

Entwicklung und Nutzung von IP-Telefonie Anwendungen auf Unix-Systemen

Ralph Ackermann <rac@KOM.tu-darmstadt.de>

Utz Roedig <utz@KOM.tu-darmstadt.de>

Prof. R. Steinmetz
<Ralf.Steinmetz@KOM.tu-darmstadt.de>

Überblick

Paketvermittelte Netze wie das IP basierte Internet werden immer häufiger nicht nur zur Datenkommunikation, sondern auch zur Übertragung von Sprachdaten eingesetzt. In diesem Zusammenhang erscheinen Schlagworte wie „Voice over IP“, „Internet-Telefonie“ oder auch „IP-Telefonie“ und die Erwartung durch eine „Konvergenz der Netze“ Kosten einzusparen sowie neue Mehrwertdienste zu etablieren.

Um entsprechende Lösungen erfolgreich umsetzen zu können, sind ein grundlegendes Verständnis der Technologie sowie adäquate Hard- und Software-Voraussetzungen notwendig. Diese, insbesondere im Bereich der Endanwender-Applikationen, waren lange Zeit auf Windows-Systeme beschränkt. Dies hat sich mittlerweile deutlich verändert und zunehmend wird leistungsfähige Hard- und Software auch für UNIX Systeme verfügbar.

Innerhalb dieses Artikels stellen wir - nach einer Einführung der generellen Konzepte der IP-Telefonie - Basiskomponenten für deren praktische Umsetzung vor. Anhand ausgewählter Beispiel-Applikationen zeigen wir, wie diese eingesetzt werden können und geben einen Ausblick auf das weiterreichende Potential der aktuellen Entwicklung.

Historie und Grundlagen

Seit etwa 1990 gibt es Versuche zur Echtzeit-Sprachkommunikation über das Internet und seit 1995 existieren auch mehrere kommerzielle Produkte. Die erste (in der Regel noch proprietäre) Generation von entsprechenden Telefonie-Lösungen ermöglichte es, mit Hilfe von Mikrofon, Lautsprecher und Soundkarte Audioverbindungen zwischen mehreren Teilnehmern herzustellen.

Die zu übertragende Sprache wird dazu in den Endgeräten digitalisiert, geeignet komprimiert und in IP-Paketen unter Einsatz des Realtime Transport Protocols RTP [14] zum korrespondierenden Gesprächspartner übertragen. Bei diesem erfolgt nach einem entsprechenden Ausgleich von Netzwerk-Laufzeitschwankungen (Jitter) die Decodierung und anschließende Wiedergabe. Verbindungen zum großen Nutzerkreis des etablierten konventionellen Telefonnetzes können über entsprechende IP-Telefonie-Gateways hergestellt werden.

Nach einer Phase eher experimentellen Betriebes bei dem von den Nutzern auch bestimmte Qualitäts- und Einschränkungen bezüglich des Umfangs der unterstützten Dienste toleriert wurden, befinden wir uns im Augenblick - getragen insbesondere von dem enormen Interesse der Anbieter etablierter Telefon-Anlagen und von Netzwerk-Equipment - im Abschnitt der Definition und Umsetzung von Diensten mit einem dem etablierten Telefonnetz vergleichbaren „Carrier Grade“ der Qualität und Stabilität. Dafür ist eine umfassende Standardisierung unbedingte Voraussetzung.

Grundlagen - Signalisierung

Die IP-Telefonie nutzt eine Reihe von in Abbildung 7 übersichtsmäßig dargestellten unterschiedlichen Signalisierungs- und Übertragungs-Protokollen.

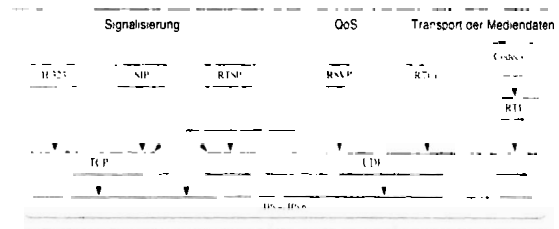


Abbildung 7: Zusammenwirken der Protokolle

Während über die Art der Übertragung der Audiodaten mit RTP Übereinstimmung besteht, existieren im Augenblick für den gerade für den Dienst-Umfang besonders wichtigen Teil der Gesprächssignalisierung mit dem von der ITU definierten H.323 [16] und dem von der IETF getragenen Session Initiation Protocol SIP [11] zwei alternative Ansätze.

H.323

Die Protokollfamilie H.323 wurde von dem Standardisierungsgremium ITU-T mit dem Ziel verabschiedet, multimediale paketbasierte Kommunikation ohne Gewährlei-

stung von Quality-of-Service, über - insbesondere IP-basierte - Netzwerke zu ermöglichen.

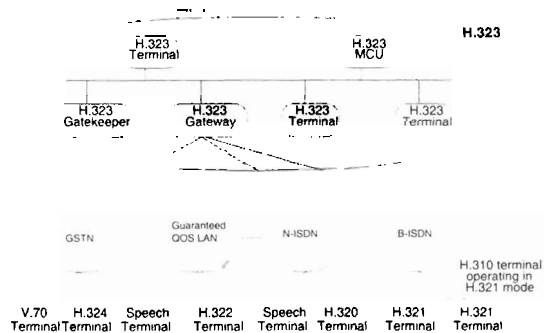


Abbildung 8: H.323 Komponenten

In der Protokoll-Definition ist ein umfassender Stack zusammengefaßt worden, der unterschiedliche Schnittstellen für die Audio-, Video- und Datenübertragung bietet. Der Standard beschreibt einzelne Komponenten, wie Terminals, Gatekeeper (GK), Gateways (GW) und Multicast Units (MCU), sowie die Mechanismen für deren Zusammenwirken.

Eine zentrale H.323 Komponente bildet dabei der sogenannte Gatekeeper, der als zentrale Instanz für alle Telefonate innerhalb einer H.323-Zone fungiert. Er übernimmt u.a. die Zuordnung von Telefonnummer und symbolischen Namen, unter denen Teilnehmer bei ihm registriert sind, zu IP-Adressen sowie das Bandbreitenmanagement. Damit ist er einer klassischen Vermittlungsanlage (PBX) vergleichbar, wird jedoch im Unterschied zu dieser in Software realisiert.

Session Initiation Protocol - SIP

Das Session Initiation Protocol (SIP) dient zum Aufbau, Modifizieren und Terminieren von Sitzungen (Multimedia-Konferenzen, Internet-Telefonie-Gesprächen). Die Teilnehmer können dabei mit Hilfe von SIP Invitations eine Benachrichtigung des angerufenen Gesprächspartners auslösen sowie Parameter (wie z.B. zu nutzende Codecs, Ports) mitteilen und aushandeln.

Das Protokoll unterstützt Mobilität durch Mechanismen zur Weiter- bzw. Umleitung von Requests an den aktuellen Aufenthaltsort eines Teilnehmers. Dazu können Benutzer ihren aktuellen Standort im System registrieren. SIP ist völlig unabhängig von anderen Konferenz-Steuerungsprotokollen sowie von den Transport-Protokollen der unteren Schichten. Damit ist

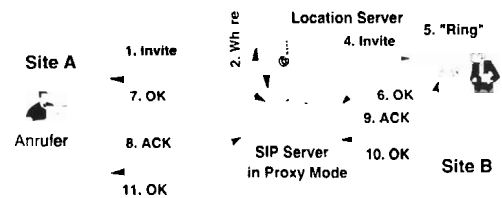


Abbildung 9: SIP-Gesprächsaufbau

es äußerst flexibel und kann leicht mit zusätzlichen Fähigkeiten ausgestattet werden. Eine Reihe von Herstellern entwickeln zur Zeit eigene SIP-fähige Endgeräte, bzw. passen vorhandene H.323 Clients und Systeme für die Interaktion mit diesem Protokoll an.

H.323/SIP Interworking

Der komplexe, nicht leicht umzusetzende und mit einer Reihe von Nachteilen behaftete H.323 Ansatz hat eine sehr starke Verbreitung gefunden und nimmt zunächst noch eine vorherrschende Rolle ein, die nicht zuletzt aus seiner bisherigen Favorisierung durch die Industrie resultiert. Es zeigt sich jedoch, daß sich das eher „leichtgewichtige“ und offene SIP zunehmend zu einer wichtigen Alternative entwickelt. In der Art eines „Umbrella-Standard“ überdecken die dahingehenden Bemühungen der äußerst aktiven IETF Working Groups mittlerweile neben der eigentlichen Gesprächs-Signalisierung auch weitergehende Aspekte wie den der Präsenz-Dienste und des Instant-Messaging, aber auch des Zusammenwirkens mit dem Call Routing, Firewalls, QoS Mechanismen und dem Billing [12][13]. Mit der Verfügbarkeit entsprechender leistungsfähiger Gateways [15][9] werden beide Protokoll-Familien zukünftig auch gemeinsam einsetzbar sein.

Praktische Nutzung

OpenH323

Sollen Applikationen unter Nutzung des H.323 Protokolls entwickelt werden, stellt deren Programmierung auf Basis des ursprünglich aus dem Versuch der Schaffung eines frei verfügbaren Unix-Pendants zu Microsoft NetMeeting entstandenen OpenH323 Stacks eine sehr leistungsfähige Option dar. Dazu sind die Pakete pwlib und openh323 von [5] bzw. einem der mittlerweile zahlreich vorhandenen Mirror-Servern herunterzuladen, zu konfigurieren und entsprechend zu übersetzen. Die so entstehende zum

Basispaket gehörende H.323 Anwendung *voxilla* zeichnet sich durch die nachfolgend aufgeführten Eigenschaften aus:

- H.323 Kompatibilität und Interoperabilität mit Microsoft NetMeeting sowie anderen Hard- (z.B. Siemens LP5100) bzw. Software-Clients
- Vielzahl leistungsfähiger Codecs und Unterstützung für spezielle IP-Telefonie-Hardware
- Möglichkeit zur H.323 Gatekeeper Nutzung
- zusätzlich zu Sprachverbindungen auch Video-Verbindungen nach dem H.261 Standard

Auf der Basis des unter der GNU Lizenz frei verfügbaren Quellcodes wurden und werden mittlerweile eine Reihe von weiteren Anwendungen entwickelt. Exemplarisch seien nachfolgend einige als Basis für eigene Erweiterungen interessante repräsentative Beispiele genannt:

- *openam* - ein H.323 basierter Anrufbeantworter
- *pstngw* - ein einfaches H.323/ISDN Gateway
- *gk* [4] - ein frei verfügbarer H.323 Gatekeeper

Der H.323 Standard wurde und wird als komplex und in der Implementierung aufwendig eingeschätzt. So ist die zu realisierende Logik umfangreich, beinhaltet eine Vielzahl von Optionen und H.323 Protokoll-Pakete (PDUs) sind in einer in Abstract Syntax Notation (ASN.1) beschriebenen Art und Weise binär kodiert. Der dazu notwendige Code kann jedoch mit Hilfe eines zum *pwlib* Paket gehörigen ASN.1 Parsers aus den entsprechenden Beschreibungen automatisch generiert werden, sodaß ein großer Teil des sonst notwendigen Programmieraufwands entfällt.

Vovida SIP Stack

Die Firma Vovida entwickelt eine Reihe leistungsfähiger Protokollstacks (RTP, H.323, SIP, Media Gateway Control Protocol MGCP[10]) und stellt diese inklusive Dokumentation und Betreuung entsprechender Entwickler-Mailing-Listen zur allgemeinen Nutzung und Weiterentwicklung zur Verfügung. Ein sehr starkes Engagement

für das Betriebssystem Linux zeigt sich dabei in einer durchgängigen Unterstützung als Entwicklungs- und Nutzungs-Plattform sowie in der Rolle als Förderer der Linux-Mitgliedschaft innerhalb des Softswitch Consortiums [8].

Die von den Autoren verwendete SIP Basissoftware ermöglicht auf Basis eines objektorientierten und eventgetriebenen Kerns von Basis-Funktionen das Erzeugen und Parsen von SIP-Nachrichten und die Entwicklung von sowohl User Agents als auch SIP Servern. Zur Nutzung als SIP basiertes Audio-Terminal ist das Programm *sua* (SIP User Agent) direkt enthalten, dieses unterstützt auch die nachfolgend beschriebene spezielle IP-Telefonie Hardware.

Anbindung spezieller Hardware

Prinzipiell ist die Nutzung von IP-Telefonie Anwendungen an jedem mit einer Soundkarte ausgestatteten Endsystem möglich. Jedoch haben ursprünglich nicht primär für einen solchen Einsatzzweck entwickelte Soundkarten (wegen der teilweise gerade bei älteren Modellen fehlenden Duplex-Fähigkeit, der für interaktive Anwendungen störenden hohen Verzögerung durch Verwendung relativ großer interner Sample-Puffer und dem Fehlen zusätzlicher Anschlußmöglichkeiten, z.B. für ein konventionelles Telefon) eine Reihe von Nachteilen.

Von der Firma Quicknet sind mit den für ISA, PCI und auch PCMCIA verfügbaren Internet Phone- bzw. Line-Jack Karten preisgünstige Produkte vorhanden, die leistungsfähige Codecs und entsprechende Interfaces (zum Anschluß von Mikrofon und Lautsprecher analog einer Soundkarte aber auch eines konventionellen Telefons bzw. im Falle der LineJack-Karten zusätzlich eines konventionellen Telefonanschlusses) zur Verfügung stellen.

Über einen als Modul in den Kernel ladbaren Treiber, der mittlerweile als Bestandteil des Linux-Telefonie-Interfaces auch in die Standard-Kernel integriert wurde, sind zusätzlich zu Aufnahme, Kodierung und Wiedergabe von Audio-Daten auch die Erkennung und Generierung von DTMF-Wahltonen, die Erzeugung der entsprechenden Signaltöne (Ruf, besetzt, etc.) sowie Rufannahme und -auslösung bei Anschluß an das PSTN möglich. Die Programmentwicklung wird dabei durch ein umfangreiches und gut dokumentiertes System Development Kit (SDK) unterstützt.

Ausgewählte Entwicklungen

Ein erweiterter SIP User Agent

Ein SIP User Agent erlaubt es dem Nutzer, sich bei einem SIP-Location-Server, einem sogenannten Registrar anzumelden und damit seinen aktuellen Standort bekanntzugeben.

Der in Abbildung 10 gezeigte User Agent i2tel (Intelligent Internet Telephony) entstand auf Basis des zum Vovida SIP Stacks vorhandenen Programms sua. Aufgrund der Wahl von Unix als Software-Plattform ist die Anbindung einer (hier mit Tcl/Tk realisierten) Nutzeroberfläche unter Nutzung von Interprocess Communications Mechanismen vergleichsweise einfach möglich. Tcl/Tk wird auch für die (auch dem Anwender mögliche) Konfiguration und Erweiterung des Funktionsumfangs durch Scripting benutzt.

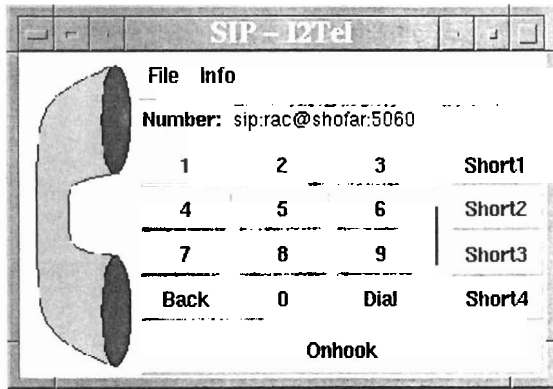


Abbildung 10: SIP User Agent i2tel

Alternativ und interoperabel zu dem von den Autoren verwendeten und erweiterten Client sind mittlerweile mit kphone [3] (als Teil des KDE2-Projektes) und dem von der Columbia University New York entwickelten und an Forschungseinrichtungen unter gewissen Restriktionen kostenlos lizenzierten Programm sipc [1] weitere SIP-Applikationen verfügbar. Das große Interesse der Industrie zeigt sich in der zunehmenden Anzahl auch kommerzieller User Agents [6].

Ein SIP/H.323-Gateway

Im Augenblick findet die Schaffung von Übergängen zwischen den Signalisierungsprotokollen H.323 und SIP eine große internationale Beachtung. Auf der Basis der entsprechenden IETF Standardisierungsvorschläge [15] wurden dafür erste Implementierungen realisiert [7][9].

Detaillierte Informationen zu einem von den Autoren unter Linux und mit Nutzung der (eingeschränkt) frei verfügbaren Komponenten OpenH323 und Vovida-SIP-Stack realisierten Lösungen können [9] entnommen werden.

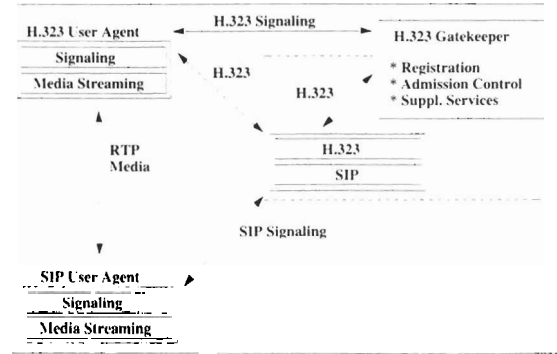


Abbildung 11: SIP-H.323 Gateway

Wie in Abbildung 11 gezeigt, wird dabei durch die Gateway-Komponente die Abbildung, Umsetzung und Weiterleitung der entsprechenden Signalisierungs-Nachrichten vorgenommen. Durch das Gateway erfolgt der Gesprächsaufbau mit seinen einzelnen Phasen (Auslösung des Rufes, „Klingeln“ beim Angerufenen, Gesprächsannahme und Aushandlung der Verbindungsparameter wie zu verwendende Ports und Codecs) in für die Teilnehmer transparenter Art. Die per RTP übertragenen Medienströme werden dann anschließend zwischen den Teilnehmern direkt versandt.

Ausblick

Wie im Beitrag gezeigt bietet Unix als Entwicklungs- und Anwendungs-Umgebung für den sich rasant entwickelnden Bereich der Nutzung paketvermittelter Netze für die Telefonie aufgrund der guten Soft- und Hardware-Unterstützung sehr gute Voraussetzungen. Dies gilt auch für den Bereich der Unterstützung von Standard-APIs für Mehrwert- und CTI (Computer Telephony Integration)-ähnliche Dienste z.B. auf Basis der Java Telephony API (JTAPI) [2] die im Beitrag aufgrund des begrenzten Umfangs nicht dargestellt werden konnten.

Allgemein wird weniger erwartet, daß sich im Rahmen der IP-Telefonie eine dedizierte „Killer-Applikation“ herauskristallisiert. Vielmehr ergibt sich das enorme Potential aus dem Zusammenwirken der unterschiedlichen Dienste. Die Vision einer „Seamless Ubiquitous Communication“ wird durch das Überbrücken von Medienbrüchen, die Integration von Instant Messaging und Präsenz-

Diensten und die Schaffung von in Unternehmensprozesse nahtlos integrierbaren Mehrwertdiensten zunehmend Realität.

Literatur

- [1] Columbia University SIP UA sipc.
<http://www.cs.columbia.edu/~hgs/sipc/>.
- [2] Java Telephony API (JTAPI).
<http://java.sun.com/products/jtapi/>.
- [3] kphone and libdissipate.
<http://www.div8.net/dissipate/>.
- [4] Openh323 Gatekeeper.
<http://www.willamowius.de/openh323gk.html>.
- [5] OpenH323, Part of the Linux VOXILLA Telecom Project.
<http://www.openh323.org/>.
- [6] Siemens SIP software for LP5100.
http://www.mediatrix.com/corpo/pressrelease/Siemens_SIP_Phone.html.
- [7] SIP-H.323 Signaling Gateway.
<http://www.cs.columbia.edu/~kns10/research/gw/>.
- [8] The Softswitch Consortium.
<http://www.softswitch.org/>.
- [9] R. Ackermann, V. Darlagiannis, and R. Steinmetz. Implementation of a H.323/SIP Gateway. *Technical Report TR-2000-02. Darmstadt University of Technology, Industrial Process and System Communications (KOM)*, July 2000.
- [10] M. Arango, A. Dugan, I. Elliott, C. Huitema, and S. Pickett. Media Gateway Control Protocol (MGCP). *RFC 2705*, October 1999.
- [11] M. Handley, H. Schulzrinne, E. Schooler, and J. Rosenberg. SIP: Session Initiation Protocol. *RFC 2543*, March 1999.
- [12] J. Rosenberg, D. Willis, R. Sparks, B. Campbell, H. Schulzrinne, J. Lennox, B. Aboba, C. Huitema, D. Gurle, and D. Oran. SIP Extensions for Instant Messaging. *Internet Draft draft-rosenberg-impp-im-00.txt*, June 2000.
- [13] J. Rosenberg, D. Willis, R. Sparks, B. Campbell, H. Schulzrinne, J. Lennox, B. Aboba, C. Huitema, D. Gurle, and D. Oran. SIP Extensions for Presence. *Internet Draft draft-rosenberg-impp-presence-00.txt*, June 2000.
- [14] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson. RTP: A Transport Protocol for Real-Time Applications. *RFC 1889*, January 1996.
- [15] K. Singh and H. Schulzrinne. Interworking between SIP/SDP and H.323. *Internet Draft draft-singh-sip-h323-01.txt*.
- [16] International Telecommunication Union. Packet based Multimedia Communication Systems. *Recommendation H.323, Telecommunication Standardization Sector of ITU, Geneva, Switzerland*, February 1998.

Systematische Analyse von Schwachstellen

M. Schumacher <schumacher@ito.tu-darmstadt.de>

M. Hurler <hurler@dvs1.informatik.tu-darmstadt.de>

Buffer Overflows sind heute, obwohl das Problem schon lange bekannt ist, immer noch eine der häufigsten Ursachen für Schwachstellen in IT Systemen. Das liegt u.a. daran, dass die Fehler zwar grundsätzlich erkannt, aber nicht universell verstanden sind. In diesem Beitrag wird versucht, einige der Ursachen zu identifizieren, die dazu führen, daß in der IT Industrie aus Fehlern in Bezug auf die IT Sicherheit manchmal nichts dazu gelernt wird und wie Lösungen für dieses Problem aussehen können. In diesem Zusammenhang werden auch kurz aktuelle Projekte und Initiativen zu diesem Thema vorgestellt, u.a. die Arbeitsgruppe sec://house des Fachbereichs Informatik an der TU Darmstadt [SHHB00b].

Das Phänomen Buffer Overflow

Nach unserem Verständnis weist eine Systemkomponente eine Schwachstelle auf, wenn sie nur in unzureichendem Maß gegen Mißbrauch geschützt ist. Wird diese Schwachstelle ausgenutzt, so ist die vorgesehene Sicherheit des betroffenen IT- Systems gefährdet. Dabei ist es unerheblich, ob hinter einem solchen Verstoß gegen die Sicherheitsrichtlinien Absicht oder Zufall steckt. Ein typischer Vertreter der Kategorie Schwachstelle ist ein *Buffer Overflow*.

Ein Puffer (*Buffer*) ist ein zusammenhängender Speicherbereich, der Daten eines bestimmten Typs enthält. Ein Buffer Overflow tritt dann auf, wenn die Menge der Daten die tatsächliche Puffergröße übersteigt