sections take a closer look at each of these fields. A modularized approach permits the re-use of any of the building blocks within other applications based on the Mbus architecture.

Mbus Modules

The conceptual outline of the current STARGATE implementation is shown in Figure 22. The depicted entities perform the following tasks:

- The H.323, SIP, and ISDN modules implement call signaling and (as far as applicable) conference control functions for the respective protocol suite. The ISDN part is assumed by an extended RAT.
- An RAT entity is instantiated whenever transcoding (e.g. for an interconnection to the telephone network) is required.
- The Call Routing module provides address and endpoint reachability resolution and, in particular, decides which protocol to route an incoming call across. This module is dubbed the “Resolver”.

- The Access Control module is used to verify that incoming calls are authorized to be completed according to the reachability decision taken by the Call Routing module (e.g. whether an IP-side caller is allowed to initiate a long-distance call via the telephone network). It module is called the "Approver".
- Finally, the Mbus controller again provides the necessary glue between all the modules forwarding call messages back and forth, keeping per call and resource utilization state, etc. In particular, it knows which control protocol entities are present and is optionally capable to translate non-standard Mbus call-control messages between the various protocols and instantiate/configure the RAT media engine(s) accordingly.
- Further Mbus entities (control applets as well as policy modules) may be introduced to provide additional functionality such as value added services based upon DTMF tones or similar signaling from the IP side.

Please note that not all of the functionality described here has been fully implemented yet, as stated in the introduction to this section.

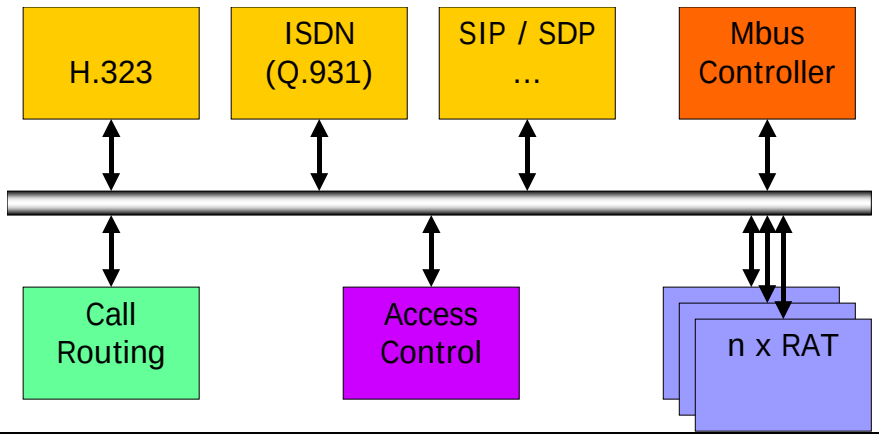

Figure 22 - Outline of StarGate

All three of the aforementioned control protocol entities share a core set of Mbus messages to set up, tear down, and monitor the progress of a call. In addition, each entity supports protocol-specific Mbus extensions that may not be (easily) mapped to other control protocols. The Mbus controller is expected to understand all these Mbus commands, route incoming messages, and optionally perform translation between different protocols. Section 6.2.4 describes the common Mbus commands so far defined for call control.

The order in which the modules are started, re-started or torn down does not matter, as they recognize each other via the Mbus command `hello` issued at frequent intervals.

Generic Call Setup

Figure 23 depicts the generic call setup within/between the STARGATE modules. It can be divided into three phases: The address resolving via the Resolver, the access control by the Approver and finally the core call setup including the capability exchange between the two STARGATE entities representing the protocol clouds of the respective endpoints to be connected. This generic call setup is the same for all protocol suites taken into account so far (and is likely to be so for any additional ones).

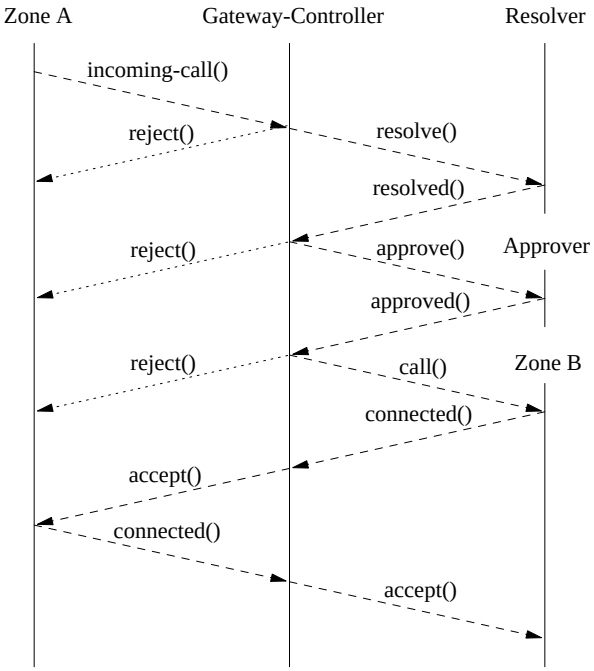

Figure 23 - StarGate generic call setup

A `resolve` command may yield a `complete`, `incomplete` or `unresolved` response, with the latter indicating the Resolver's inability to resolve the address. An `incomplete` will typically occur during two-stage dialing, when the callee's extension is not complete yet. The subsequent query to the Approver may yield an approving or disapproving answer, depending on the parameters supplied.

The STARGATE controller module needs both a Resolver and an Approver to be functional. If one of these cannot be found, then the controller (and thereby STARGATE as a whole) will turn down any `incoming-call` requests. A broken Resolver or Approver entity can be replaced on-the-fly at any time though, as long as it did not hold state for the call in question, which is seldom the case.

Once the Approver has approved the call, a `call` command is issued to the module representing the callee. After contacting the appropriate entities within its protocol cloud, it will either turn down the request or answer with a `connected` command (maybe we should have called it a *message*) that the STARGATE controller will translate into an `accept` to the module representing the caller. The subsequent `connected` and `accept` commands complete the capability exchange and thereby also complete the call setup via the gateway. The gateway then triggers the setup of the media streams directly between the endpoints or any intermediate entities representing them.

Between the initial `call` command sent to the callee and the resulting `accept` or `reject`, there can be an almost arbitrarily long exchange of `ring`, `ringing`, `proceed` and `proceeding` commands indicating that the alerting of the entities and eventually the parties involved is in progress.

The state of the STARGATE controller is maintained on a per-call basis using a generic state-machine class developed at TZI/DMN that is instantiated anew once an `incoming-call` command is received. It is described in more detail in 6.2.6. Figure 25 depicts the current STARGATE state graph.

The relatively small set of Mbus commands described here is already sufficient to map simple calls between H.323 and SIP and can thereby be considered as the "core" set of Mbus commands. Mapping to the ISDN protocol cloud requires a few additional, tool-specific commands, e.g. for the handling of two-stage dialing.

6.2.3 Control Components

While the preceding section focused on the generic call setup coordinated by STARGATE, this section describes the control components within the gateway in more detail.

Mbus Controller

The STARGATE Mbus controller is the central control unit within STARGATE. It is responsible for all of the Mbus command transcoding as well as the control of the other entities within STARGATE. As already noted, it will likely be linked even closer to the entities implementing the management functionality within the individual protocol clouds at a later point in time.

Initially passive — but not stateless — towards the modules representing the diverse protocol suites, the gateway controller only takes an active role once a timeout relating to a command sent out earlier has passed, or any other interrupt-like event has occurred.

Incoming Mbus commands are either directly processed by the controller core or are assigned to their corresponding state-machine, if they pertain to a certain call. Should the command be an `incoming-call`, a new state-machine is instantiated. Before processing the command, its syntax and origin are checked to prevent any internal/parsing or sequence failure.

Call Routing

The gateway module responsible for the address resolving has been dubbed the Resolver. Its primary role also puts it right in the middle of the call routing process, which it heavily influences. (In fact, that functional entity is completely located within the binary constituting the Resolver.)

There are two Mbus commands enabling the STARGATE controller to query the Resolver for an address resolution, described in Table 3 below.

Command name	Description
tool.stargate.resolve	The RESOLVE command is sent to the call routing module by the gateway controller to resolve the address given therein.
tool.stargate.resolved	The RESOLVED command is sent to the gateway controller by the call routing module in response to a previously received RESOLVE command. It will indicate whether the query carried out yielded either a COMPLETE, INCOMPLETE or UNRESOLVED status code as part of its result, with the latter indicating the Resolver's inability to resolve the address. An INCOMPLETE will typically occur during two-stage dialing, when the callee's extension is not complete yet.

Table 3 - Mbus commands for address resolution within StarGate

For now, the Resolver has been implemented in a rather pragmatic way, basically reading lists of the correlated (ranges of) alias/transport addresses from a plain text file, where they are stored in a table-like fashion. This approach is fully sufficient to test the core functionality of the gateway.

While the per-default relaying of incoming address resolving queries to the appropriate entities within the individual protocol clouds will be implemented, the present local-file approach will stay also and even get priority over external answers, thus enabling ad-hoc redirections and similar interventions. Therefore, certain regular expressions will be made available for use in the local resolving table(s) for the next STARGATE release at issue already.

Among the additional functionality that will be placed within the call routing module at a later date are things like checking the endpoint reachability, gateway location and other inter-domain-oriented tasks.

Access Control

The so-called Approver carries out the access control. For now, it primarily provides a rather rudimentary control scheme to prevent any misuse of the STARGATE environment, especially of the cost-intensive ISDN lines connected to it. A more detailed access control will require linking the Approver to the appropriate entities within the protocol suites allowed for by STARGATE, as already

mentioned. The extent to which the Approver has access to user profiles/data and the like is decisive in terms of the possible granularity of the access control — another reason to tightly couple STARGATE with the H.323 gatekeeper developed at TZI/DMN in the first place.

Media Stream Transcoding

The Robust Audio Tool (RAT) developed at UCL is being used to transcode different media streams as appropriate. It supports all commonly used codecs; additional ones can be added easily.

Within STARGATE, an independent instance of RAT is forked for each stream/call to be transcoded to keep things simple. Combined with the computing power needed to perform the task itself, this limits the number of concurrent calls that may involve transcoding. Still, any off-the-shelf PC available today should be capable of handling a large working group's, or even department's, full concurrent media transcoding workload — based on the assumption that most of the calling/called entities support the same common set of (partly mandatory) media encoding schemes.

Call-control Engines

The H.323, SIP, and ISDN call-control engines implement call signaling and (as far as applicable) conference control functions for the respective protocol suite. As already mentioned, the ISDN part is assumed by an extended RAT.

TZI/DMN has also developed SIP and H.323 call-control modules (based on the aforementioned protocol stacks). SIGMA, the "SIP Gateway for Mbus Applications", and a basic H.323 call-control engine.

While STARGATE and SIGMA both use the generic state-machine class developed at TZI/DMN, the H.323 engine as well as the extended RAT resort to a more or less stateless conversion between the relevant protocols.

6.2.4 Mbus Commands for Call Control

Call-control messages are intended for interaction with call-control and invitation protocols such as H.323 and SIP. They are designed to constitute the union of the call-control messaging needed by endpoints, gateways, proxies, multi-point controllers, and gatekeepers. This allows the use of the Message Bus to act as gluing mechanism to create any type of system from roughly the same building blocks.

Mbus call-control messages are based on a common basic message set, defined in Table 5 below, that will be supported by any kind of call-control protocol entity. The basic message set may be augmented by protocol-specific extensions required for protocol specific interactions between a local controller and/or local applications on one side and the respective protocol engine on the other.

A worthwhile possible future extension could be an Mbus command set for the Real-Time Streaming Protocol (RTSP) extensions. However, this is clearly left for further study at TZI/DMN.

Description of Commands

A namespace hierarchy has been defined for the generic and the protocol specific call-control commands:

Command prefix	Description
conf.call-control.	Basic call-control message set
conf.call-control.h323.	Extensions for H.323-specific call-control messages
conf.call-control.sip.	Extensions for SIP-specific call-control messages
conf.call-control.isdn.	Extensions for ISDN-specific call-control messages

Table 4 - Namespace for protocol specific call-control commands

The protocol specific extensions are currently being defined, the basic call-control command set is presented in the following section. H.323, SIP, and ISDN-specific messages are still under study.

Basic Call-control Commands

The basic set of messages is defined to provide the core functionality of initiating a call on one side, accepting or refusing it on the other, and providing progress information as well as allowing termination of the call on either side. All types of call-control engines must support the basic call-control message set.

These messages are exchanged using reliable unicast transmission between some local controlling entity and a call-control engine implementing a call-control or initiation protocol such as H.323 or SIP:

- Outgoing calls may be initiated by any local entity; the call-control engine has to keep track of the initiator of a particular call and return all responses or events relating to this call to this entity – which may be different on a per call basis. If the call-control engine notices that the controlling entity for a particular call has gone (e.g. because the Mbus reliability mechanism indicates non-delivery of a call-control message or a `bye` message was seen from this entity), these messages are forwarded to the local controller. If no local controller is available, the call is terminated.
- Indications about incoming calls are always forwarded to the local controller. If no local controller is present the call-control engine automatically rejects incoming calls.

The basic (or "core") call-control command set contains the commands for establishing a simple (point-to-point) call, along with a few messages dealing with supplementary services. The commands are defined in Table 5 below.

Command name	Description
conf.call-control.call	The CALL command is sent to the call-control engine to make the engine initiate a call to another endpoint using the parameters specified as part of the CALL command.
conf.call-control.disconnect	The DISCONNECT command is sent by the local controller to the call-control engine to indicate that the specified call is to be disconnected. It can also be used by the local controller to inform the call-control engine that a call has already been terminated by out-of-band communication, e.g. a horizontal conference control protocol. In this case a special reason code has to be passed with the command.
conf.call-control.ring	The RING command is sent by the local controller to the call-control engine. RING indicates that the controller is willing to accept the incoming call and is now alerting the user.
conf.call-control.ringing	The RINGING command is sent from the call-control engine to the entity from which it received the corresponding CALL command. RINGING indicates that one or more addresses at the far end were contacted and are now alerting the user.
conf.call-control.connected	The CONNECTED command is sent by the call-control engine to the entity that initiated the call (on the calling side) and to the local controller (on the called side) to indicate that the call was successfully established.
conf.call-control.rejected	The REJECTED command is sent by the call-control engine to the entity that initiated the call (on the calling side) and to the local controller (on the called side) to indicate that the call was rejected.
conf.call-control.disconnected	The DISCONNECTED command is sent by the call-control engine to the entity that initiated the call (on the calling side) and to the local

	controller (on the called side) to indicate that the call was disconnected.
conf.call-control.incoming-call	The INCOMING-CALL command is sent by the call-control engine to the local controller to indicate a call request from another endpoint.
conf.call-control.proceed	The PROCEED command is sent by the local controller to a call-control engine in order to indicate that the call that has been signalled with an INCOMING-CALL command is still proceeding.
conf.call-control.proceeding	The PROCEEDING command is sent by a call-control engine to a local controller in order to indicate that the call that has been initiated with a CALL command is still proceeding.
conf.call-control.accept	An ACCEPT command is sent by the local controller to the call-control engine that has indicated an INCOMING-CALL to indicate acceptance of the call.
conf.call-control.reject	A REJECT command is sent by the local controller to the call-control engine that has indicated an INCOMING-CALL to indicate rejection of the call.
conf.call-control.redirect	The REDIRECT command is sent by the local controller to the call-control engine to indicate that the specified call is to be redirected to another specified address. The third parameter determines whether the call-control engine should perform a passive redirection (by telling the caller the redirected address) or an active redirection by operating as a proxy.
conf.call-control.redirected	The REDIRECTED command is sent by a call-control engine to the local controller to indicate that the specified call has been redirected to the specified address.
conf.call-control.forward	The FORWARD command is sent by the local controller to the call-control engine to indicate that the specified call is to be forwarded to another (optionally specified) address. The FORWARD command can be used instead of REDIRECT when the end system acts as a firewall that decides which calls are to be forwarded. The forwarding can either happen with the call-control protocol's implicit semantics (e.g. SIP forwarding) or the controller can explicitly specify the forwarding address.
conf.call-control.forwarded	The FORWARDED command is sent by the call-control engine to the local controller to indicate that the specified call has been forwarded to the specified address. The local controller can decide whether the call setup should continue or be interrupted (by sending a DISCONNECT command).
conf.call-control.relayed	The RELAYED command is sent by the local controller to the call-control engine to indicate that the specified incoming call is being forwarded to the specified address via another call-control engine.
conf.call-control.caps-indicate	CAPS-INDICATE is sent be a call-control engine to a local controller upon the reception of an analogous indication from another party in its specific call-control protocol. This command can be used to initiate a capability negotiation process during or before a call and should be answered with a CAPS-SET command.
conf.call-control.caps-set	CAPS-SET is sent by a local controller to a call-control engine in order to set the media handling capabilities of the concerning endpoint. It can be used on a per call basis.

Table 5 - Basic Mbus call-control command set

Status

The basic call-control functionality has been implemented in the current STARGATE release. As this was the current release's focus, call forwarding and some other mechanisms have not been implemented yet. Still, analogous to our considerations regarding inter-domain operations, the required interfaces are already in place.

6.2.5 Call Scenarios / Call Flows

This section describes call scenarios resp. call flows in greater detail, using two examples to do so. As the general scenarios that suggest the use of a gateway are well known (see Figure 24), the attention is turned towards the internal sequence of events/commands. The first example is the almost tedious two-stage dialing, the second one the capability exchange, which is covered in section 6.2.7.

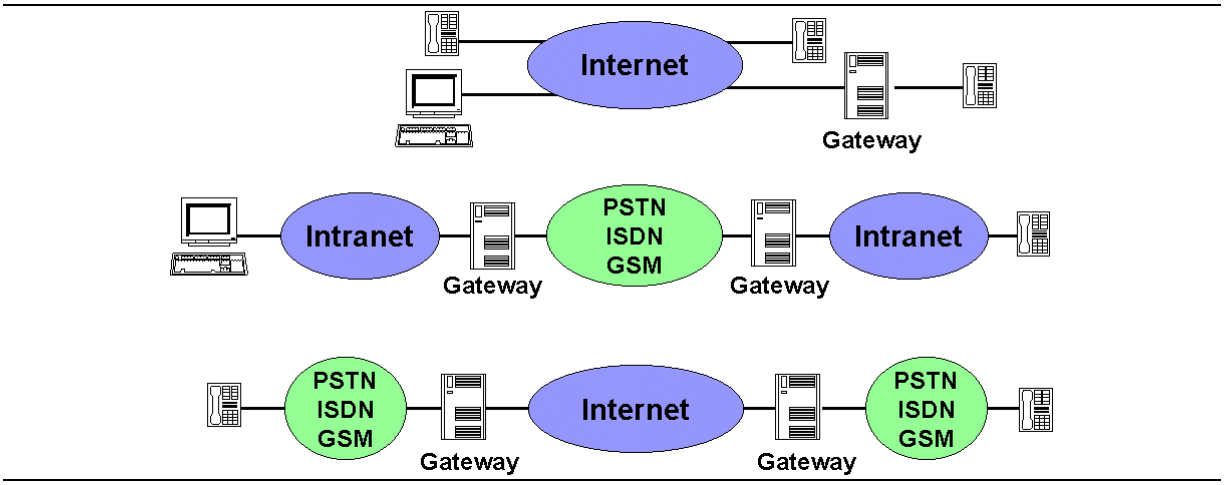

Figure 24- Generic call scenarios including a gateway

As a helpful overview, Figure 25 depicts the inner workings of STARGATE and the relations between the entities of which it is built. The modules themselves are described in more detail in sections 6.2.2 and 6.2.3 of this deliverable.

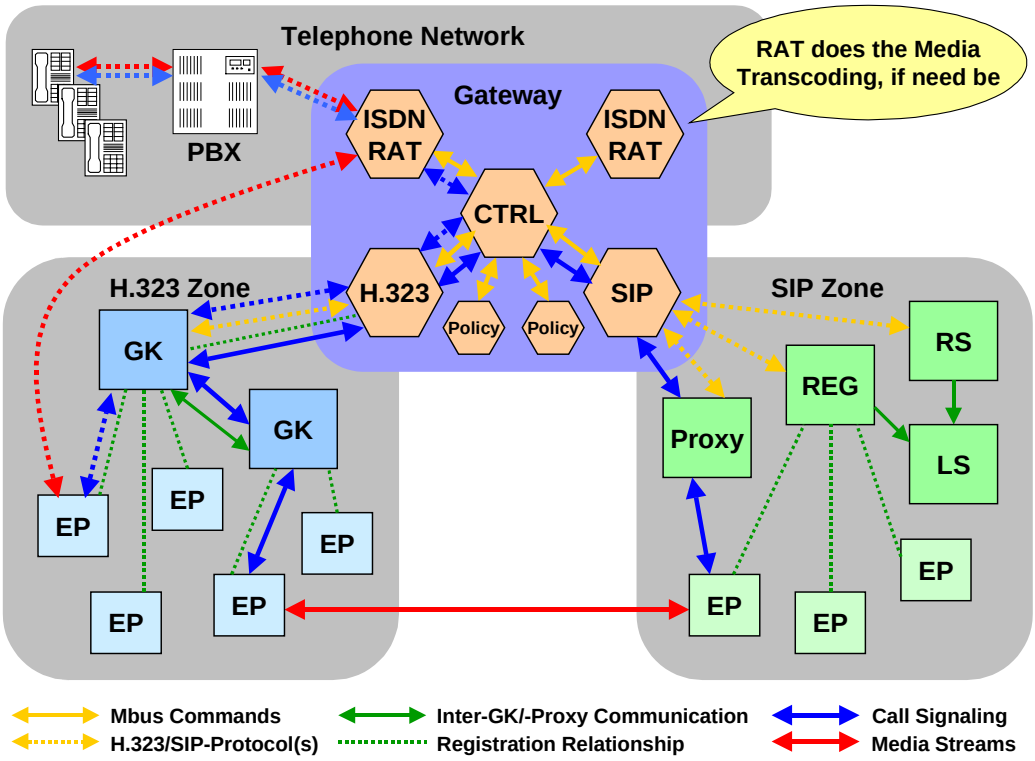

Figure 25 - The communication paths within StarGate

The figure shows two separate calls/conferences in progress. The dotted blue and red arrows indicate the communication paths for an H.323-ISDN call. The other call is an H.323-SIP one (note: without the optional media transcoding in this case) initiated/routed via the (neighboring) gatekeeper the gateway is registered with.

Example: Two-Stage Dialing

An `incoming-call` command initially sent to the StarGate controller module usually includes the alias/phone number of the desired callee. When not including an extension, but only the ambiguous access number, the subsequent `resolve` command will lead to the Resolver responding with an `incomplete` status in its `resolved` command, because although the access number is listed in its internal tables, it still needs the extension of the callee to be able to distinguish between the relevant entries and ultimately resolve the query.

This leads to a follow-up problem: Although it is possible to supply additional digits/characters via DTMF tones, this cannot be done unless the ISDN connection is accepted/opened. This forces the gateway controller to accept the incoming call, even though it couldn't perform any endpoint reachability check yet. It sends an `accept` command to the extended RAT instance representing the caller and waits for the corresponding `connected` command. This is illustrated in Figure 26.

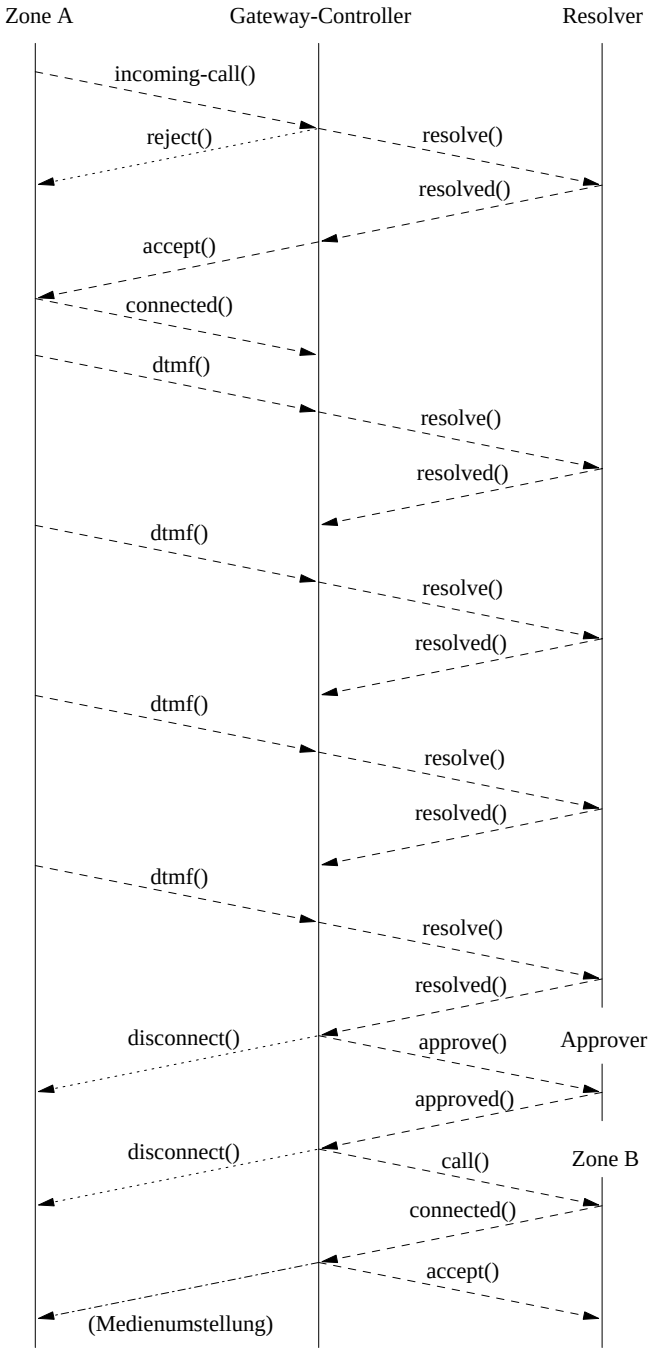

Figure 26 - StarGate call setup including two-stage dialing

The extended RAT now sends `tool.audiogate.dtmf` commands to the STARGATE controller, as the calling party enters additional digits one by one. The controller hands the concatenated address/phone number to the Resolver to re-try resolving it, until it receives either a `complete` or an `unresolved` status with the `resolved` command. The example depicted is obviously based on 4-digit extensions.

Taking two-stage dialing into account leads to the state-graph that can be seen in Figure 27. The first `incomplete` received shifts the corresponding state-machine into the INCOMPLETE state, whereas all further ones keep it at EXTENSION, which itself has been triggered by the `connected` command issued by the extended RAT representing the caller in the meantime.

After a definitive `resolved` command the call setup continues as expected, with the exception that the connection to the calling side is already open. Should there be any reason not to accept the call, the gateway controller will have to send a `disconnect` rather than a `reject` command (as the call

fan-out resulting from an `unresolved` is not fully carried out yet, the receipt of an `unresolved` status is such a case).

Building on RAT's Legacy

The fact that the module handling the ISDN connections is based on RAT with its support for various audio encoding schemes and the appertaining transcoding capabilities paves the way towards quite a few very useful extensions to STARGATE's core functionality. These range from the simple replay of arbitrary sound files to rather complex menu structures when combining RAT's functionality with the generic state-machine class used within STARGATE.

6.2.6 Technical Details and Configuration

Although the description of the transcoding functionality supplied by the gateway as a whole should be considered of most interest, this section presents more detail on the inner workings of STARGATE.

Call State-Machine

The two main data structures within the STARGATE controller are the list of active calls and the list of the entities "on" the Mbus. While both have already been mentioned, albeit on a rather abstract level, this subsection takes a closer look a the generic state-machine class that is instantiated anew for each incoming call, because the list of active calls is made up of instances of that class.

Having to carry the state of the call it represents, the class has one member function for each of the states in which a call can be (see Figure 27). These functions, along with all possible state shifts (each as an ordered pair of two states), are registered with the state-machine itself as the call is created — making it possible to even have calls with different state graphs managed by the same gateway controller.

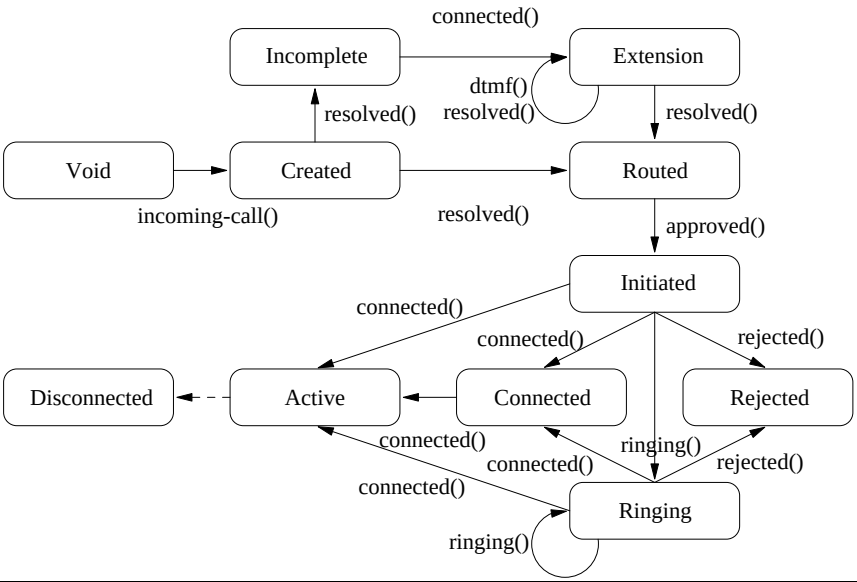

Figure 27 - StarGate state graph including two-stage dialing

When an Mbus message arrives at the controller, it is split up into the commands it carries at the Mbus layer. Those are checked for a call reference. If one is found, the command is handed over to the corresponding state-machine (or a new one is instantiated). Otherwise, the controller itself takes care of the command — mostly coordination issues with other entities within STARGATE. The dispatcher within the state-machine now checks for the state the call is currently in and "offers" the command to all functions representing one of the possible follow-up states. These functions usually perform an extended syntax check before processing the command. Should this go as anticipated, the function will return a positive result, meaning that a shift into the state it represents may take place.

Should all the member functions in question, on the contrary, “turn down” the command, either for syntax reasons or simply because processing it did not yield the expected result, a default function will catch it and determine if the corresponding calls needs to be disconnected or any other action taken.

System Requirements

Our hardware requirements for the machine hosting STARGATE were that it should be possible to plug in multiple ISDN lines and that it could handle the burden of the media transcoding required by a large working group or even a department (see also section 6.2.1). These can both be met by deploying any off-the-shelf PC available today and fitting it with a standard ISDN card. Cards with up to four connectors are readily available and the number of channels supported per card is growing steadily.

Using standard hardware also coincides with the desire to offer a solution both affordable and easy to set up even for smaller facilities, like a university working group wanting to “try things out” or be the local early-adopter.This blends remarkably well into the current and the forthcoming WIPTEL phases.

The operating system of choice for STARGATE is Linux, for several reasons. First of all, the extended RAT handling the ISDN connections is based on a specific ISDN driver that requires Linux. But even with this constraint lifted, we would still use Linux, both because as a UNIX derivative, its structures and mechanisms are well-understood in the academic field and because it can provide the overall system stability needed to run a service like an IP-telephony gateway/gatekeeper.

The machine hosting STARGATE at TZI/DMN is a 400 MHz Pentium II with 256 MB of memory and an off-the-shelf, one-socket ISDN board (STARGATE has been developed using a Teles S0 16.3 ISDN board) running a custom RedHat Linux 6.2 with Kernel 2.2.16 und HiSax 3.3. Scaling down the transcoding requirements slightly would even make it possible to co-locate STARGATE on an already existing machine, making the ISDN board the only part that has to be bought.

The ISDN board is connected to our Teles iPBX which in turn is connected to some ISDN lines. The primary H.323-cloud that was used for testing STARGATE at TZI/DMN is made up of our own H.323 gatekeeper (see above) and a couple of Siemens HiNet LP 5100 H.323-phones.

Configuration

The configuration of STARGATE is completely (and only) done by editing the `gateway` startup script and the `stargate.conf` file found in the distribution/installion tree. The required instructions for configuring the `gateway` script are provided therein. `stargate.conf` file provides the following options, that are listed in detail for illustration purposes.

Option	Description	Default
P_NATIONAL	prefix for national calls	0
P_INTERNATIONAL	prefix for international calls	00
P_SOLAR_SYSTEM	prefix for calls within the solar system	000
P_INTERGALACTIC	prefix for intergalactic calls	0000
ALLOWED	one of {PBX,LOCAL,NATIONAL,INTERNATIONAL}	NATIONAL
GW_AREACODE	prefix of region we're located in	*0421
GW_NUMBER	prefix-less phone number of the gateway	*2017023
PREFIX_GK_GW	add to number given to gatekeeper	*02
PREFIX_PBX_GW	strip from number given by PBX (ISDN)	*99
PREFIX_GW_PBX	dial to get out of PBX numbering plan	*98
PBX_ZONE	list of numbers within the pbx	*(2017028, ...)
TRANSPORT_TABLE	file name of transport-address table	*/etc/stargate.tab
GATEKEEPER	preferred H.323-gatekeeper:port	*dutchman:...:1719

The local administrator at the site using STARGATE will most likely change the options marked with a "*", since the defaults simply reflect our local installation. The adapted `stargate.conf` might then look somewhat like this:

```
[STARGATE]
ALLOWED=NATIONAL
GW_AREACODE=0421
GW_NUMBER=2017023
PREFIX_GK_GW=03
PREFIX_PBX_GW=99
PREFIX_GW_PBX=98
PBX_ZONE=(2017023,2017028,2234314,2239618)
GATEKEEPER=dutchman.informatik.uni-bremen.de:1719
```

6.2.7 Inclusion of Media Engines for Transcoding

With media transcoding being one of the tasks the gateway was built to perform, and the area of media capability description and exchange being handled significantly differently by the major protocol suites, a larger portion of STARGATE's development effort was targeted at this area. While not all of the details have been polished to sparkle yet, a fundamental system is in place which, amongst other things, prevents unnecessary resource-intensive media transcoding while never needlessly turning down an incoming call. This is described in section 5.2.7.2 below.

Capability Description

The capability descriptions currently use the Session Description Protocol (SDP). While this is useful to a certain extent, the protocol also has some severe shortcomings, suggesting a new or reworked protocol is needed for this purpose. TZI/DMN has already begun investigating possible enhancements and will continue to do so in the near future to help to swiftly fill the current gap. With SDPng, that is.

SDPng is the short name for a successor of the Session Description Protocol (SDP, RFC 2327) developed by the MMUSIC WG of the IETF. SDP has originally been designed for announcements of Mbone multimedia conferences. Its use has been expanded to include media descriptions for audio/video streaming (retrieval) controlled by RTSP (RFC 2326), media descriptions for SIP-initiated (RFC 2543) multimedia sessions and particularly IP telephone calls, and media/session descriptions exchanged between Media Gateways and Media Gateway Controllers as specified in the MEGACO WG of the IETF.

As SDP was developed for a different purpose, it falls short in meeting a number of requirements that came up with these new uses. Numerous work-arounds, sometimes heavily bending SDP syntax and semantics, have been created. Additional requirements came from the work on media packetization formats in the AVT WG of the IETF: more complex descriptions for codecs, codec parameters, and packetization formats were needed in various places. And, finally, support for multiparty negotiation of capabilities is found useful.

These and other requirements have led to the idea to develop a successor to SDP — SDPng — that is capable to provide the necessary semantical expressiveness, syntactical structure, and prcoedures yet simple enough to allow easy implementation. The development of SDPng is undertaken in the MMUSIC WG of the IETF, lead my TZI/DMN and will address the perceived shortcomings of SDP.

Capability Exchange

For now, the capability exchange is an integral part of the call signaling process. The relevant Mbus commands have been assigned the relevant parameters to attach the media capabilities to them. In a

somewhat SIP-like fashion, these are then narrowed down to what is eventually chosen, after taking account of the transcoding capabilities of the gateway in between.

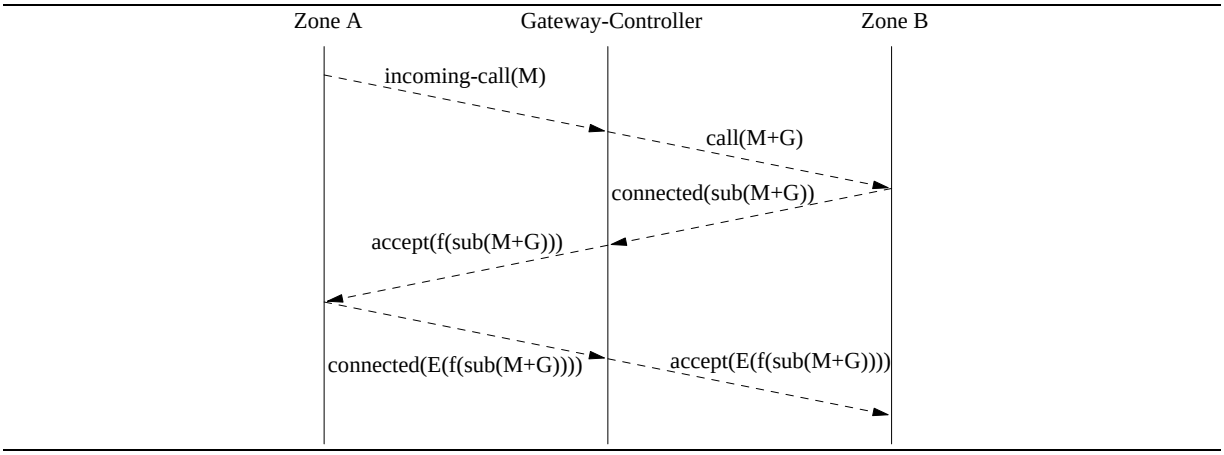

Figure 28 - Capability exchange taking transcoding into consideration

The call-control engine representing the caller communicates the set M of audio encoding schemes it is capable of to the controller by inserting it into the opening `incoming-call` Mbus command. The controller then adds the set G of encoding schemes it can transcode into one or more of those in M to M and sends their union (ordered by preference) to the call-control engine representing the callee via the `call` command. The fact that there are no combined capability descriptions, e.g. “either G.711 or G.723.1 with H.261 video” yet is making things a great deal easier here.

After receiving a subset of $M \cup G$ as part of the resulting `connected` command, the controller has to determine whether there is any element of M left. In that case, the controller removes all remaining elements of G and communicates the result — that is, the remaining elements of M — to the call-control engine that sent the `incoming-call` command as part of an `accept` command and lets it choose the encoding scheme to use eventually. Otherwise, if there are no elements of M, but only some of G left, the gateway will have to do the necessary media transcoding.

Analogous to the preceding call setup examples, the endpoints arrange that their media engines are then configured according to the capability exchange and begin to send their media streams. Should an extended RAT instance be involved, it is configured using the predefined group of `rtp`-commands.

Special Cases

There can be several exceptions to the sequence of Mbus commands or reactions to them described above. For example, the gateway controller might decide to transcode between encoding schemes, in spite of there being elements of M, to save bandwidth when it is scarce. It might also declare the capability exchange as failed, because it simply cannot provide the resources to do the necessary transcoding at the moment. A combination of these cases is also possible. Furthermore, the untimely `accept` in conjunction with an ISDN line handled by an extended RAT implies the reduction of M to exactly one of its elements — the heuristics for this remain to be discussed; they will largely depend on the audio encoding schemes mandatory within the dominating protocol suites.

It should also be noted at this point that the H.323 call-control engine developed at TZI/DMN can map the SDP-subset required for the capability exchange to H.245. It is currently being discussed whether H.245 functionality should be integrated into the controller. Otherwise, the capability exchange within the H.323 cloud, that is between the H.323 call-control engine and the respective gateway-external entities, may yield a one-element subset of $M \cup G$, forcing the gateway to do media transcoding (that would be unnecessary in the first place), if the remaining element is from G, the set that the controller added before.

The aforementioned `caps-indicate` and `caps-set` commands are regarded as an add-on to the capability exchange process described in this section for the time being.

6.2.8 Compliance with the relevant Standards

At first, it might be rather confusing to find a section labelled like this. Please note that it has been included due to standard-compliance issues with third-party implementations, not ours (e.g. the components that STARGATE is made up of).

We simply can not guarantee that STARGATE's H.323 components will work with your H.323 phone, gatekeeper, and so on. Even with the equipment that we were able to test in our lab, there have been broken PDUs and the like.

Furthermore, some third-party applications somehow seem to be unable to generate a valid RTP/RTCP-Stream. Example: Microsoft Netmeeting. As RAT is rather picky in this regard, a stable communication with Netmeeting was not possible (that is, unless you hack RAT to pieces).

This was another reason for us to include the above setup-description of StarGate, our own H.323 gatekeeper and our Siemens H.323 phones, because that has been tested to work --- while forcing us to adapt to some of Siemens' "very own" interpretations of some aspects of H.323.

Please feel free to contact us about interoperability issues as soon as you have reasonable doubt that your equipment is operating in full compliance with H.323 or any other relevant ITU-standard.

6.2.9 Conclusion and Perspective

It has been the current STARGATE release's primary goal to provide a basic version of STARGATE that implements the core functionality described herein. We have been successful in achieving this goal.

While a string of additions still needs to be made (among them a user interface Mbus module enabling a run-time/remote configuration), the advantages of a self-developed and thereby well-understood call signaling and media transcoding gateway whose source code is readily available are already clearly visible. With its modular and expandable architecture, STARGATE developed by TZI/DMN stands out from other implementations available in hardware and/or software.

Conference Control on the Mbus

The focus of the Mbus call/conference control commands is currently being shifted from a call-based towards a conference-based model. The predefined Mbus command set is being adapted accordingly so that a call between two endpoints will become only a special case of a multipoint conference. At the same time, support for call forwarding and call redirection will be added to the STARGATE controller.

Merging Gateway and Gatekeeper

As already mentioned throughout this chapter, it is highly desirable to closely integrate the STARGATE controller with the H.323 gatekeeper developed at TZI/DMN to make its functionality directly available to the entities described herein without unnecessarily distributing it within the overall system. This will be the focus of mid-term development of our call-signaling and media-transcoding gateway beyond its current release.

6.3 H.323 Client: Wipone

To spread the use of IP-telephony usage within the local department of computer science, Wipone has been placed on top of the H.323 stack developed at TZI/DMN. It provides a graphical user interface to the basic telephony services offered by the H.323 stack and other adjacent modules also developed by TZI/DMN. Wipone provides the user with "phone directory", a list of current phone calls, an interface for audio control and a possibility for transmitting DTMF-tones.

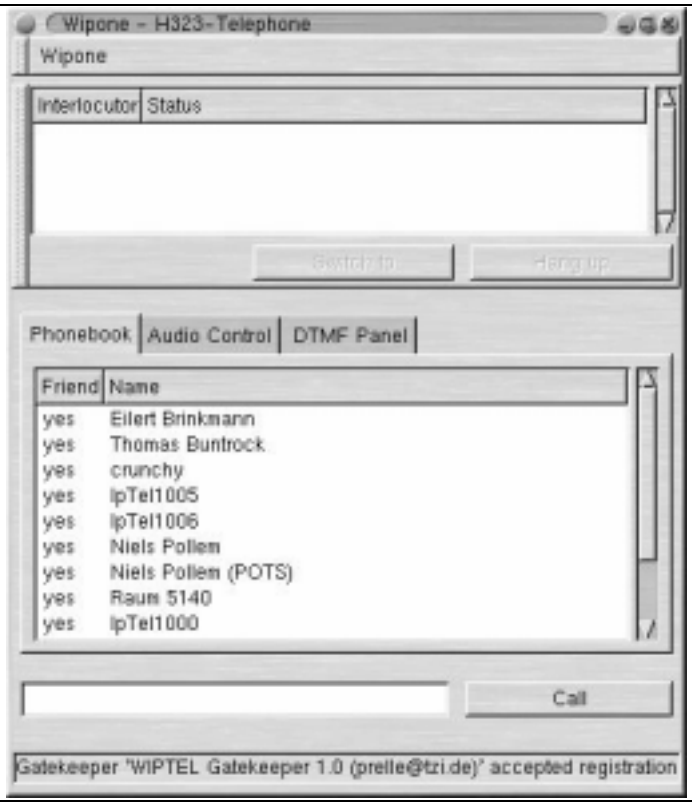

Figure 29 - Wipone GUI

Wipone takes both the control over RAT (see 6.3.3) and the H.323 call-control and therefore is the central controller in the endpoint.

6.3.1 Functionality

When started, Wipone starts some other needed modules, such as the H.323 call-control, and the RAS-client. Then, the H.323 call-control is configured over the Mbus and tries to register with a gatekeeper. The state of the registration is displayed in the status bar of the GUI. It will either read "IP dialing only", if no gatekeeper could be found, or, if a gatekeeper accepted the registration, the name of the gatekeeper who actually did so.

In the middle area of the window is a personal directory. To initiate a call, the corresponding entry has to be selected and the button “call” has to be pressed. Wipone then automatically detects whether the name could successfully be resolved into a transport-address or if Wipone itseld needs to supply the H.323 call-control with it.

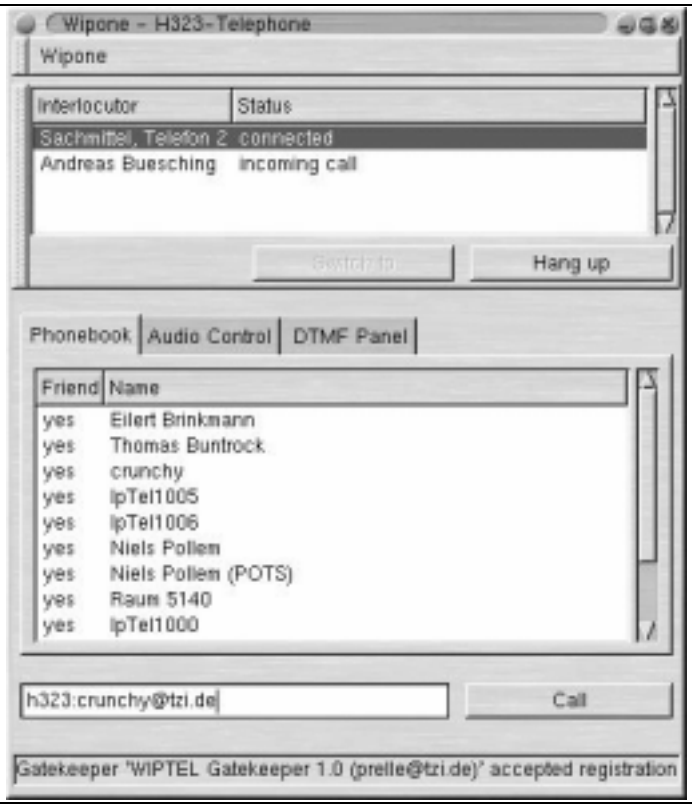

Figure 30 - Wipone GUI during a call

As an alternative, one can enter a H.323-, E.164- or transport-address directly into the field at the bottom to call persons not listed in the directory. To distinguish between the different types of addresses, it is important to use the standard-prefixes "h323:" resp. "e164:". Transport-addresses are either the IP-address in dot-notation or the name of the host and the port-number separated by a colon.

All connections, both established and upcoming, are displayed in the upper part of the window and can be interrupted at any time. It is possible to switch between established connections by selecting the desired connection and pressing the "Switch-to"-butto. Since the H.323 call-control does not yet support the supplementary service "hold", the switching is not carried out according to the standard. Instead, Wipone simply quits the audio-engine, so that no more media-stream data can neither be received nor be sent. In case the user should want to continue the call at a later point, the audio-engine will be restarted with the previous parameters and addresses.

In the middle of the window, one can also select a panel for audio-control. The upper part of the GUI still shows the current connections and a direct call is also still possible.

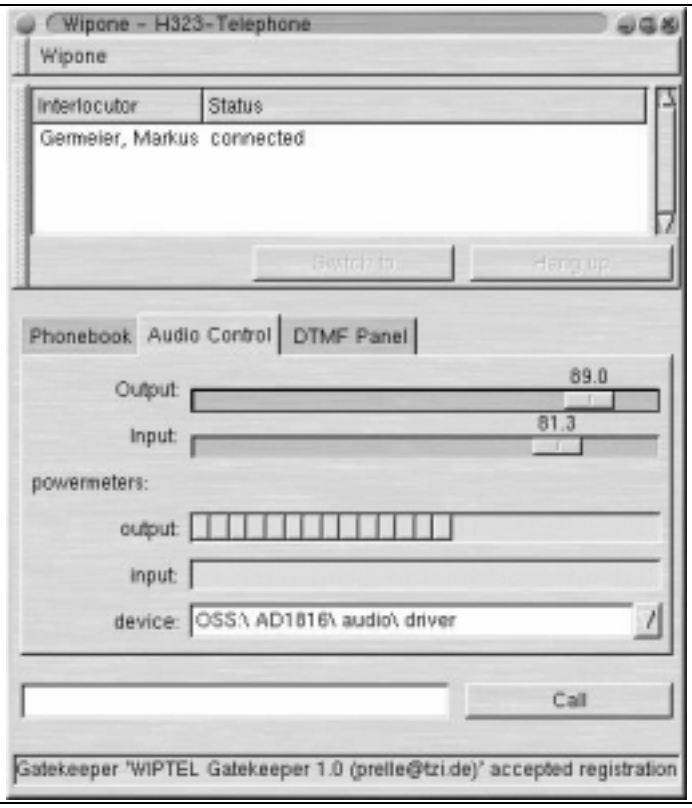

Figure 31 - Audio control in Wipone

In the middle there are two sliders to control the volume. One is for the input volume and the other is for the output volume. Underneath each slider, its deflection is shown. In addition to this, the current audio-device is also displayed. For the case of several devices, the wanted device can be chosen.

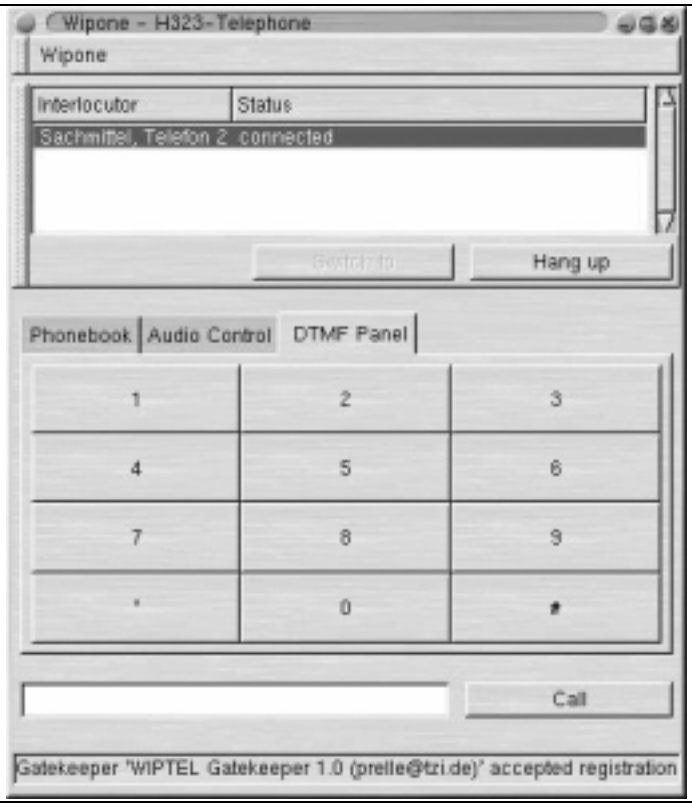

Figure 32 - Wipone DTMF panel

The third panel shows a keypad known as on normal telephones to transmit DTMF tones. To do so, the receiving party must first be selected in the upper part of the GUI. It is not necessary that the receiving party is also the current connection, it may also be on hold.

6.3.2 Configuration

Since Wipone originates from a small test-program, which was never supposed to be used somewhat interactively, a configuration from within Wipone in not yet possible.

However, some preferences can be defined in the file "`~/.wipone/options`":

- `realname`

  This field contains the name of the user. It will be transfered in the Display-Information-PDU.

- `signalling_port`

  This tells Wipone the standard call signalling port to use, if none is explicitly given. It should have the value 1720.

- `upi`

  H.323- and E.164-addresses under which the user wants to register at a Gatekeeper.

- `preferred_gks`

  A list of Gatekeepers, where the registration is to be tried first. If every Gatekeeper in the list refuses the registration, a registration request is sent out via multicast.

For example, a configuration could look like this:

```
realname "Andreas Buesching"
signalling_port 1720
upi h323:crunchy@wiptel.org e164:72839
preferred_gks dutchman.informatik.uni-bremen.de
```

The file "`~/.wipone/directory`" contains the entries of the personal directory. Every entry is placed in a seperate line beginning with the word "friend", followed by the name of the person, a transport-address (incl. port-number) and a list of H.323- and E.164-addresses. Neither a transport-address nor a H.323- or E.164-address is mandatory by itself, but at least one of these elements must be present, no matter which.

A directory could look like this:

```
friend "Andreas Büsching" "dolormin:1720" h323:crunchy@wiptel.org e164:72839
friend "Eilert Brinkmann" "dominion:1720" h323:eilert@wiptel.org
friend "Dirk Meyer" "riemen:1720" h323:dmeyer@wiptel.org h323:dmeyer@tzi.org
friend "Telefon 1000" "iptel1000:1720" h323:1000
```

Settings for RAT can currently only be set if RAT is started directly (with them) and not from within Wipone. RAT saves those settings by itself and the relevant media-engine reads and uses them once it is started from within Wipone.

6.3.3 Wipone Requirements

Wipone needs the following packages:

- gtk 1.2.x, glib 1.2.x
  `http://www.gtk.org/`
- libmbus, libnotifer (Version > 1.0.2!)
  `http://www.mbus.org/`
- UCL Robust-Audio Tool (RAT) (Version >= 4.2.9)
  `http://www-mice.cs.ucl.ac.uk/multimedia/software/rat/`
- TZI/DMN RAS-module
  `http://www.wiptel.org/`
- TZI/DMN H.323 call-control
  `http://www.wiptel.org/`
- pthread
- Java 1.2
  `http://java.sun.com/`

6.3.4 Binary Installation

At the moment, Wipone does not have any complex directory tree, so the installation is quite easy.

1. Make sure you have installed all the system requirements
2. Copy the wipone binary to any path which is listed in the environment variable PATH.
3. Check if all needed executables are copied to a path which is listed in the environment variable PATH. Needed executables: RAT audio-engine (e.g. rat-4.2.10-media), H.323 call-control and RAS-module, both from TZI/DMN.
4. Create a symbolic link to the RAT Audio-Engine you want to use named audio-engine e.g.
   `ln -s /usr/local/bin/rat-4.2.10-media /usr/local/bin/audio-engine`.

Please note: If you would like to communicate with a Siemens HiNet LP 5100 telephone, you need to patch the RAT audio-engine because this telephone does not comply with RFC 1889 regarding the RTP port, which should be an even number.

7 IP-Telephony Testbeds within WIPTEL

7.1 IP-Telephony Test Lab at TZI/DMN

This section describes the test environment at the TZI/DMN testlab of the University of Bremen. The first subsection introduces the telephony and network infrastructure of the university while the next subsection informs about the IP telephony components and their architecture.

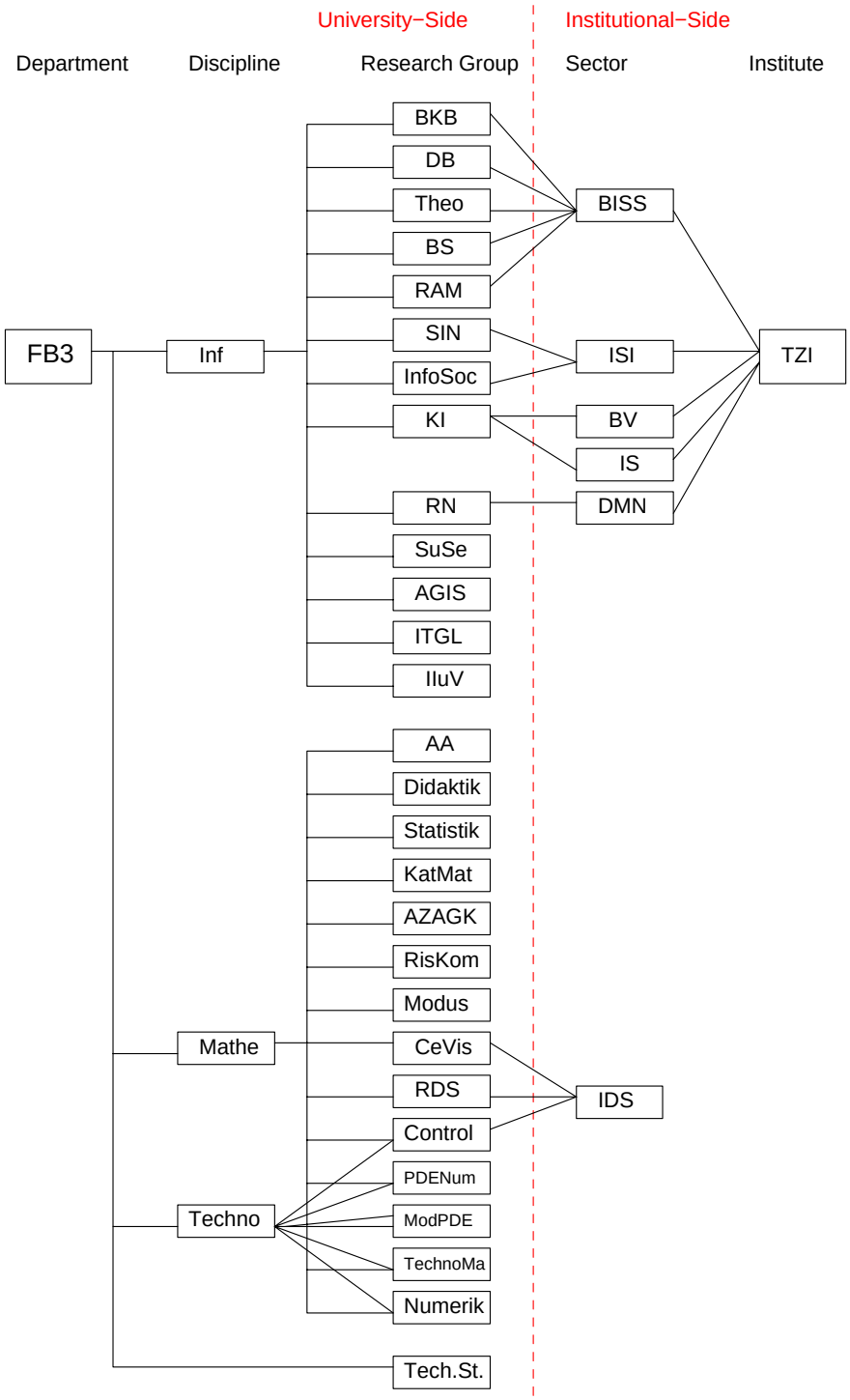

Figure 33 - Organization of the faculty of Computer Science / Mathematics

7.1.1 Structure of the University of Bremen

The University of Bremen is divided into several departments ("Fachbereiche") which host faculties. Each faculty itself is divided into several research groups which address specific subjects of a faculty. Some of these research groups join forces and form an.

An example (compare Figure 33) for the divisions introduced above is the "Fachbereich 03" which hosts the faculty "Computer Science", "Mathematics" and "Techno-Mathematics". There are 13 research groups within Computer Science and 14 in Mathematics where five of this research groups also build the faculty-variation Techno-Mathematics.

To the outside, some research groups are represented as an institute called Center for Computing Technology (Technologiezentrum Informatik, TZI). There is not necessarily a 1:1-mapping between research groups of the faculty and a sector within the TZI. There may be a n:1-mapping like the research groups and the BISS-sector ("Bremer Institut für Sichere Systeme"), a 1:n-mapping like the RG KI ("Artificial Intelligence") and the units "Picture Processing" and "Intelligent Systems" or a 1:1-mapping like the RG RN ("Computer Networks") and the sector DMN ("Digital Media and Networks").

Apart from the departments, there are other organizations the university may be divided into. There is the administration which is again divided into departments ("Dezernate") dedicated to special areas of the administrative work.

Another group is the technical staff of the university. This encompasses all those who are responsible for all work related to a specific building and those who do jobs concerning all parts of the campus.

Existing telephone infrastructure

Two kinds of networks are important for the university: the telephone and the computer network. Up to now they just coexist without any connection worth mentioning. Because this is the situation we start from, both infrastructures shall be inspected in more detail.

To reach a phone within the university from the outside you have to call +49 421 218 plus a 4-digit suffix. Theoretically there are 10000 phone-numbers but in fact there are no numbers starting with a 1 or a 6. To place a call from a university-phone to a destination outside the campus you need to dial a prefix. The type of prefix determines how a call should be handled and accounted:

- Service prefixes:

  01: This prefix indicates a call to the "worldwide" telephone network that shall be treated as an official call. Resulting costs will be paid by the university.

  02: Unassigned (formerly used similar to 01 or 03)

  03: This prefix also indicates a call to the "worldwide" telephone network but unlike the prefix 01 the primary user of the telephone has to repay the resulting costs for calls outside the local calling area. Local calls are balanced via a fixed monthly deduction form the primary user's wage.

  04: Pager

  05: Respond to pager

  06: Reserved for faxserver

  07: Emergency

  09: Using this prefix a call is targeted in a separate telephone network that connects government authorities. In this network calls are free.

- Service extension numbers:

  2000: Answeringmachine

84: Pickup groups

85: Student union ("Studentenwerk")

88: Central reception

80x: other receptions

91111: Emergency (external)

Existing network infrastructure

There are numerous domains in the scope of the university. At the top-level the domain uni-bremen.de leads to the border element gatekeeper of the University of Bremen. Most faculties have their own subdomain like informatik.uni-bremen.de for the faculty Computer Science. While the departments with a greater number of computers administrate their network themselves the departments with less computer equipment are administrated by the ZfN ("Zentrum für Netze" - Network Centre) which also is directly connected to the DFN.

There are currently about 280 subdomains for uni-bremen.de - too much to list them all in this document.

The following Figure 34 shows the simplified network infrastructure in the computer networks research group before introducing IP-Telephony. It contains workstations and telephones in a seperate telephone network, their locations and how they are connected to the rest of the world. The "Landesnetzrouter", located in the "Zentrum für Netze", is the next hop to the DFN Connection.

The room 5390 is used as a server room for several levels of the building. The host dialman is a TeleS-PBX and was bought because the PBX of the University hasn't have enough free ports to satisfy the needs of our research group. The host damn contains an ISDN card and is used for "Audiogate" - a gateway to participate in Mbone conferences with a telephone.

With the introduction of IP-Telephony there had to be some changes. Most of the rooms either weren't equipped with the required amount of network-sockets to connect the IP-Phones or just possibilities to connect 100 MBit/s-devices (and the IP Phones from Siemens only support 10 MBit/s). The last situation caused problems with the Siemens HiNet LP 5100 IP-Phones that have been bought because they only support 10 MBit/s. To solve this problem we placed 100/10-Mbit/s-switches in those rooms.

Figure 35 shows the networking infrastructure of our research group after the introduction of IP telephony. Note that there was enough bandwidth capacity so that no QoS mechanisms were needed at all to enable the research group to use IP telephony.

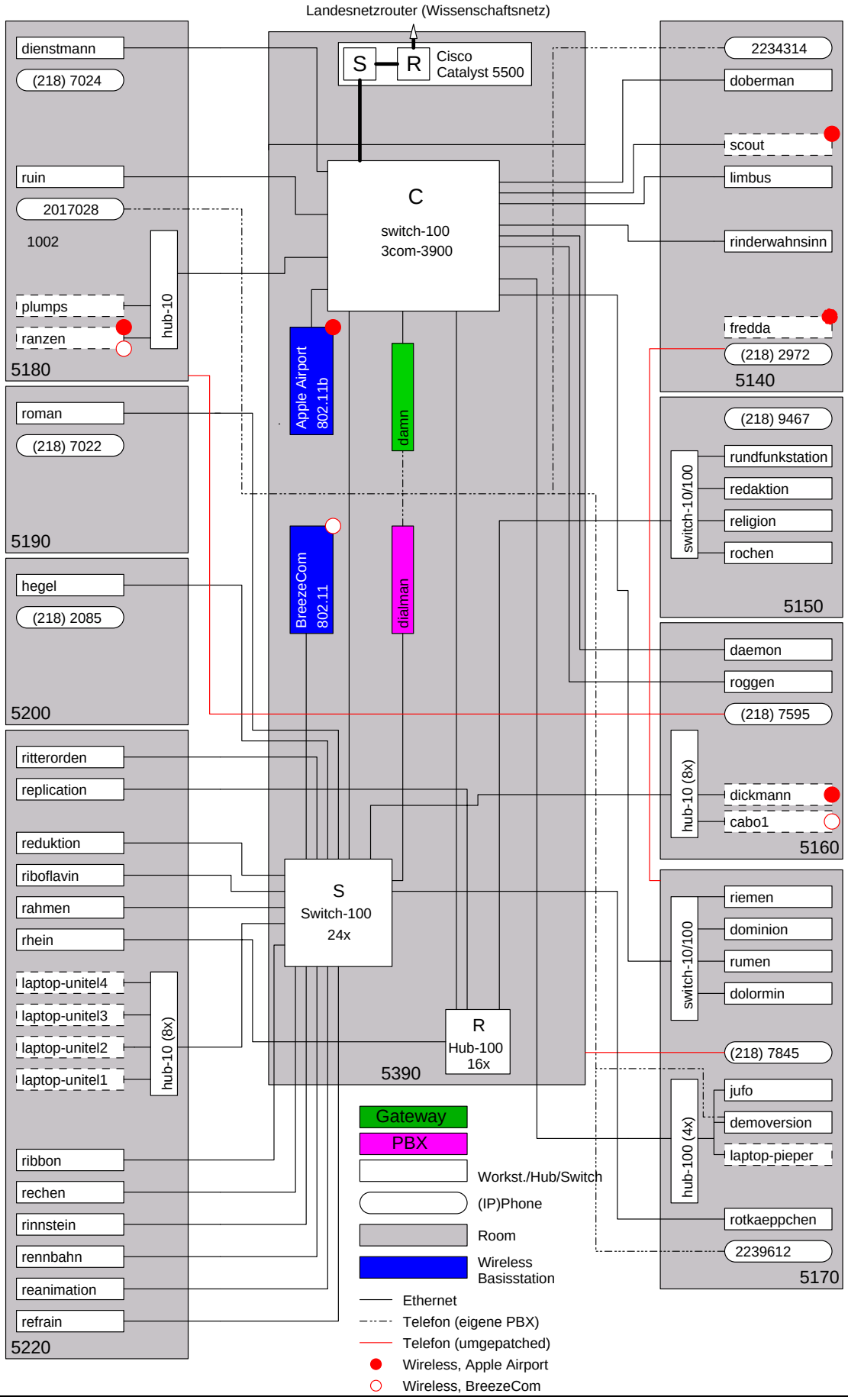

Figure 34 - Network infrastructure before the introduction of IP telephony

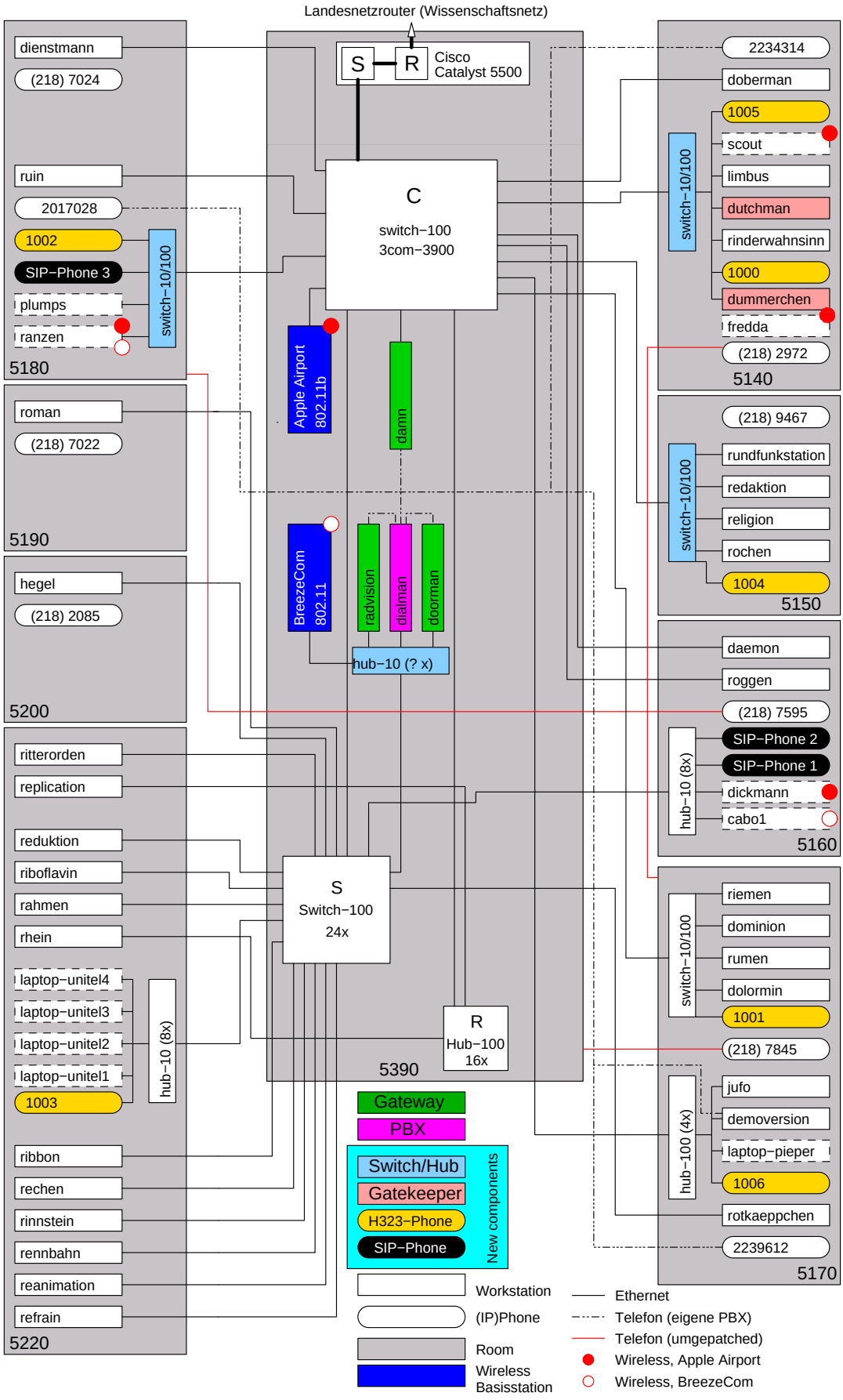

Figure 35 - Network infrastructure after introduction of IP telephony

7.1.2 Telephony testbed in the University of Bremen

The testbed of the University of Bremen includes H.323 gatekeepers, SIP proxies, hard and software clients and a Teles PBX and a Teles iGate-Gateway (see Figure 37 and Figure 36). The later is used to connect the testbed to ISDN/PSTN and GSM phones. Up to now there is no connection to the legacy phone network of the University. Such a connection is expected to be installed in early 2001. The connection to the WiN will be provided by StarGate as a signaling gateway and a TRIP border element.

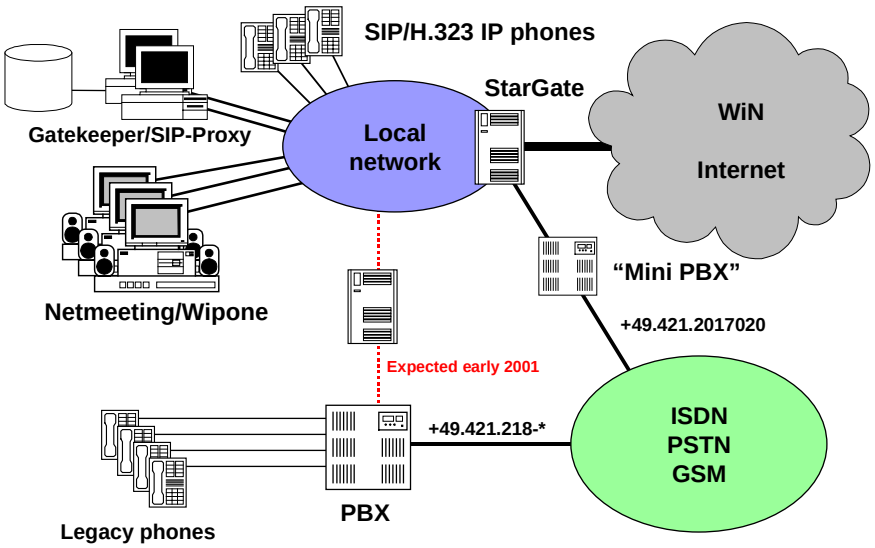

Figure 36 - IP telephony testbed in Bremen

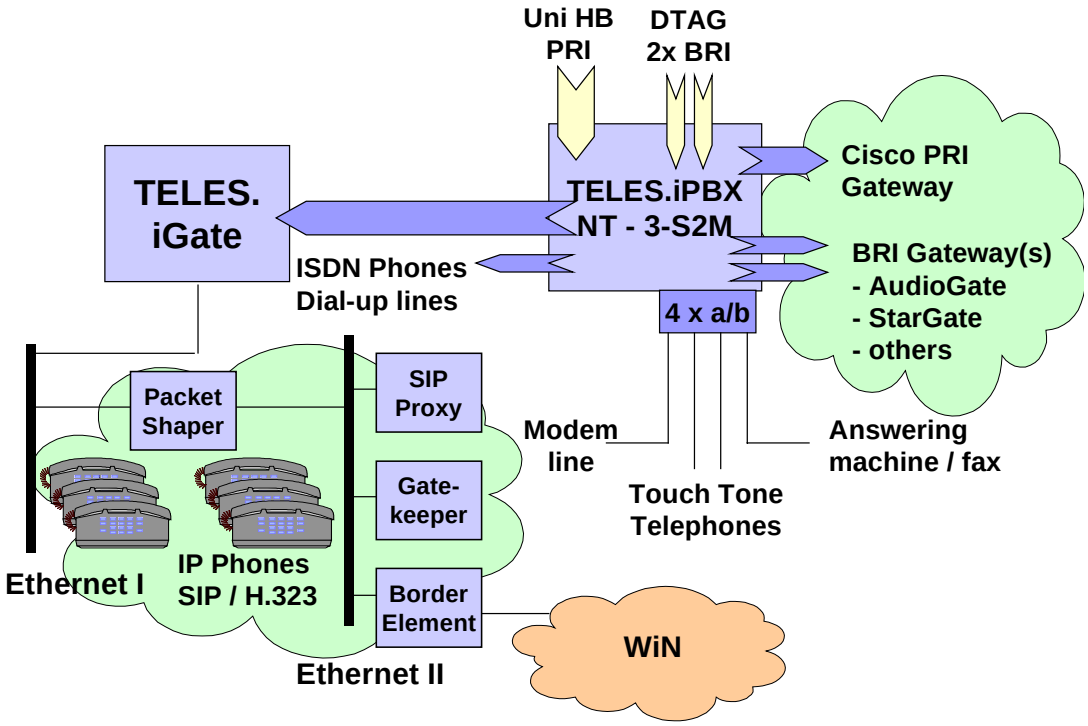

Figure 37 - DMN/TZI testlab

Following is a list of components - temporarily for testing purpose, bought or self-made - that are currently used and tested for IP telephony:

- Terminals
  - Siemens HiNet LP 5100 (Hosts 1000-1006)
  - 3com SIP Phone
  - WIPTEL software client (wipone)
  - NetMeeting 3.0
  - OpenPhone
- Terminals expected to arrive in December:
  - Siemens HiNet LP 2100 (with new H.323 Stack)
  - Innovaphone Tiptel 200
- Gateways
  - TeleS IGate (Host doorman)
  - RadVision OnLan 323 Voice (Host radvision)
  - StarGate - Own software specialized for WIPTEL (Host damn)
- Gatekeepers
  - Siemens HiNet RC 3000 (Host dummerchen)
  - Gatekeeper developed at the University of Bremen (Host dutchman)
  - OpenGatekeeper

While the H.323 components were described in a former deliverable the SIP components are described in the next section.

7.1.3 SIP testbed

This section shows in brief the tests that have been performed so far. As our local network infrastructure is considerably well administrated it does not completely reflect the problems that may arise in real world application scenarios. For example, SIP requests might have to traverse different ISP's backbone networks in order to reach their final destination. Since these are managed independently, routing issues may occur that cannot be simulated in our local environment.

Keeping this deficiency in mind, most of the tests described in the following subsections are basic feature tests used to classify each SIP component with respect to the classification scheme previously defined. Using the results of this section and the application scenarios that have been shown at the beginning, a set of more realistic test cases will be defined in a later section of this document.

Our local SIP infrastructure

Before starting with detailed descriptions of the tested components, a short overview of our test environment is provided. As mentioned before, the tests have been performed using the heterogeneous campus network at the University of Bremen. The available SIP components have been spread over several networks in different subdomains to achieve more realistic results.

Whenever possible, automatic configuration services like DHCP have been used to set basic Internet access parameters–netmask, gateway, DNS server to name a few. As none of the hardware user agents has support for autoconfiguration of SIP-specific parameters, these have been set manually.

In order to create meaningful output traces that can easily be analyzed, user agents have been configured to use a single outbound proxy, if applicable. Figure shows one of the configurations applied during tests.

As indicated in the figure, several SIP implementations, both hardware and software, from different vendors are available. The implementations range from nearly product-state SIP IP phones to early

releases of experimental user agents used to create customized test scenarios. This environment does not only allow for typical application scenarios as expected for day-to-day use of SIP user agents to be tested. In addition, certain error-prone configurations that probably do not arise during normal use in our laboratory (e.g. high jitter rates due to network latency) can be produced in a deterministic manner.

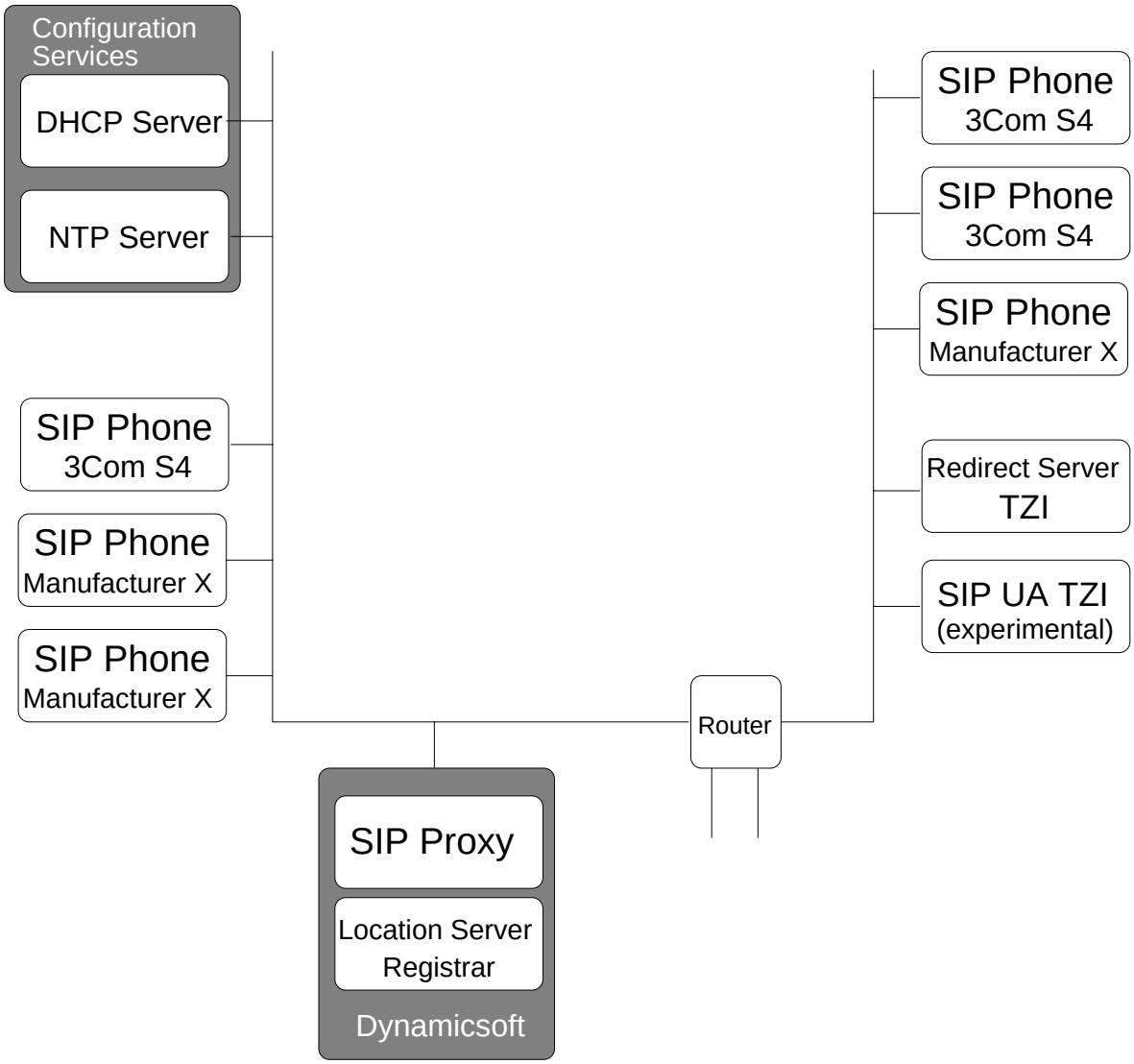

Figure 38 - Local SIP infrastructure

Test Environment

As our testing currently aims at basic feature tests, we have to avoid too complex testing scenarios. In order to reduce the possibility of errors induced by the testing framework, the most important testing tools are rather simple: A packet sniffer is been used to trace the call-signalling path while small SIP messages are generated manually (using a command-line interface). For more complex transactions, the TZI SIP implementation is used.

Up to now, most of the feature tests have been performed with the participating components being in the same network segment. Doing so, it was possible to log the network traffic with a packet sniffer. IP addresses have been assigned dynamically via DHCP. Since all tested user agents depend on having a dedicated SIP proxy, they have been manually configured with the proxy's address.

The remaining part of this section gives a short overview of the tools that have been involved in tests or are considered to be part of the framework. For some of these tools, more detailed experience reports are given in subsequent sections.

TZI SIP implementation

Most of the tests have been performed using the TZI's own SIP implementation in order to detect interoperability issues as early as possible. This section briefly describes the features of the TZI stack. As it is designed to act as protocol engine for a component-based SIP-H.323-gateway, it has a message oriented command-line interface accessible by several scripting languages.

The core component of that stack is a SIP UAS capable of TCP and UDP (multicast and unicast). Incoming messages are parsed and mapped to call-control commands as specified in *The Message Bus: Messages and Procdedures*[2].

Dynamicsoft SIP Server

Dynamicsoft [3]offers a broad range of high-end SIP components. We obtained an evaluation license for their proxy and location servers. In addition, user agent implementations in C++ (for Solaris and Windows) and Java are offered to download for a free 60-day trial.

The server distribution contains several tools to facilitate administration and test of the dynamicsoft components. The stack is implemented in pure Java using the JDBC-interface to an external database. It should be possible to configure the registrar to use any DB system for which JDBC support is available. The dynamicsoft server is shipped with configuration files for Oracle 8i, Standard Edition.

The database is used by the registrar/location server to store and lookup user registrations. At registration time, a user may upload a CPL script that is evaluated when the location service is invoked.

The proxy's feature list shows almost full compliance to RFC 2543, however lacking support of DNS SRV records and PGP authentication. The server can run as stateful and stateless proxy as well as redirect server. Moreover, call routing to external telephony gateways is supported for tel- and SIP-telephony-URLs.

Though we have not yet been able to perform systematic feature test with this SIP stack, we decided to provide a short description of our observations made when getting the system's components to work.

3Com SIP Phone S4

By now, most of our tests have been performed using the 3Com SIP telephone S4 bought directly at the 3Com US branch. Already at product state, this telephony hardware has turned out to be very useful to interactively generate typical call flows as input for test cases. As many typical supplementary services enabled by SIP are already implemented, we were able to create more complex test cases resulting in real world applications than with manual scripting.

The phone's quality was better than expected, most of the bugs that we have found in the protocol implementation were fixed in one of the regularly announced firmware upgrades prepared by the vendor.

7.2 H.323 Partner University of Hannover

The "Regionales Rechenzentrum Hannover (RRZN)" is one of the partner sites in WIPTEL. The RRZN already managed to convince their administration to get access to the PBX with their H.323

[2] http://www.mbus.org/drafts/mbusprot.html

[3] http://www.dynamicsoft.com/

products (Cisco CallManager). It was planned to enable H.323 call routing via Bremen and Hannover. The RRZN set up a computer with and the gatekeeper of the University of Bremen for being a border element to exchange routing information. Furthermore the RRZN used Microsoft NetMeeting as a software client for H.323. The gatekeepers in Bremen and Hannover used a preliminary TRIP implementation to exchange routing information.

A practical test of calling one another was successful to a certain degree. Both parties were able to talk to another but at least the quality of the audio stream that was received in Bremen was pretty low and the speaker in the RRZN hard to understand. The reason for this may be NetMeeting or the quality of the Headset used in the RRZN. This is for further study.

7.3 VoIP - Connectivity between the University of Hanover and the Technical University of Brunswick

The VoIP testbed between Hanover and Brunswick is based on two Cisco 2600 routers, both enhanced with VoIP gateways. As shown in Figure 39, the VoIP are connected via H.323. In both sites, the connection between the VoIP gateway its respective PBX is established via a 2 Mbit/s link. The protocols employed on this link are ITU-T G.703/ G.704 in layer 1 and 2, Qsig basic call as signaling protocol in layer 3. In order to check the availability of the remote partner gateway as well as the link quality between the two, a call fallback feature was implemented, showing the behaviour described below.

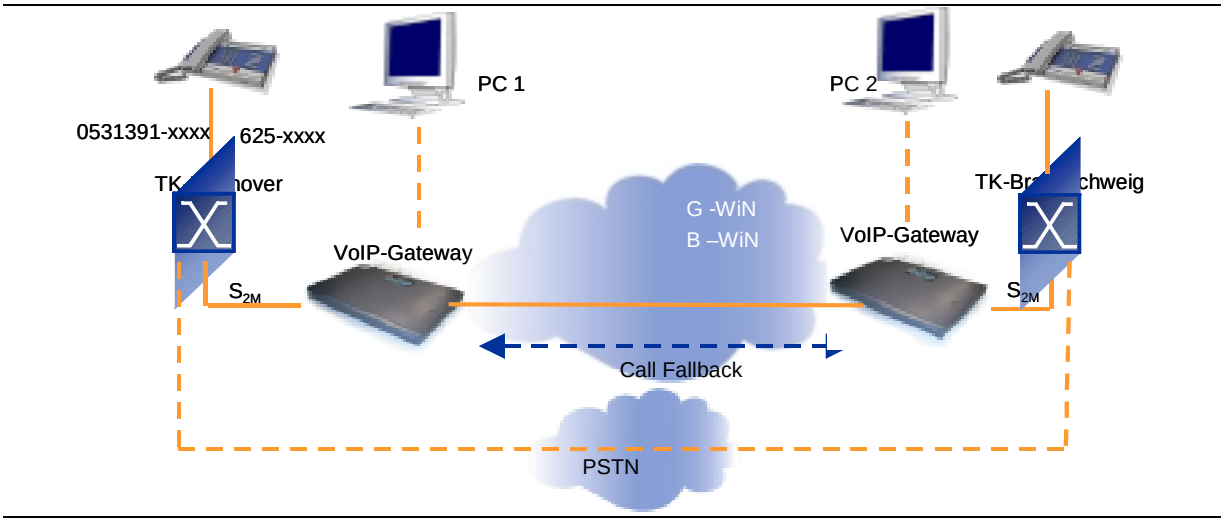

Figure 39 - VoIP-Testbed Hanover-Brunswick

Every time a call setup from the PBX is received, an internal call fallback cache will be checked first. A call fallback cache entry stores the delay and loss rate of previous quality of service measurements for a given dial peer. If no corresponding entry in the call fallback cache exists, a service assurance agent (SAA) measures the required parameters by creating a test loop to the dial peer. The delay and loss rate, either derived from the cache or from the SAA, will be checked against the configured thresholds. If all values are within bounds, an H.323 call setup is sent to the peer. If the preconfigured thresholds are exceeded, a call disconnect with an appropriate cause message will be sent to the PBX.

Due to the limitation that all PBXes accept exactly one specific cause message for the purpose of rerouting, our installation maps all messages that belong to the class "quality of service unavailable" onto the message "no route to destination". As a result the PBX will try to establish an alternative connection via the public switched telephone network (PSTN).

The VoIP link can be reached in two ways: explicitly via dialing prefix or automatically via least cost routing in the PBX. In our installation, 625-xxxx is the prefix to activate the gateway to Brunswick. If the PBX least cost router in Hanover encounters the 0531-391-xxxx prefix, the call is rerouted to use the VoIP gateway.

Our tests revealed that as long as the one-way delays don't exceed 120 ms and packet losses are below a 7 % threshold, a service level not worse than the PSTN can be achieved. Using the above specified threshold, facsimile transmission employing VoIP networks were also tested successfully.

7.4 Fokus SIP Testbed

To check the suitability of the SIP concept and the current implementations it is planned to setup a testbed. Within this testbed it should be possible to deploy all standard scenarios (PC-to-PC, phone-to-PC, phone-to-phone) with regard to inter-domain communication and supplementary services.

The following figure introduces the planned infrastructrue at GMD Fokus. It consists of existing as well as internally developed solutions. The core of our SIP infrastructure consists of multiple SIP proxies. These provide services for the endpoints, external participants and a ISDN gateway that allows communication with external as well as internal PSTN participants. A SIP-H.323 signaling gateway provides interoperability with H.323 systems.

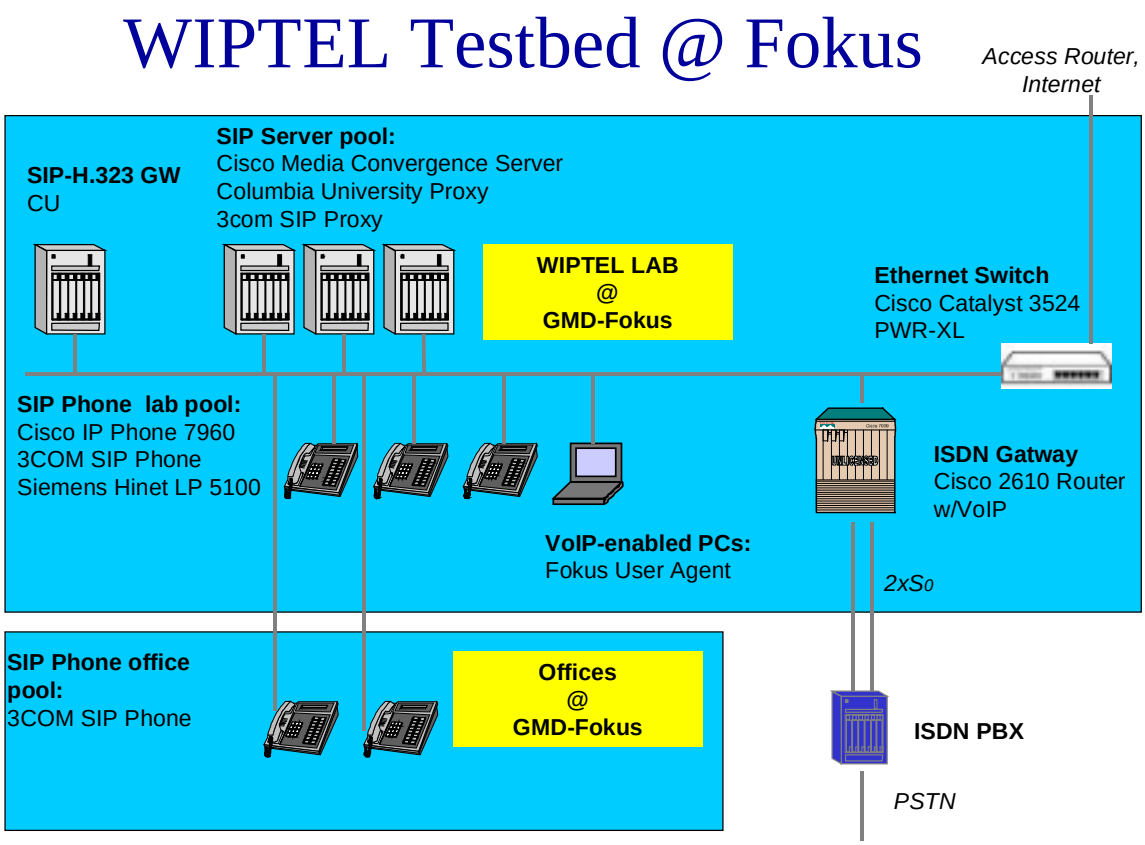

Figure 40 - GMD Fokus testbed

Depending on the implementation status an RTSP announcement server, dynamic configurable firewalls and an own SIP user agent which are developed within other Fokus projects can be integrated into the testbed.

7.5 Other observations

Checking the logfiles of the gatekeeper implemented in the University of Bremen revealed some H.323 activities from other institutions. Occasionally some sites started H.323 applications that used a TTL high enough to reach the University of Bremen to discover their gatekeeper. Most of this applications only supported H.323 version 1 which means that they were presumably 2 or more years old.

If a gatekeeper confirmed such a request some of these endpoints tried to register with the gatekeeper. If then the registration request is confirmed too the endpoints registered with the gatekeeper in Bremen - which presumably wasn't intended because it may prevent those endpoints from reaching other endpoints in their site.

An example for this is the Clinic of the University of Regensburg. For some still unknown reason the H.323/ISDN gateway in Regensburg always directly tried to register with the gatekeeper in Bremen. If the gatekeeper rejected the request the gateway in Regensburg wasn't operational and thus no medical conferences could be held. The customer support of the distributor was unable to tell Regensburg how to change the gatekeeper settings so the temporary solution was to enable some hosts in Regensburg to register in Bremen. This "solution" is still in use.

8 Conclusions

This project deliverable has elaborated on the site infrastructure as well as the WiN infrastructure for VoIP communication developed within WIPTEL. Furthermore, it has reported on the initial steps taken towards and also the first experiences made and lessons learned from the WIPTEL testbed, formed by the RRZN at the University of Hanover, GMD Fokus in Berlin, and, of course, TZI in Bremen, where all central services are located. We would like to mention again that there are additional institutions that have expressed an interest in being included into the testbed. We will accommodate them as soon as the current testbed has reached a state in which it can be used in a production-system-like manner by all staff on the participating working groups.

Also included in this deliverable was a wrap-up of the project goals and main activities throughout the whole of WIPTEL, as well as the past deliverables, to help better understand the overall results that we have presented in this 6th deliverable, the *"Infrastructure and Deployment Report"*.

8.1 Overall Results

We are glad to report that WIPTEL, headed by TZI, has reached all of its major goals, albeit the current testbed phase will have to continue beyond the scheduled end of the project. (As has been mentioned in the introduction, a separate 7th deliverable summarizing its results will be handed in in 2001 by TZI and its participating partner institutions.)

- Regarding both the overall site and the WiN infrastructure, we have proposed, implemented, and thereby also verified a protocol-agnostic IP-telephony architecture that fully respects the autonomy of the institutions connected to the German Research Network, while abiding to the major VoIP standards.
- We have proposed a name/numbering plan and its integration with the one of the current telephone system, especially taking into account the proprietary PBXs present in many institutions.
- We have tested an increasing range of IP-telephony components, including hardware and software endpoints and adapted own implementations to fill open gaps where needed. As a result, we were able to install (and document) a reference site infrastructure at the TZI that has been in a production-system-like use by our staff for quite some time.
- With H.323 and SIP, two important standards for call signaling have been taken into account. We have shown where and how to integrate H.323-SIP-(POTS) gateways and the like into the overall infrastructure. Avoiding proprietary solutions has been an important decision.
- TZI staff has actively participated in ITU-T as well as IETF activities throughout the whole of the project. This has, for example, led to an intimate understanding of TRIP and one of the first TRIP implementations becoming available. The proposed WiN infrastructure has greatly benefited from these activities.

For a more detailed description of the archievements throughout the project's lifetime, please refer to the wrap-up in this deliverable's introduction or to the individual deliverables.

8.2 Open Issues

While the major project goals have been reached, we need to point out that there are some open issues even at the end of the scheduled project lifetime. These can be categorized into such that where impossible to reach given the circumstances encountered and such that will be part of the ongoing work on the WIPTEL IP-telephony testbed in the German Research Network (WiN) run by the DFN.

Regarding the overall product availability, the implementation of security standards and protocols is generally not well-progressed. This currently limits the production use of IP-telephony in all cost-

incurring areas. Also, quality-of-service (QoS) services are not currently available on a broad scale within the DFN's WiN. We were therefore largely unable to go further into these directions.

Other problems stem from our difficulties with obtaining more test equipment (that is, in the end, with support from the vendors). With more available recently, we will catch up during the remainder of the testbed phase, though. Please also note that while we had to cut the number of participating institutions to two instead of the planned three, both H.323 and SIP components will be included.

8.3 Next Steps

As already stated, the current testbed phase will have to continue beyond the scheduled end of the WIPTEL project. A separate 7th deliverable summarizing its results will be handed in in 2001 by TZI/DMN and its participating partner institutions.

We will continue to support the current WIPTEL testbed beyond the scheduled project end to ensure its growth and consolidation, and eventually the adoption of IP-telephony connectivity and services offered by DFN by more and more of its member institutions. The continuing active participation of TZI staff in standardization activities will play an important role in this process.

A RRZN Hannover: Netzübergreifende H.323-Infrastruktur

A.1 Theoretische Grundlagen für die Migration von POT (Plain old Telephone) zu VoIP (Voice over IP)

A.1.1 Schritt 1: Kopplung von Telefonanlagen über ein IP-Netz

Die in vielen Einrichtungen eingesetzten Telefonanlagen sind an das öffentliche Telefonnetz angeschlossen. Es ist in Europa üblich, dass Telefonkosten nach tatsächlicher Verbindungszeit abgerechnet werden. Arbeiten z.B. mehrere Einrichtungen an einem gemeinsamen Projekt, so sind häufige (und oft lange) Telefonate unvermeidbar. Bei einer zeitabhängigen Abrechnung der Telefonate entstehen in solchen Fällen erhebliche Kosten.

Existieren Festverbindungen zwischen einzelnen Einrichtungen bzw. zwischen einer Forschungseinrichtung und z.B. dem internen Telefonnetz einer Behörde, so besteht in der Regel die Möglichkeit, die einzelnen TK-Anlagen direkt miteinander zu koppeln. Dabei werden sowohl die Daten als auch die Signalisierung digital übertragen. Da bei einer Kopplung von Telefonanlagen nicht nur die rudimentären Dienste wie Verbindungsauf- und -abbau, sondern auch zusätzliche Leistungsmerkmale (sogenannte Supplementary Services), siehe Bild 1, übertragen werden sollen, wird für solche Querverbindungen in der Regel nicht das Euro-ISDN Protokoll, sondern ein propriäteres Protokoll eingesetzt. Mitte der 80-er Jahre haben sich einige Hersteller von TK-Anlagen auf einen Industrie-Standard, das sogenannte QSig-Protokoll, geeinigt. Dieser ist aber im Laufe der Zeit von unterschiedlichen TK-Anlagen-Herstellern erweitert und modifiziert worden.

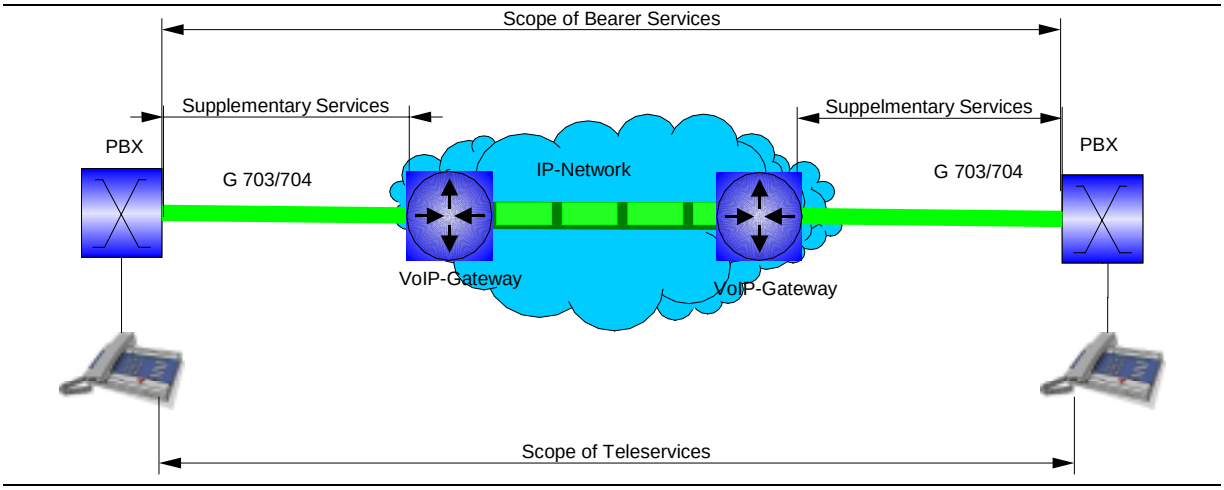

Abbildung 41 - Gültigkeitsbereich einzelner Dienste

Das QSig – Protokoll kann man in 3 wesentliche Bestandteile teilen:

– **QSig-Basic Call**: kontrolliert den Verbindungsauf- und abbau sowie die Übertragung der Telefonnummer des rufenden Teilnehmers (CallerID). Dieser Teil ist sehr stark an das Signalisierungsprotokoll des Euro-ISDN nach ITU-T Q931 angelehnt.

- **QSig-Generic Functions**: Dieser Teil des QSig-Protokolls beschreibt im Wesentlichen das Format der Einkapselung von Informationselementen in das Signalisierungsprotokoll. Kann eine Übertragungseinrichtung den Inhalt der in die Generic-Functions eingebetteten Elemente nicht interpretieren, so werden diese unverändert weiter ans Ziel übermittelt. Die Motivation zur Einführung der Generic Functions war es, spezielle Leistungsmerkmale zwischen zwei Vermittlungseinrichtungen eines Herstellers über eine Vermittlungseinrichtung eines dritten Herstellers transparent zu übertragen. Siehe Bild 2.

– **Herstellerspezifische Funktionen**: Das QSig-Protokoll sollte dazu dienen, viele Supplementary Services, die im EDSS1-Protokoll nicht enthalten sind, über Querverbindungen zwischen einzelnen TK-Anlagen zu realisieren. Es ist aber bislang dabei geblieben, dass jeder Hersteller von TK-Anlagen eigene Leistungsmerkmale präferiert. Dieses führte dazu, dass unterschiedliche Versionen und unterschiedliche Implementierungen des QSig-Protokolls zueinander unkompatibel sind.

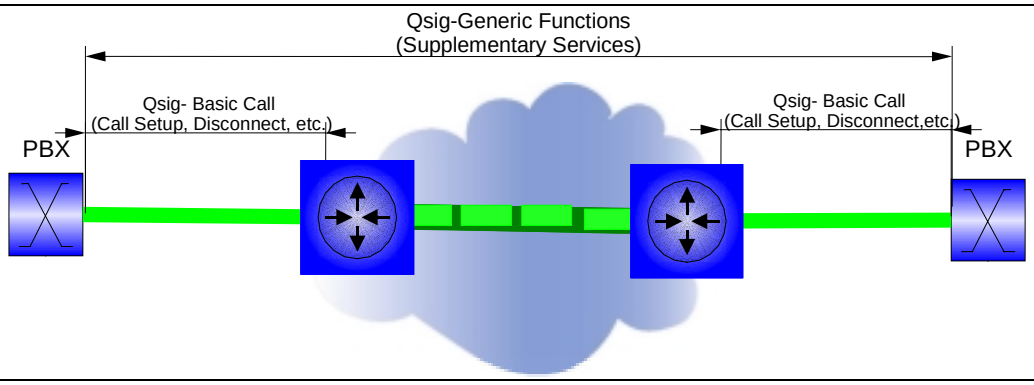

Abbildung 42- Austausch von QSig Basic Call und Generic Functions Elementen

Koppelt man also Vermittlungsstellen unterschiedlicher Hersteller über das QSig- bzw. über das EDSS1 – Protokoll, so ist diese Implementierung im Wesentlichen auf den im ITU-T Q.931 beschriebenen Basic Call beschränkt.

Standleitung vs. Virtual Leased Line

Wie schon in der Einleitung erwähnt, benötigt man für eine Kopplung von unterschiedlichen TK-Anlagen eine digitale Übertragungsinfrastruktur. Eine direkte Kopplung von TK-Anlagen über eine Querverbindung ist nur in den Fällen vom Interesse, wo eine kostengünstige Infrastruktur zur digitalen Datenübertragung zur Verfügung steht. Dabei ist darauf zu achten, dass die Verfügbarkeit des Telefondienstes durch eventuellen Ausfall der Querverbindungen nicht wesentlich beeinträchtigt wird.

Zusätzlich zu berücksichtigen ist die Verkehrscharakteristik der Sprachverbindungen zwischen einzelnen Einrichtungen, siehe Abbildung 43:

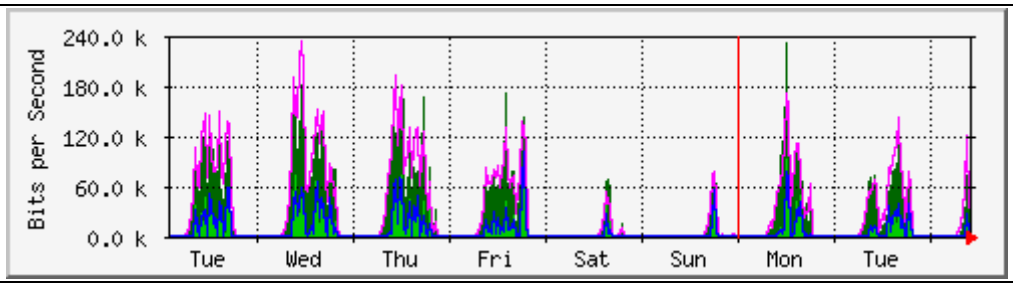

Abbildung 43 – Wochenstatisktik der Nutzung der VoIP Strecke

Der Sprachverkehr weist eine sehr hohe Burstiness auf. Die Übertragungsrate in der sogenannten Hauptverkehrsstunde ist vielfach höher als die über einen Tag gemittelte Übertragungsrate. Verbindet man die Telefonanlagen über eine Standleitung, so muss sie zwangsläufig für die Hauptverkehrsstunde dimensioniert werden. Eine solche Dimensionierung hat jedoch zur Folge, dass die Standleitung wirtschaftlich schlecht ausgenutzt bleibt.

Mit dem Ausbau des Wissenschaftsnetzes vom B-WiN zum G-WiN steht eine hochratige Infrastruktur zur digitalen Datenübertragung zur Verfügung. In Anbetracht geringer Datenraten bei einer Telefonverbindung sind die Kosten für die Kopplung der TK-Anlagen (Datenraten von unter 2 Mbit/s) gering. Es sind aber noch notwendige Vorkehrungen für die Einhaltung von QoS zu schaffen.

Nutzt man zur Kopplung von TK-Anlagen ein Datennetz, so kann die unterschiedliche Verkehrscharakteristik der Datenströme im IP-Netz ausgenutzt werden. In Zeiten geringen Sprachverkehrs kann die für Sprache reservierte Bandbreite im Übertragungskanal für die Datenübertragung genutzt werden. Dabei muss aber gewährleistet werden, dass sobald Anforderungen für Telefonverbindungen im IP-Netz ankommen, eine bestimmte minimale Bandbreite zur Sprachdaten-Übertragung zur Verfügung steht. Dieses kann durch die Einführung eines QoS-Dienstes im IP-Netz realisiert werde. Außerdem dürfen bestimmte QoS-Parameter der Strecke nicht überschritten werden. Die Grenzwerte für die Parameter der Übertragunsstrecke sind in den ITU-Standards G.113 und G.114 beschrieben, siehe Tabelle 6 und Tabelle 7.

Ta (ms)	*Idd* (eif)
150	0
200	3
250	10
300	15
400	25
500	30
600	35
800	40

Tabelle 6 - Relationship between one-way delay and Idd

Least Cost Routing Funktionalität

Die Demonopolisierung des Telefonmarktes in Deutschland im Jahre 1998 hatte auch die Einführung neuer Features hinsichtlich des Routing in den TK-Anlagen zur Folge. Die Routing-Entscheidungen können nun nicht nur aus geografischen, sondern vielmehr aus preislichen Gesichtspunkten getroffen werden. Folglich wurden Least Cost Routing-Funktonen mit neuen Software-Updates von den meisten Herstellern von TK-Anlagen angeboten. Unter diesem Aspekt kann eine IP-Kopplung als eine alternative kostengünstige Querverbindung in den TK-Anlagen konfiguriert werden.

Upper limit for *Icpif*	Speech communication quality
5	Very good
10	Good
20	Adequate
30	Limiting case
45	Exceptional limiting case
55	Customers likely to react strongly (complaints, change of network operator)
NOTE – *Itot,* in the equation *Icpif Itot – A,* is very near in numerical value to the decrease in *R*-rating, caused by similar impairments, of the Bellcore Transmission Rating model, described in Supplement No. 3 to the P-Series Recommendations.	

Tabelle 7 - Quality levels as function of the total impairment vaue Icpif

A.1.2 Schritt 2: Einführung einer IP-Telefonie-Umgebung im lokalen Netz

Im Gegensatz zum Schritt 1 geht es bei der Integration von IP-Telefonen in ein lokales Netz im Wesentlichen um eine Anbindung von Endgeräten unterschiedlicher Hersteller in eine Telefonie – Umgebung. Dabei werden die Endgeräte nicht über eine separate Telefon-Verkabelung, sondern über das Datennetz miteinander und mit einer Vermittlungs- bzw. Management-Einrichtung verbunden. Zu diesem Zweck sind standardkonforme Endgeräte einzusetzen. In den vergangenen Jahren hat sich im Bereich der IP-Telefonie der ITU-T-Standard H 323 durchgesetzt. Dieser bietet aber bislang einige Einschränkungen bei einer Abbildung eines POT-Netzes auf eine IP-Umgebung, siehe Kap. 3. Alternativ dazu können Systemtelefone eines einzelnen Herstellers eingesetzt werden. Diese können dann nur über ein entsprechendes Gateway mit anderen Netzen kommunizieren. Wichtig dabei ist, dass der eingesetzte Gateway nach außen als ein standardkonformes Endgerät bzw. Gateway agiert. Zwei wesentliche Merkmale charakterisieren eine standardkonforme lokale IP-Telefonie-Umgebung:

- Die Interoperabilität mit Geräten anderer Hersteller ist relativ einfach realisierbar. Es ist eine Anbindung an Informationssysteme wie z.B. zentrale Adressdatenbanken etc. über weitere offene Schnittstellen realisierbar.
- Die Endgeräte haben eine gewisse Komplexität und besitzen in der Regel einen eigenen Prozessor. Somit lassen sich auch zusätzliche Dienste, wie z.B. Abfrage von Telefonnummern aus einem Directory, Integration von Unified Message Systeme realisieren. Außerdem besteht die Möglichkeit zur Realisierung eines IP-Telefones als Anwendungsprogramm auf einem PC bzw. einer Workstation.

Bezüglich der QoS bestehen hier dieselben Anforderung wie beim Schritt 1. Da es sich bei der Netzverwaltung i.d. Regel um eine administrative Einheit handelt, ist die Einrichtung und Einhaltung bestimmter Dienstgüte einfacher als in einem WAN realisierbar. Außerdem ist auch ein Overprovisioning mit einem geringeren Kostenaufwand als in einem bundesweiten Netz realisierbar.

A.1.3 Schritt 3: Migration zur netzübergreifenden H 323-Infrastruktur

Die Integation der Schritte 1 und 2 führt zu einer einheitlichen H.323- Installation. Dieses kann z.B. über einen H 323-Gateway stattfinden. Bild 4 zeigt die Struktur einer solchen netzübergreifenden VoIP-Lösung. Hierbei ist auf einen konsistenten Rufnummernplan sowie auf eine Möglichkeit zur Ausscheidung in das öffentliche Telefonnetz im Falle der Verbindungsunterbrechungen im WAN zu achten. Außerdem müssen zusätzliche Maßnahmen zur Bereitstellung der Notdienste bei Stromausfall oder im Katastrophenfall ergriffen werden.

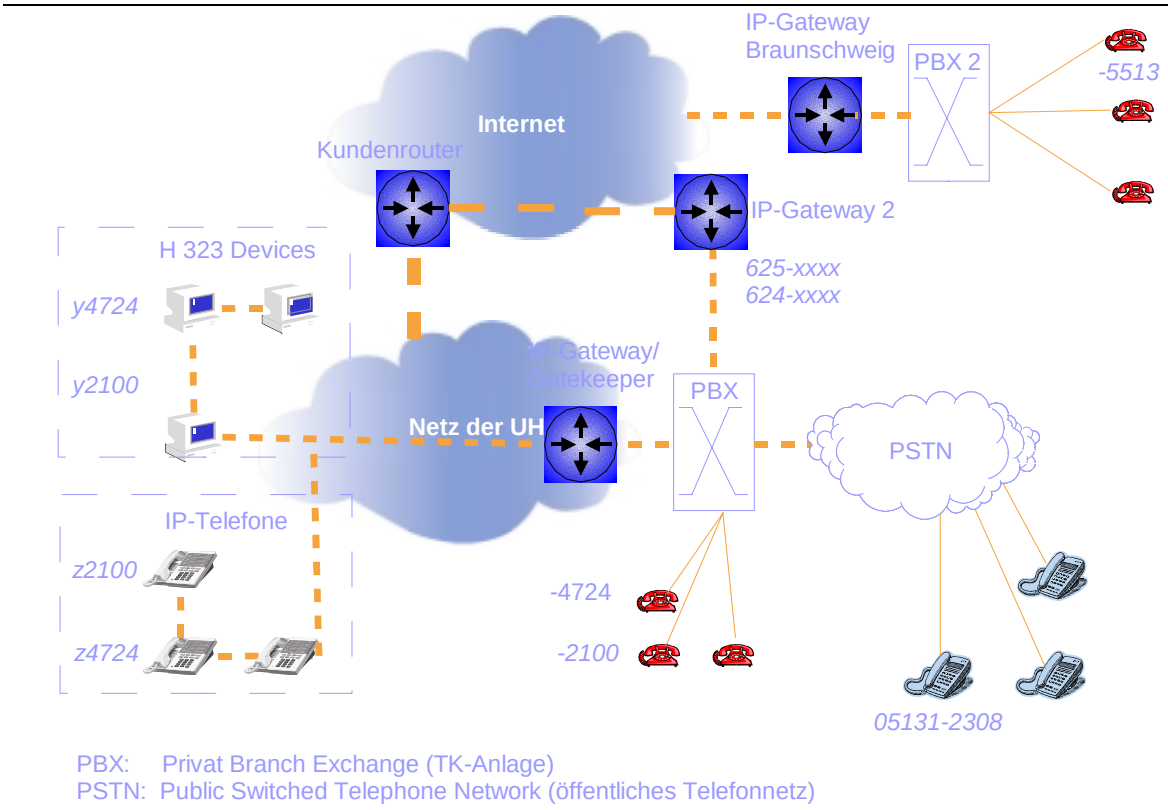

Abbildung 44 - Strukturbild für ein IP-Netz im vollen Ausbau

A.2 TK-Anlagen-Kopplung Hannover-Braunschweig

Ein erster Migrationsschritt zu einer VoIP-Lösung ist zwischen der Universität Hannover und der Technischen Braunschweig vorgenommen worden. Abbildung 45 zeigt das Strukturbild zu der beschriebenen Implementierung.

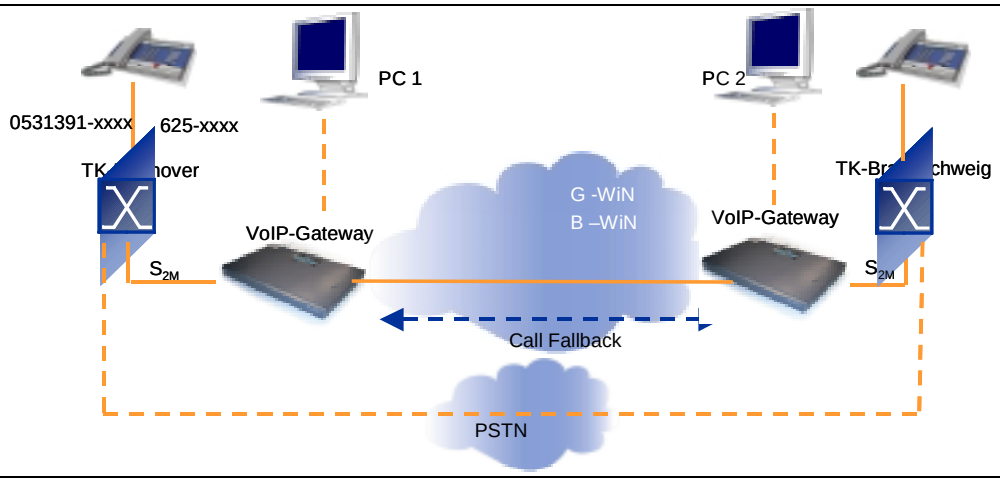

Abbildung 45 - Strukturbild der TK-Anlagen-Kopplung zw. Hannover und Braunschweig

A.2.1 Aufbau der TK-Kopplung zwischen der Universität Hannover und der TU Braunschweig

Eingesetzte Komponenten:

1. Cisco 2610
2. Cisco 2621
3. Voice Gateway S2m
4. Pri-Interface für den Cisco 26XX
5. TK-Anlage SOPHO 3070 (Philips)
6. TK-Anlage HiCom 300 (Siemens)

Die VoIP-Gateways sind sowohl in Hannover als auch in Braunschweig über jeweils einen Catalyst-Switch an einen B-WiN/G-WiN-Router direkt angeschlossen. Da das G-WiN zur Zeit keine QoS-Unterstützung bietet, läuft die Verbindung durch das G-WiN- Netz als Best-Effort-Datenstrom. Somit stehen die Datenpakete der IP-Telefonie in Konkurrenz zu anderen Datenströmen.

Zwischen den VoIP-Gateways wird unbedingt ein geswitchtes Netz benötigt, d.h.. es gibt in der installierten Umgebung keine Shared-Media Links. Dieses ist eine Mindestbedingung für ein managebares oder sogenanntes „predictable“ Netz. Die Tests liefen in Braunschweig über das ältere B-WiN. Dieses war in der Hauptverkehrsstunde sehr stark belastet, so dass die Belastung in den Zeiten zwischen 14 und 16 Uhr auf den Links die 5-minutigen Mittelwerte von 60% überstieg. Zeitweise kam es auch zu Paketverlusten.

Zwischen den TK-Anlagen und den VoIP-Gateways wurden jeweils synchrone digitale Übertragungsstrecken aufgebaut. Als Übertragugsprotokoll der Schicht 1 und Schicht 2 kommt ITU-T G. 703 und G.704 zum Einsatz. Als Signalisierungsprotokloll wird QSig eingesetzt. Als ISDN-Protokoll wird Primär Multiplex mit QSig als D-Kanal-Protokoll eingesetzt. Als Supplementary Services ist nur die Rufnummernübermittlung des rufenden Teilnehmers eingesetzt. Weitere Leistungsmerkmale wurden nicht getestet, da die gekoppelten Anlagen keine weiteren gemeinsamen Services unterstützen. Die VoIP-Implementierung ist aus Sicht der jeweiligen TK-Anlage eine Querverbindung mit 2 Mbit/s, über die einzelne B-Kanäle in Richtung Braunschweig vermittelt werden.

A.2.2 Konfiguration einzelner Komponenten für die TK-Anlagen-Kopplung

Wie bereits oben erwähnt, wird die Verbindung zwischen VoIP-Gateway und TK-Anlage als Bündel zu einer anderen TK-Anlage konfiguriert. Somit kann dieses Bündel explizit über eine Querverbindungs-Kennzahl angewählt werden. Die Berechtigungen werden für das ganze Bündel gesetzt. Jedes Telefon im Universitätsnetz, das eine Querverbindung anwählen darf, kann über die Kennzahl 625 in das Netz der TU Braunschweig aussteigen. Da sowohl die TK-Anlage in Hannover als auch in Braunschweig das Least Cost Routing unterstützt, ist die Querverbindung als der kostengünstigste Weg für das Präfix 0531-391- (die Vorwahl für Braunschweig und die Kennziffer für die TU Braunschweig) in der TK-Anlage der Universität Hannover konfiguriert worden. Somit wird die IP-Verbindung auch für normale „Amtsverbindungen“ als erster Weg angewählt.

Außerdem sind Querverbindungen zwischen der TK-Anlage der Universität Hannover und dem Behördennetz in Hannover konfiguriert. Diese Bündel sind auch berechtigt, die Querverbindung nach Braunschweig zu nutzen. Folglich ist es möglich, von einem Telefonanschluß im Behördennetz über das Netz der Universität Hannover die Technische Universität Braunschweig anzuwählen. Dafür wird erst die Kennzahl für die Querverbindung zum Universitätsnetz und anschließend die Vorwahl für die IP - Verbindung nach Braunschweig gewählt. Wird ein Least Cost Routing ähnlich zur Konfiguration in der Universität Hannover konfiguriert, so kann die Anwendung von speziellen Kennzahlen für die Querverbindungen entfallen.

Das Voice over IP –Gateway erledigt im wesentlichen folgende Aufgaben:

- Beim Verbindungsaufbau wird die im QSig-Protokoll empfangene Information ausgewertet. Anhand der Ziel – Rufnummer wird eine Routing-Entscheidung getroffen. Dabei kann die Gegenstelle sowohl ein VoIP-Gateway als auch ein H 323- Endgerät sein. Dafür wird für jedes Rufnummernmuster ein sogenanntes Dial-Peer im Gateway angelegt. Jedes so angelegte Dial Peer wird dann einzeln konfiguriert. Es kann somit für jedes Ziel ein bestimmter Codec, Wählverfahren Time-Outs, u.v.m. konfiguriert werden.
- Es können außerdem für jedes Dial-Peer bestimmte Regeln zur Manipulation der Ziel- oder Quell-Rufnummer definiert werden. (Siehe dazu den Abschnitt „Realisierung der Caller-ID-7“). Wichtig ist zu berücksichtigen, dass das Voice over IP-Gateway lediglich die Informationen, die den Verbindungsauf- und -abbau betreffen, interpretiert. Werden weitere, vom Cisco-Gateway nicht erkannte Nachrichtenelemente von einer Anlage gesendet, so werden diese als Generic Functions transparent an die Gegenstellen übermittelt.
- Die Abtastwerte werden gebuffert und ggf. mit dem gewählten Kodierungsverfahren umkodiert. Die kodierten Sprachpakete werden dann in IP-Pakete eingekapselt und über das IP-Netz an das Dial-Peer gesendet. Entsprechend werden die Sprachpakete an der Gegenseite empfangen, die Integrität geprüft und dann als synchroner Bitstrom auf die S_{2M}-Leitung gegeben. Die Kommunikation zwischen den Dial-Peers läuft nach dem H-323 - Standard ab.

Besondere Vorkehrungen bei der Wahl unterschiedlicher Codecs.

Die Abtastwerte der Sprachsignale werden in den Telefonnetzen gemäß ITU-T G 711, anhand der A-Law-Kompandierungskennlinie kodiert. Wird dieser Codec im VoIP-Gateway eingesetzt, so werden die Abtastwerte im Gateway lediglich gebuffert und in das IP-Protokoll eingekapselt. Dabei entsteht in jedem Gateway eine Verzögerung von 20 ms und ein zusätzlicher Overhead von ca. 16 kbit/s pro Sprachverbindung. Die Brutto – Datentrate pro Sprachverbindung beträgt dabei ca. 80 kbit/s .

Es besteht auch die Möglichkeit, den Sprach-Datenstrom weiter zu komprimieren, wenn z.B. Codecs nach den Standards ITU-T G.728, G.729 und G.723 eingesetzt werden. Dabei ist aber mit einigen Beeinträchtigungen des Dienstes rechnen:

- Die Qualität der Sprachverbindung wird schlechter
- Da die oben aufgelisteten Codecs die Statistik menschlicher Sprache ausnutzen, kann die Fax-Übertragung sowie die Dual Tone Modulation Frequency Erkennung (DTMF-Erkennung) bei einer Verbindung zu Interactive Voice Responce - Systemen beeinträchtigt werden. Um diese Probleme zu umgehen, gibt es die Möglichkeit, das Fax- und DTMF-Code-Relaying im Cisco-VoIP-Gateway zu aktivieren. Weiterführende Tests dieser Features wurden bislang nicht durchgeführt.

Die an das Übertragungsnetz gestellte Anforderungen bezüglich QoS steigen bei stark komprimierenden Verbindungen.

A.2.3 Die Call-Fallback-Implementierung

Damit über das IP-Netz ein hoch verfügbarer Dienst angeboten werden kann, ist bei der TK-Kopplung zwischen Hannover und Braunschweig eine Call Fallback Lösung implementiert worden.

Dabei wird in beiden VoIP-Gateways ein sogenannter Service Assurance Agent (SAA) installiert. Dieser sendet auf Anforderung Testpakete an die Gegenstelle. Anhand der Antwort der Gegenstelle werden Statistiken bezüglich der Strecke, wie z.B. Einweg-Verzögerung, Paketverluste, oder eingesetzter Codec für die Sprachverbindung, etc. in einen internen Cache geschrieben. Kommt ein Ruf seitens der TK-Anlage beim Gateway an, so wird, bevor ein Call Setup an die Gegenstelle gesendet wird, der Cache abgefragt. Befinden sich keine Messergebnisse im Cache so wird mit dem

SAA eine Messung durchgeführt. Ist das Messergebnis positiv, so wird ein H 323 Call Setup an die Gegenstelle gesendet. Sind die Ergebnisse negativ, so wird ein Disconnect mit einer entsprechenden Cause Message an die TK-Anlage gesendet. Die TK-Anlage erkennt anhand der Cause Message, dass die Verbindung wegen eines Fehlers im Netz abgewiesen ist, worauf hin eine Verbindung über das öffentliche Netz aufgebaut wird. Man kann wahlweise konfigurieren, ob feste Grenzwerte für die Einweg-Verzögerung oder Paket-Verlust-Rate dem VoIP-Gateway vorgegeben werden. Wird einer dieser Werte überschritten, so werden neue Verbindungsanforderungen abgewiesen. Bestehende Verbindungen werden davon nicht betroffen. Alternativ kann man dem Gateway vorgeben, dass aufgrund der Messung ein skalarer Wert – I_{cpif} nach dem ITU-T Standard G 113 berechnet wird. Außer der Verzögerung und den Paketverlusten wird dann zusätzlich der Einfluß der Codecs in den I_{cpif} - Faktor eingerechnet. Im praktischen Betrieb hat es sich aber erwiesen, dass nur der erste Fall sichere Ergebnisse liefert.

Der Einsatz des SAA hat folgende Vorteile:

1. Dieser Vorgang ist für den Teilnehmer transparent. Man muss keine Neuwahl beim Ausfall der IP-Strecke vornehmen.
2. Durch das Cachen der Messergebnisse reduziert sich bei hoher Netzbelastung die Verzögerung für den Verbindungsaufbau.
3. Die Verfügbarkeit der Strecke zwischen zwei Endpunkten ist mindestens so hoch wie die Verfügbarkeit im öffentlichen Netz.

Auf der anderen Seite weist dieses Verfahren folgende Probleme auf:

1. Wird ein Re-Routing über das öffentliche Netz vorgenommen, entstehen Kosten, von denen der Nutzer keine Kenntnis hat. Auch für den Betreiber s der Kopplung stellt sich das Problem der genauen Gebührenzuordnung für solche umgeleitete Verbindungen. Dieser Punkt bedarf noch weiterer Untersuchungen.
2. Es existiert keine einfache Möglichkeit den Call Fallback Cache abzufragen, z.B. mit SNMP. Als Workaround ist im Rahmen des Projekts ein Tool geschrieben worden, das von einem externen Rechner gestartet wird und in regelmäßigen Abständen den Call Fallback Cache über das CLI (Command Line Interface) abfragt und auswertet. Überschreiten die Messergebnisse die konfigurierten Grenzwerte, so wird eine rudimentäre Untersuchung der Strecke vorgenommen und das Ergebnis mit dem Inhalt des Call Fallback Caches an den Systemadministrator per Email gesendet.

A.2.4 Erfahrungsbericht über den praktischen Betrieb

Zusammenschaltung der TK-Anlage mit dem VoIP-Gateway

Die Zusammenschaltung der TK- Anlage mit dem VoIP- Gateway wurde über eine G.703/G.704 Schnittstelle realisiert. Dabei sind mehrere Optionen zu berücksichtigen, so z.B. die Zählung der B-Kanäle 1-15 und 16-31 oder ein kontinuierliches Durchzählen von 1 bis 30. Stimmen die Einstellungen auf beiden Seiten nicht überein, so werden von beiden Seiten unterschiedliche B-Kanäle belegt. Die TK-Anlagen lassen sich hierzu meistens nicht konfigurieren, somit muss der VoIP-Gateway die entsprechende Flexibilität ermöglichen.

Dasselbe betrifft auch die CRC-Prüfung auf Schicht 2. Man muss sich auf einem Verfahren einigen, damit der Rahmen von der Gegenstelle erkannt wird.

Bei einer Kommunikation zwischen Vermittlungseinrichtungen über das QSig- Protokoll besteht zwischen diesen immer eine Master-Slave Beziehung. Die VoIP-Gateways lassen sich zur Zeit nur im Slave-Modus betreiben.

Sowohl im QSig als auch im EDSS1 sind insgesamt 127 unterschiedliche Cause- Messages definiert, die die Ursache eines Disconnects genauer beschreiben. Eine Klasse daraus bilden die „QoS-

Unavailable“-Messages. Diese werden standardmäßig als Ursache für ein Disconnect bei einem negativen SAA-Ergebnis an die TK-Anlage übermittelt. Die meisten TK-Anlagen können aber nur eine einzige Cause-Message für das Re-Routing verwenden. Folglich mussten bei der Implementierung in Hannover alle Cause Messages aus der „QoS-Unavailable“-Klasse auf die Message „No route to destination“ abgebildet werden. Nur so lässt sich bei einer SOPHO-Anlage eine Fallback Lösung realisieren.

A.2.5 Übermittlung der CallerID

Eines der mit dem QSig Basic Call realisierbaren Features ist die Übermittlung der rufenden Nummer an den gerufenen Teilnehmer (CallerID). Die CallerID wird wie die Ziel-Rufnummer beim Call Setup an das entfernte VoIP- Gateway übermittelt. Dabei ist darauf zu achten, dass die Querverbindungs-Kennzahl für die IP-Strecke bzw. das Präfix für die Einrichtung (z.B. 0511-762- bei der Universität Hannover) von der Vermittlungstelle mitgesendet werden. Findet dies nicht statt, so könnte das Präfix im Router mit einer Translation Rule vorangestellt werden. Wird das Voranstellen der Querverbindungs-Kennzahl vor die CallerID im VoIP Gateway vorgenommen, so tritt das Problem auf, dass die Quell-Rufnummer dem Quell-Netz (Universitätsnetz vs. Berhördennetz) nicht mehr eindeutig zuzuordnen ist. Man kann aber sehr wohl die Querverbindungs-Kennzahl im VoIP-Gateway durch den PSTN-Präfix für das entsprechende Netz ersetzen (z.B. wird 625- durch eine 0511-762- ersetzt).

A.3 Betrieb eines Cisco Call Manager mit Selsius Telefonen

Als lokale VoIP - Infrastruktur im Universitätsnetz ist ein Call Manager der Firma Cisco beschafft und installiert worden. Dabei handelt es sich im Wesentlichen um eine proprietäre Lösung, die sich bei der Kommunikation mit anderen VoIP-Zonen wie eine H.323 - Komponente verhält. Der Call Manager wurde in der Version 2.1 getestet.

A.3.1 Gründe zur Einführung des Skinny Station Protokolls

Zur Kommunikation zwischen den Celsius Telefonen und dem Call Manager wurde bei Cisco ein hauseigenes, sogenanntes Skinny Station Protokoll, entwickelt. Da der Call Manager als Ersatz für TK-Anlagen in Firmennetzen konzipiert ist, sind einige Änderungen gegenüber dem H.323 in das Skinny Station Protokoll eingeführt worden. Nachfolgend werden einige Einschränkungen genauer erläutert:

- Die IP-Telefone sind vollständig vom Call Manager abhängig und können mit einen anderen Telefon keine direkte Verbindung aufnehmen.
- Beim Anschalten des Telefons an das Datennetz wird mit Hilfe einer Discovery Routine nach einem Cisco Call Manager im Netz gesucht. Dieses kann z.B. über einen DNS-Eintrag oder über einen DHCP-Server erfolgen. Alternativ kann die IP-Adresse des Call Managers manuell im Gerät eingetragen werden, um beliebige Netzlösungen zwischen den IP-Telefinen und dem Call Manager zu erlauben.Wird ein Call Manager gefunden, so wird die restliche Konfiguration des Telefons vom Call Manager mit Hilfe des TFTP- Protokolls vorgenommen. Im Gegensatz zum Ablauf im H.323 Protokoll, kann das Telefon sich nicht mit einer eigenen Telefonnummer am Call Manager anmelden. Das von Cisco verwendete Modell der Rufnummernvergabe entspricht in etwa dem Modell in klassischen Telefonnetzen.
- Als eindeutige Kennzeichnung eines Telefons dient für den Call Manager die MAC-Adresse eines Selsius-Telefons. Damit eine Identifikation über beliebige Netze ermöglicht wird, wird die MAC-Adresse im Skinny Station Protokoll an den Call Manager übertragen. Dieses kann als Sicherheitslücke betrachtet werden, da die MAC-Adresse von einem beliebigen anderen Gerät dem Call Manager vorgetäuscht werden kann.
- Eine User-basierte Vergabe von Rufnummern und Berechtigungen ist im Call Manager dieser Version nicht möglich. Dieses ist aber für spätere Versionen des Call Managers angekündigt worden.

- Wesentlich ist auch der Unterschied bei der Rufkontrolle im H.323 und dem Skinny Station Protokoll. Im H.323 gibt es zwei Möglichkeiten einer Verbindung zwischen H.323 – Endgeräten:
  - Bei einem Gatekeeper-routed Call laufen die Verbindungsdaten, Kontrolldaten sowie die Gesprächsdaten über den Gatekeeper. Dieser Modus ist ineffizient bezüglich der Lastverteilung, da der Gatekeeper alle Sprachpakete vermitteln muss. Findet ein Netzausfall zwischen zwei am Gespräch beteiligten Komponenten so fällt auch die Verbindung aus. Dieses mag als Nachteil erscheinen, ist aber aus Sicht eines Netzbetreibers eher positiv zu betrachten, da die Integrität der Call Detail Records hierbei gewährleistet ist. Es ist aber offensichtlich, dass diese Lösung für große VoIP-Installationen nicht skalieren kann.
  - Beim direct-routed Call wird lediglich der Verbindungsaufbau und –abbau vom Gatekeeper kontrolliert. Dieses ergibt einen hohen Performancegewinn gegenüber den Gatekeeper Routed Calls. Fällt aber eine Verbindung zwischen dem Gatekeeper und einem Endgerät aus, so ist dadurch das Telefonat nicht zwangläufig unterbrochen. Folglich kann eine Inkonsistenz bei den Call Detail Records stattfinden. Besonders bei zeitabhängiger Tarifierung der Gespräche wird dieses zu einem Problem.

Beim Skinny Station Protokoll werden die Sprachpakete zwar direkt zwischen den Endgeräten ausgetauscht, es laufen aber ständig sogenannte Keep Alive Messages zwischen dem Call Manager und den Selsius-Telefonen. Fallen die Keep Alive Messages mehrfach nacheinander aus, so wird das Gespräch vom Selsius-Telefon beendet.

A.3.2 Erfahrungen aus dem Testbetrieb des Call Managers

Der Testbetrieb des Call Managers in der Version 2.0 mit 10 IP-Telefonen Selsius 12SP+ und Selsius 30 VIP hat folgendes gezeigt:

- Sobald eine Installation abgeschlossen ist, läuft diese robust und zuverlässig. Man konnte keinerlei Beeinträchtigungen aufgrund von Überlastungen im Netz feststellen. Dieses kann einerseits auf die kleine Größe der VoIP-Pakete und andererseits auf die geringe Datenrate in Relation zur Link-Datenrate zurückgeführt werden.
- Wird die IP-Adresse des Call Managers manuell eingetragen, so muss diese bei einer Änderung der IP-Adresse des Call Managers in allen IP-Telefonen neu eingetragen werden. Folglich ist für mittlere bis große Installationen die Auflösung der Call Manager Adresse über einen DHCP-Server oder über einen DNS-Eintrag zu bevorzugen.
- Der Call Manager läuft unter dem Betriebssystem Windows NT. Folglich müssen nach den meisten Konfigurationsänderungen die Services bzw. der ganze Call Manager neu gestartet werden, damit die Änderungen wirksam werden.
- Eine Kommunikation zwischen einem H 323 – Gatekeeper und dem Call Manager in der Version 2.1 ist nicht möglich. In der Version 3.0 soll so eine Kommunikation mittels eines Cisco VoIP-Gateways möglich sein.

A.3.3 Redundanz und Ausfallsicherheit im Call Manager 3.0

Eine der zentralen Anforderungen an eine Telefon-Installation ist die Ausfallsicherheit. Um diese zu erhöhen, ist ab der Version 3.0 ein Clustering von Call Managern möglich. Damit wird jedem IP-Telefon ein Primary - und ein Backup – Call Manager zugeordnet. Fällt der primary Call Manager aus, so übernimmt der Backup Call Manager die Überwachung und Steuerung des IP-Telefon-Clusters.

Damit der Betrieb der IP-Telefone auch bei Stromausfall möglich ist, können die Cisco IP-Telefone der neuen Generation (Cisco 7960 und Cisco 7910) ihre Stromversorgung über den Datennetz-

Anschluss beziehen. Entsprechend sind auch Switch-Boards, die eine solche Stromversorgung liefern, verfügbar. Da zum Zeitpunkt der Erstellung dieses Berichts noch kein Call Manager der Version 3.0 mit den IP-Telefonen Cisco 7960 und 7910 zur Verfügung stand, muss dieser im Rahmen zukünftiger Test untersucht werden.

A.4 Bericht zur Verbindung zwischen zwei H323-Zonen

Im Rahmen des Interoperabilität - Tests wurde versucht, Verbindungen über die Grenzen einer Voice over IP-Zone herzustellen.

Dafür kam in Hannover NetMeeting 3.01 als H 323-Client und der im Rahmen des Projekts vom TZI entwickelte Gatekeeper GKNG zum Einsatz. Abbildung 46 und Abbildung 47 zeigen den Aufbau der Testumgebung für die Gatekeeper- Connectivität.

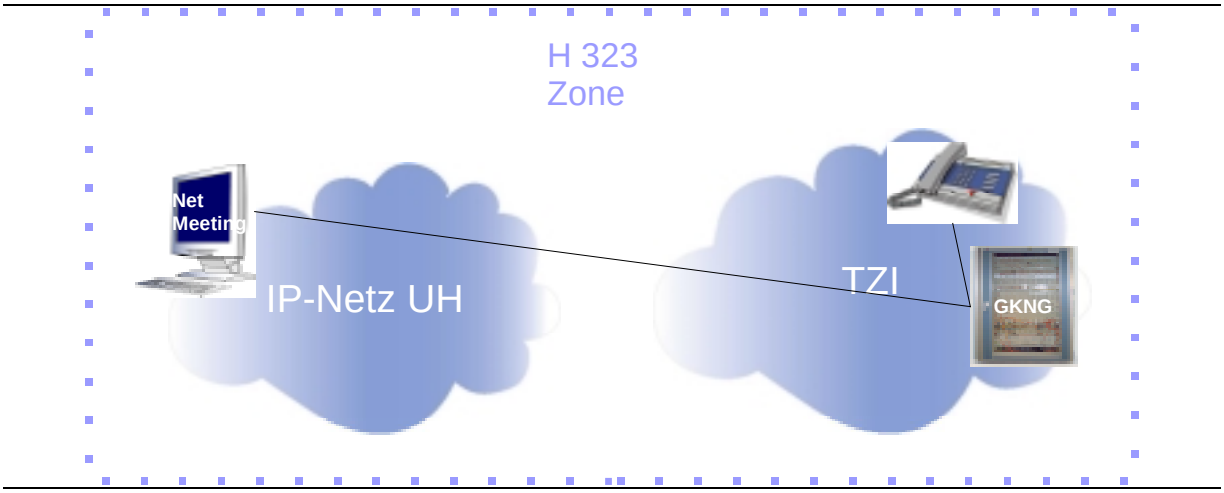

Abbildung 46 - Verbindung innerhalb einer H.323 Zone

Im ersten Schritt ist versucht worden, den H 323 Client in Hannover über den Gatekeeper in Bremen mit einem Client in Bremen zu verbinden. Diese Verbindung kam zustande. Ebenso funktionierte die Verbindung in der Gegenrichtung. Voraussetzung hierfür war, dass der Client in Hannover beim Gatekeeper in Bremen angemeldet war. Dabei wurde der Rufnummernbereich 1xxx für die Zone Bremen und 2xxx für die Zone Hannover vorgesehen.

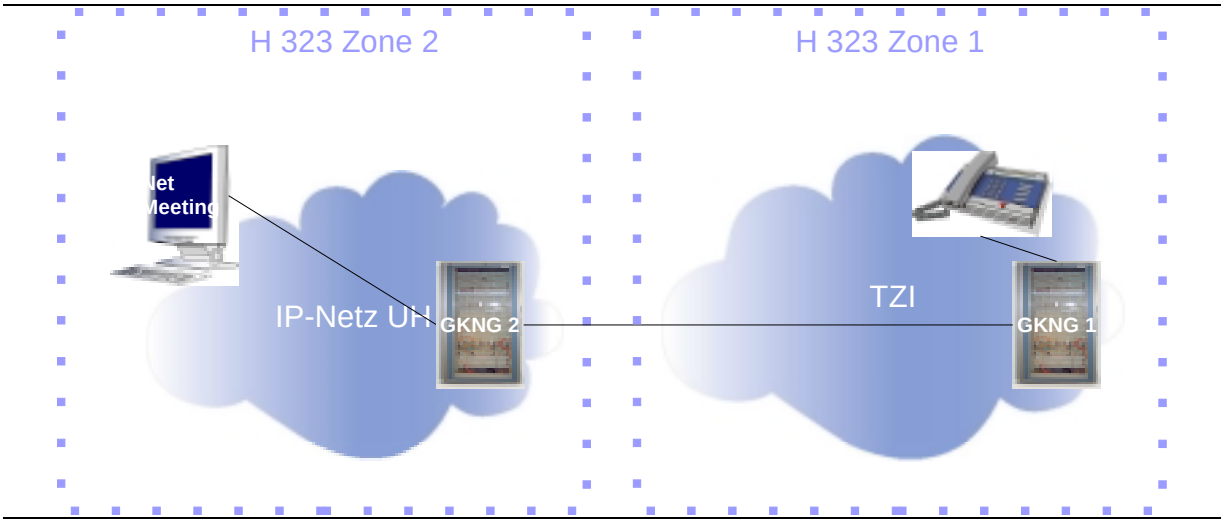

Abbildung 47 - Verbindung über die Grenzen einer H.323-Zone hinweg

Im zweiten Schritt wurde ein Gatekeeper GKNG im Netz der Universität Hannover aufgesetzt. Auch hier konnten sich die H 323 Clients beim Gatekeeper registrieren und die Rufnummern auflösen. Somit konnte auch in Hannover lokal mit Hilfe des Gatekeepers telefoniert werden.

Im letzten Schritt wurde versucht, von einem Client in Hannover mittels des Gatekeepers in Hannover die Adresse eines IP-Telefons in Bremen aufzulösen. Bei diesem Versuch konnte eine Verbindung vom Gatekeeper in Hannover zum Gatekeeper in Bremen aufgebaut werden. In der Gegenrichtung konnte jedoch keine Verbindung aufgebaut werden. Es sind in dieser Hinsicht noch weitere Tests erforderlich.

B SIP Telephony: Current Status, Challenges and Solutions

B.1 Hintergrund -IP-Telefonie mit SIP

B.1.1 Protokollabläufe und Komponenten

SIP ist ein *Request-Response* Protokoll. Will ein Benutzer erreichbar sein, muss er sich zuerst mit Hilfe seiner Signalisierungssoftware (die *User Agent* genannt wird) bei einem *Registrar Server* mit einer E-mail ähnlichen Adresse anmelden. Danach können ankommende Gespräche an seinen jeweiligen Standort geleitet werden. Initiiert der Benutzer einen Anruf selbst, muss sein *User Agent* zuerst den Server finden, bei dem sich der Angerufene registriert hat. Dies erfolgt mit Hilfe von DNS anhand der SIP Adresse des Angerufenen. An diesen Server wird eine SIP Einladung geschickt (Schritt 1 in Abbildung 48). Der Server findet den aktuellen Benutzerstandort des Angerufenen und leitet die Einladung an ihn weiter (das sogenannte *Proxying* - 4). Es ist auch möglich, daß die Einladung an mehrere Zielstellen gleichzeitig weitergeleitet wird, damit der Angerufene auch in dem Fall gefunden wird, in dem er über mehrere Stellen erreichbar sein will. Alternativ zum Proxying könnte der SIP Server die Zieladresse an den Rufenden weiterleiten, damit er sich selbst mit dem Angerufenem in Verbindung setzt. Dieses Verfahren wird als *"Redirection"* (Umleitung) bezeichnet.

Erreicht die Einladung schließlich den Angerufenen, kann er sich entscheiden, ob er sie entgegennimmt, umleitet, oder ablehnt. Das kann mit Hilfe einer Anrufbearbeitungssoftware automatisiert werden, so dass z.B. Anrufe von unbekannten Personen ohne Eingriff des Nutzers an einen Anrufbeantworter umgeleitet werden können. Entscheidet sich der Angerufene, den Anruf entgegen zunehmen, schickt er eine positive Antwort zurück (5). Daraufhin beginnen beide Gesprächsteilnehmer mit dem Senden von Datenströmen.

Kompliziertere Fälle können auch den Einsatz mehrerer Proxies erforderlich machen – z.B. wenn ein Unternehmen sehr groß ist und mehrere Proxies zur Lastverteilung einsetzt. Oder es wird verlangt, daß der Anruf auch durch den Proxy Server des Rufenden geleitet wird. Dies ist besonders dann wichtig, wenn dieser zusätzliche Signalisierungsdienste anbietet (z.B. Umleitung eines Anrufs in PSTN über das kostengünstigste Gateway) oder Firewall Kontrolle ausübt.

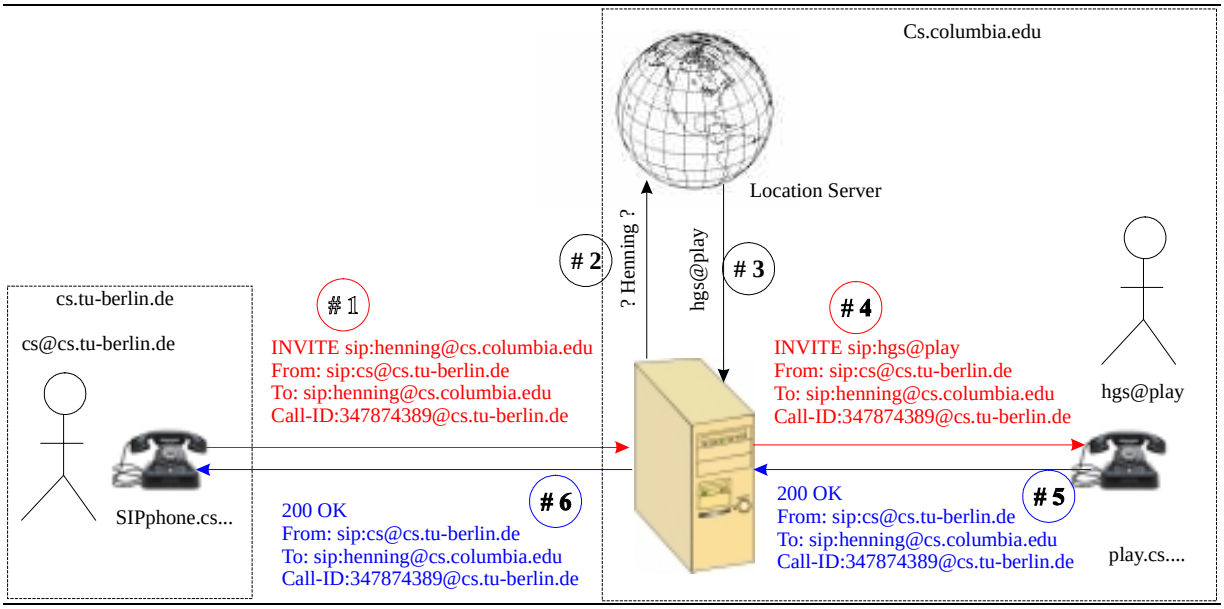

Abbildung 48 - Ein SIP Anrufszenario

B.1.2 SIP Dienstmodel

Eine der fundamentalen Eigenschaften, auf denen das SIP Konzept beruht, ist das end-to-end Kommunikationsmodel. Dieses Model sieht vor, daß sich die Anwendungsintelligenz in Endgeräten

(sowohl in Clients als Servern) befindet, während das Netzwerk ausschließlich eine Transportfunktion ausübt. Das heißt, daß jeder mit IP Konnektivität imstande ist, Dienste für andere Internetbenutzer anzubieten. Wir glauben, daß diese Offenheit eine Voraussetzung für die Verbreitung der Internet Telefonie ist. Nur so können schnell innovative Dienste durch Dritte bereitgestellt werden. So kann zum Beispiel ein Mehrwertdienstanbieter einen SIP Proxy betreiben, der eine Liste von Spamquellen führt, und alle Anrufe, von einer bekannten Spamquelle eliminiert. Die Lokalisierung von preisgünstigen PSTN Gateways könnte ebenfalls von einem SIP Proxy als zusätzlicher Dienst angeboten werden.

Die für Internet Telefonie erforderlichen Dienste lassen sich in folgende Klassen einteilen:

- *Transport Infrastruktur*: IP Zugang, DNS Dienste (inkl. ENUM), QoS Dienste.
- *Signalisierungsdienste*: Basissignalisierungsdienste, IN-ähnliche Dienste (alle Arten von Umleitungen, Antispam, Auswahl des preisgünstigsten PSTN Gateways, geographischorientierte Dienste, usw.), weitere Dienste, die auf SIP aufbauen (z.B. Instant Messaging and Presence)
- *Ergänzende Dienste*: Z.B. Anbindung an Voice Mail, Web Click-to-Dial, usw.
- *Zugang zu PSTN Gateways*; (inkl. Routing Infrastruktur)
- *Vertrauensvermittlungsdienste* - mit diesen durch dritte Seite gewährleisteten Diensten ist es möglich, eine Geschäftsverbindung zwischen mehreren Teilnehmern aufzubauen, auch wenn sie gegenseitig in keiner bilateralen Verbindung stehen. Dies ist zum Beispiel für domänenübergreifende Szenarios mit QoS Unterstützung hilfreich.

Einzelne Dienste können auch von unterschiedlichen Anbieter erbracht werden. So kann zum Beispiel ein Anbieter für IP Konnektivität sorgen, während ein anderer die Funktionalität eines SIP Servers bereitstellt, und ein dritter Zugang zu PSTN Gateways gewährleistet.

B.1.3 Vergleich mit H.323

Im Folgenden vergleichen wir SIP mit dem konkurrierenden Standard H.323. Wir verzichten auf den Vergleich mit MGCP/Megaco, da sie nach unserer Meinung in erster Linie nur als Lösungen zur Dekomposition von Telefonie Gateways angesehen werden können. Wir meinen, dass derartige Master-Slave Protokolle für den Einsatz zur Signalisierung zwischen Internet Telefonie Endgeräten nicht zu empfehlen sind, da sie die Entwicklung neuer Telefonie Diensten erschweren und nicht kompatibel mit dem Internet end-to-end Konzept sind.

Komplexität

Protokolle und Optionen

H.323 ist vergleichsweise komplex und beinhaltet mehrere Protokolle für unterschiedliche Aufgaben u.a.:

- H.225 für Anrufsignalisierung und Registrierung
- H.245 für Endsystemparameter und Mediensteuerung
- H.235 für Authentifizierung
- H.246 für Interoperabilität

Aus Gründen der Abwärtskompatibilität enthält H.323 viele Optionen (z.B. für die Zusammenarbeit von H.225 und H.245). Sie müssen sowohl von Endsystemen, als auch von Servern unterstützt werden. Bestimmte Funktionalitäten sind oft mehrfach vorhanden (z.B. bieten H.245 und RTCP ähnliche Mechanismen für Medien- und Konferenzkontrolle). Aus Dienstsicht ist keine klare Trennung zwischen den Protokollen zu erkennen.

Die Aufgaben von SIP konzentrieren sich vor allem auf Signalisierung und Benutzerlokalisierung. SIP Nachrichten enthalten alle notwendigen Informationen für den Aufbau und die Steuerung einer Multimediaverbindung. Zusätzliche Funktionen (z.B. Übertragung von Medien oder Konferenzmanagement) sind nicht Bestandteil von SIP und werden von anderen Protokollen (RTP, RTCP, RSVP usw.) realisiert. SIP hat nur wenige Methoden und Optionen und die Unterstützung dafür ist hauptsächlich in den Endsystemen notwendig.

Generierung/Kodierung/Parsing

H.323 verwendet ein binäres Format für den Aufbau von Protokollnachrichten. Die Generierung und das Parsen erfolgt über Komponenten, die zuvor aus einer textuellen Beschreibung der Syntax (ASN.1) erzeugt werden müssen. Um eine plattform-unabhängige Verarbeitung zu gewährleisten, müssen spezielle Kodierungsregeln eingehalten werden (ASN.1 Packet Encoding Rules).

SIP und SDP sind vollständig text-basierte Protokolle. Sie verwenden den ISO 10646 Zeichensatz (Unicode) und es ist kein Aufwand für zusätzliche Kodierungen nötig. Der Aufbau einer SIP Nachricht ist ähnlich zu HTTP und die Analyse kann durch einen einfachen Text-Parser erfolgen.

Implementierung/Debugging

Das Debugging von H.323 Anwendungen erfordert die Umsetzung von H.323 Nachrichten in eine lesbare Form. Die dafür verwendeten Hilfsmittel müssen bei Änderungen der Spezifikation an das neue Nachrichtenformat angepasst werden.

Debugging von SIP ist einfach, da es sich um reine Textnachrichten handelt. Es sind keine speziellen Hilfsmittel erforderlich. Zusätzlicher Aufwand bei Änderungen am Protokoll ist nicht vorhanden. Die Bezeichnung von Protokollelementen ist meist selbst-dokumentierend.

Erweiterbarkeit

Kompatibilität

Sowohl SIP, als auch H.323 verwenden Versionsnummern als Mechanismus zur Kontrolle von Protokollerweiterungen.

SIP stellt keine besonderen Anforderungen an die Kompatibilität zwischen unterschiedlichen Versionen. Unbekannte oder nicht-unterstützte Elemente des Protokolls werden ignoriert. SIP stellt Funktionen zur Überprüfung der von einer Implementierung unterstützten Methoden (*Require, Supported* header) bereit.

H.323 fordert zur kontinuierlichen Unterstützung existierender Dienste und Funktionen, die vollständige Kompatibilität zu älteren Versionen des Standards.

Protokoll- und Diensterweiterung

Bei H.323 sind die Basisdienste wie *Call hold* oder *Call transfer* fest in der Spezifikation verankert. Zusätzliche Protokollerweiterungen erfordern die Änderung des Standards. H.323 stellt für herstellerspezifische Anpassungen *NonStandardParameter* Strukturen zur Verfügung. Sie bestehen aus einem Herstellercode und zugeordneten Daten. Der Herstellercode ist Bestandteil der Protokolldefinitionen. Neue Hersteller müssen deshalb zur Einführung eigener Erweiterungen die Änderung des Standards beantragen.

Statt ISDN-ähnliche Dienste als Bestandteil des Standards zu spezifizieren, werden sie bei SIP mit der Basisfunktionalität des Protokolls realisiert. So kann z.B. der Dienst *Call forward on busy subscriber* mit Hilfe der in SIP eingebauten Benutzerlokalisierungs- und Umleitungsfunktion implementiert werden ohne die Spezifikation ändern zu müssen.

Direkte Erweiterungen von SIP erfolgen in der Regel durch Hinzufügen neuer Protokollheader. Der aktuelle SIP RFC definiert nur Standardmethoden und -header. Erweiterungen werden in separaten

RFCs oder IETF-drafts festgelegt. Hersteller-spezifische Anpassungen werden unterstützt, indem ihre Bezeichnungen in einem hierarchisch organisierten Namensraum verwaltet und bei der Internet Assigned Numbers Authority (IANA) registriert werden. Neue Methoden und Erweiterungen können so der Öffentlichkeit zugänglich gemacht und benutzt werden.

Skalierbarkeit

Teilnehmeranzahl

Ursprüngliches Einsatzgebiet von H.323 waren LANs. Um den Betrieb in weitverzweigten Netzen zu ermöglichen, wurde das Konzept der H.323 Zone eingeführt. H.323 beschreibt in "Annex G" die Kommunikation zwischen H.323 Zonen, und es gibt Prozeduren für die Benutzerlokalisierung anhand einer Email-Adresse. H.323 schlägt Methoden zur Behandlung von Signalisierungs-Schleifen vor, deren Umsetzung allerdings problematisch für die Skalierbarkeit sind.

SIP wurde für die Benutzung im Internet entworfen und unterstützt von Hause aus die Adressierung in weiterverzweigten Netzen. Sind mehrere Server im Ablauf eines Anrufs involviert, kommt ein ähnlicher Algorithmus wie BGP zur Erkennung von Signalisierungs-Schleifen (*loop detection*) zum Einsatz. Auswirkungen auf Skalierbarkeit ergeben sich nicht, da die Umsetzung zustandslos erfolgt. Für die Benutzerlokalisierung werden SIP Registrar und Redirect Server eingesetzt.

Zustandsverwaltung

Typischerweise wirkt sich bei Anrufen die Verwaltung von Zuständen in Servern negativ auf die Skalierbarkeit und Performance eines Multimediasystems aus. So ist es dann nicht mehr möglich, während eines laufenden Anrufs den Server zu wechseln, und beim Neustart des Server müssen zuvor alle aktuelle Zustände gerettet werden.

Obwohl H.323 zustandslose Verarbeitung erlaubt, sind die meisten Implementierungen für H.323 Gatekeeper oder Gateways zustandsorientiert.

SIP Server können sowohl zustandslos, als auch zustandsorientiert betrieben werden. Die meisten SIP Proxy Implementationen sind zustandslos.

Transportprotokolle

Die Eigenschaften des verwendeten Transportprotokolls wirken sich ebenfalls auf die Skalierbarkeit aus, da hier ggf. Zustände der Transportverbindung gehalten werden müssen.

Die Signalisierung in H.323 erfolgt hauptsächlich in TCP, obwohl auch die Unterstützung von UDP möglich ist. Die Benutzung von TCP erschwert den zustandslosen Betrieb von H.323 Servern.

SIP kann sowohl über UDP, als auch über TCP oder SCTP betrieben werden. Die Verwendung von UDP wird allerdings bevorzugt. SIP stellt keine Anforderungen an die Zuverlässigkeit des Transportprotokolls und besitzt eigene Mittel zu Behandlung von Paketverlusten.

Konferenzgrößen

Eine zentrale Verwaltung von Konferenzen mit hoher Teilnehmeranzahl ist in aller Regel unzureichend. Zur Unterstützung von großen Konferenzen ist ein verteilter Ansatz gefordert.

H.323 Konferenzen basieren auf dem Einsatz von zentralen *Multipoint Control Units (MCU)*. Zur Unterstützung von großen Konferenzen erlaubt H.323 den Einsatz von Multicast zur Medienübertragung. Für Mediensteuerung und Feedback wird das H.245 Protokoll benutzt, das kein Multicast unterstützt. Dies erschwert die Skalierung in sehr großen Konferenzen. Die H.332 Spezifikation enthält Erweiterungen zur Behandlung dieser Probleme.

Da SIP ursprünglich für den Einsatz in großen Konferenzen entwickelt wurde, bereiten hohe Teilnehmeranzahlen weniger Schwierigkeiten. Die Medienübertragung erfolgt in Multicast. Feedback

und Status werden mit RTCP behandelt, das ebenfalls verteilt arbeitet. Skalierungsprobleme ergeben sich hier nicht. Falls gewünscht können Konferenzen ebenfalls über MCUs geführt werden.

Dienste

Unterstützte Dienste

Name	H.323	SIP
Call Hold	✔	✔
Call Transfer	✔	✔
Call Forwarding	✔	✔
Call Waiting	✔	✔
Conferencing	✔	✔
Call Park	✔	✔
Call Pickup	✔	✔
Call Completion on Busy Subscriber	✔	✘
Calling Line ID	✔	✔
Message Waiting Indication	✔	✔
Directed Call Pickup	✘	✔
Call Return	✘	✔
Follow-Me	✘	✔
Find-Me	✘	✔
Camp On	✘	✔
Call Queuing	✘	✔
Automatic Call Distribution	✘	✔
Do Not Disturb	✘	✔
Third Party Call Control	✘	✔

Austausch von Endsystemparametern

H.323 benutzt das H.245 Protokoll zum Austausch von Endsystemparametern (*Capability Sets).* Die Capability Sets beschreiben in einer relativ kompakten Struktur, welche Fähigkeiten ein Endgerät besitzt und welche Medien empfangen und verarbeitet werden.

SIP benutzt SDP zum Austausch von Medien- und Endsystemparametern. Weiterhin existieren mit der *OPTION* Methode und *Required/Supported* Headern die Möglichkeit zur Abfrage der Fähigkeiten eines Endsystems oder Servers.

Paketverluste und Fehlerbehandlung

H.323 Version 1 bis 2 basieren vollständig auf dem gesicherten Transportprotokoll TCP. Seit Version 3 ist zusätzlich die Unterstützung von UDP mit eigenen Prozeduren für Sicherung vorhanden.

SIP ist vollständig unabhängig vom verwendeten Transportprotokoll. Es besitzt eigene (timer-basierte) Funktionen zur Behandlung von Datenverlusten. Für die Übertragung kommt hauptsächlich UDP zum Einsatz. Die Verwendung von TCP oder SCTP ist aber auch möglich.

Zusammenfassung

Zur Zeit bieten die Standards H.323 und SIP vergleichbare Dienste an. Die erste Version von H.323 wurde jedoch früher veröffentlicht und konnte so eine schnellere Verbreitung finden.

Den grundlegenden Vorteil von SIP erkennen wir in seiner Flexibilität, die den Einsatz von neuen Diensten ermöglicht. So wurden kürzlich Vorschläge und Implementierungen von *Instant Messaging* und *3-rd Party Call Control* vorgestellt. Diese Dienste können in bestehende SIP-Infrastrukturen integriert werden, ohne daß diese verändert werden müssen.

Ebenso ermöglicht SIP die Integration von zahlreichen innovativen Diensten wie *reply with a webpage* mit Standard- Telefoniediensten.

B.2 Aktuelle Probleme

In diesem Abschnitt fassen wir alle Themen zusammen, die wir für einen kommerziellen Einsatz von Internet Telefonie wichtig halten, und die bislang immer noch offen bleiben. Jeder Einführung ins Problem folgt ein Kurzbericht über den aktuellen Status.

B.2.1 Quality of Service (QoS)

Die Unterstützung von QoS gilt als eine der wesentlichen Aufgaben beim kommerziellen Einsatz von SIP Telefonie. Die Autoren von SIP waren sich dessen bewusst und haben zu diesem Zweck den Begriff *Preconditions* eingeführt. Die Grundidee besteht darin, von den jeweiligen konkreten QoS Mechanismen zu abstrahieren, und die Signalisierung erst dann abzuschließen, wenn alle Vorgaben (*Preconditions*) erfüllt sind.

Solche Preconditions können z.B. die Bereitstellung einer gewünschten Bandbreite oder die Existenz einer verschlüsselten Verbindung sein. Ein Hauptvorteil dieser Methode besteht vor allem in der Trennung der Telefoniesignalisierung von den benutzten QoS Architektur bzw. Sicherheitsmechanismen.

Aktuell kommen als QoS Mechanismen RSVP Reservierung und die etwas leichtgewichtigere DiffServ Methode in Frage. Möglich ist auch die Kombination von den beiden Methoden, in dem RSVP zur Flexibilität in Zugangsnetzwerken eingesetzt wird, während DiffServ in Backbone Netzwerken zur besseren Skalierbarkeit verwendet werden kann. Sinnreich et al haben Einzelheiten eines kompletten domänenübergreiffenden Szenarios mit RSVP Unterstützung in beschrieben.

Aktueller Stand

Die Mechanismen zur Synchronisation von Telefoniesignalisierung mit QoS Mechanismen werden immer noch standardisiert. Zur Zeit sind uns nur wenige Implementierungen von Internet Telefonie Geräten bekannt, die über QoS Unterstützung verfügen. Bei manchen wird "TOS Marking" implementiert, einige prototypische Implementierungen (wie z.B. das von GMD-Fokus im Auftrag des DFNs entwickelte Multimedia Terminal) verfügen über RSVP Unterstützung.

B.2.2 Koexistenz von SIP mit Firewalls

SIP Sitzungen über mit Firewalls versehenen Netzwerken aufzubauen, bleibt immer noch ein schwerwiegendes Problem. Die Ursache dafür besteht darin, daß IP Adressen und Port Nummern aller Sitzungsteilnehmer mit Hilfe von SIP dynamisch ausgehandelt werden. Firewalls, die nur IP Ströme zu bekannten Adressen bzw. Port Nummern zulassen, sind dadurch nicht in der Lage, SIP Sitzungen freizuschalten. Hier müssen sie zusätzlich auch SIP interpretieren und Filterregeln dynamisch verändern. Das Problem wird noch komplizierter, wenn Network Address Translation (NAT, Adressenumsetzung) eingesetzt wird. Dann muß auch der Inhalt aller SIP Nachrichten verändert

werden, damit überhaupt eine Sitzung zustande kommen kann. Rosenberg, Drew und Schulzrinne haben diese Probleme und Lösungen in [SR00] erläutert.

Aktueller Stand

Zur Zeit sind uns nur wenige Firewall/NAT Produkte bekannt, die über SIP Unterstützung verfügen (In der Dokumentation der neuesten Cisco PIX Firewalls ist SIP Unterstützung erwähnt, es gibt eine Firewall der Firma Aravox und es gibt ein NAT Modul für den Linux Kernel). Das heißt, daß die meisten Internet Telefonie Benutzer, die sich hinter einem Firewall/NAT befinden, zur Zeit aus dem Internet-weiten Betrieb effektiv ausgeschlossen sind.

Wir untersuchen die Möglichkeit, SIP-Intelligenz, die sich bereits in SIP Servern befindet, für externe Kontrolle der Firewalls zu benutzen. Dieser Lösungsansatz befreit Firewalls/NATs von SIP Kenntnissen, führt dadurch zur besseren Performanz und Verwaltbarkeit, und ermöglicht hop-by-hop Sicherheit. Allerdings befindet sich diese Initiative erst am Anfang -- sie wird vermutlich Ende 2000 gestartet.

Eine andere Lösung wurde vor kurzem von Rosenberg und Schulzrinne in [SR00A] vorgeschlagen. Die Grundidee besteht darin, die Firewall-policy mit Hilfe von externen Proxies zu umzugehen, und zählt dadurch eher zu den "Hack-Lösungen".

B.2.3 Call Routing

Soll eine Verbindung von einem SIP User Agent (unter diesem Begriff kann sich auch ein SIP-fähiges PSTN Gateway verbergen) zu einem PSTN Teilnehmer aufgebaut werden, muß der Anruf durch ein *Gateway* geleitet werden. Ein Gateway setzt SIP Signalisierung und RTP in das für das PSTN geeignete Format um.

Da es sehr unwahrscheinlich ist, daß ein Anbieter alle mögliche Anrufziele weltweit mit eigener Infrastruktur abdecken kann, wird eine Kooperation zwischen mehreren Anbietern angenommen. Sie erfolgt dadurch, daß verfügbare Gateways sowohl Benutzern als auch kooperierenden Anbietern zur Verfügung gestellt werden. Um dem Rufenden die Auswahl eines für ein Gespräch geeigneten Gateways zu erleichtern, wurde das "Telephony Routing over IP (TRIP)" Protokoll entworfen. Mithilfe dieses Protokolls ist es allen kooperierenden Gateways möglich, Informationen über PSTN Rufnummern, die über diese Gateways zugänglich sind, gegenseitig auszutauschen.

Die Entscheidung, über welches Gateway ein Anruf geleitet wird, kann sowohl direkt von einem Endgerät, als auch über einen SIP Server erfolgen. Die erste Variante setzt die Möglichkeit voraus, die durch TRIP ermittelten Routingtabellen abzufragen. Die letztere Möglichkeit bietet sich an, um Mehrwertdienste durch Dritte anzubieten (Z.B. "Wähle das preisgünstigste Gateway) bzw. gewisse Regelung durchzusetzen (Z.B. "Leite Anrufe nur über Gateways, die zur gleichen Anbieterallianz gehören.) Die Routing-Entscheidung hängt typischerweise von der Vorwahl der Zielnummer ab, das Protokoll stellt aber keine Einschränkungen bezüglich der Parameter, die in den Entscheidungsprozeß eingezogen werden können. So kann zum Beispiel die Entscheidung auch von Angebot an unterstützten Signalisierungs- bzw. Verschlüsselungsprotokollen abhängig gemacht werden.

Das TRIP Protokoll wurde absichtlich so konzipiert, daß es auch unter großer Last sehr gut skaliert. Routinginformationen werden aggregiert und selektiv weitergeleitet, alle Veränderungen werden schrittweise vorgenommen.

Eine mögliche Alternative zum Auffinden einer Internet Adresse anhand einer PSTN Nummer beruht auf Umsetzung von E.164 Nummern in DNS Einträge (wie beschrieben in RFC 2916). Diese Lösung wird bei der Arbeitsgruppe ENUM entwickelt und macht es möglich, gewöhnliche E.164 Nummer in eine oder auch mehrere URIs umzusetzen. Die Hauptanwendung dieses Mechanismus sehen wir darin, Internet Benutzer für Rufende aus dem PSTN erreichbar zu machen.

Aktueller Stand

TRIP wird immer noch standardisiert und es sind uns der Zeit keine Produkte mit TRIP Unterstützung bekannt. Daher erwarten wird, daß in absehbaren Zukunft statische, in Geräten vorkonfigurierte Routingtabellen benutzt werden.

B.2.4 Authentication, Authorization

In SIP sind bereits Mittel für die Authentifizierung vorhanden. SIP Clients können sich einem Server gegenüber mit Hilfe von Basic bzw. Digest Methoden authentifizieren. SIP Server greifen typischerweise auf externe Datenbanken zu, um die Richtigkeit der Authentifizierungsanforderungen zu überprüfen.

Noch immer fehlen Mechanismen, um die Authentifizierungsanforderungen an die Signalisierungslogik zu koppeln. Dies ist z.B. dann wichtig, wenn ein SIP Server den Zugang zu einem PSTN Gateway regelt und unterschiedliche Benutzergruppen mit verschiedenen Berechtigungen vorsieht. Eine solche Regelung kann z.B. den öffentlichen Zugang zu einer "Free-call" Rufnummer ermöglichen, während für andere Anrufe die Zugehörigkeit zu einer Authentifizierungsgruppe und Kenntnisse der entsprechenden Authentifizierungsnachweise erforderlich ist.

Erwähnenswert ist, daß SIP Server, die einen externen AAA Server mit ROAMOPS Unterstützung benutzen, imstande sind, AAA Dienste für Benutzer aus kooperierenden Netzwerken zu erbringen. Dies ist insbesondere in mobilen Szenarien von Bedeutung, in denen sich Benutzer in fremde Netzwerken einwählen.

Aktueller Stand

Der Bedarf für die Ankopplung von Authentifizierung an die Signalisierungslogik wurde in [KU00A] beschrieben und Unterstützung von "authentication switching" in CPL als Lösung vorgeschlagen. Auch wenn in der IPTEL WG Einverständnis über den Bedarf für eine derartige Lösung herrscht, befindet sich die Diskussion darüber noch in einem sehr frühen Stadium.

B.2.5 Accounting und Charging

Damit IP-Telefonie kommerziell betrieben werden kann, müssen Anbieter imstande sein, Gebühren für angebotene Dienste in Rechnung zu stellen. Auch wenn es noch eine offene Frage ist, welche Tariffierungsmodelle für welche Teildienste einzusetzen sind, wurden bereits zwei Dienste erkannt, bei denen eine pauschale Abrechnung nicht als optimal angesehen wird und Abrechnugsmechanismen notwendig sind. Diese Dienste sind Zugang zu Telefonie Gateways und QoS Reservierung.

Beim Zugang zu Gateways ist der Abrechnungsprozeß unter Einsatz von generischen Abrechnugsmechanism realisierbar. Das Geschäftsverhältnis wickelt sich ausschließlich zwischen dem Dienstanbieter und seinem Kunden ab und es reicht, wenn ein Gateway Authentifizierung und Autorisierung durchführt und Signalisierungsereignisse an ein Abrechnungsserver meldet. Dazu kann ein bestehendes AAA Protokoll wie Radius oder Diameter verwendet werden. Der Abrechnugsserver sammelt alle "Call Detail Records (CDR)" und liefert sie an ein Billingsystem.

Ist es einem Benutzer viel zu kompliziert, Geschäftsverhältnisse mit zahlreichen Anbietern zu pflegen, ist es denkbar, daß er Gebrauch von einem "Aggregationsdienst" macht. Solche Dienste vereinfachen den Umgang mit mehreren Anbietern, indem sie eine Datenbank über Dienstzugriffsdaten verwalten. Sie können durch Software in Endgeräten oder auch als ein Netzwerkdienst realisiert werden.

Das Abrechnugsverfahren ist für QoS Reservierung wesentlich komplizierter. Die Ursache dafür ist, daß ein QoS Dienst von einer Reihe von Dienstanbietern, die sich entlang eines Datenübermittlungspfades befinden, erbracht wird. Weil es nicht anzunehmen ist, daß alle Anbieter untereinander in einem Geschäftsverhältnis stehen, wurde von Sinnreich et al. eine auf *Clearing Houses* basierende Lösung vorgeschlagen. Die Clearing Houses dienen als "Trust Broker", d.h. sie tauschen sicherheitssensitive Daten wie die CDRs zwischen Anbietern aus und haften für ihre

Vertrauenswürdigkeit. Zum Datenaustausch wird das bei ETSI entwickelte *Open Settlement Protocol* vorgesehen.

Aktueller Stand:

Die meisten SIP Server verfügen heutzutage über die Möglichkeit, Signalisierungsereignisse über ein AAA Protokoll zu protokollieren. Lösungsansätze für domänenübergreifende und abrechenbare QoS Unterstützung wurden vorgeschlagen, bislang gab es allerdings wenig Akzeptanz.

B.2.6 H.323 Interoperabilität

Insbesondere im europäischen Raum hat H.323 durch eine schnellere Standardverabschiedung den breiteren Einsatz gefunden. Um die Interoperabilität der H.323 Infrastruktur mit kommender SIP Infrastruktur zu gewährleisten, werden Lösungen gesucht, die Koexistenz der beiden Ansätze ermöglichen. Ansätze von derartigen Lösungen wurden in beschrieben, die Arbeit wird weiterhin beim "SIP-H.323 Design Team" fortgesetzt.

Aktueller Stand:

Uns sind derzeit zwei prototypische Implementierungen von SIP-H.323 Signalisierungsgateways bekannt: eine wird an Columbia Universtity, die andere an der TU Darmstadt entwickelt.

B.2.7 Konfiguration und Management

Um SIP Geräte konfigurierbar und verwaltbar zu machen, wurden folgende Protokollerweiterungen entworfen: Simple Network Management Protocol (SNMP) MIB, Dynamic Host Configuration Protocol (DHCP) Option und Service Location Protocol (SLP) Templates zum Auffinden eines lokalen Proxys.

Aktueller Stand

Die Standardisierung von DHCP und SNMP Unterstützung ist beinahe abgeschlossen und es ist zu erwarten, das beides in absehbaren Zukunft in Endgeräten zu finden sein wird.

B.3 Aktueller Status von SIP -- Zusammenfassung

Auch wenn immer noch zahlreiche SIP Erweiterung bei IETF diskutiert werden, scheint die Grundspezifikation ihre Stabilität erreicht zu haben. An der aktuellen Version wird noch immer gearbeitet, sie bleibt aber kompatibel zum RFC2543, und die meisten Veränderungen bestehen aus verbesserten Kommentaren.

An SIP Lösungen wird bei allen bekannten Herstellern (Z.B. Cisco, 3com, Nortel Networks, usw.) gearbeitet, viele dieser Produkte sind bereits lieferbar. Durch Einfachheit des SIP Konzepts wurde die Interoperabilität unterschiedlichster Produkten schnell und erfolgreich demonstriert. Beim letzten "Bake-off" in August 2000 haben 45 Hersteller mit Erfolg teilgenommen.

Es gibt erste kommerziell betriebene SIP Netzwerke und bei bedeutenden Netzwerkanbietern laufen Feldversuche im raschen Tempo. Nicht zuletzt gewinnt SIP an Bedeutung dadurch, daß es als Signalisierungsprotokoll für UMTS (aufgrund seiner Erweiterbarkeit) gewählt wurde.

Zu den aktuellen Forschungsthemen, zählen zur Zeit vor allem unterschiedlichste Mobilitätsaspekte, Koexistenz mit Firewalls und NATs, domänen-übergreifende Abrechnungsdienste, und Unterstützung für Dienstqualität.

C References

[AAA99] J. Vollbrecht, P. Calhoun, S. Farrell, L. Gommans, G. Gross, B. de Bruijn, C. de Laat, M. Holdrege and D. Spence. “AAA Authorization Framework.” Work in Progress. Internet-Draft draft-ietf-aaa-authz-arch-00.txt. October 1999.

[AR00] Agrawal et al. “ SIP-H.323 Interworking Requirements.” Work in Progress. Internet-Draft. July 2000.

[AVT00] Schulzrinne, Casner, Frederick, Jacobson. “RTP: A Transport Protocol for Real-Time Applications”. Work in Progress. Internet-Draft draft-ietf-avt-rtp-new-07.txt. March 2000.

[AVT00A] Schulzrinne, Casner. “RTP Profile for Audio and Video Conferences with Minimal Cotrol”. Work in Progress. Internet-Draft draft-ietf-avt-profile-new-08.txt, January 2000.

[AVT00B] S. Casner. “SDP Bandwidth Modifiers for RTCP Bandwidth”. Work in Progress. Internet-Draft draft-ietf-avt-rtcp-bw-01.txt. March 2000

[BE00] B. Beser. “Codec Capabilities Attribute for SDP.” Work in Progress. Internet-Draft draft-beser-mmusic-capabilities-00.txt

[BOR96] Carsten Bormann, Jörg Ott, and Christoph Reichert. “Simple Conference Control Protocol.” Work in Progress. Internet-Draft draft-ietf-mmusic-sccp-01.txt. 1996.

[BRO00] A. Brown. “ENUM Requirements.” Work in Progress. Internet-Draft draft-ietf-enum-rqmts-01.txt. June 2000.

[CAM99] Gonzalo Camarillo. “Best Current Practice for ISUP to SIP mapping.” Work in Progress. Internet-Draft draft-camarillo-mmusic-sip-isup-bcp-00.txt. August 1999.

[CO00] G. Klyne. “Identifying composite media feature.” Work in Progress. Internet-Draft draft-ietf-conneg-feature-hash-05.txt. June 2000

[CO00A] G. Klyne. “A revised media feature set matching alogorithm.” Work in Progress. Internet Draft draft-klyne-conneg-feature-match-02.txt. April 2000.

[DCS99] W. Marshall, K. Ramakrishnan, E. Miller, G. Russell, B. Beser, M. Mannette, K. Steinbrenner, D. Oran, J. Pickens, P. Lalwaney, J. Fellows, D. Evans, K. Kelly and F. Andreasen. “Integration of Resource Management and Call Signaling for IP Telephony.” Work in Progress. Internet-Draft draft-dcsgroup-sip-resource-00.txt. October 1999.

[DCS00a] W. Marshall, E. Miller, B. Beser, D. Oran, J. Pickens, P. Lalwaney, J. Fellows, D. Evans and K. Kelly. “SIP Extensions for Media Authorization.” Work in Progress. Internet-Draft draft-dcsgroup-sip-call-auth-02.txt. June 2000.

[DCS00b] W. Marshall, E. Miller, B. Beser, D. Oran, J. Pickens, P. Lalwaney, J. Fellows, D. Evans and K. Kelly. “Architectural Considerations for Providing Carrier Class Telephony Services Utilizing SIP-based Distributed Call Control Mechanisms.” Work in Progress. Internet-Draft draft-dcsgroup-sip-arch-03.txt. November 2000.

[DCS00c] W. Marshall, E. Miller, B. Beser, D. Oran, J. Pickens, P. Lalwaney, J. Fellows, D. Evans and K. Kelly. “SIP Extensions for Caller Identity and Privacy.” Work in Progress. Internet-Draft draft-dcsgroup-sip-privacy-02.txt. June 2000.

[DCS00d] W. Marshall, E. Miller, B. Beser, K. Steinbrenner, D. Oran, J. Pickens, P. Lalwaney, J. Fellows, D. Evans and K. Kelly. “SIP Extensions for supporting

Distributed Call State." Work in Progress. Internet-Draft draft-dcsgroup-sip-state-02.txt. July 2000.

[DCS00e] W. Marshall, E. Miller, B. Beser, K. Steinbrenner, D. Oran, J. Pickens, P. Lalwaney, J. Fellows, D. Evans and K. Kelly. "SIP proxy-to-proxy extensions for supporting DCS." Work in Progress. Internet-Draft draft-dcsgroup-sip-proxy-proxy-03.txt. November 2000.

[CA00] G. Camarillo, J. Häller. "SDP media alignment in SIP." Work in Progress. Internet-Draft draft-camarillo-sip-sdp-00.txt. June 2000.

[RFC 2916] P. Falstrom. "E.164 numbers and DNS." Request For Comments 2916. Proposed Standard. September 2000.

[GSM 06.10] European Telecommunications Standards Institute (ETSI). "GSM Full Rate Speech Transcoding." Recommendation GSM 06.10.

[GSM 06.20] European Telecommunications Standards Institute (ETSI). "GSM Half Rate Speech Transcoding." Recommendation GSM 06.20.

[GSM 06.60] European Telecommunications Standards Institute (ETSI). "GSM Enhanced Full Rate Speech Transcoding." Recommendation GSM 06.60.

[H.225.0] International Telecommunication Union (ITU). "Call signalling protocols and media stream packetization for packet-based multimedia communication systems". ITU-T-Recommendation H.225.0. Version 4. November 2000.

[H.235] International Telecommunication Union (ITU). "Security and encryption for H-Series (H.323 and other H.245-based) multimedia terminals". ITU-T-Recommendation H.235. Version 2. November 2000.

[H.245] International Telecommunication Union (ITU). "Control protocol for multimedia communication". ITU-T-Recommendation H.245. Version 7. November 2000.

[H.248] International Telecommunication Union (ITU). "Gateway Control Protocol". ITU-T-Recommendation H.248. October 1999.

[H.310] International Telecommunication Union (ITU). "Broadband audiovisual communication systems and terminals". ITU-T-Recommendation H.310. September 1998.

[H.320] International Telecommunication Union (ITU). "Narrow-band telephone systems and terminal equipment". ITU-T-Recommendation H.320. May 1999.

[H.321] International Telecommunication Union (ITU). "Adaptation of H.320 visual telephone terminals to B-ISDN environments". ITU-T-Recommendation H.321. February 1998.

[H.322] International Telecommunication Union (ITU). "Visual telephone systems and terminal equipment for local area networks which provide a guaranteed quality of service". ITU-T-Recommendation H.322. March 1996.

[H.323] International Telecommunication Union (ITU). "Packet-based multimedia communications systems". ITU-T-Recommendation H.323. Version 4. November 2000.

[H.324] International Telecommunication Union (ITU). "Terminal for low bit-rate multimedia communication". ITU-T-Recommendation H.324. February 1998.

[H.332] International Telecommunication Union (ITU). "Visual telephone systems and terminal equipment for local area networks which provide a guaranteed quality of service". ITU-T-Recommendation H.332. September 1998.

[H.341] International Telecommunication Union (ITU). "Multimedia Management Information Base". ITU-T-Recommendation H.341. May 1999.

[H.450.1] International Telecommunication Union (ITU). “Generic functional protocol for the support of supplementary services in H.323”. ITU-T-Recommendation H.450.1. February 1998.

[H.450.2] International Telecommunication Union (ITU). “Call transfer supplementary service for H.323”. ITU-T-Recommendation H.450.2. February 1998.

[H.450.3] International Telecommunication Union (ITU). “Call diversion supplementary service for H.323”. ITU-T-Recommendation H.450.3. February 1998.

[H.450.4] International Telecommunication Union (ITU). “Call hold supplementary service for H.323 ”. ITU-T-Recommendation H.450.4. May 1999.

[H.450.5] International Telecommunication Union (ITU). “Call Park and Call Pickup Supplementary Services for H.323”. ITU-T-Recommendation H.450.5. May 1999.

[H.450.6] International Telecommunication Union (ITU). “Call Waiting Supplementary Services for H.323”. ITU-T-Recommendation H.450.6. May 1999.

[H.450.7] International Telecommunication Union (ITU). “Message Waiting Indication Supplementary Service for H.323”. ITU-T-Recommendation H.450.7. May 1999.

[H.450.8] International Telecommunication Union (ITU). “Name Identification Services”. ITU-T-Recommendation H.450.8. February 2000.

[H.450.9] International Telecommunication Union (ITU). “Call Completion Supplementary Services for H.323”. ITU-T-Recommendation H.450.9. Novemeber 2000.

[KO98] H. Honko, P. Koskelainen. “SDP syntax for H.263 options.” Work in Progress. Internet-Draft draft-koskelainen-sdp263-02.txt. June 1998

[KU00] D. Kutscher, J. Ott, C. Bormann. “Requirements for Session Description and Capability Negotiation.” Work in Progress. Internet-Draft. draft-kutscher-mmusic-sdpng-req-00.txt. July 2000.

[KU00A] J. Kuthan. “CPL Authentication and Database Access Extensions.” Work in Progress. Internet-Draft. November 2000.

[LS00] Jonathan Lennox and Henning Schulzrinne. “CPL: A Language for User Control of Internet Telephony Services.” Work in Progress. Internet-Draft draft-ietf-iptel-cpl-04.txt. November 2000.

[MM99] J. Rosenberg, H. Schulzrinne, S. Donovan. “Establishing QoS and Security Preconditions for SDP Session.” Work in Progress. Internet-Draft draft-ietf-mmusic-sdp-qos-00.txt. June 1999

[MM00] R. Finlayson. “Describing session directories in SDP.” Work in Progress. Internet-Draft draft-ietf-mmusic-sdp-directory-type-01.txt. June 2000.

[MM00A] B. Quinn. “SDP Source-Filters.” Work in Progress. Internet-Draft draft-ietf-mmusic-sdp-srcfilter-00.txt. May 2000.

[OT99] J. Ott, D. Kutscher, C. Bormann. “Capability description for group cooperation.“ Work in Progress. Internet-Draft draft-ott-mmusic-cap-00.txt. June 1999.

[Q.700] International Telecommunication Union (ITU). “Introduction to CCITT Signalling System No. 7”. ITU-T-Recommendation Q.700. March 1993.

[RA00] R. Kumar, M. Mostafa. “Extension of the Session Description Protocol (SDP) for ATM-based Narrowband Telephony.” For in Progress. Internet-Draft draft-rajeshkumar-sdp-atm-01.txt. March 2000.

[RFC 1006] M.T. Rose, D.E. Cass. “ISO transport services on top of the TCP: Version 3.” Request For Comments 1006. Standard. May 1987.

[RFC 1439] C. Finseth. “The Uniqueness of Unique Identifiers.” Request For Comments 1439. Informational. March 1993.

[RFC 1633] R. Braden, D. Clark, S. Shenker. “Integrated Services in the Internet Architecture: an Overview.” Request For Comments 1633. Informational. June 1994.

[RFC 1889] H. Schulzrinne, S. Casner, R. Frederick, V. Jacobson. “RTP: A Transport Protocol for Real-Time Applications.” Request For Comments 1889. Proposed Standard. January 1996.

[RFC 1890] H. Schulzrinne. “RTP Profile for Audio and Video Conferences with Minimal Control.” Request For Comments 1890. Proposed Standard. January 1996.

[RFC 2126] Y. Pouffary, A. Young. “ISO Transport Service on top of TCP (ITOT).” Request For Comments 2126. Proposed Standard. March 1997.

[RFC 2198] C. Perkins, I. Kouvelas, O. Hodson, V. Hardman, M. Handley, J.C. Bolot, A. Vega-Garcia, S. Fosse-Parisis. “RTP Payload for Redundant Audio Data.” Request For Comments 2198. Proposed Standard. September 1997.

[RFC 2205] R. Braden, Ed., L. Zhang, S. Berson, S. Herzog, S. Jamin. “Resource ReSerVation Protocol (RSVP). Version 1. Functional Specification.” Request For Comments 2205. Proposed Standard. September 1997.

[RFC 2206] F. Baker, J. Krawczyk, A. Sastry. “RSVP Management Information Base using SMIv2.” Request For Comments 2206. Proposed Standard. September 1997.

[RFC 2207] L. Berger, T. O'Malley. “RSVP Extensions for IPSEC Data Flows.” Request For Comments 2207. Proposed Standard. September 1997.

[RFC 2208] A. Mankin, Ed., F. Baker, B. Braden, S. Bradner, M. O`Dell, A. Romanow, A. Weinrib, L. Zhang. “Resource ReSerVation Protocol (RSVP). Version 1. Applicability Statement Some Guidelines on Deployment.” Request For Comments 2208. Informational. September 1997.

[RFC 2209] R. Braden, L. Zhang. “Resource ReSerVation Protocol (RSVP). Version 1. Message Processing Rules.” Request For Comments 2209. Informational. September 1997.

[RFC 2210] J. Wroclawski. “The Use of RSVP with IETF Integrated Services.” Request For Comments 2210. Proposed Standard. September 1997.

[RFC 2326] H. Schulzrinne, A. Rao, R. Lanphier. “Real Time Streaming Protocol (RTSP).” Request For Comments 2326. Proposed Standard. April 1998.

[RFC 2327] M. Handley, V. Jacobson. “SDP: Session Description Protocol.” Request For Comments 2327. Proposed Standard. April 1998.

[RFC 2354] C. Perkins, O. Hodson. “Options for Repair of Streaming Media.” Request For Comments 2354. Informational. June 1998.

[RFC 2430] T. Li, Y. Rekhter. “A Provider Architecture for Differentiated Services and Traffic Engineering (PASTE).” Request For Comments 2430. Informational. October 1998.

[RFC 2474] K. Nichols, S. Blake, F. Baker, D. Black. “Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers.” Request For Comments 2474. Informational. December 1998.

[RFC 2475] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, W. Weiss. “An Architecture for Differentiated Service.” Request For Comments 2475. Informational. December 1998.

[RFC 2507] M. Degermark, B. Nordgren, S. Pink. “IP Header Compression.” Request For Comments 2507. Proposed Standard. February 1999.

[RFC 2508] S. Casner, V. Jacobson. “Compressing IP/UDP/RTP Headers for Low-Speed Serial Links.” Request For Comments 2508. Proposed Standard. February 1999.

[RFC 2509] M. Engan, S. Casner, C. Bormann. “IP Header Compression over PPP.” Request For Comments 2509. Proposed Standard. February 1999.

[RFC 2533] G. Klyne. “A Syntax for Describing Media Feature Sets.” Request For Comments 2533. Proposed Standard. March 1999

[RFC 2543] M. Handley, H. Schulzrinne, E. Schooler, J. Rosenberg. “SIP: Session Initiation Protocol.” Request For Comments 2543. Proposed Standard. March 1999.

[RFC 2686] C. Bormann. “The Multi-Class Extension to Multi-Link PPP.” Request For Comments 2686. Proposed Standard. September 1999.

[RFC 2687] C. Bormann. “PPP in a Real-time Oriented HDLC-like Framing.” Request For Comments 2687. Proposed Standard. September 1999.

[RFC 2689] C. Bormann. ”Providing Integrated Services over Low-bitrate Links.” Request For Comments 2689. Informational. September 1999.

[RFC 2703] G. Klyne. “Protocol-independent Content Negotiation Framework.” Request For Comments 2703. Informal. September 1999

[RFC 2705] M. Arango, A. Dugan, I. Elliott, C. Huitema, S. Pickett. “Media Gateway Control Protocol (MGCP).” Request For Comments 2705. Informational. October 1999.

[RFC 2733] J. Rosenberg, H. Schulzrinne. “An RTP Payload Format for Generic Forward Error Correction.” Request For Comments 2733. Proposed Standard. December 1999.

[RFC 2750] S. Herzog. “RSVP Extensions for Policy Control.” Request For Comments 2750. Proposed Standard. January 2000.

[RFC 2782] A. Gulbrandsen, P. Vixie, L. Esibov. “A DNS RR for specifying the location of services (DNS SRV).” Request For Comments 2782. Proposed Standard. February 2000.

[RFC 2824] J. Lennox, H. Schulzrinne. “Call Processing Language Framework and Requirements.” Request For Comments 2824. Informational. May 2000.

[RFC 2865] C. Rigney, S. Willens, A. Rubens, W. Simpson. ”Remote Authentication Dial In User Service (RADIUS).” Request For Comments 2865. Draft Standard. June 2000.

[ROS00] J. Rosenberg, H. Salama and M. Squire. “Telephony Routing over IP (TRIP).” Work in Progress. Internet-Draft draft-ietf-iptel-trip-04.txt. November 2000.

[SR98] H. Schulzrinne, J. Rosenberg. "A Comparison of SIP and H.323 for Internet Telephony.” July 1998

[SR99] Henning Schulzrinne and Jonathan Rosenberg. “SIP Call Control Services.” Work in Progress. Internet-Draft draft-ietf-mmusic-sip-cc-01.txt. June 1999.

[SR00] J. Rosenberg, D. Drew, H. Schulzrinne. “Getting SIP SIP through Firewalls and NATs.” Work in Progress. Internet-Draft. February 2000

[SR00A] J. Rosenberg, H. Schulzrinne. "SIP Traversal through Residential and Enterprise NATs and Firewalls" Work in Progress. Internet-Draft. November 2000.

[SS00] K. Singh, H. Schulzrinne. "Interworking Between SIP/SDP and H.323", Work in Progress. Internet-Draft, May 2000.

[SS00A] K. Sin Singh, H. Schulzrinne. " Interworking between SIP/SDP and H.323 Architecture", in Proceedings of the 1st IP Telephony Workshop, April 2000.

[T.120] International Telecommunication Union (ITU). "Data protocols for multimedia conferencing". ITU-T-Recommendation T.120. July 1996.

[ZI99] Eric Zimmerer, Aparna Vemuri, Vijay Nadkarni, Brian Morgan, Steven Mayer and Gonzalo Camarillo. "SIP Best Current Practice for Telephony Interworking." Work in Progress. Internet-Draft draft-zimmerer-sip-bcp-t-00.txt. October 1999.