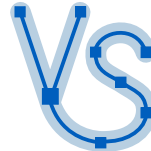


Universität
Stuttgart



Institut für Parallele
und Verteilte Systeme



Abteilung
Verteilte Systeme

Seminar:

**Moderne Kommunikationsmechanismen
in Rechnernetzen**

Thema:

**Internet-Telephonie und Instant Messaging:
Ein Überblick über die Geschichte und Technologien**

Autor: Alexander Wobser
8. Fachsemester

Betreuer Dominique Dudkowski

erstellt am: 20. Juli 2007

letzte Änderung am: 23. September 2007

Zusammenfassung—Diese Ausarbeitung beschäftigt sich mit den Themen Instant Messaging und Internettelephonie. Hierbei wird im Detail zunächst auf die geschichtliche Entwicklung sowie anschließend die hieraus praktisch entstandenen Technologien des Instant Messaging eingegangen. Es werden die relevantesten Protokolle und deren Funktionsweise umrissen und spezielle heutzutage allgegenwärtige Funktionen kurz in allgemeinverständlicher Sprache erläutert. Anschließend wird das Thema Internettelephonie aufgegriffen. Es werden hier die fundamentalen geschichtlichen Entwicklungen, Protokolle und bekanntesten Clients vorgestellt. Nach einem kurzen Ausblick auf moderne Entwicklungen in den beiden Gebieten wird diese Arbeit mit einem zusammenfassenden Fazit abgerundet.

Stichworte—Instant Messaging, Internet-Telephonie, XMPP, ICQ, IRC, TelNet, Sichtbarkeiten, Multi-User-Konferenz, SIP, STUN, Skype, Google Talk, TeamSpeak

I. EINLEITENDE WORTE

Seit jeher ist Kommunikation ein zentrales Mittel in der zwischenmenschlichen Interaktion. Kommunikation beschreibt den Austausch von Nachrichten zwischen verschiedenen Parteien, den Kommunikationspartnern. Mit Hilfe von Kommunikation ist es uns möglich unsere eigenen Gedanken zu reflektieren zu äußern oder in sozialen Strukturen einzusetzen um Einfluss auf andere zu nehmen.

War es vor Jahrtausenden noch das einzelne gesprochene Wort, das einen großen Fortschritt gegenüber der Lautsprache wie sie auch Primaten benutzen, darstellte, so entwickelte sich hieraus jedoch rasch eine Sprache die tieferen Gesetzmäßigkeiten bzw. einer Grammatik gehorchte. Je detaillierter die Sprache ausfiel, desto präziser konnte kommuniziert werden und desto leichter fiel es soziale Kontakte zu pflegen. Für die damalige Zeit waren diese bestehenden Kommunikationsstrukturen vollkommen ausreichend. Dies änderte sich jedoch im Laufe der Geschichte. Aus den einfachen sozialen Strukturen der Anfänge entwickelten sich komplexe Systeme, die hierarchisch aufgebaut waren und immer großflächigere Gebiete abdecken mussten. Das einfache Kommunikationsparadigma der Sprache war überholt.

Mit der Erfindung der Schrift zur Konservierung von Sprache, gelang der Menschheit der nächste große Schritt der Kommunikation. Ob im Einsatz in der Diplomatie, bei Feldherren, die Schriftstücke mittels Boten transferierten, oder einfach in der Wissenschaft bzw. in familiären Strukturen. Die Schrift bzw. die Post ermöglichte nun den Austausch von Informationen über große Distanzen hinweg. Wiederum blieben diese Kommunikationsmethoden viele Jahre aktuell, und genügten den Bedürfnissen der Zeit.

Die industrielle Revolution brachte hier eine Neuerung. Der fortschreitende Wandel der Gesellschaft bedingte es, dass kurze Entscheidungswege und geschwindige Umsetzung von Ideen einen wirtschaftlichen Vorteil mit sich brachten. Solche schnellen kurzfristigen Entscheidungen konnten über das Schicksal ganzer Unternehmungen bestimmen. Ein Schlüssel zu solchen Entscheidungen stellte hierbei die flüssige Kommunikation über große Distanzen hinweg dar.

Diesem Wunsch der Zeit entsprang als nächster entscheidender Schritt die Erfindung der Telegraphie. Durch sie wurde erstmals direkte Kommunikation über weite Strecken umgesetzt. Die Menschheit war im Telekommunikationszeitalter angekommen.

Weitere Erfindungen brachten immer praktischere Anwendungen mit sich, und trugen der sich rasant entwickelnden Industrie Rechnung. Telefon und Telefax wurden erfunden. Und schließlich stellte Konrad Zuse mit seinem Z3, den ersten frei programmierbaren Computer der Welt vor. Erneut brach ein neues Zeitalter an. Das Zeitalter der multimedialen Inhalte auf dem Weg zur Informationsgesellschaft, die wir heute längst sind.

Als Personalcomputer allgemein verfügbar wurden, entwickelte sich in den Menschen das Bedürfnis, dass neu gewonnene Medium auch zur Kommunikation zu nutzen. Die Umsetzungen zur Befriedigung dieser Bedürfnisse gipfelten schließlich in die heute bekannten Instant Messaging und Internet-Telephonie-Technologien, welche dieses Dokument zum Inhalt hat.

II. BEGRIFFSFINDUNG

Um einer Sache auf den Grund gehen zu können, und deren Inhalte beschreiben zu können, bedarf es zunächst einer Erläuterung worum es sich bei Instant Messaging bzw. Internet-Telephonie überhaupt handelt.

„Instant Messaging“ ist ein Anglizismus, der Einzug in die deutsche Sprache gehalten hat. Wörtlich korrekt übersetzt würde es soviel wie „sofortige Nachrichtenübermittlung,“ heißen.

Ein Instant Messaging Dienst, ist ein Dienst der es ermöglicht in Echtzeit mit anderen Teilnehmern zu kommunizieren. Dieser Vorgang wird im Allgemeinen auch als „chatten,“ bezeichnet. Beim Chatten werden kurze Textmitteilungen an einen Empfänger geschickt, auf die dieser unmittelbar antworten kann.

„Internet-Telephonie“ beschreibt das Telefonieren über Computernetzwerke die nach Internet-Standards aufgebaut sind. Da bei einem solchen Standard in der Regel die IP eines Rechners, als dessen Adressierung im Telephonnetzwerk dient, die IP also quasi eine Art „Telephonnummer“ repräsentiert, wird Internet-Telephonie auch „IP-Telephonie“ oder „Voice over IP (VoIP)“ genannt.

Bei VoIP werden für Telephonie typische Informationen, d.h. Sprache und Steuerinformationen für z.B. den Verbindungsaufbau ausgetauscht. Zum Einsatz kommen sowohl Computer, spezialisierte Telephonendgeräte, als auch über spezielle Adapter angeschlossene klassische Telephone die die Verbindung ins Telephonnetz herstellen.

III. INSTANT MESSAGING

A. Die Geschichte des Instant Messaging

Die Idee des Instant Messaging ist wie bereits in der Einleitung zum Ausdruck gebracht, keine Neue. Zum Anfang

des Telekommunikationszeitalters durch die Telegraphie entstanden und geschürt, reifte der Wunsch der Portierung des Prinzips des Echtzeitnachrichtenaustausches auf die neu aufkommende Technologie der Computer.

Ende der Sechziger Jahre entstand das ARPANET. Das ARPANET, der Vorgänger des heutigen Internets, stellte ein Pilotprojekt dar, in dem erstmals versucht wurde ein dezentrales Netzwerk zu realisieren. Ein solches Netzwerk stellt eine essentielle Grundlage für den Einsatz von Instant Messaging Technologien dar.

In diesem sowie ähnlich verknüpften Netzwerken wurde Anfang der 70'er Jahre mit dem TelNet-Protokoll ein Mechanismus geschaffen, der es Benutzern erlaubte Fernzugriffe auf entfernten Rechnern durchzuführen, als wären sie vor Ort lokal an einem Terminal eingeloggt. Der Benutzer bekam hierbei den Bildschirminhalt übertragen, den er auch sähe, wenn er sich lokal angemeldet hätte. Ein solcher Fernzugriff bot die Möglichkeit zur Ausführung sämtlicher Tätigkeiten, die auch lokal möglich gewesen wären, und somit bestand auch die Möglichkeit die Unixbefehle „talk“ und „w“ bzw. „who“ zu nutzen. Dies war die eigentliche Geburt des Instant Messaging. Ein Benutzer hatte somit die Möglichkeit sich mit „w“ bzw. „who“ die am Rechner angemeldeten Personen anzeigen zu lassen. Und mit „talk“ die Möglichkeit eine Chat-session mit einer dieser Personen zu initiieren.

Es gab im Verlaufe der 70'er Jahre einige weitere Bestrebungen Instant Messaging mittels Rechnern zu etablieren. Viele dieser Bestrebungen beeinflussten die Richtung der Entwicklung von Messengern. Neue Entwicklungen gebrauchten hierbei kontinuierlich die Erkenntnisse ihrer Vorgänger und machten sich diese bei der Übertragung auf andere Netzwerke oder Protokolle bzw. durch neue erweiterte Funktionalität zu Nutze. Bestrebungen dieser Art stellten Projekte wie „EMISARI“ (Emergency Management Information System And Reference Index), „PLANET“ (Planning Network), „PLATO - TERM-talk“, und „EIES“ (Electronic Information Exchange System) dar. Details zu diesen Systemen lassen sich bei Interesse im Anhang der Präsentationsfolien des zum Thema gehörigen Vortrages nachlesen. Zusammenfassend lässt sich sagen, dass am Ende dieser Entwicklung Systeme vorlagen, die nun Elemente wie Short-Cuts, Darstellungsmodi, Menüs, Dateitransfers oder Portierungen bzw. Protokollanpassungen umgesetzt hatten. [1] [2] [3] [4] [5]

Nach diesen vielen „Small Steps“ in den Siebzigern erfolgte der nächste „Big Step“ in der Entwicklung derartiger Anwendungen im Jahre 1985. Der Entwickler Jeff Kell programmierte eine Anwendung, die in das BitNet, ein Netzwerk zwischen Universitätsgroßrechnern, fest integriert war. Diese trug den Namen BitNet Relay Chat. Dank der Basisintegration in die Netzwerktechnologie des BitNet, konnte in dieser Anwendung ohne zusätzlichen Client gechattet werden. Außerdem neuartig war das Chatprinzip. Es erlaubte eine Nachrichtenverbreitung an mehrere User mittels

Multicast. Hierdurch wurde es möglich in „Chaträumen“ gleichzeitig mit mehreren Personen zu chatten. Aufgegriffen und auf das ARPANET übertragen hat diese Idee schließlich der Finne Jarkko Oikarinen. Seine Realisierung besaß einen zusätzlichen Client und konnte auf durchschnittlichen Personalcomputern die sich im Netzwerk befanden eingesetzt werden. Die Anwendung trägt den Namen „IRC (Internet Relay Chat)“. Diese Chattechnologie und die zugehörigen Netzwerke erfreuen sich auch heute noch reger Beliebtheit, trotzdem sich die Funktionen seit der damaligen Zeit nicht wirklich geändert haben. Heutzutage existieren dutzende verschiedener Clients, die auf der IRC-Netzwerktechnologie beruhen. [6] [7] [8] [9] [10]

1996 waren es 4 israelische Studenten, welche die Firma Mirabilis gründeten. Diese Firma bzw. das Produkt für die sie steht, revolutionierte das Instant Messaging und schuf erstmalig einen Instant Messaging Dienst, so wie wir ihn heute kennen. Die vier Studenten nannten das Produkt ICQ. Dieses Buchstabenkürzel steht für „I seek you“, was soviel bedeutet wie „Ich suche dich“. Das wirklich Neue an ICQ war die Buddyliste, die es ermöglichte Chat-Kontakte dauerhaft zu verwalten. Diese Funktion und die Kombination der vielen kleinen revolutionären Eigenschaften früherer Chat-Tools, die sich auch in ICQ wiederfanden, führten zu einer weiten Verbreitung der Anwendung. Der ICQ-Client zum damaligen Zeitpunkt war freie Software, und konnte kostenlos heruntergeladen werden. Die flächendeckende Einführung von ISDN, in der zweiten Hälfte der Neunziger, und die somit zunehmende Verbreitung des Internet verschaffte ICQ gegenüber den in kurzen Abständen danach auf dem Markt erscheinenden Konkurrenzprodukten, wie dem AIM („AOL Instant Messenger“) bzw. Yahoo-Messenger oder MSN-Messenger einen unvergleichbaren Marktvorteil, da das ICQ-Netzwerk somit natürlich die breiteste Nutzerfront besaß. Wechsler zwischen den Messaging Systemen sind und waren eher unüblich, und somit musste wer später kam, zwangsläufig den ICQ-Client benutzen, um mit Bekannten chatten zu können. Ein ähnlicher Vorteil lässt sich heute in der IP-Telephonie für den Client „Skype“ der im Jahre 2004 erschien, ausmachen.

Zum gegenwärtigen Zeitpunkt lässt sich ein deutlicher Trend im Instant Messaging erkennen, der zu Clients geht, die es erlauben, in verschiedenen Netzwerken gleichzeitig aktiv zu sein. Solche Clients bezeichnet man als Multi-Protokoll-Clients. Diese Programme überwinden bestehende Protokollgrenzen der verschiedenen Systeme, und ermöglichen so, ohne viele verschiedene Programme installieren zu müssen, die Kommunikation mit Personen in den verschiedenen Netzen. Somit sind auch die zuvor disjunkten Nutzergruppen, über ein und denselben Client erreichbar. Man benötigt jedoch für jedes Netzwerk einen eigenen Account.

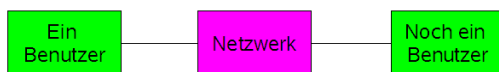
Die Problematik der disjunkten Netze erkannte auch AOL frühzeitig, weshalb sie den Marktführer ICQ 1998 kur-

zerhand aufkauften und die vier israelischen Studenten zu Multimillionären machten. In Folge vereinheitlichte AOL bis heute die beiden einst heterogenen Netzwerke strukturell und verschaffte sich somit den Vorteil der Kompatibilität. AIM und ICQ Nutzer profitieren nun also davon, indem sie untereinander chatten können. Die Problematik der nicht öffentlichen Messenger-Dienste, mit ihren proprietären Protokollen wurde seit 1999 versucht durch die Einführung eines Standards zu bekämpfen. Das Ergebnis erschien im Jahre 2004 und nennt sich „XMPP“ („Extensible Messaging & Presence Protocol“). XMPP, auch als Jabber-Protokoll bezeichnet, ist ein XML-Dialekt und wurde standardisiert in den Standards RFC 3920 - 3923 durch die IETF. Einige der neusten Instant Messenger wie zum Beispiel **Google Talk** basieren auf dem Jabber-Protokoll.

B. Theoretische Grundlagen

Um zu verstehen wie Instant Messaging realisiert wird, muss man sich zunächst bewusst machen, vor welchen technischen Voraussetzungen man steht. **Abbildung 1** zeigt dies recht eingänglich. Im Allgemeinen ist es so, dass ein Nutzer (in der Abbildung links) mit einem anderen Nutzer (in der Abbildung rechts) in Kontakt treten will. Abhängig von den benutzten Diensteanbietern und Clients findet dann eine Kommunikation über ein Netzwerk statt. Dieses Netzwerk wird in der Regel das Internet sein. Auf dessen Technologie wird dann beispielsweise das ICQ Netz oder das MSN (Microsoft Network) etabliert.

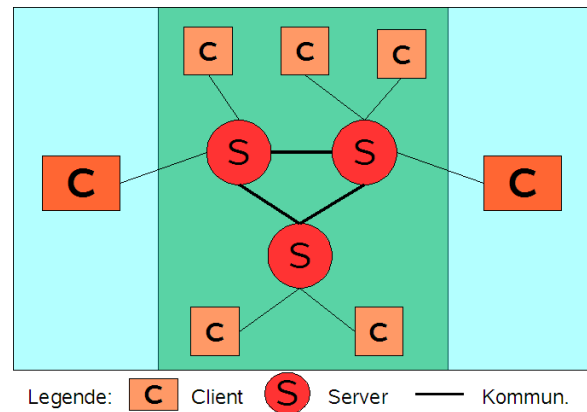
Abbildung 1. Schematische Darstellung der Voraussetzung der Kommunikation beim Instant Messaging



Die Komponenten, welche ein solches Netzwerk enthält, sind hierbei stets die gleichen. In der **Abbildung 2** sind diese zu betrachten. Es handelt sich bei den Komponenten um Clients und um Server. Clients sind hierbei Programme mit denen Benutzer an das Netzwerk angeschlossen sind. Sie befinden sich also stets in der „Peripherie“ des Netzwerkes. Sie nutzen von ihrer Position aus, die Dienste und Funktionalitäten, die von den Servern des Netzes angeboten werden. Die Server selbst stellen also den „Kern“ eines Netzwerkes dar. Um alle Dienste realisieren zu können, ist es im Normalfall auch nötig, dass die Server selbst ebenfalls untereinander kommunizieren, beispielsweise um die Liste der im Moment im Netzwerk verfügbaren Clients aktuell zu halten. Denn wie die Abbildung gut zeigt, müssen Clients nicht zwangsläufig direkt mit ein und dem selben Server verbunden sein. Hierbei reicht die Verbindung zu einem beliebigen Server. Das Gleiche gilt im Übrigen

auch für die Server. Die Abbildung suggeriert hier zwar Gegenteiliges, jedoch muss ein Server nicht in allen Instant Messaging Netzwerken alle weiteren Server kennen. Dies ist implementierungsabhängig, und zum Beispiel einer der Gründe, weshalb verschiedene Netzwerke nicht kompatibel sind. Viel schwerer wiegt hier jedoch der Grund, dass in den Netzen unterschiedliche „Sprachen“ gesprochen werden, also unterschiedliche **Protokolle** zum Einsatz kommen. [11]

Abbildung 2. Die Komponenten im Aufbau eines Netzwerkes



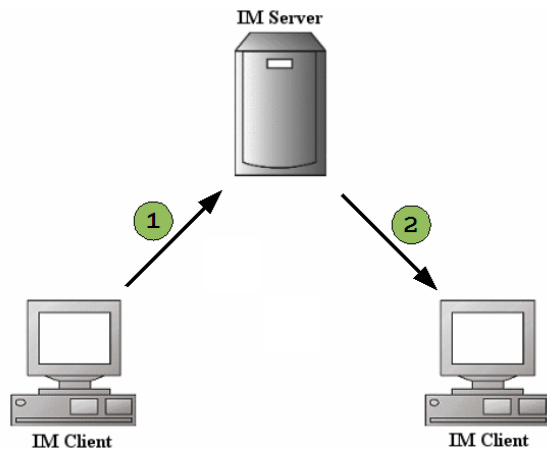
C. Kommunikationsschemata

Die in Abschnitt **III-B: Theoretische Grundlagen** beschriebenen Komponenten müssen zur Realisierung der Kommunikation, nach einem gemeinsamen Schema vorgehen. Dieses Schema wird neben anderem durch das verwendete Protokoll vorgegeben. Im Instant Messaging haben sich 2 verschiedene Kommunikationsschemata durchgesetzt. Die Schemata sind „**Client-Server-Client**“ (auch „Client-to-Server-Prinzip“) und das Schema „**Peer to Peer**“ (auch „Client-to-Client-Prinzip“).

1) **Client-Server-Client**: Dieses Schema geht von einem zentralen Knotenpunkt aus, bei dem jeglicher Datenverkehr über mind. einen Server geschickt wird. Die Clients nehmen bei diesem Schema niemals direkt Kontakt miteinander auf. In **Abbildung 3** wird dieses Schema graphisch illustriert. Pfeil 1 zeigt hierbei das Daten mit Zielbestimmung an einen Server übermittelt werden. Netzwerkabhängig kann es jetzt vorkommen, dass die Daten noch über weitere Intermediates (Zwischenstationen) geschickt werden, bis schließlich die gewünschten Daten von einem Server an den Empfänger weitergeleitet werden (Pfeil 2).

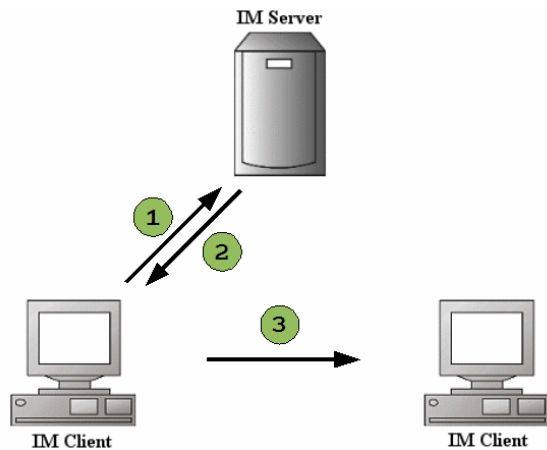
2) **Peer to Peer**: Sind beim Client-Server-Client-Schema noch die Server die alles regelnden Knotenpunkte, so wird bei der Peer-to-Peer Realisierung auf direkte Verbindungen der Clients gesetzt. In **Abbildung 4** ist der Ablauf der Kommunikation einmal dargestellt. Hier sieht man das die Kommunikation durch die Anfrage an einen Server eingeleitet wird (Pfeil 1). Ein Client erbittet sich die Zieladresse bzw. die Verbindungsdaten des gewünschten Clients des Kommunikationspartners. Der Server antwortet und übermittelt die

Abbildung 3. Schematische Darstellung des Client-to-Server-Prinzips.
 1: Senden einer Nachricht inkl. Empfängerbezeichner an Server
 2: Server leitet Nachricht an entsprechenden Empfänger



entsprechenden Daten (Pfeil 2) an den Client zurück. Der Client ist nun in der Lage direkt Kontakt mit dem Zielclient aufzunehmen.

Abbildung 4. Schematische Darstellung des Client-to-Client-Prinzips.
 1: Client stellt Anfrage nach Daten eines anderen Clients
 2: Server liefert gewünschte Daten zurück
 3: Client leitet direkte Kommunikation mit anderem Client ein.



3) *Vergleich der Schemata:* In der Praxis hat sich gezeigt, dass sowohl das eine als auch das andere Schema jeweils Vor- und Nachteile bietet. Ein klarer Vorteil vom Peer-to-Peer-Aufbau ist eine Verminderung von Datenverkehr. Dies entlastet somit den „Kern“ des Netzes (die Server). Hierdurch skalieren Peer-to-Peer-Netze deutlich besser als solche die nach dem Client-to-Server-Prinzip arbeiten.

Auch die Sicherheit ist ein Aspekt. Bei der Peer-to-Peer-Technik erlangt der Server von den Inhalten der Kommunikation der Gesprächspartner keine Kenntnis. Das heißt das Ausspähen der Daten durch derartige Server wird unterbunden. Dies gilt ebenso für Dateitransfer. Andere Funktionen wie zum Beispiel „Offline Nachrichten Versand“ lassen sich

im Client-Server-Client-Prinzip leichter umsetzen. Für die Unternehmen die hinter den Messengern stecken, bietet das Client-Server-Client Prinzip einige Vorteile. Das verwendete Protokoll wird zum Beispiel besser verschleiert, und somit vor Nachahmung geschützt. Dies liegt unter anderem daran dass nur wenige Kommunikationselemente offenbart werden, und Steuerinformation so schwerer zu entschlüsseln sind. Auch wurden die Benutzerrichtlinien von einigen Diensteanbietern (Bsp: von AOL für AIM und ICQ) so weithin geändert, dass diese sich vorbehalten die gesendeten Daten zu analysieren, was natürlich in der Theorie individualisierte Werbung und somit Mehreinnahmen verspricht. In wie weit diese Mechanismen in der Praxis tatsächlich Anwendung finden, ist nicht klar. In einigen Ländern machte sich ein Anbieter mit einem solchen Vorgehen aus datenschutzrechtlicher Sicht strafbar (z.B. Deutschland §88 TKG - Fernmeldegeheimnis), was an einer Umsetzung dieser Theorie zweifeln lässt.

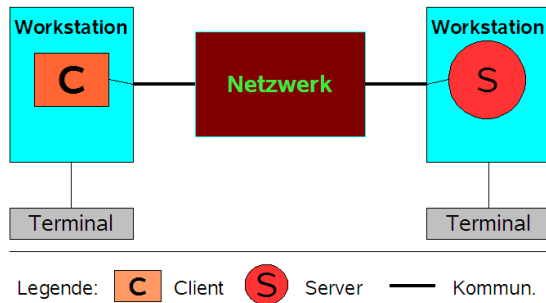
D. Protokolle

Protokolle sind Regeln, welche das Format, den Inhalt, die Bedeutung und die Reihenfolge gesendeter Nachrichten zwischen verschiedenen Instanzen der gleichen Schicht des ISO/OSI-Schichtenmodells festlegen. Mit anderen Worten, es handelt sich mit einem Protokoll um eine „Sprache“, die alle an der Kommunikation beteiligten Instanzen verstehen, verarbeiten und sprechen können müssen. Unter Instanz versteht man in Bezug auf Instant Messenger, Clients bzw. Server des jeweiligen Netzes. Im Folgenden werden hier nun in der Reihenfolge des geschichtlichen Auftretens verschiedene Methodiken des Instant Messagings anhand der jeweils populären Protokolle angeführt. [11]

1) *TelNet (Telecommunication Network):* Das TelNet Protokoll ist ein Protokoll zur Durchführung von Fernzugriffen auf entfernte Rechner. **Abbildung 5** zeigt den Aufbau der ein Chatten ermöglichte. Voraussetzung für TelNet ist, dass 2 Rechner (Workstations) über ein Netzwerk miteinander verbunden sind. Das TelNet Protokoll setzt auf diesem bestehenden Datenkanal auf und ist nun in der Lage, die Informationen von der Workstation auf der der TelNet-Server läuft, sprich Bildschirminformationen über das Netzwerk an den TelNet-Client zu übermitteln. Mit Hilfe des Clients können wiederum Befehle an den Server übermittelt werden, die auf der Server-Workstation auszuführen sind. Auf diese Art und Weise, bekommt der Benutzer am Terminal der Client-Workstation die Möglichkeit auf der Server-Workstation zu arbeiten, als ob er sich lokal eingeloggt hätte. Arbeitet an der Workstation noch ein anderer Nutzer, beispielsweise über ein lokales Terminal, so kann dieser mit dem Befehl „who“ ausfindig gemacht werden, und mit „talk“ ein Chat initiiert werden.[12] [13]

2) *IRC (Internet Relay Chat):* Das IRC ist ein Netzwerk in dem mehrere Benutzer gleichzeitig miteinander kommunizieren können. Die Kommunikation geht hier mit Hilfe

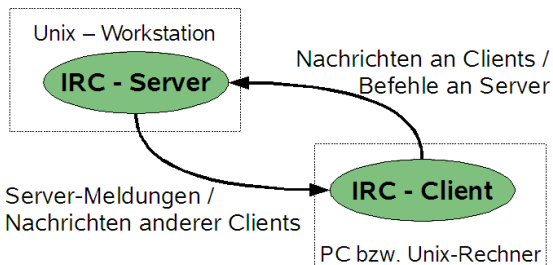
Abbildung 5. Schematische Darstellung der Funktionsweise des TelNet-Protokolls



von Multicasting von statten. Zum Chatten im IRC ist ein Client notwendig.

Abbildung 6 zeigt schematisch wie hierbei welche Art von Kommunikation möglich ist. Im Gesamten lässt sich erkennen, dass dem Client im Wesentlichen 2 Möglichkeiten, um mit dem Server in Verbindung zu treten, zur Verfügung stehen. Zum Einen kann er System-Befehle an den Server schicken, welche dieser verarbeiten muss/kann, zum Anderen bleibt die Möglichkeit Nachrichten an andere Clients zu schicken. Das IRC ist somit eine Realisierung des **Client-to-Server-Prinzips**. Organisiert wird hierbei der Nachrichtenversand wie schon erwähnt mittels Multicast. User eines Client können im IRC so genannten Chat-Räumen beitreten, und beim Versenden einer Nachricht, wird vom Server die Nachricht des Senders nun bspw. per Flooding an die Nutzer/Empfänger im Chat weitergeleitet. Neben den Nachrichten anderer Clients ist ein IRC-Server auch in der Lage, entsprechende Antwortmeldungen auf Befehle bzw. Statusmeldungen zurückzugeben.

Abbildung 6. Schematische Darstellung der Kommunikation der IRC-Komponenten



Das IRC-Protokoll ist ein textbasiertes Protokoll, dass in seiner Notation an die BNF erinnert. Es erlaubt heutzutage nach der Einführung zusätzlicher Kommandos, die es zu Beginn des IRC nicht gab, auch Dateitransfer. Die Nachrichten die selbst in Textform vorliegen, werden geparkt und im Server entsprechend der steuernden Protokoll-Inhalte weiter verarbeitet. Einen genaueren Blick auf die Syntax dieses Protokolls liefert Folie 44 im Anhang der Folien des zu

diesem Thema gehörigen Vortrages. [14] [10]

3) **ICQ/AIM - OSCAR**: OSCAR (Open System for Communication in Realtime) ist das proprietäre Protokoll von AOL das in den Instant Messengern AIM und den aktuellen Clients des ICQ-Netzwerkes verwandt wird. Da AOL das Protokoll an für sich unter Beschlag hält, und keine Daten über die Funktionsweise von OSCAR preisgibt, ist man wenn man etwas über dieses Protokoll erfahren will, auf die einschlägigen Seiten der entsprechenden Szene angewiesen. Durch Monitoren der ein- und ausgehenden Daten ließ sich die Funktionsweise des Protokolles bereits weitestgehend entschlüsseln. Demnach ist OSCAR ein Protokoll das die Kommunikation über TCP abwickelt und von Hause aus Dateitransfer sowie das Einbinden von Bild- und Musik-Dateien in die Nachrichten erlaubt. Ein wichtiger Punkt der Funktionsweise von OSCAR ist die Erkenntnis, dass es sich bei OSCAR eigentlich um einen Protokoll-Stack handelt, welcher sich aus den beiden Teilen „SNAC“ und „FLAP“ zusammensetzt.

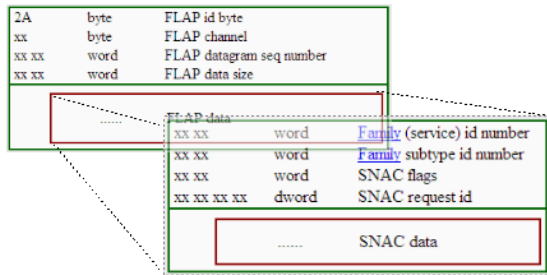
Abbildung 7 soll den Aufbau dieses Stacks veranschaulichen. Man sieht das SNAC auf FLAP aufsetzt. Wenn man die Daten des FLAP Protokolls betrachtet erkennt man, das hier unter Anderem ein Kanal („channel“) übergeben wird. Dieser Kanal gibt an, welche rudimentäre Funktion des Protokolls durch die Nachricht ausgeführt werden soll. Zu diesen Funktion gehört zum Beispiel der „Keep-Alive“ mit Kanal 4, „Initiate Session“ mit Kanal 1 oder „SNAC Daten folgen“ auf Kanal 2.

Beim Versenden einer standardmäßigen Nachricht, enthält das „FLAP-Datenfeld“, neben zusätzlichen SNAC-Steuerinformationen dann auch die Nachricht an für sich. Die Nachricht befindet sich hierbei wiederum im „SNAC-Datenfeld“. Die SNAC-Steuerinformation beinhalten u.A. den Zusatz der gewünschten Funktion, bei ICQ „Service Family“ genannt. „Chatnachrichtenübermittlung“ wäre hierbei z.B. „Service Family“ 0004. Eine umfangreiche Liste der Services befindet sich auf den Vortragsfolien 46 und 47. Das Beispiel mit dem Versenden einer Nachricht lässt erkennen, dass auch OSCAR auf dem **Client-to-Server-Prinzip** beruht. [15] [16]

Das Ursprüngliche ICQ-Protokoll hingegen war **Peer-to-Peer** basiert. Weitere Informationen zu diesem Protokoll lassen sich bei Interesse im Vortrag von Folie 45 entnehmen. [17]

4) **Jabber - XMPP**: XMPP (extensible Messaging & Presence Protocol) ist ein von der IETF in den RFC's 3920-3923 standardisiertes Protokoll. Konzipiert ist dieses Protokoll für XML-Routing, findet praktisch aber hauptsächlich Anwendung im Instand-Messaging. Das Jabber-Protokoll wie XMPP auch genannt wird, ist ein XML-Dialekt und dementsprechend erweiterbar. Das Protokoll geht von einer Client-to-Server-Architektur aus. Die Funktionsweise des Protokolles soll **Abbildung 8** veranschaulichen. Clients werden im Netzwerk über den Server identifiziert bei dem

Abbildung 7. Aufbau des OSCAR-Protokoll-Stack



sie angemeldet sind. Die Struktur der ID ist ähnlich einer E-Mail-Adresse aufgebaut (Abbildung links mitte). Will ein Client einen Dienst eines Jabber-Servers in Anspruch nehmen, macht er dies, indem er die XMPP-Daten an den Dienst des Servers adressiert. Identifiziert werden diese Dienste über eine URI bei der der gewünschte Dienst als „Subdomain“ angefügt wird (Abbildung links unten). Somit bietet das Jabber-Protokoll eine optimale Erweiterbarkeit was neue/eigene Dienste betrifft. Sobald diese im Server implementiert sind, können diese auf den Servern über entsprechende Zusätze in der URI von Clients in Anspruch genommen werden.

XMPP ist ein Protokoll zum Echtzeit-XML-Streaming, dem entsprechend läuft die Kommunikation selbst in sogenannten Streams ab. Ein Datenaustausch zwischen den Kommunikationspartnern ist nur während des Bestehens eines solchen Streams möglich. **Abbildung 8** zeigt das umschließende Strukturelement (links oben). Es heißt „<stream>“. Der Inhalt bestimmt dann die weitere Funktion, als Strukturelemente in der Grafik sind „<presence>“ und „<message>“ beispielhaft erwähnt. „<presence>“ diente im Gebrauch dem Ändern oder Abfragen des Online-Status, und „<message>“ dem Versand von Nachrichten.

In der Abbildung ist rechts eine Chat-Sitzung beispielhaft dargelegt. Ist der Stream einmal geöffnet, so hat der Stream Bestand, bis er wieder geschlossen wird. Es können allerlei Informationen ausgetauscht werden. Ist die Kommunikation beendet, wird der Stream geschlossen.

[18], [19], [20], [21]

5) *Weitere Protokolle:* Die Bereits aufgeführten Protokolle stellen selbstverständlich nur einen Ausschnitt der am Markt verfügbaren IM-Protokolle dar. Auf den Folien im Anhang des Vortrages finden sich noch stichpunktartige Erläuterungen sowie Links zu den Protokollen des MSN Messengers, des ursprünglichen ICQ sowie Informationen zu YMSG, dem Protokoll des Yahoo! Messengers. Der Markt an Messenger-Technologien kennt aber durchaus noch weitere Protokolle, die auch dort keine Erwähnung finden. Ein Beispiel hierfür wäre das Protokoll „Gadu-Gadu“ das hauptsächlich in Polen verbreitet ist.

Abbildung 8. Darstellung der Struktur einer XMPP-Sitzung (links oben), der Identifikation von Nutzern und Diensten eines Jabber-Servers (links mitte/unten), sowie ein Beispiel einer Chatsitzung (rechts)

'C:' ist Nachricht von Client zu Server
'S:' ist Nachricht von Server zu Client

XMPP Struktur	Beispiel einer Jabber-Sitzung
<pre><stream> <presence> <show/> </presence> <message to='foo'> <body/> </message> ... </stream></pre>	<pre>C: <?xml version='1.0'?> <stream:stream to='example.com' xmlns='jabber:client' version='1.0'> S: <?xml version='1.0'?> <stream:stream from='example.com' id='someid' xmlns='jabber:client' version='1.0'></pre>
Beispiel einer Jabber-ID	...encryption, binding...
JabberID@JabberServer.org	C: <message from='juliet@example.com' to='romeo@example.com'>
Beispiele der Identifikation von speziellen angebotenen Diensten eines Jabber-Servers	<body> Romeo, oh Romeo! </body> </message>
Allgemein:	S: <message from='romeo@example.com' to='juliet@example.com'>
Service.JabberServer.org	<body> Ja, Juliet? </body> </message>
Bsp. Multiuser-Konferenz: conference.example.com	C: </stream:stream>
Bsp. Definition eigener Dienst: aims.example.com	S: </stream:stream>

E. Funktionalitäten

Die Funktionen eines aktuellen Instant Messengers sind dank der starken Konkurrenz verschiedener Produkte im Umfang im Vergleich zu den frühen Jahren deutlich angewachsen. Standardmäßig besitzen die Messenger neben dem Nachrichten-Versand und -Empfang heutzutage auch die Funktionen des Verschlüsselten Nachrichten- und Dateiversands, Online-Spiele, verschiedene Sichtbarkeiten, Multi-User-Konferenzen, SMS-verschicken, eine History, eine Buddyliste und vieles vieles mehr. Im Folgenden werden exemplarisch die Funktionen der **Sichtbarkeiten** und die **Multi-User-Konferenz** erklärt. [22] [23] [24]

1) *Sichtbarkeiten:* Sichtbarkeiten stellen eine Gruppierung von Funktionen dar. Ein Benutzer eines Clients besitzt die Möglichkeit an seinem Client eine Einstellung so zu ändern, dass er anderen Nutzern im Netzwerk unter einem bestimmten Präsenz-Zustand angezeigt wird. Die verfügbaren Zustände unterscheiden sich von Messenger zu Messenger, allgemein üblich sind jedoch Zustände wie „Online“, „Nicht Verfügbar“, „Nicht Stören“ oder „Unsichtbar“. Mit den verschiedenen Präsenz-Zuständen können je nach Messenger auch verschiedene Funktionen verbunden sein. So kann es vorkommen das im „Nicht Stören“-Modus eingehende Nachrichten blockiert werden, oder im „Unsichtbar“-Modus die Kommunikation total unterbunden wird, der Nutzer also nur Kontaktlisteninformationen erhält. Bei den gängigen Instant Messengern wird die Verwaltung

der Präsenzen der Clients serverseitig vorgenommen. Die Clients verwalten dementsprechend nur ihren eigenen Status. Zur Verwaltung hält ein Server zusätzlich zu der Liste der aktiven Nutzer auch noch deren aktuellen Status vor. Will der Nutzer seinen Präsenz-Zustand ändern, so wählt er die entsprechende Funktion seines Client aus, und der Client sendet eine neue Präsenz-Information an den Server, dieser verarbeitet diese, und führt in seiner Liste den Status des Client entsprechend der Information nach. Ein anderer Client der in regelmäßigen Abständen entweder seine Kontaktliste aktualisieren lässt, oder über Statusänderungen aktiv vom Server informiert wird (je nach Protokoll), sieht nach der Aktualisierung dann den aktuellen Status der Präsenz seiner Kontakte.

2) *Multi-User-Konferenz*: Die Praktische Umsetzung dieser Funktionalität ist im Detail nicht nur Protokoll-abhängig, sondern auch davon abhängig welches Kommunikationsschema verwandt wird. Handelt es sich bei der zugrunde liegenden Architektur um das Client-Server-Client-Schema, so übernimmt der Server in diesem Moment einfach die Funktion eines Multicasters, ähnlich der Umsetzung die im IRC Anwendung findet. Ein Client übergibt hierbei eine Nachricht mitsamt einer Liste von Empfängern an den Server, und dieser Versendet die Nachricht dem entsprechend weiter. Alle aktuell weit verbreiteten Systeme verfolgen dieses Konzept. Handelt es sich bei dem zugrundeliegenden Schema um Peer-To-Peer, ist dieses Verfahren nicht möglich. Das Original-ICQ-Protokoll basierte auf diesem Schema, und bot von sich aus keine echt Multi-User-Konferenz an. Mit der Zeit wurden in Multi-Protokoll-Messengern Behelfslösungen implementiert, bei der der Nutzer aus der Kontaktliste eine Gruppe von Kontakten auswählte, der Client die Eingabefenster zuerst koppelte und später integrierte, sodaß der Client selbst die Verteilung der Nachrichten an die Empfänger vornahm. Bei entsprechend vielen Empfängern führte das für einige User somit zu zeitlichen Verzögerungen, im Erhalt der Nachricht einfach aufgrund der zur Verfügung stehenden Bandbreite. Ein weiterer Nachteil war, dass der Raum lokal etabliert war, und andere Nutzer bei sich den Chat in einem Chatfenster mit lediglich dem Versender angezeigt bekamen. Mir ist zu diesem Zeitpunkt keine Peer-To-Peer-Lösung im Instant Messaging bekannt, die eine Multi-User-Konferenz unterstützt, wie sie bei Client-Server-Client-Lösungen üblich sind.

F. Clients

Wie man der Beschreibung der Funktionalitäten schon entnehmen kann, ist die Art der angebotenen Funktionen in erster Linie von den Clients und der Struktur der zu Grunde liegenden Netzwerke abhängig. Somit stellen sich die Fragen: „Welche Arten von Clients gibt es?“ und „Welche Clients bieten welche Vorzüge?“.

Im Wesentlichen lassen sich 2 große Gruppen von Clients differenzieren. Das sind zum Einen die **kommer-**

ziellen/proprietären Produkte, und die **nicht kommerziellen/proprietären also offenen Produkte**. Beide Arten von Clients haben jeweils Vor- und Nachteile.

1) *proprietäre/kommerzielle Clients*: Proprietäre Produkte bieten in der Regel die Vorteile des vollständigen Funktionsumfangs und der Kompatibilität zwischen den verschiedenen Versionen des Protokolles. Das Letztere ist wichtig, sollten Änderungen am Protokoll vorgenommen werden. Hier bleibt das Programm in der Regel funktionsfähig oder es wird zumindest ein Update angeboten, dass die Funktionsfähigkeit umgehend wieder herstellt. Die kommerziellen Originalclients der jeweiligen Protokolle bieten aus User-Sicht auch einige Nachteile. Der massivste Nachteil dürfte wohl sein, dass jedes Protokoll seinen eigenen Client mit sich bringt. Hat man einen breit gefächerten Bekanntenkreis, oder muss man aus sonstigen Gründen ein zweites oder drittes Protokoll benutzen, so ist man hier auch gezwungen sich einen zweiten oder dritten Client zu installieren. Multi-Protokoll-Messenger versprechen hier Abhilfe. Trillian wäre hier als kommerzielles Beispiel zu nennen. Hierbei ist aber bei Protokollumstellungen in Kauf zu nehmen, dass das Produkt nicht mehr 100% funktionsfähig bleiben muss. Desweiteren sind kommerzielle proprietäre Produkte oftmals durch Werbung mitfinanziert, und dem entsprechend mit Werbung versehen.

2) *nicht proprietäre/kommerzielle Clients*: Die Szene der nicht kommerzielle Zwecke verfolgenden Clients ist vor allem durch Multi-Protokoll-Messenger geprägt, obwohl es auch Clients gibt die speziell nur ein Protokoll unterstützen (z.B. SPARK unterstützt nur Jabber). Diese freien Clients bieten die Vorteile, dass sie im Allgemeinen werbefrei sind und mehrere Protokolle gleichzeitig verstehen können. Man erspart sich somit ein Sammelsurium an verschiedenen Clients. Ein wesentlicher Nachteil ist, dass diese Messenger oft nicht den gesamten originalen Funktionsumfang besitzen. Dies liegt nicht zuletzt daran, dass so gut wie kein proprietäres Protokoll vollständig entschlüsselt ist. Ein großer Vorteil ist jedoch, dass diese Clients oft über Plugin-Schnittstellen verfügen, über die man die Funktionalität des Clients mit eigenen oder heruntergeladenen Modulen erweitern kann. Somit bieten viele der freien Clients auch Funktionalitäten die originale Clients noch vermissen lassen. Beispiele für derartige Clients sind zum Beispiel: Miranda, Pidgin (früher Gaim), CenterICQ. Eine detaillierte Liste zu Messenger-Clients findet sich auch in der freien Enzyklopedie Wikipedia ([25]). Diese Liste macht dabei auch Aussagen, für welche Plattformen die jeweiligen Messenger verfügbar sind. Ein großes Plus der freien Programme ist, dass es für eigentlich jedes Betriebssystem eine Anwendung für jedes Netz zu geben scheint.

IV. INTERNET-TELEPHONIE

A. Die Geschichte der Internet-Telephonie

Die Geschichte der Internet-Telephonie war im Gegensatz zur Geschichte der Instant Messenger nicht geprägt durch zahlreiche Produkte, sondern konzentrierte sich vielmehr auf die Entwicklung von verschiedenen Standards, die eine VoIP Umsetzung immer besser oder auf andere Art und Weise ermöglichten. Die Anfänge der digitalen Übertragung von Sprache stellt wohl die Einführung der PCM30-Technik [26] dar. Gegen 1960 wurden auf Basis dieser Technik automatisierte Vermittlungsanlagen und Selbstwahlsysteme entwickelt, die Übertragung der Daten zwischen den Telephonzentralen erfolgte dann mit PCM30 digital [27] [26]. Von den Telephonzentralen zum Benutzer hingegen lief alles weiterhin analog ab. 1973 gelang schließlich die erste Übertragung digitaler Sprache im ARPANET mittels NVP (Network Voice Protocol). Hierbei kamen verschiedene Codierungstechniken zum Einsatz [28]. Ziel des Projektes war ursprünglich nur zu demonstrieren, dass eine qualitativ hochwertige Übertragung von Sprache möglich ist. Von speziell militärischem Interesse war dies, da eine derartige Übertragung auch Verschlüsselungsfähig und somit abhörsicher war [29]. 1977 wurde das NVP schließlich durch die IETF standardisiert. Als dann 1980 das IP (Internet Protocol) als Standard verabschiedet worden war, und selbiges 1982 im ARPANET implementiert wurde, waren die Voraussetzungen für Internet-Telephonie wie wir es heute kennen geschaffen. Die Bezeichnung IP-Telephonie trägt dieser Abhängigkeit Rechnung. Parallel zum Internet Protocol fand eine Standardisierungsbewegung des CCITT (heute ITU) statt. Ziel dieser Bewegung war es die noch mechanischen Vermittlungsstellen der Telephonnetze zu ersetzen. Somit sollte eine bessere Auslastung der Leitungen und mehr Komfort für die Benutzer erreicht werden [30]. Das Ergebnis der Entwicklungen war schließlich der ISDN (Integrated Services Digital Network) Standard. 1992 erblickte mit den ersten GSM (Global System for Mobile Communications) Netzen eine weitere Technologie in Deutschland das Tageslicht. GSM war der erste Mobilfunkstandard der auf einem Austausch digitaler Signale beruhte. Die Datenübertragung im GSM-Standard erfolgt paketorientiert. Nach dem ab 1989 ISDN schrittweise aufgebaut wurde, stockte die Entwicklung der IP-Telephonie für einige Jahre, aufgrund der technischen Umgebungsbedingungen. Das Internet steckte noch in den Kinderschuhen, und bot zu geringe Datenübertragungsraten, beziehungsweise noch keine ausreichende Attraktivität, da die Anschlussstellen in der Anzahl noch recht begrenzt waren. Nachdem selbige Anzahl jedoch ab 1989 sprunghaft anstieg und ab 1995 ISDN und somit eine konstante Datenübertragungsrate von 64 kBit/s möglich und flächendeckend verfügbar wurde erhielt auch die IP-Telephonie einen Schub. Das Jahr 1995 gilt in Bezug auf IP-Telephonie oft als Geburtsstunde, da es dieses Jahr war, in dem die israelische

Firma VocalTec die erste Anwendung auf den Markt brachte, die IP-Telephonie nutzte. Die Software trug den Namen „Internet Phone“. [31] Diese Technologie arbeitete zum Einführungszeitpunkt noch im Halbduplexbetrieb. Das bedeutet, dass die kommunizierenden Gesprächspartner nicht beide gleichzeitig sprechen konnten, sondern eine Kommunikation wie sie auch aus dem CB-Funk bekannt ist stattfand. Man konnte hier also zunächst nur abwechselnd miteinander sprechen. Dieses System funktionierte mehr schlecht als recht, da es keine entsprechenden Standards gab, und alle diese Gesprächspartner diese Software brauchten. Ein weiteres Problem war, dass ISDN zwar eine konstante Datenrate zur Verfügung stellte, jedoch reichten die Kapazitäten bei Weitem nicht aus, um Echtzeit-Anwendungen wie sie bei VoIP zum Einsatz kommen performant umzusetzen. Die Lösung war die Sprachdaten so weit verlustbehaftet zu komprimieren damit überhaupt eine Übertragung möglich wurde. In Kombination mit der Tatsache, dass das IP keine „Quality of Services“ bietet und nach dem „Best Effort“ („Größte Bemühungen“) Prinzip arbeitet, gingen viele Datenpakete bei der Übermittlung verloren. Die Sprachqualität genügte somit keinen alltäglichen Anforderungen. Verbindungsabbrüche und Aussetzer waren an der Tagesordnung. Bereits ein Jahr später im Jahre 1996 erschien mit „QuickTime Conferencing“ das erste Programm das Kommunikation in Voll-duplex erlaubt. Mit H.323 verabschiedet ebenfalls 1996 die ITU eine erste Norm für Multimedia-Kommunikation über paketorientierte Netze und die IETF mit RFC 1889 (heute RFC 3550) das Realtime Transport Protocol (RTP) welches in H.323 zur Übertragung von Audiodaten benutzt wird. Die aus den neuen Standards resultierenden Produkte kamen dann ab 1997 auf den Markt. H.323 beschrieb unter anderem Gateways die nun umgesetzt wurden, und mit deren Hilfe erstmals die heterogenen Netze des Internets und dem öffentlich Festnetz verbunden werden konnten. Somit konnten nun alle Kombinationen von Telephonendgerät und PC umgesetzt werden. Teure Minutenpreise in Ferngesprächen konnten beispielsweise auf diese Art umgangen werden. 1998 wurden weitere Elemente von H.323 umgesetzt, so genannte Gatekeeper (Server), die in lokal betriebenen Netzen die Bereitstellung der Leistungen und Gesprächsvermittlung übernahmen. 1999 wurde mit dem **SIP (Session Initiation Protocol)** von der IETF ein Protokoll standardisiert (RFC 2543), dass vollständig mit dem Standard H.323 konkurrieren konnte. Zu diesem Zeitpunkt sank das öffentliche Interesse in Deutschland an VoIP aber schon wieder, was mit der Liberalisierung des Telekommunikationsmarktes und dem entsprechenden geringeren Einsparpotentials durch VoIP zusammenhing. Während der Hype um IP-Telephonie im Jahre 2000 quasi vollkommen zum Erliegen kam, erschien in RFC 2916 von der IETF das ENUM (Telephon Number Mapping) Protokoll, welches die Umsetzung zwischen Telephonnummer und IP Adressen regelt. Nach bisher mäßigem Erfolg des SIP regelt die IETF in RFC 3261 Neuerungen am SIP Protokoll

im Jahre 2002. Die neue Version wird zum Erfolg und schon 2004 setzen nahezu alle VoIP Anbieter nunmehr auf SIP und nicht wie bisher auf H.323. Zuvor im Jahre 2003 wurden der heutigen Denic der Betrieb von ENUM übertragen, womit eine deutschlandweite VoIP-Nummernvergabe möglich war. Im selben Jahr wurde mit **STUN (Simple Traversal of UDP over NAT's)** auch ein Protokoll vorgestellt, das für SIP die letzte Hürde nahm, so genannte NAT-Firewalls, fortan konnten Telephonate über Router hinweg aufgebaut werden. Im Jahre 2004 erscheint die Software Skype, sie verwendet ein eigenes nicht offengelegtes Protokoll. Während andere kommerzielle Diensteanbieter ihre Endgeräte auf SIP ausrichten, und ab 2004 von den in hoher Zahl ansteigenden Breitbandzugängen profitieren, die VoIP attraktiv machen, verfolgt **Skype** weiterhin das eigene Protokoll. Hat Damit allerdings ebenfalls bisher Erfolg. Im Jahre 2006 ging ENUM aus dem Testbetrieb in den Wirkbetrieb über, und konnte bis Dato über 1 Million vergebener ENUM Telephonnummern melden.

B. Theoretische Grundlagen

Um zu wissen wie VoIP funktioniert sollte man zunächst wissen, wie ein herkömmliches Telefongespräch funktioniert. Bei einem herkömmlichen Telefonat, wählt ein Teilnehmer eine Nummer mit seinem Telefon, diese Nummer wird dann zur örtlichen Vermittlungsstelle geleitet, welche die Telefonnummer identifiziert, und an die Vermittlungsstelle weiterleitet, welche für das Zielnetz zuständig ist. Der Vermittlungsrechner im Zielnetz nimmt dann das Gespräch an, und leitet es an den entsprechenden Festnetzanschluss weiter, wo daraufhin das Telefon zu klingeln anfängt. Beim herkömmlichen Verbindungsaufbau müssen also mehrere Stationen durchlaufen werden, bis eine Verbindung zustande kommen kann. VoIP wählt hier eine grundsätzlich andere Strategie. Bei VoIP wird zunächst eine direkte Verbindung mit dem Empfänger über das Netzwerk aufgebaut. Der Aufbau einer Verbindung ist hierbei generell anders, was mit der Architektur des zugrunde liegenden Internets zusammenhängt. Es gibt keine Vermittlungsstellen in dem Sinne, sondern vielmehr werden die Sprachinformationen vor dem Versenden in Datenpakete zerteilt. Diese Datenpakete werden nun wie jeder andere Datenverkehr mittels geeigneten Transportprotokoll über das Netzwerk übertragen. Somit ist nicht garantiert, dass jedes Datenpaket den selben Weg durch das Netzwerk nimmt, bzw. die Reihenfolge der Pakete erhalten bleibt. Somit ist wie bereits oben erwähnt keine gleich bleibende Qualität zwischen den Gegenstellen möglich. Sollte es nämlich vorkommen das Datenpakete verworfen werden müssen, und diese Sprachdaten enthalten, geht deren Information verloren, und es kommt zu Aussetzern im Gespräch. Da es sich um Echtzeitdaten handelt macht hier ein erneutes Senden wenig Sinn.

C. Protokolle

Die Entwicklung der IP-Telephonie ist das Resultat einer Evolution von Standardisierungen. Dementsprechend zahlreich sind die in diesem Bereich aufgetretenen Standards. Im Folgenden seien daher nur 2 bedeutende Standards erklärt, die zum Verständnis des Themas einen wichtigen Beitrag leisten können.

Erwähnenswert ist zusätzlich noch die XMPP-Protokollerweiterung „Jingle“, da **XMPP** bereits bei Instant Messaging thematisiert wurde. Jingle erweitert hierbei XMPP um Voice-Over-IP und Peer-To-Peer-Funktionen. [32] [33] Der Instant Messenger **Google Talk** benutzt diese Erweiterung, um eine VoIP Lösung anbieten zu können. [34]. Für weitere Protokolle sei auf die entsprechende Lektüre verwiesen. Auch wird die Funktionsweise der Protokolle nicht im Detail erklärt, da SIP und Skype Thema einer eigenen Ausarbeitung sind.

1) **SIP (Session Initiation Protocol)**: Das SIP ist ein Netzwerkprotokoll, welches den Aufbau einer Kommunikationssitzung zwischen mindestens zwei Teilnehmern regelt. Das Protokoll wird in RFC 3261 spezifiziert. Benutzt wird das Protokoll hierbei, um beliebig lange Sessions von Multimediaströmen verwalten zu können. Es ist also nicht auf den Einsatz für die IP-Telephonie beschränkt. Wenn man es im ISO/OSI-Schichtenmodell einordnen würde, wäre SIP ein Protokoll auf der Anwendungsschicht, also Schicht 7. Dem entsprechend benötigt SIP im Einsatz ein zugrunde liegendes Transportprotokoll wie z.B. RTP, TCP oder UDP.

Abbildung 9. Darstellung des Ablaufes einer Kommunikation mittels SIP.
alice@192.168.2.5 SIP Proxy bob@192.168.1.10

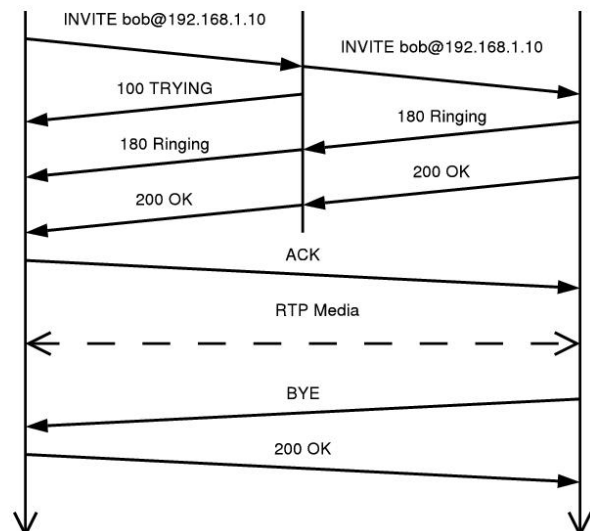


Abbildung 9 zeigt den Ablauf einer SIP-Sitzung vom Rufaufbau bis zum Beenden der Übertragung. Wie man in ihr erkennt sendet ein VoIP-Teilnehmer zum Aufbau eines Gespräches, eine Gesprächseinladung für einen anderen

VoIP-Teilnehmer zunächst an einen SIP-Server der die Daten als Proxy an den Gesprächsempfänger weiterleitet und dem einladenden Teilnehmer (in der Grafik als Alice bezeichnet) den Status des Versuches des Verbindungsaufbaus zurückgibt. Da der Empfänger (in der Grafik Bob) online ist, liefert sein SIP-Client an den Proxy zurück, dass er es klingeln lässt. Diese Information wird an Alice-Client weitergeforwardet, sodass sie eine Rückmeldung des aktuellen Status erhält. Je nach Implementierung kann dies analog zum normalen Festnetztelefon als Freizeichen signalisiert werden. Sobald Bob das Gespräch nun entgegennimmt, schickt sein Client dem Proxy dies als Information. Der Proxy leitet diese wiederum an Alices Client weiter. Durch ein ACK signalisiert Alice-Client nun direkt an Bobs Client, dass er wahrgenommen hat, dass Bob abgehoben hat. Er etabliert einen SIP-Mediastream, der über RTP übertragen wird. In dem Beispiel der Grafik beendet Bob das Gespräch. Sein Client schließt anschließend die SIP-Session, und der Client von Alice gibt nur eine Statusmeldung zurück, das erfolgreich geschlossen werden konnte. [35] [36]

2) *STUN (Simple Traversal of UDP over NAT's)*: STUN ist wie bereits oben erwähnt ein Netzwerkprotokoll zum Erkennen von Firewalls bzw. Routern. Der Zweck von STUN liegt darin das Überwinden der Selbigen zu erleichtern, und erfüllt anbei auch gleich die Funktionalität der Ermittlung der eigenen IP-Adresse.

STUN's Funktionsweise ist folgende: durch Verbindungsversuche mit verschiedenen Konfigurationen wird versucht zu ermitteln welche Art von Datenpaketen den Empfänger erreichen, der vermutlich hinter einer Firewall oder NAT sitzt. So sieht der Algorithmus [37] [38] von STUN an einer bestimmten Stelle zum Beispiel vor, die öffentliche IP des Empfängers mit der gelinkten IP des Empfängers zu vergleichen. Unterscheiden sich beide, ist verifiziert das der User hinter einem Router sitzen muss. Ist keine NAT/kein Router vorhanden, stimmen diese überein. Mit einer Abfolge von derartigen Requests lässt sich für einen Receiver die Art des Netzwerkes und dessen Schutzes ordentlich ermitteln. Entsprechend des ermittelten Statuses des Netzwerkes kann nun SIP reagieren, und versuchen eine Kommunikation aufzubauen. [38] [37]

D. Funktionalitäten

Die heute gängigen IP-Telephoniedienste allen voran **Skype** bieten bereits als Basisfunktionen im Wesentlichen alle Funktionen die auch die gängigen Instant Messenger besitzen. Ihr Alleinstellungsmerkmal ist dann jedoch die Möglichkeit Telefongespräche führen zu können. Ebenso haben sich für alle klassischen Telephondienste Online-Derivate herausgebildet, die deren Funktionalität im virtuellen Raum nachbilden. So existieren zum Beispiel in den Programmen auch integrierte Anrufbeantworter, oder es gibt die Möglichkeit Konferenzgespräche zu starten. Auch Programme

mit integrierter Sprachsteuerung gibt es, sowohl zur direkten Steuerung als auch der Fernsteuerung.

E. Clients

Die Bekanntesten Clients der Internet-Telephonie sind wohl **Skype**, **Google Talk** und **TeamSpeak**.

1) *Skype*: Skype ist eine kostenlos erhältliche VoIP-Software. Das verwandte Protokoll ist proprietär. Skype unterscheidet sich von den meisten anderen VoIP-Clients dadurch, dass es die Möglichkeit bietet, wenn auch gebührenpflichtig mit SkypeOut ins Festnetz oder den Mobilfunk zu telefonieren, oder mit SkypeIn ebenfalls gebührenpflichtig die Erreichbarkeit aus dem herkömmlichen Telephonnetz herzustellen. Für nähere Informationen zu Skype sei hierbei auf die Seminararbeit „SIP und Skype“ dieses Seminars verwiesen. [39]

2) *Google Talk*: Bei Google Talk handelt es sich um ein Instant Messaging System, welches zunächst auf Textnachrichtenaustausch ausgerichtet war. Es basiert auf dem XMPP-Protokoll. Durch Google Talk bekommen Gmail-Benutzer die Möglichkeit in Echtzeit zu kommunizieren. Nach der Erweiterung des XMPP Protokolls durch Jingle, ist es nun auch möglich den Client für VoIP Dienste zu nutzen. Google Talk ist aufgrund des verwandten Protokolls desweiteren auch mit jedem beliebigen anderen Jabber-Client nutzbar. Seit Anfang 2006 ist Jabber mit Google Talk vollständig kompatibel, sodass die Nutzer der verschiedenen Netze mit einander kommunizieren können. Ein aktuelles Bestreben verfolgt das Ziel die Kompatibilität von Skype und Google Talk zu erhöhen bzw. langfristig zu ermöglichen. [40] [41]

3) *TeamSpeak*: TeamSpeak (TS) ist eine proprietäre Sprachkonferenzsoftware, welche für die Nutzung in Online-Spielen optimiert ist. Die Software hat somit eine geringe Prozessorauslastung und verursacht verhältnismäßig wenig Datenverkehr. Für nicht kommerzielle Nutzung ist die Software frei verfügbar.

Da TS sein eigenes nicht offenes Protokoll verwendet, ist es zu den herkömmlichen VoIP Protokollen wie SIP und H.323 inkompatibel. Das Konzept besteht aus Servern und Clients, wobei auf jedem Server sogenannte Channels existieren. Die Benutzer können nun auf dem Server je nach ihren Rechten Räume eröffnen oder mit Passwort schützen. Auch kann bestehenden Räumen beigetreten werden. [42] [43]

V. ABSCHLIESSENDE ZUSAMMENFASSUNG

Dieses Dokument beschäftigte sich nach einer kurzen **Einleitung** mit den Themen **Instant Messaging** und **Internet-Telephonie**. Hierbei wurde zunächst das Thema Instant Messaging behandelt. Es wurde die **Geschichte des Instant Messaging** beleuchtet, genauso wie die **Technologien** die zur Umsetzung verwandt werden. Nachdem die möglichen **Kommunikationsschemata** dargelegt waren, wurde an praktischen

Protokollen (TelNet, IRC, ICQ, XMPP) die Umsetzung der Technologien exemplarisch beschrieben. Nach einer kurzen Erwähnung der heutzutage zur Verfügung stehenden **Funktionen** wurden 2 dieser Funktionen näher erläutert. Dieses Thema abrundend wurden die **proprietären Clients** den **frei verfügbaren Multi-Protokoll-Clients** gegenübergestellt, und das Für und Wieder der Clients diskutiert. Der zweite Teil dieser Arbeit beinhaltet die **Internet-Telephonie**. Auch bei diesem Thema wurde zunächst die **geschichtliche Entwicklung** dargelegt, woran sich ebenfalls eine **Einweisung** in die Technologie anschloss. Es wurden folgend die Protokolle **SIP** und **STUN** erläutert, auf die bestehenden Funktionalitäten eingegangen, sowie die wahrscheinlich 3 am weitesten verbreiteten Software-**Clients** für Internet-Telephonie vorgestellt. Nach dieser kurzen Zusammenfassung wird dieses Dokument schließlich durch ein letztes **Fazit** mit Ausblick abgerundet werden.

VI. FAZIT

Abschließend lässt sich für die beiden Themen Instant Messaging und Internet-Telephonie festhalten, dass beiden Zweigen in der heutigen Wissens- und Telekommunikationsgesellschaft eine rosige Zukunft bevorsteht. Den kommerziellen Unternehmen dürften, in diesem Bereich noch hohe Gewinnmargen bevorstehen. So erfolgreich beide Themen langfristig in Zukunft auch sein werden, kurzfristig dürfte sich die Entwicklung hier noch unterscheiden. Nachdem die Technologien im Instant Messaging recht ausgereizt scheinen, wird es hier wohl nur noch zu Kompatibilitätsbestrebungen kommen. Der Massenmarkt scheint durch die verfügbaren Produkte hier gesättigt. Es zeichnet sich ab, dass neue Messengern mit sehr innovativen Ideen aufwarten müssen, um sich am Markt noch halten zu können. Beispiele für solch innovative Ideen liefern Messenger wie „X-Fire“ oder „C-Space“. X-Fire soll z.B. speziell „Gamer“ ansprechen, diese bekommen dann in der Kontaktliste zu den Kontakten das aktuelle Online-Spiel und den Server auf dem sie sich befinden angezeigt. Über Tastenkombinationen kann hiermit auch während des Spieles gechattet werden [44]. C-Space etabliert VPN's unter Buddys, und ermöglicht somit neben dem Chatten auch ein anonymes relativ sicheres Filesharing [45]. Es wird spannend sein die Entwicklung von Nischenprodukten zu verfolgen, und sich von den Ideen der Entwickler überraschen zu lassen. Bei VoIP ist eine solche Überraschung zunächst sicher nicht primäres Ziel. Bei VoIP steht wohl eine Weiterentwicklung der Technologien im Vordergrund, sowie eine Steigerung bei den „Quality of Services“.

VII. ABKÜRZUNGEN

AIM - AOL Instant Messenger

AOL - America OnLine

ARPA - Advanced Research Projects Agency

ARPANET - Advanced Research Projects Agency Network

BitNet - Because it's time Network

BNF - Backus Naur Form

CCITT - Comité Consultatif Internationale Télégraphique et Téléphonique

EIES - Electronic Information Exchange System

EMISARI - Emergency Management Information Systems And Reference Index

ENUM - tElephon NUmber Mapping

FLAP - keine konkrete Bedeutung

GSM - Global System for Mobile Communications

ICQ - „I seek you“

IETF - Internet Engineering Task Force

IM - Instant Messaging / Instant Messenger

IP - Internet Protocol

IRC - Internet Relay Chat

ISDN - Integrated Services Digital Network

ISO - International Standardization Organisation

ITU - International Telecommunication Union

MSN - MicroSoft Network

NAT - Network Address Translation

NVP - Network Voice Protocol

OSCAR - Open System for CommunicAtion in Realtime

OSI - Open System Interconnection

PCM 30 - Puls Code Modulation (mit 30 Kanälen)

PLANET - PLAnning NETwork

RFC - Requests For Comments

RTP - Realtime Transport Protocol

SIP - Session Initiation Protocol

SNAC - keine konkrete Bedeutung

STUN - Simple Traversal of UDP over NATs

TCP - Transmission Control Protocol

TelNet - Telecommunication Network

UDP - User Datagram Protocol

URI - Unified Resource Identifier

VoIP - Voice over IP

VPN - Virtuell Private Network

XML - eXtensible Markup Language

XMPP - eXtensible Messaging and Presence Protocol

YMSG - keine konkrete Bedeutung

VIII. QUELLEN

Dieses Kapitel enthält Verweise auf Quellen die das gesamte Dokument beeinflussten oder jedoch große Teile daraus und nicht an den entsprechenden Stellen angeführt werden konnten. Alle Quellen finden sich ebenso wie die zugeordneten Quellen innerhalb des Dokumentes im Literaturnachweis gelistet.

a) *Allgemein Instant-Messaging:* [46], [47], [48], [49]

b) *Allgemein IP-Telephonie:* [50], [27], [31], [51], [52]

c) *Grafiken:* **Abbildung 3** und **Abbildung 4** sind angelehnt an: [48] ; **Abbildung 9** ist von [53]

LITERATUR

- [1] [Online]. Available: http://www.livinginternet.com/r/ri_emisari.htm
- [2] [Online]. Available: http://www.livinginternet.com/r/ri_planet.htm
- [3] [Online]. Available: <http://www.platopeople.com/>
- [4] [Online]. Available: http://www.livinginternet.com/r/ri_eies.htm
- [5] [Online]. Available: <http://www.njit.edu/v2/CCCC/eies/eiesinfo.html>
- [6] [Online]. Available: <http://web.inter.nl.net/users/fred/relay/>
- [7] [Online]. Available: <http://nethistory.dumbentia.com/relayhist.html>
- [8] Wikipedia, "Bitnet relay, englisch," 2007. [Online]. Available: http://en.wikipedia.org/wiki/Bitnet_Relay
- [9] —, "Bitnet relay, deutsch," 2007. [Online]. Available: http://de.wikipedia.org/wiki/Bitnet_Relay
- [10] —, "Internet relay chat," 2007. [Online]. Available: http://de.wikipedia.org/wiki/Internet_Relay_Chat
- [11] A. Wobser, "'nachrichtenaustausch und interaktionssysteme' seminararbeit im studienprojekt b," 2007.
- [12] [Online]. Available: http://www.clever-linux.de/LinuxGrundlagen/telnet_srv.htm
- [13] J.Postel and D.Reynolds, "Telnet protocol specification," 1983. [Online]. Available: <http://tools.ietf.org/html/rfc854>
- [14] J.Oikarinen and D.Reed, "Internet relay chat protocol," 1993. [Online]. Available: <http://www.faqs.org/rfcs/rfc1459.html>
- [15] Seite leider nicht mehr verfügbar. [Online]. Available: <http://www.micq.org/ICQ-OSCAR-Protocol-v7-v8-v9/index.html>
- [16] Open system for communication in realtime. [Online]. Available: <http://iserverdl.khstu.ru/oscar/>
- [17] [Online]. Available: <http://www.ihse.net/icq/>
- [18] P. Saint-Andre, "Extensible messaging and presence protocol (xmpp): Core," 2006. [Online]. Available: <http://tools.ietf.org/html/rfc3920>
- [19] S. Brunk and H. Mühleisen, "Vortrag zur ermittlung der eignung von xmpp zur umsetzung im projekt aims," 2007.
- [20] [Online]. Available: <http://www.xmpp.org>
- [21] Wikipedia, "Extensible messaging and presence protocol," 2007. [Online]. Available: <http://de.wikipedia.org/wiki/XMPP>
- [22] P. Saint-Andre, "Invisibility," 2005. [Online]. Available: <http://www.xmpp.org/extensions/xep-0126.html>
- [23] —, "Best practices for handling offline messages," 2006. [Online]. Available: <http://www.xmpp.org/extensions/xep-0160.html>
- [24] M. Muralidharan and R. Shah, "Message moderated conferences," 2007. [Online]. Available: http://www.xmpp.org/extensions/inbox/msg_moderate.html
- [25] Wikipedia, "Multi-protokoll-client," 2007. [Online]. Available: <http://de.wikipedia.org/wiki/Multi-Protokoll-Client>
- [26] —, "Pcm30," 2006. [Online]. Available: <http://de.wikipedia.org/wiki/PCM30>
- [27] Information über die geschichte voip bis 1995. [Online]. Available: <http://www.voip-information.de/voip-mittelstand/geschichte-ip-telefonie.html>
- [28] IETF. Network voice protocol. [Online]. Available: <http://tools.ietf.org/html/rfc741>
- [29] Wikipedia, "Network voice protocol," 2006. [Online]. Available: http://en.wikipedia.org/wiki/Network_Voice_Protocol
- [30] —, "Isdn," 2006. [Online]. Available: <http://de.wikipedia.org/wiki/ISDN>
- [31] Information über die geschichte voip ab 1995. [Online]. Available: <http://www.voip-information.de/voip-geschichte.html>
- [32] "Xmpp erweiterung jingle (xep-0166)," 2006. [Online]. Available: <http://www.xmpp.org/extensions/xep-0166.html>
- [33] "Xmpp erweiterung jingle audio(xep-0166)," 2007. [Online]. Available: <http://www.xmpp.org/extensions/xep-0167.html>
- [34] Wikipedia, "Jingle erweiterung in wikipedia," 2006. [Online]. Available: <http://de.wikipedia.org/wiki/Jabber#Jingle-Erweiterung>
- [35] —, "Session initiation protocol," 2006. [Online]. Available: http://de.wikipedia.org/wiki/Session_Initiation_Protocol
- [36] IETF, J.Rosenberg, H.Schulzrinne, G.Camarillo, A. Johnston, J.Peterson, R. Sparks, M. Handley, and E.Schooler. Sip: Session initiation protocol. [Online]. Available: <http://tools.ietf.org/html/rfc3261>
- [37] IETF, J.Rosenberg, J.Weinberger, C.Huitema, and R.Mahy. Simple traversal of user datagram protocol (udp) through network address translators (nats). [Online]. Available: <http://tools.ietf.org/html/rfc3489>
- [38] Wikipedia, "Stun," 2006. [Online]. Available: <http://de.wikipedia.org/wiki/STUN>
- [39] —, "Skype." [Online]. Available: <http://de.wikipedia.org/wiki/Skype>
- [40] Homepage von google talk. [Online]. Available: <http://www.google.com/talk/intl/de/>
- [41] Wikipedia, "Google talk." [Online]. Available: http://de.wikipedia.org/wiki/Google_Talk
- [42] —, "Homepage von teamspeak." [Online]. Available: <http://www.teamspeak.com>
- [43] —, "Teamspeak in der wikipedia." [Online]. Available: <http://de.wikipedia.org/wiki/ISDN>
- [44] F. Parusel, "X-fire: Multifunktions game-client für zocker," 2005. [Online]. Available: <http://www.netzwelt.de/news/73273-xfire-multifunktions-gameclient-fuer-zocker.html>
- [45] —, "Cspace: Filesharing, instant messenger und vnc-client in einem tool," 2005. [Online]. Available: <http://www.netzwelt.de/news/74500-cspace-filesharing-instant-messenger-und.html>
- [46] G. Schnider, "Instant messaging technologien in learning management systemen," 2004. [Online]. Available: http://www.ifi.unizh.ch/archive/masterthesen/DA_Arbeiten_2005/Schnider_Guido.pdf
- [47] A. Holzhauser, "Chat-technologien in der arbeitswelt," 2003. [Online]. Available: http://chat.themenplattform.com/_0_11840739951293428711915192535708_/fsDownload/Diplomarbeit_AHolzhauser.pdf?forumid=228&v=1&id=126557
- [48] C. Nell, "Instant messaging," 2005. [Online]. Available: <http://www-vs.informatik.uni-ulm.de/teach/ss05/vp/Ausarbeitung/Christina%20Nell%20-%20Instant%20Messaging.pdf>
- [49] IETF. A model for presence and instant messaging. [Online]. Available: <http://www.ietf.org/rfc/rfc2778.txt>
- [50] Wikipedia, "Ip-telefonie," 2007. [Online]. Available: <http://de.wikipedia.org/wiki/IP-Telefonie>
- [51] Voip - das internet erobert den telefonmarkt. [Online]. Available: <http://www.dsl-magazin.de/voip/infos/>
- [52] Telefonie im internet. [Online]. Available: <http://www.netplanet.org/telefonie/index.shtml>
- [53] "Grafische illustration des kommunikationsablaufes mit sip," 2006. [Online]. Available: http://graphics.cs.uni-sb.de/VoIP/fopra/sip_dia.jpg