HTWK Leipzig, Fakultät Informatik, Mathematik und Naturwissenschaften

Voice over IP

Aufbau und Analyse eines VoIP-Systems

Tim Georg Ebert, René Martin

2013

Inhaltsverzeichnis

1. Einleitung

Bereits seit einigen Jahren etabliert sich im Kommunikations-Bereich die Audioübertragung über andere Netzwerke als das herkömmliche Telefonnetz, wie z.B. das Internet. Bereits 2006 verkündete die Computerzeitschrift „c't“: der Marktanteil von VoIP - der Audio-Kommunikation über das Internet-Protokoll - machte im vorrangegangenem Jahr in China über 50% aus und stellt damit eine zunehmende Konkurrenz zur analogen Telefonie dar [1]. Zu der Zeit betrug der Marktanteil in Deutschland 12% [2]. Bis heute ist zu verfolgen, dass dieser Anteil weltweit stetig angestiegen ist und 2012 in Deutschland 55,4% betrug (siehe Abbildung 1.1).

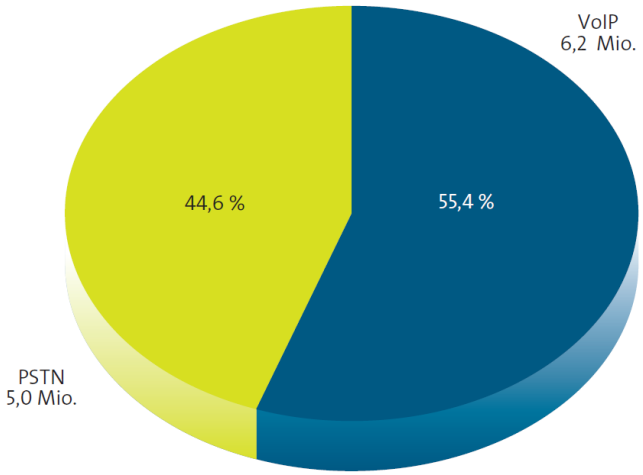

Abbildung 1.1 Marktanteil von VoIP und PSTN (analoge Telefonie) bei Telekomkommunikations-Wettbewerbsunternehmen von 2012; entnommen aus Quelle 14

Aufgrund der nicht Umfangreichen Integration in bestehende Telefon-Anlagen, der Flexibilität und der geografischen Unabhängigkeit der Benutzer wird VoIP bereits von diversen Telefonkonzernen als starke Konkurrenz empfunden [3].

Wegen des steigenden Marktanteiles ist es notwendig, VoIP genauer zu untersuchen, um Stärken und Schwächen der neuen Generation der Telefonie zu erkennen. Feststellbar ist, dass VoIP die herkömmliche Telefonie bereits in fast allen Anwendungsbereichen ersetzen kann. Der weiterhin steigende Marktanteil der IP Telefonie bestärkt ebenfalls diese These.

Im Rahmen der Arbeit wurden verschiedene VoIP-Systeme betrachtet, von denen einige der herkömmlichen Telefonie besonders im Vergleich auf Sprachqualität nicht nachstehen. Unterstützend zu den Untersuchungen werden in den Kapiteln 2.3 - 2.6 die Grundlagen von VoIP und seine einzelnen Protokolle vorgestellt. Wichtig sind dabei bereits existierende und in Internetanwendungen integrierte Protokolle wie das Session Initiation Protocol (Kapitel 2.4) zum Management multimedialen Konferenzen sowie das Real-time Transport Protokoll (Kapitel 2.5) zum Transport der multimedialen Daten. Ebenfalls wird die Umwandlung der Audiodaten bei VoIP näher beschrieben (Kapitel 2.6).

Im Zuge der Untersuchungen wurde das Open Source Programm „Peers“ analysiert. Diese Analyse beschäftigt sich insbesondere mit dem Kommunikationsverlauf zwischen den einzelnen Anwendern, in dem einzelne Pakete der beteiligten Protokolle genauer analysiert werden (Kapitel 3.3, 3.4). Außerdem folgt eine Analyse des in „Peers“ verwendeten Sprachkodierungsverfahrens (Kapitel 3.5). Besonders Interessant hierbei ist der Vergleich der Funktionen und Anforderungen Verschiedener VoIP-Kommunikationssysteme mit „Peers“. Dabei wird deutlich, dass „Peers“ nur fundamentale Funktionen von VoIP unterstützt und dies sich auch in der benötigten Rechenleistung wiederspiegelt.

Im Abschluss der Arbeit werden die analysierten Daten ausgewertet, um einerseits einen Vergleich von „Peers“ mit anderen VoIP-Systemen zu ermöglichen (Kapitel 3.6) und daraus Unterschiede im VoIP-Bereich zu ermitteln.

2. Grundlagen von VoIP

VoIP ist die Abkürzung für **Voice over Internet Protocol** und steht für die Kommunikation über ein auf dem Internet Protokoll basierenden Netzwerk (z.B. Internet oder Intranet) zur Übermittelung von Audiodaten. VoIP wird in Publikationen auch als IP Telefonie, Internet Telefonie oder digitales telefonieren bezeichnet.

2.1 Funktionsweise

Im Gegensatz zur herkömmlichen Telefonie, bei welcher kontinuierlich Daten über das Telefonnetz gesendet werden, zerlegt man die Audiodaten bei VoIP in Pakete, welche über eines der genannten IP-Netzwerke versendet werden. Da die Pakete über unterschiedliche Wege im Netz versendet werden, weisen sie oftmals unterschiedliche Laufzeiten auf. Dadurch besteht die Möglichkeit, dass Pakete beim Empfänger nicht in der richtigen Reihenfolge eingehen und sortiert werden müssen. Daher erwartet man einen gewissen Grad an QoS (Quality of Service).

QoS bei VoIP beschreibt die Anforderungen an IP-Netze bezüglich der Qualität der Sprachübertragung. Die wichtigsten Anforderungen, die an ein IP-Netz, welches für VoIP verwendet werden soll, gestellt werden, sind demzufolge eine ausreichende Bandbreite, eine geringe Verzögerung, Reduktion von Jitter-Auswirkungen sowie Verhinderung von Paket-verlusten.

2.2 Vor- und Nachteile von VoIP

VoIP wird zunehmend aus Kostengründen als Ersatz zur bisherigen Telefonie eingesetzt. Zusätzlich zu den grundlegenden Funktionen bietet VoIP einige zusätzliche Funktionen, welche bisherige Methoden nicht ohne weiteres realisieren können. Dazu gehören unter anderem das Führen mehrerer Telefonate gleichzeitig über eine Rufnummer oder die Ortsunabhänigkeit des Benutzers [3]. Dadurch ist die Rufnummer nicht mehr an eine spezielle Verbindung gebunden und kann theoretisch weltweit benutzt werden.

Allerdings bringt VoIP auch einige Nachteile mit sich. Die Qualität des Gespräches wird stark von der Stabilität der Internetverbindung beeinflusst, und es ist in der Regel eine Breitband Verbindung zum gewünschten Netzwerk erforderlich, um ähnlichen Komfort wie mit herkömmlicher analoger Telefonie zu erhalten [3]. Außerdem müssen die für eine VoIP Verbindung nötigen Geräte über das jeweilige Stromnetz des Hauses versorgt werden, was zu zusätzlichen Kosten führt und im Fall einer Katastrophe ist die Verfügbarkeit von Notrufen dadurch nicht gesichert. Die bereits genannte Ortsunabhängigkeit des Anrufers bringt auch den Nachteil mit sich, dass dieser nicht lokalisiert werden kann, was bei Notrufen problematisch ist. Ein weiterer Kritikpunkt ist die Übertragung über das Internet Protokoll an sich. Dieses kann aufgrund der Übertragungsart abgehört, gestoppt oder sogar verweigert werden.

2.2 Aufbau einer VoIP-Verbindung

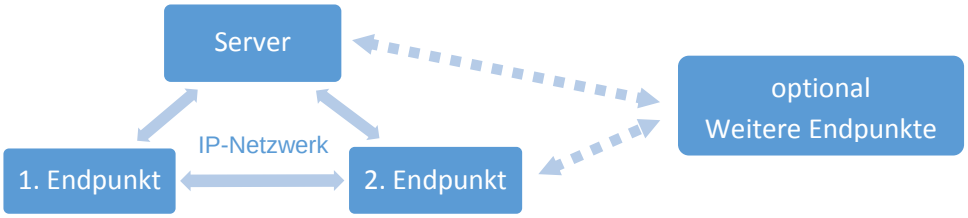

Abbildung 2.2.1 benötigte Teilnehmer zum Aufbau einer VoIP Verbindung; es können ebenfalls noch weitere Endpunkte involviert sein, wenn beispielsweise eine Telefon-konferenzen abgehalten wird

Es gibt verschiedene Möglichkeiten VoIP-Verbindungen aufzubauen. Diverse Software-Angebote im Internet nutzen eigene Protokolle, wie z.B. Skype oder ICQ [4]. VoIP wird allerdings auch im Telefonie-bereich eingesetzt und dafür wird in der Regel ein Verbindungsaufbau in 2 Schritten gewählt, da man zum Zeitpunkt des Anrufs nicht die genaue Position des 2. Endpunktes kennt.

Den ersten Schritt beim Aufbau einer Telefonverbindung über VoIP stellt die Suche nach den weiteren Teilnehmern mit Hilfe des Servers und Anfrage einer Kommunikation mit diesen Endpunkten dar (siehe Abbildung 2.2.1). Dies kann z.B. durch das Session Initiation Protocol realisiert werden.

Im zweiten Schritt wird zwischen allen Teilnehmern eine direkte Verbindung aufgebaut, welche die Audiodaten der einzelnen Personen in Echtzeit überträgt. Dies kann durch das Real-time Transport Protocol geschehen. Im Allgemeinen werden die Audiodaten für die Übertragung komprimiert um den entstehenden Datenstrom zu minimieren.

Diese Protokolle können sowohl von Computern mit entsprechender Software (sog. Softphones), als auch von modernen VoIP-fähigen Telefonen interpretiert werden und es besteht sogar die Möglichkeit zwischen verschieden-artigen Geräten eine Verbindung herzustellen, wenn sie die gleichen Standards unterstützen.

2.3 Session Initiation Protocol

Das erstmals 1999 unter dem Namen RFC 2543 veröffentlichte Session Initiation Protocol (SIP) ist ein einfach strukturiertes, textbasiertes Protokoll, welches die Übertragung von Echtzeitmedien wie Audio und Video über ein bestehendes IP-Netz ermöglicht [5]. Für VoIP wird es als Signalisierungsprotokoll verwendet. Es dient dem Aufbau und dem Abbau einer sogenannten Session, im VoIP einer Sprachübertragung zwischen zwei IP-Telefonen. Als Transportprotokoll wird hauptsächlich UDP verwendet, da es aufgrund seiner geringen Eigenlast für Echtzeitmedien geeignet ist. UDP ist unzuverlässiger als TCP, jedoch effizienter im Aufbau einer Session, da SIP selbst über Mechanismen zur Fehlerkontrolle verfügt.

SIP ist an SMTP und HTTP angelehnt, was die Integration in Internetanwendungen ohne höheren Aufwand ermöglicht. SIP funktioniert nach dem Request/Response-Prinzip auf Basis eines Client/Server-Prinzips. Die Adressierung erfolgt über den Internetstandart URL (Uniform Ressource Location), die Adressen ähneln hierbei denen von E-Mails.

Auf- und Abbau von SIP-Sessions erfolgt über aufeinanderfolgende Request und Response Benachrichtigungen in Form von SIP-Paketen. In der Basisspezifikation von SIP RFC3261 werden die sechs in Tabelle 2.3.1 ausgeführten Request-Typen definiert [6].

INVITE	Initiiert einen Anruf und enthält unter anderem die Adressen der Session Teilnehmer sowie optional Grund und Priorität des Anrufes
BYE	Initiiert den Abbau einer bestehenden Session
ACK	Bestätigt die Annahme eines Anrufes und initiiert die eigentliche Medien-übertragung
OPTIONS	Bietet die Möglichkeit abzufragen, welche unterschiedlichen Fähigkeiten (zum Beispiel Einbindung von Video oder Codierungsverfahren) den Session-Teilnehmern zur Verfügung stehen
CANCEL	Bricht den Verbindungsaufbau zu einer initiierten Medienübertragung ab
REGISTER	Übermittelt die Position eines Teilnehmers an den Server, der den SIP-Service bereitstellt

Tabelle 2.3.1 Request-Typen nach RFC3261

Wenn ein Nutzer beispielsweise einen anderen Nutzer anrufen möchte, so sendet er einen Request vom Typ INVITE. Ein Request vom Typ REGISTER wird verschickt, wenn der Nutzer sich bei seinem SIP-Provider anmeldet um anschließend von diesem weitere Requests empfangen zu können.

Die Response-Pakete beginnen mit einem dreistelligen Zahlencode, wobei die erste Ziffer den Typen der Klasse angibt. Response-Nachrichten sind immer antworten auf Requests [6].

Informational (1xx)	Teilt dem Absender mit, dass der Request bearbeitet wird
Success (2xx)	Bestätigt den erfolgreichen Empfang eines Requests
Redirection (3xx)	Teilt dem Absender mit, dass der Request weitergeleitet werden muss
Client-Error (4xx)	Signalisiert, dass der Request ungültig ist oder vom Server nicht ausgeführt werden kann
Server-Error (5xx)	Teilt dem Absender mit, dass der Server nicht in der Lage war, den Request auszuführen
Global-Failure (6xx)	Teilt dem Absender mit, dass der Request auf keinem Server ausführbar war

Tabelle 2.3.2 Response-Klassen nach RFC3261

In Tabelle 2.3.2 sind alle möglichen Response-Klassen dargestellt. Erhält beispielsweise ein Proxy-Server des SIP-Providers einen INVITE Request, so antwortet er mit „180 Trying“, was zur Klasse Informational gehört, und sendet einen Request an einen weiteren Proxy-Server oder den Zielnutzer, falls dieser beim Proxy-Server angemeldet ist. Erhält dieser Nutzer einen INVITE Request und möchte den Anruf annehmen, so generiert er die Antwort „200 OK“, welche zur Success Klasse gehört.

Request und Response Nachrichten unterscheiden sich prinzipiell nur in ihrer ersten Zeile. Bei Requests ist diese folgendermaßen aufgebaut:

	Methode	Request-URI	SIP-Version
Beispiel:	INVITE	sip:Teilnehmer@abc.de	SIP/2.0

Response Startzeilen sind wie folgt strukturiert:

	SIP-Version	Status-Code	Reason-Phrase
Beispiel:	SIP/2.0	200	OK

Dieses Beispiel signalisiert die Bestätigung einer erfolgreichen Request-Bearbeitung.

Nach der Startzeile folgen beliebig viele Header-Felder, in welchen weitere Parameter übergeben werden. Ein solcher Header ist beispielsweise in Abbildung 2.3.1 dargestellt. Abschließend, durch eine Leerzeile getrennt, folgt ein optionaler Message Body, in welchem Besonderheiten der RTP-Session näher beschrieben werden.

```
INVITE sip:tgebert@sip2sip.info SIP/2.0
Via: SIP/2.0/UDP 192.168.2.100:6060;rport;branch=z9hG4bKZtzMKXf1v
Route: <sip:proxy.sipthor.net>
Max-Forwards: 70
To: <sip:tgebert@sip2sip.info>
From: <sip:lordo@sip2sip.info>;tag=KNUEq7wS
Call-ID: tnG0flVB-1375013449222@Windrunner
CSeq: 3 INVITE
Content-Length: 212
Content-Type: application/sdp
Contact: <sip:lordo@192.168.2.100:6060;transport=UDP>
```

Abbildung 2.3.1 Beispiel eines SIP-Requests von lordo@sip2sip.info an tgebert@sip2sip.info; Call-ID Identifiziert den Anruf, CSeq nummeriert und beschreibt den Request-Typ, Via enthält Informationen zum Protokoll und dem Absender, Max-Forwards gibt an, wie viele Proxy-Server maximal zwischen den Telefonen durchlaufen werden dürfen

Ein Rechner, welcher mithilfe eines Softphones als VoIP-Telefon verwendet wird, bezeichnet man als User Agent. Man unterscheidet zwischen dem User-Agent-Client (UAC), dem Anrufer, und User-Agent-Servern (UAS), welche angerufene Komponenten darstellen.

Zur Lokalisierung von SIP-Telefonen wird ein Registrar-Server verwendet. Dieser empfängt von den User Agents in regelmäßigen Abständen eine REGISTER-Nachricht, welche deren SIP- und IP-Adressen beinhaltet. Diese Adressen werden in einem Location-Server abgespeichert [6].

Zum endgültigen Verbindungsaufbau zwischen zwei User Agents werden verschiedene Server verwendet. Die wichtigsten sogenannten SIP-Server sind Proxy-Server und Redirect-Server. Im praktischen Einsatz besteht die Struktur des benutzen IP-Netzes aus mehreren gemischten Servern [5].

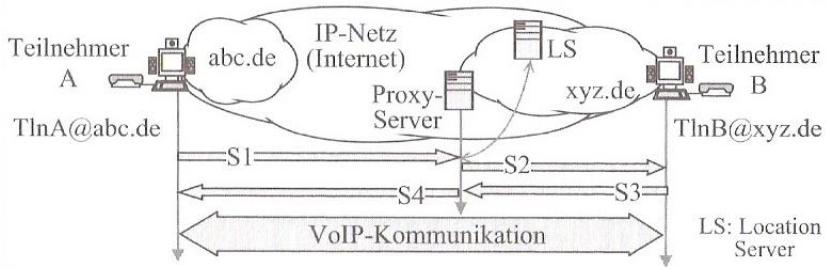

Abbildung 2.3.1 Call-Flow eines Verbindungsaufbaus über einen Proxy-Server; Teilnehmer A möchte Teilnehmer B anrufen und sendet eine Anfrage an einen Proxy-Server, welcher diese weiterhin bearbeitet und A mitteilt, wie die Antwort von B ist.; entnommen aus Quelle 5

Der Einsatz eines Proxy-Servers ist in Abbildung 2.3.1 zu sehen. Teilnehmer A kontaktiert den Proxy-Server und gibt an, in welcher Internet-Domain sich Teilnehmer B befindet, allerdings weiß er nicht, welchen Rechner (welches Telefon) der Angerufene in dieser Domain benutzt. Der Proxy-Server nimmt den Anruf entgegen und bezieht von einem Location-Server (in Abbildung als „LS“ dargestellt) die Adresse von Teilnehmer B innerhalb der Domain. Daraufhin leitet der Proxy-Server den Anruf an die Zieladresse weiter [5].

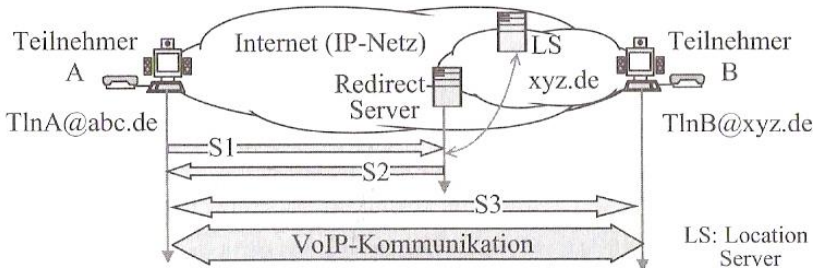

Abbildung 2.3.2 Call-Flow eines Verbindungsaufbaus zu einem Redirect-Server; Teilnehmer A möchte Teilnehmer B anrufen und erhält von einem Redirect Server dessen aktuelle Adresse für die Audio-Kommunikation; entnommen aus Quelle 5

Die Aufgabe eines Redirect-Servers besteht darin, den Anrufer direkt über die Zieladresse des zweiten Gesprächsteilnehmers zu informieren, wie in Abbildung 2.3.2 dargestellt ist [5]. Teilnehmer A ruft zunächst den Redirect-Server an, um die Zieladresse und somit Position von Teilnehmer B zu erfahren. Danach wird eine direkte Verbindung zwischen den beiden Geräten aufgebaut. Weiterhin ist es möglich, mit Hilfe von SIP Anrufverzweigungen einzurichten. Sollten mehrere Telefone gleichzeitig verwendet werden, wird der Anrufer nach dem Erhalten der Zieladressen an mehrere Telefone gleichzeitig weitergeleitet.

Eine weitere integrierbare Funktion ist der Einsatz eines Voice-Mail-Servers, welcher als Anrufbeantworter dienen kann [5]. Kommt es bei der Initiation eines Anrufes zu einem Timeout seitens des Angerufenen, leitet dessen Location bzw. Redirect-Server den Anruf an ein Voice-Mail-System weiter. Dieses signalisiert dem Anrufer, dass der Anruf ersatzweise von ihm entgegengenommen wird.

2.4 Real-time Transport Protocol

Bei der Verwendung des Real-time Transport Protocol (RTP) werden Echtzeitmedien als eine Folge von RTP-Paketen übertragen. Im RTP-Header sind notwendige Daten zum transportierten Medium angegeben. In Abbildung 2.4.1 ist der Aufbau eines typischen RTP-Paketes zu sehen.

RTP-Header (min. 12 Bytes) enthält: • RTP-Version • Padding-Bit • Extension-Bit • CSRC Count • Marker-Bit • **Payload-Type** • **Sequence Number** • **SSRC-Identifier** • CSRC-Identifiers (optional) • Header-Extension (optional)	Payload (Nutzlast) Enthält komprimierte Audio- bzw. Video-Daten Entsprechend des im Payload-Type festgelegten Typs

Abbildung 2.4.1 allgemeiner Aufbau eines RTP-Pakets; im Header-Bereich werden verschiedene Informationen zum Payload und zur Quelle der Daten übermittelt

Der Payload-Typ gibt an, um was für ein Medium es sich handelt bzw. wie dieses kodiert wurde. Ein Zeitstempel, welcher in Abhängigkeit des Payload-Typen initiiert wurde, ermöglicht es, Jitter in der Übertragung erkennen und ausgleichen zu können [7].

Die Sequenznummer der Pakete gibt an, in welcher Reihenfolge die Pakete losgeschickt wurden und erlaubt es dem Empfänger, Paketverluste zu erkennen bzw. gegebenenfalls die richtige Reihenfolge der Pakete wieder herzustellen [7].

Der „Synchronization Source Identifier“ (SSRC-Identifier) enthält Angaben zur Quelle des Mediums. Dies erlaubt es dem Empfänger, Datenströme unterschiedlicher eingehender Medien zu sortieren. Wenn der Datenstrom über Zwischensysteme verändert und weitergesendet worden ist, dann folgt eine Liste von Originalquellen („Contributing Source Identifier“) [7]. Insgesamt besitzt ein RTP-Header eine Größe von mindestens 12 Bytes.

Das im RTP integrierte RTCP (Real-time Transport Control Protocol) enthält Informationen über einzelne Gesprächsteilnehmer, Quellen von Medien und den Verlauf einer bestehenden Konferenz [5]. Es ermöglicht die Überwachung der Übertragungsqualität sowie die Organisation von Konferenzen mit mehr als zwei Teilnehmern, indem es alle Konferenzteilnehmer über Zu- und Abgänge anderer Benutzer informiert. Die Qualitätsüberwachung wird durch sender- und empfängerbezogene Report-Pakete realisiert. Diese werden periodisch versendet und teilen den restlichen Gesprächsteilnehmern Informationen über die aktuelle Qualität der Konferenz mit, indem sie beispielsweise Informationen über die eigene Empfangsqualität bzw. Paketverlust übermitteln.

2.5 Sprachverarbeitung bei VoIP

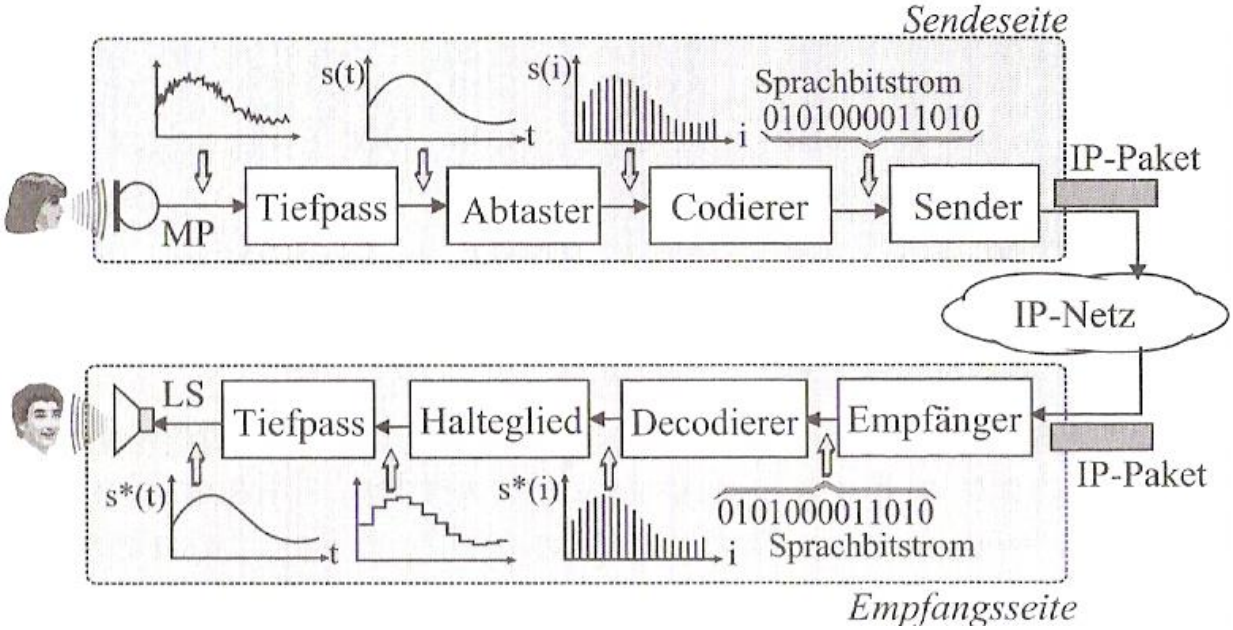

Abbildung 2.5.1 Schema der Audioverarbeitung und -übertragung bei VoIP; entnommen aus Quelle 5

In Abbildung 2.5.1 ist die Aufnahme, Verabeitung, Übermittlung und Ausgabe von Audiosignalen bei VoIP abgebildet. Eingehende Audiosignale werden über ein Mikrofon aufgenommen, in einem Tiefpassfilter geglättet und anschließend mit einem Abtaster in regelmäßigen, zeitlich kurzen Abständen (für Audio-Kommunikation meist 0,125 ms) abgetastet [5]. Die Folge der gemessenen Werte wird daraufhin quantisiert (komprimiert durch z.B. Runden) und in binäre Codewörter umgewandelt. Aus dem entstehenden Datenstrom werden IP-Pakete gebildet, welche über IP-Netze versendet und empfangen werden können.

Auf Grund der hohen Abtastfrequenz (in der Regel 8000 Hz) ist der entstehende Datenstrom zu groß, um ihn nicht komprimiert beispielsweise über eine ISDN-Leitung versenden zu können. Gerade wenn zusätzlich zu der Audioübertragung eine Videoübertragung erwünscht wird, ist eine Komprimierung der Abtastwerte durch Quantisierung unabdingbar, um die entstehenden Datenmenge in Echtzeit übertragen zu können, auch wenn diese Komprimierung einen Verlust der Originalqualität bedingt.

Die vom Empfänger erhaltenen Pakete werden bei diesem in Codefragmente zerlegt, welche jeweils eine bestimmte Spannung kodieren. Bis zu einem Spannungswechsel wird die aktuelle Spannung durch ein Halteglied gehalten. Zuletzt wird erneut ein Tiefpassfilter verwendet, um eine Nachbildung des Originalen Audiosignals zu erhalten [5].

Man unterscheidet zwischen Abtastwert-orientierter und Segment-orientierter Sprachcodierung. Abtastwert-orientiert bedeutet, dass eingehende Sprachsignal wird mit einer bestimmten Frequenz abgetastet werden, woraufhin für jede Abtastung ein meist durch Quantisierung komprimierter Wert (Sample) kodiert wird. Segment-orientierte Sprachcodierung basiert auf der Zerlegung des eingehenden Sprachsignales in Zeitsegmente mit einer Länge zwischen 10 und 30 ms. Aus den analysierten Segmenten ergeben sich Parameter, welche codiert und versendet werden. Der Empfänger rekonstruiert die Segmente synthetisch auf Basis der übermittelten Parameter.

3. Aufbau und Analyse eines VoIP-Systems

Für den Aufbau des VoIP-Systems wird als Softphone das Sourceforge Open-Source Projekt “Peers” von Yohann Martinau verwendet [8]. Zusätzlich benötigt man einen SIP-Provider, welcher einen Registrar-Server und Proxy-Server zur Verfügung stellt. Dazu wird der kostenlose Service von sip2sip.info verwendet [9]. Alternativ können aufgrund des in „Peers“ verwendeten SIP-Standards die ebenfalls kostenlosen Services von ippi.fr oder ekiga.net genutzt werden. Zur späteren Analyse der Software wird der Open-Source Packet-Analysierer Wireshark Version 1.10.0 verwendet [10].

3.1 Erstellen von „Peers“

Der Java-Sourcecode des Opensource Programms „Peers“ ist auf der Projektseite bei sourceforge.com frei verfügbar [11]. Zum Erstellen des Projekts wurde im Rahmen dieser Arbeit der in die Entwicklungsumgebung Eclipse eingebaute Java-Compiler [12] mit dem „Java Runtime Enviroment 1.7“ genutzt und als Startklasse wurde net.sourceforge.peers.gui.MainFrame verwendet. „Peers“ verwendet die in Abbildung 3.1.1 dargestellte die graphische Benutzeroberfläche. Nach der Wahl des Menüpunktes „Edit“ und anschließend „Account“ wird die der Eingabe der SIP-Verbindungsinformationen ermöglicht (siehe Kapitel 3.2). In der Account Ansicht sind die zur Registrierung beim SIP-Provider notwendigen Daten einzugeben. Anschließend ist der Nutzer in der Lage VoIP-anrufe durch Eingabe einer SIP-Adresse zu tätigen. Während des Gesprächs ist es außerdem möglich über die Anruf-Ansicht weitere Nummern zu wählen oder das Gespräch zu beenden.

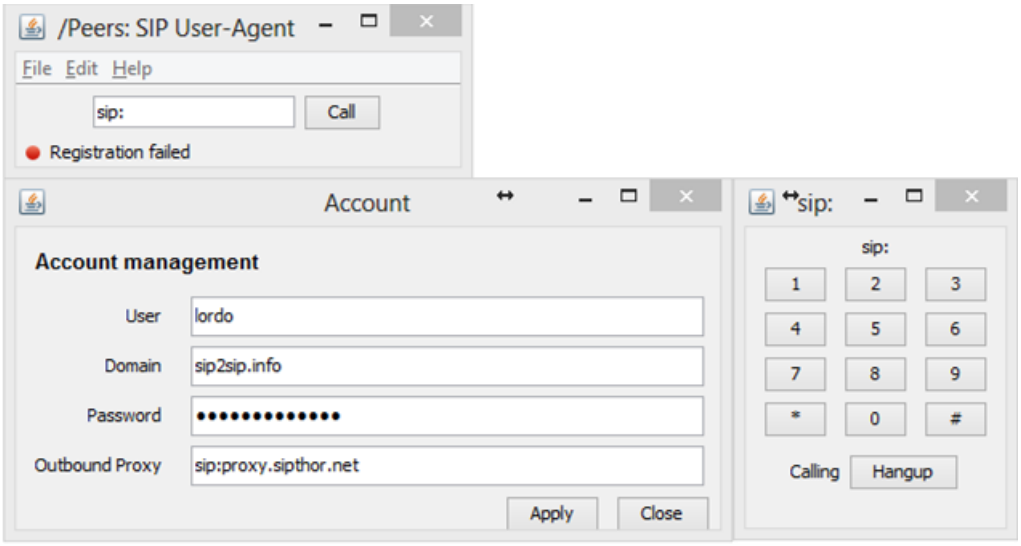

Abbildung 3.1.1 Graphische Benutzeroberfläche von „Peers“

3.2 Einrichten einer SIP-Verbindung

Zunächst ist es erforderlich auf der Webseite von sip2sip.info einen Account anzulegen, um sich später über das SIP beim Registrar-Server anzumelden. Eine ansicht der Website sip2sip.info ist in Abbidlung 3.2.1 zu sehen.

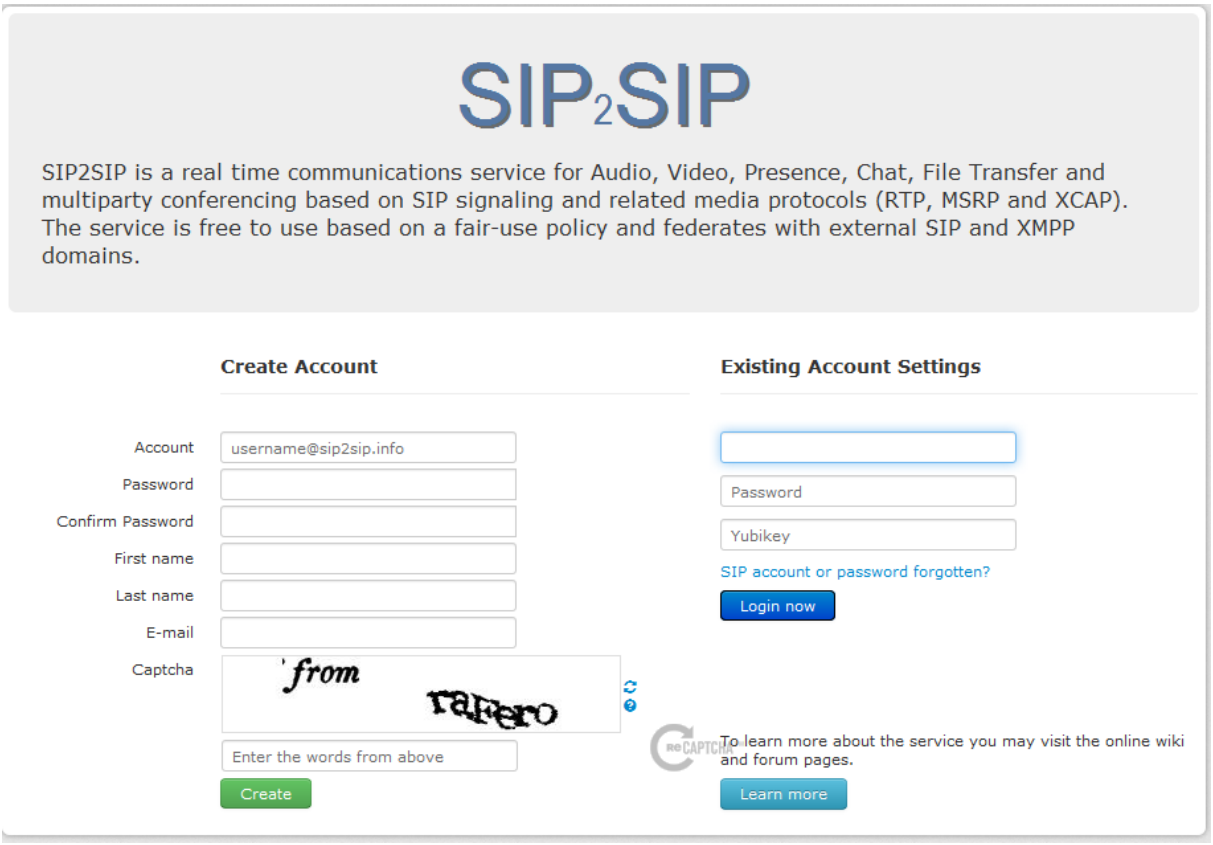

Abbildung 3.2.1 Ansicht von sip2sip.info beim Registrieren eines Accounts für den SIP-Service; entnommen aus Quelle 9

Nach der Registrierung und der Bestätigung eines Accounts ist es möglich das SIP-Softphone mit den entsprechenden Daten einzurichten. Dazu ist es notwendig, die vom SIP-Provider gegebenen Daten in der Account Ansicht von Peers einzutragen. Im Falle von sip2sip.info als Provider gibt werden als User und Password die selbst gewählten Daten eingegeben, als Domain „sip2sip.info“ und als Outbound Proxy „sip:proxy.sipthor.net“. Beim anschließenden Wählen des Apply-Buttons erscheint die Nachricht „Registered“, falls die Registrierung erfolgreich war. Ist dies nicht der Fall, sind die Account-Daten noch einmal zu überprüfen.

3.3 Analyse der SIP-Schicht von „Peers“

Die in „Peers“ gewählte SIP-Implementierung entspricht dem Standard RFC3261 und wird über das UDP vom Port 6060 (lokal) zum Port 5060 (Registrar-Server) transportiert. Dies setzt eine Registrierung des User Agent Client (UAC) beim Registrar-Server voraus.

```
REGISTER sip:sip2sip.info SIP/2.0
Via: SIP/2.0/UDP 192.168.2.100:6060;rport;branch=z9hG4bKsFNPBSGna
Route: <sip:proxy.sipthor.net>
Max-Forwards: 70
To: <sip:lordo@sip2sip.info>
From: <sip:lordo@sip2sip.info>;tag=8H0zSmfu
Call-ID: yuzqSIru-1374936303793@Windrunner
CSeq: 1 REGISTER
Contact: <sip:lordo@192.168.2.100:6060;transport=UDP>

Response:
SIP/2.0 401 Unauthorized
Via: SIP/2.0/UDP 192.168.2.100:6060;received=88.74.130.81;
     rport=6060;branch=z9hG4bKsFNPBSGna
To: <sip:lordo@sip2sip.info>;tag=f2f36c970cbb772c79daadc5c9d6e415.cae4
From: <sip:lordo@sip2sip.info>;tag=8H0zSmfu
Call-ID: yuzqSIru-1374936303793@Windrunner
CSeq: 1 REGISTER
WWW-Authenticate: Digest realm="sip2sip.info",
                  nonce="51f3dd0f95a25db73abdf88f0ae9cb22090aea9c"
Server: SIP Thor on OpenSIPS XS 1.9.0
Content-Length: 0
```

Abbildung 3.3.1 Auszug der von dem Programm Wireshark registrierten Header der SIP-Pakete, die auf dem Port 5060 verschickt wurden; SIP-Paket zur Registrierung des Nutzers beim Registrar-Server und dessen Antwort, dass der Zugriff ohne Passwort nicht gestattet und stattdessen eine Digest-Authentifizierung von Nöten ist

In Abbildung 3.3.1 wird gezeigt, dass der Registrar-Server eine Digest-Authentifizierung erfordert, bevor der SIP-Service genutzt werden kann. Dazu wird der im WWW-Autheticate Header nonce-Wert genutzt um mithilfe des MD5-Verschlüsselungsalgorithmus ein response-Wert zu berechnen und diesen in einem weiteren SIP-Request im Authorization Header zu übermitteln. Wenn dieser Wert mit dem des lokal beim Registrar-Server berechneten übereinstimmt entspricht der Status-Code der Antwort „200 OK“.

```
NOTIFY sip:88.74.130.81:6060 SIP/2.0
Via: SIP/2.0/UDP 81.23.228.129:5060;branch=0
From: sip:keepalive@81.23.228.129;tag=3d3dddec
To: sip:88.74.130.81:6060
Call-ID: 6d063a46-23beeb16-25fe8@81.23.228.129
CSeq: 1 NOTIFY
Event: keep-alive
Content-Length: 0

Response:
BSIP/2.0 405 Method Not Allowed
From: sip:keepalive@81.23.228.129;tag=3d3d6f93
Call-ID: 6d063a46-23be7cbd-2571e@81.23.228.129
CSeq: 1 NOTIFY
Via: SIP/2.0/UDP 81.23.228.129:5060;branch=0
To: sip:88.74.130.81:6060
Allow: INVITE, ACK, CANCEL, OPTIONS, BYE
Contact: <sip:192.168.2.100:6060;transport=UDP>
```

Abbildung 3.3.2 Auszug der von dem Programm Wireshark registrierten Header der SIP-Pakete, die auf dem Port 5060 verschickt wurden; SIP-Request vom Registrar-Server an den User-Agent, um dessen weitere Anwesenheit zu prüfen und dessen Antwort, dass die NOTIFY-Methode beim Standard RFC3265 nicht erlaubt sei

Anschließend sendet der Registrar-Server ca. alle 50 Sekunden ein Notify-Request (siehe Abblidung 3.3.2), mit welchem „Peers“ nicht umgehen kann wie man an dem Status-Code der Antwort („405 Method not allowed“) sehen kann. Dies liegt daran, dass Notify-Requests erst in RFC3265 definiert werden, welches in „Peers“ nicht implementiert ist [13]. Der Server von sip2sip.info erkennt an der Antwort jedoch trotzdem, dass der User-Agent noch aktiv ist und erhält die Verbindung weiterhin aufrecht.

Anrufe in „Peers“ verlaufen nach dem in Abbildung 3.3.3 dargestellten Call-Flow Diagramms. Zu Beginn eines Anrufes sendet der UAC zu seinem registrierten Proxy-Server ein INVITE-Request, welches zunächst mit „407 Proxy Authentification Required“ beantwortet wird, und wiederrum ist eine Digest-Authentifizierung erforderlich. Daraufhin wird ein weiterer INVITE-Request mit entsprechender Authentifizierung versendet. Dieser wird zunächst mit „100 Trying“ und „100 Giving a try“ beantwortet. Falls der entsprechende Ziel-UAC erreicht wird auch mit „180 Ringing“ und beim Bestätigen der Verbindung mit „200 OK“ beantwortet. Anschließend wird vom 2. UAC noch ein ACK Request gesendet, woraufhin die Verbindung der RTP-Session aufgebaut werden kann. In den INVITE-Requests werden zusätzlich zum Header noch Informationen im Content Bereich des SIP gesendet. Diese basieren auf dem Session Description Protocol und geben Auskunft über die gewünschte Medienverbindung. Dabei werden mögliche Audio-Sample Raten, Encoding-Typen und die lokale RTP-Portnummer übermittelt. Im „200 OK“ Response zum

entsprechenden INVITE-Request wird daraufhin ebenfalls über das SDP die akzeptierte Form der Medienübertragung definiert.

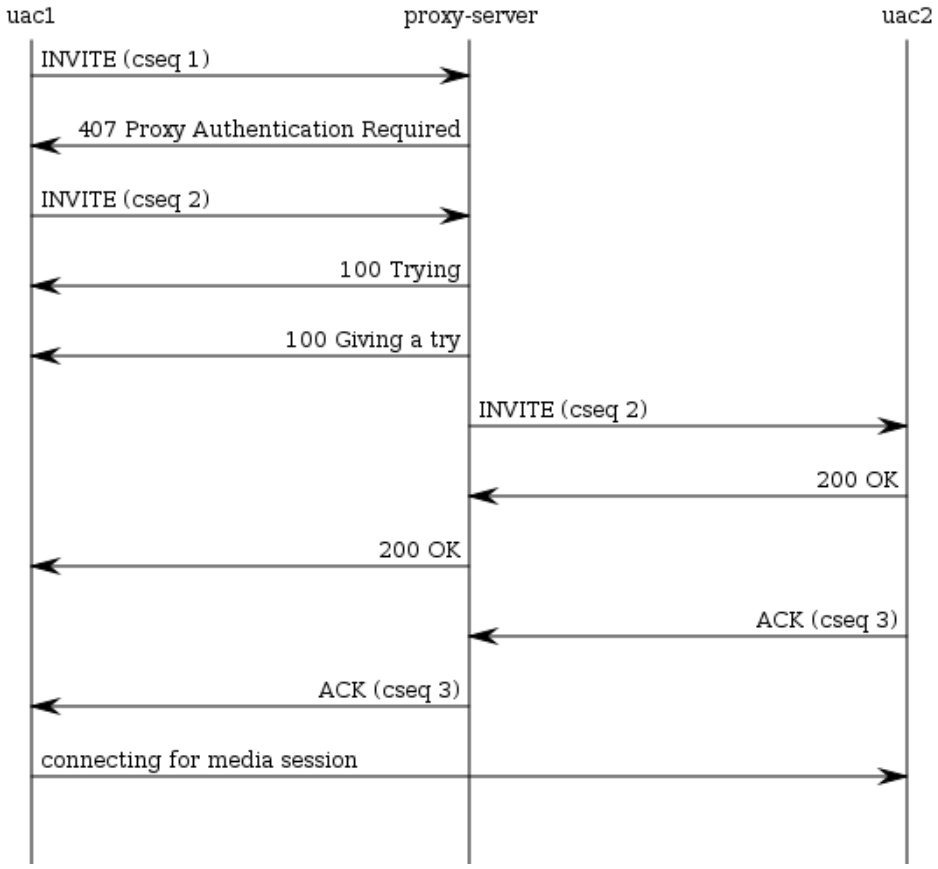

Abbildung 3.3.3 Call-Flow Diagramm eines einfachen Medien-Verbindungsaufbaus in „Peers“, Beschriftungen mit Großbuchstaben stehen für Requests und Beschriftungen mit Zahlen beginnend stehen für Antworten auf Requests

Sobald einer der UAC die Verbindung beenden möchte sendet er ein BYE-Request über seinen Proxy an den anderen UAC. Dies ist notwendig, damit das Abbrechen der RTP-Verbindung nicht als Fehler bei der Übertragung interpretiert wird.

3.4 Analyse der RTP-Schicht von „Peers“

Die RTP-Implementierung von Peers befindet sich im Quelltext größtenteils unter dem Namespace **net.sourceforge.peers.rtp**. Dieser beinhaltet die Klassen RFC3551, RFC4733, RTPListener, RTPPacket, RTPParser und RTPSession. In den Klassen RFC3551 und RFC4733 werden Konstanten der gewählten Standards definiert. RTPListener ist ein Interface zum Verwalten des Package Listener. RTPPacket enthält Daten eines Pakets wie z.B. dessen Payload-Type und RTPParser ist eine Klasse welche ein RTPPacket-Objekt in die eigentliche

binäre Form eines RTP-Pakets kodieren kann und ein RTP-Paket dekodieren kann. Die Klasse RTPSession erhält die eigentliche RTP Sitzung der Teilnehmer aufrecht, indem sie RTP-Pakete über den entsprechenden RTPListener versendet und empfängt.

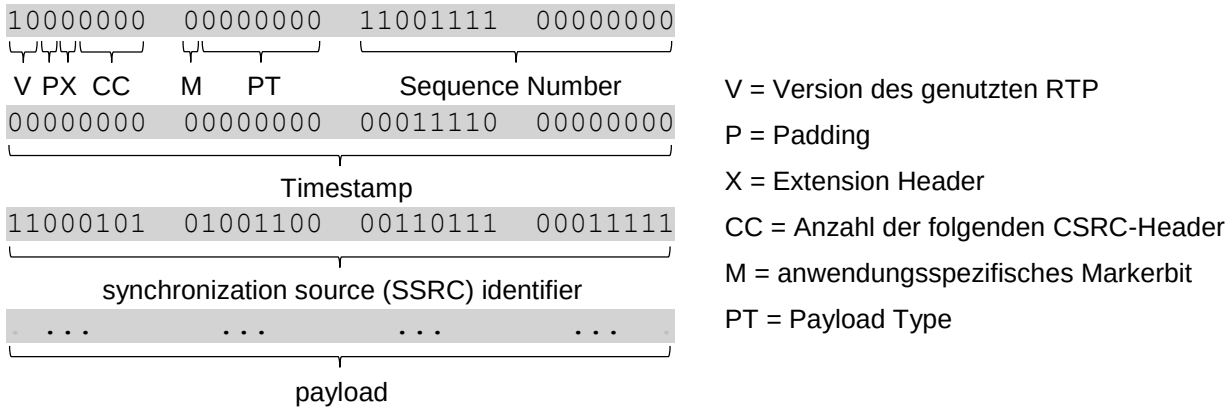

Abbildung 3.4.1 Auszug der ersten Bytes eines von „Peers“ gesendeten RTP-Pakets; grau hinterlegt sind die eigentlichen Inhalte des Pakets und zusätzlich sind die Bedeutungen der Daten beschrieben; zwischen dem SSRC Identifier und Payload befinden sich keine Contributing Source Identifiers und Header-Erweiterungen, da „Peers“ von ihnen keinen Gebrauch macht

In Abbildung 3.4.1 ist der Header (die ersten 12 Bytes) eines von „Peers“ gesendeten und von „Wireshark“ auf Port 8000 aufgenommenen RTP-Paketes dargestellt. Das gesamte Paket hat eine Länge von 172 Bytes, wobei die 160 nicht dargestellten Bytes den payload, also die kodierten Audiodaten beinhalten. Hierbei ist ersichtlich, dass „Peers“ die RTP-Version 2 verwendet, welches durch den Standard RFC3551 definiert wurde. Ein Padding ist nicht von Nöten, da das gewählte Verschlüsselungsverfahren (siehe Kapitel 3.5) keine feste Paketlänge erfordert und ein möglicher Extension Header wird ebenfalls nicht verwendet. Des Weiteren nimmt „Peers“ lediglich über die Java-Sound-Bibliothek Audiodaten auf und verändert diese nicht, weshalb die Anzahl der folgenden CSRC-Header gleich null ist. Das Marker-Bit wird bei ausgehenden Paketen nicht gesetzt und als Payload-Type wird standardmäßig null was für Pulse Code Modulation mit U-Law Codec - PCMU genannt - (bzw. acht was für Pulse Code Modulation mit A-Law Codec - PCMA genannt -) steht. Die verwendete Methode kann in der Konfigurationsdatei umgestellt werden. Die Sequence Number und Timestamp zählen den Paketen entsprechend weiter. Als SSRC Identifier generiert „Peers“ beim Verbindungsaufbau eine Zufallszahl, welche für alle ausgehenden Pakete verwendet wird. Dadurch kann der Empfänger bei mehreren eingehenden Verbindungen die Quellen der Daten festgestellt werden.

Bei eingehenden Paketen verarbeitet „Peers“ keinerlei Header-Informationen, die mit einem Paket mitgeliefert werden, außer dem Payload Type, aufgrund dessen eine PCMU- bzw. PCMA-Dekodierung erfolgt. Außerdem werden keinerlei RTCP-Pakete verschickt bzw. verarbeitet. Demzufolge ist nur eine simple Audio-Kommunikation mit einem einzelnen Teilnehmer zu jeder Zeit möglich.

3.5 Analyse der Verschlüsselung von „Peers“

Aus „Peers“ Quelltext in der Klasse **net.soucerforge.peers.media.PcmuEncoder** geht hervor, dass die Verschlüsselung bei Peers über den Codec G.711 realisiert wird. Es handelt sich um eine Abtastwert-orientierte Sprachcodierung mit einer Abtastfrequenz von 8000 Hz. Die gemessenen Samples werden mittels Quantisierung auf eine Datenmenge von 8 Bit pro Abtastung verlustbehaftet komprimiert. Dies resultiert in einer benötigten Bandbreite von 64kBit/s für die Audio-Daten und gemeinsam mit den IP-, UDP- und RTP-Header bei ca. 80kBit/s.

Zur Komprimierung in G.711 können das A-Law sowie das µ-Law Verfahren verwendet werden. In Peers ist das µ-Law Verfahren voreingestellt, kann jedoch variiert werden. µ-Law codiert 14 Bit lange Audio-Samples:

Eingehendes Audio-Sample	Komprimiert
v00000001abcdr	v000abcd
v0000001abcdrr	v001abcd
v000001abcdrrr	v010abcd
v00001abcdrrrr	v011abcd
v0001abcdrrrrr	v100abcd
v001abcdrrrrrr	v101abcd
v01abcdrrrrrrr	v110abcd
v1abcdrrrrrrrr	v111abcd

Tabelle 3.5.1 Komprimierungstabelle bei dem µ-Law Algorithmus zur Komprimierung von Audiodaten; ein eingehendes Audiosample wird von einer Länge von 14 Bit verlustbehaftet auf 8 Bit komprimiert

Auf die Magnitude des eingehenden Bitstroms wird 32 (binär: `100000`) addiert. Anschließend wird die Anzahl der Nullen, welche nach dem Vorzeichenbit (`v`) folgen, gezählt. In der komprimierten Ausgabe folgt nach dem Vorzeichenbit das Komplement der Anzahl der Nullen vor der ersten Eins. Die nächsten 4 Stellen (`a`, `b`, `c`, `d`) nach dieser werden ebenfalls übernommen, der Rest (`r`) wird abgeschnitten, da dieser vergleichsmäßig kleine Werte (32 Mal kleiner als die erste Eins) kodiert. Dieses Verfahren kann verkürzt mit der in Tabelle 3.5.1 gezeigten Schreibweise dargestellt werden.

Beispiel:
Aus dem gemessenen Sample ergibt sich ein Wert von 12 (1100). Die zu codierenden 14 Bit sind dementsprechend 00000000001100, hinzuaddiert werden 32: 00000000101100. Hier ist erkennbar, dass die Addition der 32 notwendig ist, da die Anzahl der Nullen bei Werten betragsmäßig kleiner als 32 mehr als 7 beträgt. Jedoch sind für das Komplement dieser Anzahl nur drei Bits vorgesehen. Die Anzahl der Nullen nach dem Vorzeichen (0, da positiv) und vor der ersten Eins im Beispiel beträgt 7, binär 111, im Komplement 000. Zusammen mit den folgenden 4 Ziffern des Eingangsstroms ergibt sich komprimiert: 00000110. Das letzte Bit (0) fällt weg.

Durch die Addition der 32 kann es bei betragsmäßig großen Werten zu einem Overflow-Fehler kommen. Daher müssen die eingehenden Werte betragsmäßig kleiner als 8159 sein (8159 + 32 = 8191, 8192 kann nicht mehr durch 13 Bit dargestellt werden). In diesem Bereich werden Frequenzen zwischen 300 Hz und 3400 Hz realisiert.

3.6 Vergleich der Leistungsanforderung von „Peers“ zu „Skype und „Jitsi“

In den Kapiteln 3.3 – 3.5 wurden „Peers“ Funktionen vielseitig betrachtet und man kann zu dem Schluss kommen, dass zwar in der Lage ist eine VoIP-Verbindung aufzubauen und erhalten, jedoch weitere in der VoIP-Telefonie verwendete Funktionen, wie z.B. Konferenzgespräche oder Anrufweiterleitung nicht implementiert sind. Zur Analyse der Leistungsanforderungen von VoIP-Softphones werden daher noch die ebenfalls kostenlosen Programme „Jitsi“ und „Skype“ herangezogen. „Jitsi“ ist ein Java basierter Open-Source SIP-Kommunikator, welches zu Gruppengesprächen und Anrufweiterleitungen in der Lage ist. „Skype“ wird von Microsoft zur Verfügung gestellt, nutzt ein eigenes Übertragungsprotokoll und unterstützt Gruppen. Alle in diesem Kapitel gemessenen Werte wurden auf einem Computer mit dem Betriebssystem „Microsoft Windows 8“ (Build 9800) im Ressourcenmonitor ermittelt.

Programm	Ruhezustand	Telefonat mit 2 Teilnehmern	Konferrenzgespräch
„Skype“	0,11% (0,02%)	1% (0,15%)	1,33% (0,18%)
„Jitsi“	0,01% (0,01%)	2,32% (0,07%)	2,98% (0,08%)
„Peers“	<0,01% (<0,01%)	0,26% (0,02%)	-

Tabelle 3.6.1 gemessene Prozessor-Auslastung der Programme „Skype“, „Jitsi“ und „Peers“ im Ruhezustand, Telefonat mit 2 Teilnehmern und Konferenzgespräch mit 3 Teilnehmern; relative Auslastung eines Intel i7-3770 Prozessor-Kern; Werte in Klammern entsprechen der mittleren quadratischen Abweichung

In Tabelle 3.6.1 werden die CPU-Auslastungen der geprüften Programme im Ruhezustand, bei Telefonaten mit 2 Teilnehmern und falls möglich Konferenzen mit 3 Teilnehmern verglichen. Man erkennt, dass alle Softphones im Ruhezustand den Prozessor erwartungsgemäß kaum auslasten. Auch während Audiokommunikationen sind die CPU-Auslastungen durchweg gering. Allerdings sieht man, dass „Peers“ den geringsten Zuwachs aufweist, was durch den vergleichsweise einfachen Komprimierungsalgorithmus zu erklären ist.

Programm	Ruhezustand	Telefonat mit 2 Teilnehmern	Konferrenzgespräch
„Skype“	185.800 KB (700 KB)	216.200 KB (1.140 KB)	214.900 KB (1.180 KB)
„Jitsi“	104.500 KB (150 KB)	124.700 KB (400 KB)	127.400 KB
„Peers“	70.700 KB (90 KB)	130.000 KB (250 KB)	-

Tabelle 3.6.2 gemessene Hauptspeicherbelegung der Programme „Skype“, „Jitsi“ und „Peers“ im Ruhezustand, Telefonat mit 2 Teilnehmern und Konferenzgespräch mit 3 Teilnehmern; gerundeter Mittelwert von 60 Sekunden Messung; Werte in Klammern entsprechen der mittleren quadratischen Abweichung

Der benötigte Hauptspeicher der untersuchten Programme ist in Tabelle 3.6.2 dargestellt. Im Vergleich zu anderer Software sind die Anforderungen in diesem Bereich gering. Bei Gesprächen wird ein Teil der Daten im Hauptspeicher zwischengelagert, bis er abgespielt werden muss. Diesen „Vorlauf“ nennt man Audio-Buffer. Auffallend ist, dass bei Konferenzen mit mehreren Teilnehmern die Hauptspeicherbelegung nicht weiter ansteigt. Dies ist damit zu begründen, dass die erhaltenen Daten in Echtzeit verarbeitet werden und der Audio-Buffer nicht weiterhin vergrößert werden muss.

Programm	Ruhezustand	Telefonat mit 2 Teilnehmern	Konferrenzgespräch
„Skype“	100 B/s (30 B/s)	9.400 B/s (520 B/s)	20.700 B/s (1.530 B/s)
„Jitsi“	34 B/s (7 B/s)	6.100 B/s (370 B/s)	11.800 B/s (480 B/s)
„Peers“	14 B/s (3 B/s)	17.200 B/s (2.130 B/s	-

Tabelle 3.6.3 gemessene Bandbreiten-Anforderung der Programme „Skype“, „Jitsi“ und „Peers“ im Ruhezustand, Telefonat mit 2 Teilnehmern und Konferenzgespräch mit 3 Teilnehmern; gerundeter Mittelwert von 60 Sekunden Messung; Werte in Klammern entsprechen der mittleren quadratischen Abweichung

In Tabelle 3.6.3 sieht man die Netzwerkauslastung der analysierten Softphones. Alle Programme weisen im Ruhezustand eine geringe Bandbreiten-Anforderung vor, was daran liegt, dass sie mit ihrem jeweiligen Registrar-Server bzw. Login-Server bei Skype Informationen austauschen. Im Vergleich mit Tabelle 3.6.1 sieht man, dass bei einem Gespräch mit 2 Teilnehmern Programme mit geringerer CPU-Auslastung eine höhere Bandbreite benötigen. Trotzdem kann jedes Programm selbst mit einer ISDN-Leitung verwendet werden. Die Netzwerkauslastung nimmt mit

jedem weiteren Teilnehmer zu. Dies liegt daran, dass jeder die Audio-Daten an jeden anderen versendet.

Abschließend ist zu sagen, dass „Peers“ weniger Funktionen als andere Softphones bietet und auch nicht leistungssparender als ähnliche Software ist. Aufgrund seiner Plattformunabhängigkeit bietet es jedoch eine solide Grundlage für weiterführende Projekte.

4. Zusammenfassung und Ausblick

VoIP ist eine vielseitige Technologie, welche auch im zunehmenden Maße im Telefonie-Bereich eingesetzt wird. Viele Anbieter setzen hierbei auf eine Kombination des SIP und RTP, welche gemeinsam vielseitige Funktionen bereitstellen, wie z.B. das Simulieren von Anrufbeantwortern, Gruppenkonferenzen, Halten von Telefonaten oder auch gleichzeitige Telefonate.

Im Rahmen dieser Arbeit sind diese Protokolle anhand eines VoIP-Systems analysiert. Das Opensource-Programm „Peers“, welches in Java entwickelt wurde, agierte hierbei als Softphone und stellte gemeinsam mit dem SIP-Provider sip2sip.info das Analysesystem dar. SIP ist ursprünglich mit dem Standard RFC3261 definiert und inzwischen vielfach erweitert. „Peers“ Implementierung unterstützt hierbei lediglich die grundlegenden Funktionen wie in Kapitel 3.3 nachzulesen ist. Die RTP-Schicht ist ähnlich simpel gehalten. In Kapitel 3.4 wurde festgestellt, dass außer dem Payload-Type keine weiteren Informationen des RTP verarbeitet werden. Auch das RTCP findet bei „Peers“ keine Verwendung, wodurch z.B. die Qualitätsüberwachung der Verbindung nicht gegeben ist. „Peers“ nutzt das in Kapitel 3.5 vorgestellte Komprimierungsverfahren G711 auf Grundlage einer A-Law bzw. µ-Law Kodierung. Damit kann bei VoIP-Kommunikation sichergestellt werden, dass selbst bei geringen Bandbreiten eine Verbindung möglich ist.

Letztendlich zeigt die Analyse, dass „Peers“ zwar in der Lage ist einfache Audio-Kommunikation zu betreiben und dies theoretisch auch mit VoIP-fähigen Telefonen die dieselben Protokolle und Kodierungsverfahren nutzen, jedoch nicht die vollen Möglichkeiten ausschöpft, welche mit den gegebenen Techniken möglich sind wie in Kapitel 3.6 gezeigt wurde.

Bei dieser Analyse wurde die eigentliche Transport-Schicht von „Peers“ nicht betrachtet und auch nicht die damit verbundene NAT-Gateway-Implementierung. Dies ist durchaus in weiterführenden Arbeiten denkbar. Des Weiteren gibt es viele verschiedene Softphones wie Kapitel 3.6 zeigt, welche auch in „Peers“ nicht enthaltene Funktionen bieten, wie z.B. das ebenfalls Java basierte Opensource-Program „Jitsi“. Diese bieten weitreichende Funktionen, wie Konferenzgespräche, Gespräche halten und Anrufbeantworter. Außerdem sind die benötigten Rechenleistungen all dieser Systeme durchweg gering (z.B. benötigt produziert keines der Programme mehr als 22 KB/s Netzwerklast oder 250 MB Hauptspeicherbelegung).

In dieser Arbeit wurde gezeigt, wie eine grundlegende Implementierung von VoIP-Kommunikation funktionieren kann und wie „Peers“ als Grundlage einer solchen dienen könnte. Dabei sind verschiedene Aspekte noch zu beachten, wie z.B. die Verbesserung der Leistung und Funktionalität.

5. Tabellen und Abbildungsverzeichnis

6. Glossar

- CSRC (Contributing Source) Identifier
  Header-Bestandteil des RTP mit dem die Quellen identifiziert werden, welche den übertragenen Medienstrom verändert haben.
- Location-Server
  Der Location-Server eines SIP-Providers weiß von jedem derzeit registriertem User-Agent-Client die tatsächliche IP-Adresse und über welchen Proxy-Server er zu erreichen ist. Daher dient er zur Vermittlung von Anfragen zwischen Proxy-Servern.
- Proxy-Server
  Ein Proxy-Server dient einem User-Agent-Client als Anfrage-Punkt für SIP-Requests und vermittelt diese an entsprechende andere Server weiter oder stellt einkommende Anfragen an einem ihm zugewiesenen User-Agent-Client.
- Redirect-Server
  Redirect-Server können als Endpunkt bei einem SIP-INVITE erreicht werden und vermitteln die eingehenden Anrufe an eine oder mehrere andere Stellen weiter.
- Registrar-Server
  Ein Reigstrar-Server dient zur Registrierung eines User-Agent-Client bei seinem SIP-Provider. Bei diesem Vorgang wird dem Server mitgeteilt, unter welcher IP-Adresse der Nutzer derzeit zu finden ist und dies wird in den Location-Server übertragen.
- RTP (Real-time Transport Protocol)
  RTP ist ein Protokoll, welches dazu dient Echtzeitmedien zu übertragen und eventuell auch weitergehende Informationen zwischen Teilnehmern einer Medienübertragung auszutauschen. Es wird in der Regel als Inhalt eines UDP verwendet.
- SIP (Session Initiation Protocol)
  SIP ist ein Protokoll, welches häufig genutzt wird um eine VoIP-Verbindung zu initiieren und die physikalische Adresse eines jeden Teilnehmers den anderen mitzuteilen. Es kann sowohl in einem TCP oder UDP eingebettet sein.
- SIP-Provider
  Unter einem SIP-Provider versteht man einen Dienstleister, welcher in der Lage ist bei Nutzer und ihre SIP-Anfragen an andere Nutzer zu vermitteln. Andere Nutzer können unter Umständen auch bei anderen SIP-Providern registriert sein und somit muss auch die Verbindung zwischen diesen gewährleistet werden.
- SIP-Request
  Ein SIP-Request beschreibt eine Art von Paketen beim SIP, welche Anfragen eines User-Agent-Client an einen User-Agent-Server darstellen. Diese können z.B. Anfragen für einen Anruf darstellen.
- SIP-Response
  Ein SIP-Response beschreibt eine Art von Paketen beim SIP, welche als Antworten auf bestimmte Requests eines User-Agent-Client von einem User-Agent-Server zurückgeschickt werden. Sie geben nähere Auskünfte über den Ausgang es gestellten Anfrage, wie z.B. dass ein Anruf-Anfrage angenommen wurde.

- Softphone
  Also Softphones bezeichnet man jegliche Computer Software, welche als virtuelles Telefon dient und in der Lage ist eine VoIP Verbindung mit anderen Endgeräten aufzunehmen. Ein Beispiel dafür ist die Opensource-Software „Peers“.
- SSRC (Synchronization Source) Identifier
  Header-Bestandteil des RTP mit dem die ursprüngliche Quelle des übertragenen Medienstroms identifiziert wird. Dadurch können verschiedene Teilnehmer unterscheiden werden.
- TCP (Transmission Control Protocol)
  TCP ist ein verbindungsorientiertes Netzwerkprotokoll, zum Übertragen von Daten. Im Gegensatz zum UDP wird auf die korrekte Reihenfolge der Pakete geachtet und fehlende nachgefordert.
- UDP (User Datagram Protocol)
  UDP ist ein Netzwerkprotokoll, welches als Transport für Daten verwendet werden kann und nur eine geringe Eigenlast aufweist. Im Gegensatz zu TCP wird nicht auf die Reihenfolge oder korrekte Zustellung der Pakete geachtet.
- User-Agent-Client
  Als User-Agent-Client bezeichnet man beim SIP einen Nutzer, welcher Anfragen an einen SIP-Provider stellt.
- User-Agent-Server
  Als User-Agent-Server wird jegliche Instanz bezeichnet, die SIP-Anfragen bearbeitet. Beispiele hierfür sind der Registrar-Server oder der Location-Server eines SIP-Providers.
- VoIP (Voice over Internet Protocol)
  Unter VoIP versteht man die Audio-Kommunikation über ein IP-basiertes Netzwerk. Dies kann z.B. über das SIP und RTP Protokoll realisiert werden.

7. Quellenverzeichnis

1 Urs Mansmann, http://www.heise.de/ct/artikel/Gleichstand-290272.html, 2006

2 Jens Ihlenfeld, http://www.golem.de/0508/39595.html, 2005

3 Lutz Düvel, http://www.voip-sip.de/faq/voip-faq-artikel-36-rubrik-2.htm, 2005

4 Torben Leuschner, http://www.torbenleuschner.de/blog/208/skype-links-praktisch-dank-eigenem-protokoll/, 2011

5 Anatol Badach, Voice over IP Die Technik – Grundladgen, Protokolle, Anwendungen, Migration, Sicherheit, Carl Hanser Verlag, München/Wien, 2007

6 J. Rosenberg, H. Schulzrinne, G. Camarillo und andere, http://tools.ietf.org/html/rfc3261, 2002

7 H. Schulzrinne, S. Casner, R. Frederick, V. Jacobson, http://tools.ietf.org/html/rfc3550, 2003

8 Yohann Martinau, http://peers.sourceforge.net/, 2013

9 AG-Projects, https://mdns.sipthor.net/register_sip_account.phtml, 2013

10 Wireshark Foundation, http://www.wireshark.org/, 2013

11 Yohann Martinau, http://sourceforge.net/projects/peers/files/peers/0.4.3/peers-0.4.3-src.zip/download, 2013

12 Eclipse Foundation, http://www.eclipse.org/, 2013

13 A. B. Roachm, http://tools.ietf.org/html/rfc3265, 2002

14 Dialog Consult / VATM, 14. TK-Marktanalyse Deutschland 2012, 2012