Skype

Nicolai Ruess

November 2006

1 Enstehungsgeschichte Skypes

Der *Voice over IP* (VoIP) Markt ist heutzutage ein ernstzunehmender Wirtschaftszweig. Nach Veröffentlichung der ersten VoIP Software im Jahr 1995 von dem israelischen Unternehmen VocalTec hochgelobt verschwand die Branche lange Zeit in der Unbedeutsamkeit [1]. Dies hatte mehrere Ursachen, die beiden Hauptgründe waren einerseits die fehlende Internet Bandbreite der Privatnutzer, und andererseits die unausgereifte Technik der damaligen VoIP Programme. Auch die Etablierung von Breitband Internetanschlüssen konnte daran nichts ändern. Die damals standartmäßig serverbasierten Programme boten aufgrund der ständig steigenden Serverkosten bei erhötem Nutzeraufkomen kaum Vorteile gegenüber herkömmlicher Telefonie. Genau an diesem Punkt sollte Skype ansetzen.

Mit der Intention ein VoIP Programm zu entwickeln welches der Telefonbranche ernsthafte Konkurrenz bieten konnte wurde Skype am 29 August 2003 von den Gründern Niklas Zennström und Janus Friis, den Entwicklern des Filesharing Netzwerkes KaZaa, in Form einer Beta veröffentlicht. Auch wenn sich Skype einiger Aspekte herkömmlicher VoIP Programme bediente so war die Netzwerkstruktur in Form eines dezentralisierten Peer-to-Peer (P2P) Netzwerkes eine völlig neue. Die folgende Liste zeigt die Schlüsselfunktionen von Skype auf:

- Übersichtliche Benutzeroberfläche
- Keine Konfiguration nötig
- Verbale Kommunikation sowie Senden und Empfangen von Text-Nachrichten
- Dezentralisiertes P2P Netzwerk
- NAT/Firewall Problemlösung
- Globales dezentralisiertes Benutzerverzeichnis
- Datenverschlüsselung

Aufgrund der innovativen Technik und der intuitiven Benutzeroberfläche kann Skype 3 Jahre nach seiner erstmaligen Veröffentlichung über 180 Millionen Downloads verzeichnen. Im Durchschnitt benutzen Skype gleichzeitig 3 Millionen Menschen, es werden 20.000 Gespräche von insgesamt 58 Millionen registrierten Benutzern zeitgleich getätigt [2, 3].

Da die Netzwerkstruktur Skypes grundsätzlich der des P2P Programmes KaZaa ähnelt, wird auf den nächsten Seiten erstmal auf eben dieses von Niklas Zennström und Janus Friis erfundene Filesharing Programm eingegangen. Im Anschluss wird die von Skype verwendete Technik untersucht und ein Einblick in die Skype API gegeben. Abschliessend werden einige mögliche Zukunftsaspekte der VoIP Branche untersucht.

2 Das KaZaa Netzwerk

Da wie erwähnt die Netzwerkstruktur Skypes auf der der KaZaas aufbaut, gehen wir im Folgenden auf die Struktur des KaZaa Netzes kurz ein.

Durch die späteren Gründer von Skype im Jahr 2001 veröffentlicht wurde das Programm bis zum heutigen Tag laut offizieller Seite über 380 Millionen Mal heruntergeladen. Zum Aufbau des Netzwerkes macht es sich hierfür das eigens entwickelte *Fasttrack Protokol* zu Nutze welches eine durchschnittliche Benutzerzahl von 2,4 Millionen vorzuweisen hat [4]. Bei diesem handelt es sich um ein so genanntes P2P Protokol der zweiten Generation. Das Netzwerk benutzt *Supernodes* um ein flexibles und skalierendes Umfeld zu garantieren. Der Unterschied zwischen einer Supernode und einem normalen Client (Node) ist, dass die Supernode als Server für eine bestimmte Anzahl Clients dient. Dieses Feature ist in jeden Client eingebaut, so dass schnelle Computer mit einer hohen Bandbreite sich automatisch in eine solche Supernode umwandeln, dieses kann jedoch vom Benutzer auch unterbunden werden.

Wenn ein Client sich das erste Mal mit dem KaZaa Netzwerk verbindet, ruft er eine programminterne Liste von Supernodes ab und versucht sich mit einer dieser zu verbinden. Sobald dies erfolgreich geschehen ist fordert er von der verbundenen Supernode eine aktuelle Liste von Supernodes an, um ein späteres Verbinden gewährleisten zu können. Um sich nun ins Netzwerk einzubinden schickt der Client dieser Supernode eine Liste von Dateien die er zum Verteilen freigegeben hat. Alle Suchanfragen des Clients werden über die entsprechende Supernode abgewickelt. Um Suchanfragen erfolgreich bearbeiten zu können stehen Supernodes in direktem Kontakt mit einigen ihrer benachbarten Supernodes. Sobald die entprechenden Peers gefunden wurden verbindet sich der Suchanfrager mit diesen. Hierbei Fungieren die Peers als Server um den Datenverkehr direkt über HTTP abzuwickeln. Abbildung 1 zeigt exemplarisch den Aufbau des KaZaa Netzes. Anzumerken ist, dass jede Supernode

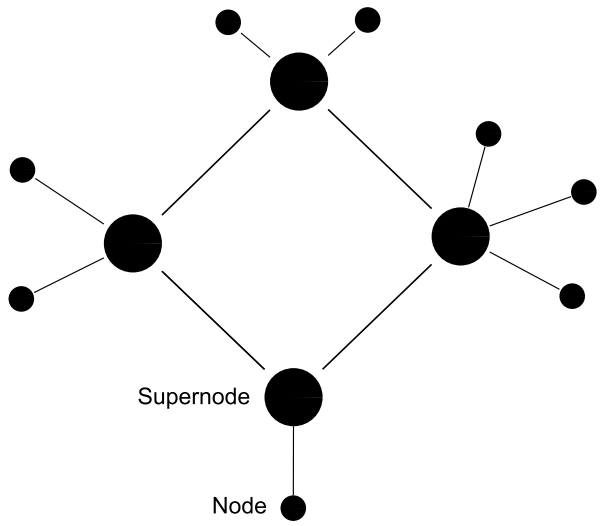

Abbildung 1: Verbindung zwischen Peers und Supernodes

in einem Intervall von 10 Minuten die mit ihm verbundenen Supernodes wechselt. Dieser Wechsel findet statt um größere Teile des Netzwerkes zu Erreichen und somit eine Fragmentierung in Form von unerreichbaren Netzteilen entgegenzuwirken [5]. Durch diesen Mechanismus erhalten Clients Zugriff auf mehr Dateien von vielfältigeren Quellen.

3 Skype: Technische Details

Im Folgenden werden die technischen Grundlagen von Skype erörtert. Es wird zuerst auf die von Skype verwendete Netzwerk Architektur eingegangen. Anschliessend wird die Sicherheit des Netzes und der verwendeten Komponenten untersucht. Darauf Folgend werden einige Informationen zu den Codes zur Sprachübertragung gegeben. Außerdem werden Routinen wie der Programmstart mit dem damit verbundenen Login, das Tätigen von Anrufen inklusive von Sprachkonferenzen und das Suchen von Benutzern näher beschrieben. Es ist zu beachten dass einige der erwähnten Punkte rein durch Untersuchungen erschlossen wurden, da es keine verlässlichen Infos zu den entsprechenden Themen gibt. Die folgenden Daten erheben sich grösstenteils auf *An Analysis of the Skype Peer-to-Peer Internet Telephony Protocol* von Salman A. Baset und Henning Schulzrinne [6].

3.1 Architektur des Skype Netzwerks

Skype baut wie das KaZaa Netzwerk auf dem *Fasttrack Protokoll* auf. Fasttrack ist ein *semi-dezentrales* P2P Netzwerkprotokoll. Das Skype Netzwerk organisiert sich in hohem Maße selbst, lediglich die Benutzerauthentifizierung wird über einen zentralen Server abgewickelt.

Diese im vorigen Kapitel bereits erwähnten Supernodes dienen als Knotenpunkte im Netzwerk. Im wesentlichen setzt sich das Skype Netz aus Nodes und Supernodes zusammen. Als Node bezeichnet man hierbei einen herkömmlichen Skypeclient, welcher anders als im KaZaa Netzwerk nicht unterbinden kann zu einer Supernode zu werden. Somit ist jeder Client mit entsprechender Bandbreite und Ressourcen ein potentieller Kandidat für einen Knotenpunkt im Netzwerk. Jeder herkömmliche Client stellt über eine solche Supernode Kontakt zum Skypenetz her. Diese Supernode dient dem ensprechenden Client als erste Anlaufstelle bei Suchanfragen und versorgt ihn zur Sicherung der Konnektivität mit Daten anderer derzeit erreichbarer Supernodes. In Abbildung 2 ist ein exemplarischer Aufbau des Skype Netzes zu sehen, anzumerken ist dass wie die Supernodes vermutlich wie schon im KaZaa Netz in regelmäßigen Intervallen die mit ihnen verbundenen Supernodes wechseln um ein dynamisches Netzwerk zu garantieren. Der abbgeblidete Login Server wird später genauer beschrieben. Die Verbindungen zwischem diesem und den Clients sind exemplarisch und für jeden Client vorhanden

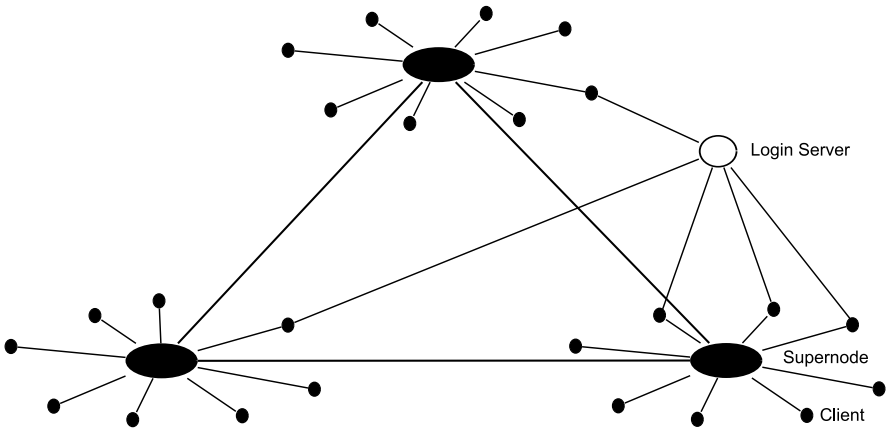

Abbildung 2: Struktur des Skype Netzes

3.2 Sicherheit im Skype Netzwerk

Da *Firewalls* und *Network Adress Translation* fähige Router heutzutage breite Verwendung finden ist es ein wichtiger Punkt Skypes die mit diesen Geräten verbundenen Probleme im Verbindungsaufbau und Informationsaustausch zu lösen. Ein erster Ansatzpunkt zur Lösung des Problems ist dass Skype zusätzlich zu seinem bei der Installation frei gewählten UDP und TCP Ports, TCP Listening Ports an den Port Nummern 80 und 443 legt, welche den standartmäsigen HTTP und HTTPS Ports entstprechen. Dies hat den Vorteil dass eben diese 2 Ports meist offen für Verbindungen sind.

Ein herkömmliches Problem von VoIP Programmen war dass 2 Benutzer hinter verschiedenen NAT Routern bzw Firewalls erhebliche Probleme beim Verbindungsaufbau erfahren können. Grund hierfür ist, dass ein NAT fähiger Router nur eingehende Pakete akzeptiert und an den vorgesehenen Empfänger weiterleitet, wenn dieser kurz zuvor eine Verbindung mit dem Absender eröffnete und somit eine offene *Session* in der Routingtabelle des NAT vermerkt ist [7]. Skype löst dieses Problem indem es Anrufe, die von oder an NAT beziehungsweise Firewall Benutzer gehen, nicht direkt zwischen den Beiden aufbaut sondern über eine dritte Node abwickelt, welche somit als Server/Router für das Gespräch fungiert. Dieser Mechanimsus wird später detailliert beschrieben.

Da Gespräche mithilfe von Skype über das öffentliche Internet und gegebenenfalls über einen dritten Skype Benutzer geleitet werden ergeben sich Sicherheitsheitsrisiken die es zu beseitigen gilt. Skype verwendet daher eine End-to-End Verschlüsselung jeglicher Datenpakete, wie zum Beispiel Text- oder Sprach-Nachrichten. Hierfür wird die bekannte Verschlüsselungsmethode AES (Advanced Encryption Standard) mit 256-Bit-Schlüssel verwendet. Diese Schlüssel, über die jeder Benutzer verfügt, müssen zudem vom zentralen Login Server beim Enloggen bestätigt werden. Durch diese Maßnamen hält Skype einen allgemein sehr hohen Sicherheitsstandart.

Aufgrund eben dieser Verschlüsselung stellt Skype in einigen Firmennetzwerk ein nicht zu unterschätzendes Sicherheitsrisiko dar. Einige Unternehmen wie zum Beispiel die Finanzbranche müssen jede Transaktion und sämtliche Telefonanrufe aufzeichnen und auch die Benutzung des Internets überwachen. Eben dieses unterbindet die Verschlüsselung von Skype jedoch. Zudem können virenbefallene Dateien mithilfe der Verschlüsselung dieser an zentralen Virenscannern vorbeischlüpfen und ins Firmennetz gelangen. Aus diesen Gründen haben einige IT Unternehmen begonnen die Verwendung von Skype im Firmennetzwerk zu unterbinden [8]. Da das Abhören von Skype Gesprächen, wenn überhaupt möglich, nur mit sehr hohem Aufwand zu bewerkstelligen ist, stellt Skype sogar für Regierungen, Behörden und auch Geheimdienste ein ernstzunehmendes Problem dar.

3.3 Codecs

Skype benutzt insgesamt zwei Codecs zum Transfer von Medienpaketen. Je nach Bandbreite der verbundenen Personen wählt Skype selbst den optimalen Codec aus um eine optimale Tonqualität zu garantieren [9]. Im Durchschnitt benötigt Skype eine Bandbreite von 3-16 Kb/s um eine optimale Tonqualität zu gewährleisten wobei für eine Gesprächsdauer von einer Stunde in etwa 30 Mb Traffic anfallen [2], bei einer maximalen Bandbreite von 1,5 Kb/s wirkt die Stimme verzerrt und unverständlich. Skype benutzt den iLB [10] und den ISAC [11] Codec. Bei letzterem ist es nicht sicher ob es sich tatsächlich um diesen handelt [6]. Die Frequenzen der verwendeten Codecs liegen zwischen 50 und 8000 Hz.

3.4 Szenarien

Der folgende Abschnitt geht vertieft auf einzelne Prozeduren der Skype Software ein.

3.4.1 Der Login Prozess

Wie bereits kurz angesprochen ist der *Login Server* das einzige zentrale Glied im ansonsten dezentralen Skype Netzwerk, daher stellt der Login Prozess eine der kritischsten Funktionen da. Der Login Server ist verantwortich für die Authentifizierung jeglicher Skypeclients. Es werden alle Login relevanten Daten auf ihm gespeichert, was zudem eine Einzigartigkeit der Benutzernamen garantiert. Für einen erfolgreichen Login in das Skype Netz muss sich der Skype Client erfolgreich beim Login Server authentifizieren. Daher ist der Loginserver das verwundbarste Glied im Skype Netz, sollte er nicht ordnungsgemäß funktionieren schlägt der Login fehl und der Client kann sich nicht ins Netz einbinden. Während dieser Routine meldet der Client auch seine Anwesenheit bei anderen Supernodes und den Benutzern seiner Freundesliste an. Desweiteren frägt er Informationen zu anderen Supernodes ab, um die Funktionalität bei Ausfall seiner derzeitigen Supernode zu gewährleisten. Zudem bestimmt er zu dieser Zeit ob er sich hinter einer Firewall oder einem NAT Router befindet. Dies geschieht während des Logins durch Nachrichtenaustausch mit der verbunden Supernode oder aber durch Nachrichtenaustausch mit anderen Nodes sobald es Verbindung mit der persönlichen Supernode unter Zuhilfenahme einer Variation des STUN [12] und TURN [13] Protokolls.

Skype speichert die Daten der eigenen Supernode sowie die der erfragten Supernodes in einer Datei namens *Host Cache*. Diese Liste wird regelmäßig auf dem neusten Stand gehalten und kann maximal 200 Einträge fassen. Sie ist von grundlegender Bedeutung für die Konnektiviät des Clients. Man muss grundsätzlich zwischen dem ersten Login der Installation und weiteren Logins unterscheiden. Der oben erwähnte *Host Cache* ist vor dem ersten Verbindungsveruch leer, somit hätte Skype eigentlich keine Möglichkeit sich erfolgreich zu verbinden. Jedoch haben Tests gezeigt dass Skype sich beim ersten Login immer zu bestimmten Supernodes verbindet, den sogennanten *Bootstrap Supernodes*, diese sind entweder im Code verankert oder aber verschlüsselt gespeichert da sie nicht lokal zu finden waren und nur einmalig abrufbar. Ein Löschen der Hostliste zu einem späteren Zeitpunkt führt unwiederruflich zur Verbindungsunfähigkeit des Clienten. Die Bootstrap Supernodes sind nötig, um zu garantieren dass der Nutzer eine erste Anlaufstelle zum Abrufen aktueller Supernodes zum Füllen des Host Caches findet. Es ist zudem anzunehmen dass der Client bei diesen die IP des Loginservers erfragt. Bei herkömmlichen Logins des Skype Benutzers, versucht der Client,wie in Abbildung 3 zu sehen, sich mit einer der Supernodes in seinem Hostcache zu verbinden, falls er erfolgreich eine Verbindung herstellen kann authentifiziert er sich bei dem Login Server und sendet abschliessend UDP Nachrichten an 22 verschiedene Nodes um seine Ankuft im Netzwerk bekanntzugeben. Wie bereits erwähnt führt ein Löschen beziehungsweise ein Füllen der Hostliste mit falschen Einträgen zu einer Verbindungsunfähigkeit.

Jedoch kann man durch Löschen der Einträge einige Einblicke in die Verbindungsprozedur erhalten. Abbildung 3 zeigt diese. Zuerst versucht Skype die Supernode über ihren Standart UDP Port zu kontaktieren. Reagiert diese nicht innerhalb von 5 Sekunden, so nimmt Skype an, hinter einer UDP beschränkten Firewall oder einem NAT Router zu stecken und versucht es erneut über ihren Standart TCP Port. Sollte die Supernode wieder nicht reagieren, werden nacheinander die Ports 80 (Http Port) und Port 443 (Https) Port getestet, die oft in Firewall und NAT geschützten Clients freigeschaltet sind. Sobald in einem dieser Schritte die Verbindung erfolgreich war bricht die Schleife ab, falls der erste Durchlauf jedoch erfolglos blieb wiederholt Skype die Prozedur vier mal bevor ein Scheitern verkündet wird.

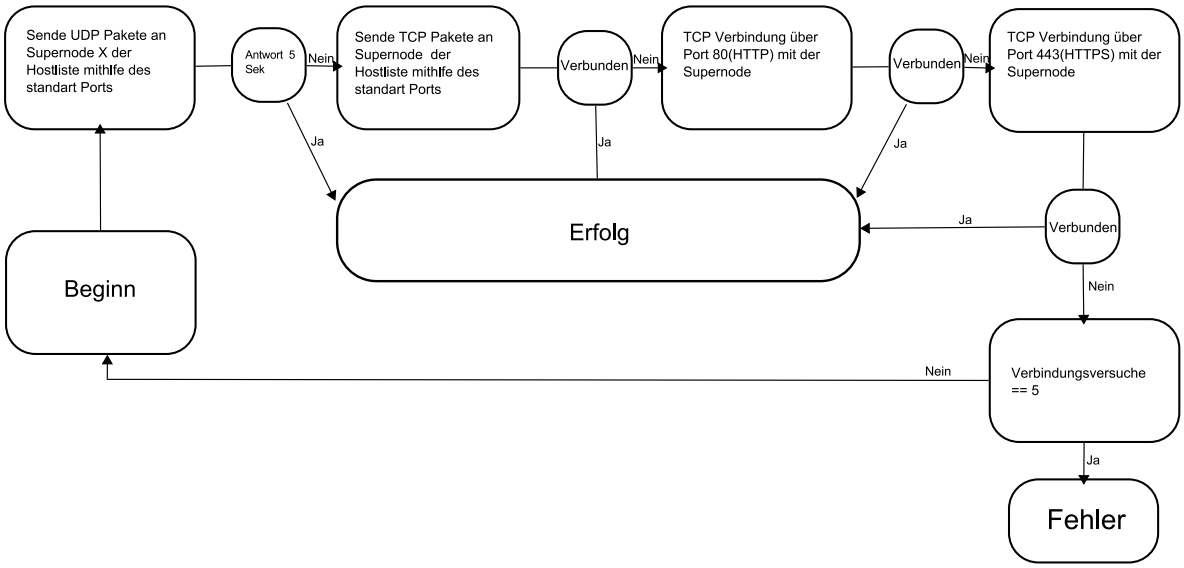

Abbildung 3: Der Login Prozess des Skype Clients

3.4.2 Benutzersuche

Da es in P2P Netzen problematisch sein kann, alle Benutzer des Netzes ausfindig zu machen muss die Benutzersuche in Skype auf besondere Weise konzipiert sein. Skype benutzt hierbei die sogenannte *Global Index* Technologie, welche laut Hersteller jeden Benutzer finden kann der in den letzten 72 Stunden online war. Da alle Nachrichten in Skype verschlüsselt sind kann man den Suchvorgang nicht weiter als bis zu seiner Supernode verfolgen. Jedoch scheint es als ob die Supernode auf eine Suchanfrage dem Skype Clienten falls er über eine öffentlichen Internetzugang verfügt die IP von vier Nodes übermittelt, die der Client kontaktieren soll. Falls dieser Vorgang nicht erfolgreich war so übergibt ihm die Supernode die IP von acht weiteren Nodes. Dies wiederholt sich bis der entsprechende Benutzer entweder gefunden wurde oder das Programm zu dem Schluss kommt, dass der gesuchte nicht existiert. Im Durchschnitt werden acht Nodes kontaktiert bis der Benutzer gefunden werden konnte. Falls der Benutzer einen NAT Router und eine Firewall die UDP Nachrichten blockt benutzt, so scheint die Supernode den Suchvorgang für den Nutzer zu übernehmen. Die Benutzersuche scheint zudem indiziert zu werden, da sich die Dauer von wiederholten Suchanfragen nach dem selben Benutzer vom ersten Durchlauf auf den zweiten von durchschnittlich 8 auf 4 Sekunden halbiert. Die Programminterne *Buddy Liste* Skypes wird wie der Host Cache lokal gespeichert. Um zu bestimmen ob ein Kontakt derzeit online ist, wird zunähst geprüft ab die entsprechenden IP Einträge noch korrekt sind, falls nicht werden wie oben beschrieben Suchanfragen zu jedem Kontakt abgearbeitet.

3.4.3 Rufaufbau

Im Folgenden wird zunächst auf den Rufaufbau zwischen zwei Personen eingegangen. Darauf folgend wird das Geschehen bei Audio-Konferenzen betrachtet. Beim Rufaufbau zweier Personen, muss man zwischen drei verschiedenen Szenarien unterscheiden.

1. Beide Benutzer verfügen über einen offenen uneingeschränkten Internetzugang.
2. Ein Benutzer befindet sich hinter einem NAT Router während der andere direkt mit dem Internet verbunden ist.
3. Beide Benutzer befinden sich sowohl hinter einem NAT Router als auch hinter einer UDP eingeschränkten Firewall.

Im ersten Fall, in dem beide Nutzer direkt mit dem Internet verbunden sind, baut der Skypeclient des Anrufers auf direktem Weg eine TCP Verbindung mit dem Angerufenen auf. Der Anrufaufbau und der damit verbundene Signalaustausch entsteht wie zu sehen über TCP, dies gilt auch für die restlichen 2 Szenarien. Dieser Nachrichtenaustausch deutet auf die Existenz eines *challenge-response* [14] Mechanismus hin. Bei diesem handelt es sich um ein Authentifizierungsverfahren welches eingesetzt wird um die Integrität beider Benutzer zu gewährleisten. Falls beide Benutzer über einen direkten und uneingeschränkten Internetzugang verfügen, findet der Austausch der Mediapakete zwischen den beiden Skypeclients direkt über die in den Optionen angegebenen UDP Ports statt.

Für den Fall, dass sich der Anrufende hinter einem Port beschränkten NAT Router befindet und der Angerufene über eine direkt Anbindung verfügt, läuft sowohl der Rufaufbau als auch der Medientransfer nicht direkt zwischen den beiden ab. Stattdessen werden sämtliche Nachrichten über eine dritte Skype Node geleitet. Der Rufaufbau geschieht wieder über TCP, während sämtliche Sprachpakete über UDP verschickt werden. Diese Konstellation gilt bis auf eine Ausnahme auch, falls sich beide Nutzer hinter einem NAT Router einer UDP eingeschränkten Firewall befinden. Da in diesem Szenario weder der Anrufer noch der Angerufene Daten über UDP empfangen können werden Signal Nachrichten und Medienpakete über die dritte Skype Node mithilfe von TCP versendet und empfangen. Das Verwenden der dritten Node in Fall 2. und Fall 3. umgeht das Problem, dass Nutzer die über ein NAT mit dem Internet verbunden sind nur Daten von IPs empfangen können an die Sie selbst Anfragen gestellt haben. So wird die beiden bekannte Node als Knotenpunkt für das Gespräch benutzt, und ermöglicht es ihnen Daten miteinander auszutauschen. Der Nachteil dieser Methode ist, dass die Node die als Server für das Gespräch fungiert mit nicht unerheblichem Traffic belastet wird, was gerade für Benutzer ohne Internet Flatrate kostenspielig werden kann.

3.4.4 Audio-Konferenzen

Im folgenden wird auf Konferenzen eingegangen die aus Telefonaten der oben aufgelisteten Szenarios entstehen. Falls zwei Benutzer, im folgenden als A und B bezeichnet, mit direktem Internatanschluss miteinander telefonieren und einen dritten Benutzer (C), welcher auch direkt verbunden ist, zu ihrem Gespräch einladen fungiert der leistungsstärkste Rechner, in unserem Fall Benutzer C, als Host für die Konferenz. A sendet in diesem Fall seine Sprachpakete über UDP an C , welcher diese mit seinen eigenen Sprachpakten mixt und an Benutzer B weiterleitet. Dies gilt umgekehrt auch für Benutzer B. In Abbildung vier wird dies nochmal grafisch veranschaulicht.

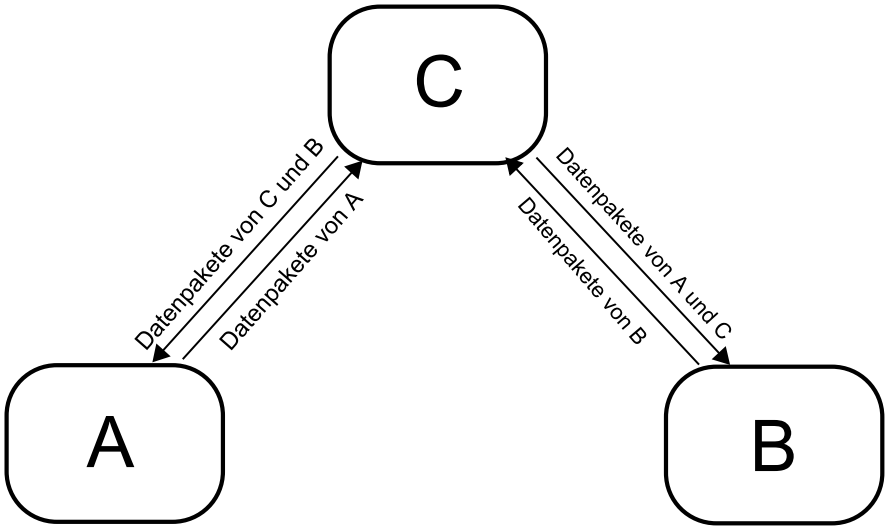

Abbildung 4: Datenfluss einer Skype Konferenz

Für den Fall, dass Benutzer A über einen direkten Internetzugang verfügt und B sich hinter einem NAT Router befindet so wird wie bereits geschrieben jedglicher Nachrichtenverkehr über eine dritte Node geleitet. Wenn nun Benutzer A einen dritten auch hinter einem NAT Router befindlichen Benutzer in das Gespräch einlädt, so wird die dritte Node die bisher für die Weiterleitung der Daten verantwortlich war ausgeschlossen und der Gesprächsteilnehmer mit dem direkten Internetzugang wird Host des Gesprächs. Das Mixen und Verteilen der Medienpakete funktioniert hierbei wie im vorherigen Abschnitt.

Dasselbe gilt falls A Und B beide hinter UDP eingeschränkten NAT Routern sitzen mit dem Unterschied dass Benutzer C, falls er einen öffentlichen Zugang besitzt, die Pakete über TCP statt UDP sendet.

Falls sich alle Benutzer hinter NAT Routern befinden so fungiert die Node die den Traffic des ursprünglichen Gesprächs weitergeleitet hat als Host für die Konferenz.

3.5 Die Skype API

Das Skype *Application Programming Interface* ermöglicht es eigenen Programmen mit Skype zu kommunizieren. So ist es zum Beispiel möglich, USB Telefone für die Benutzung von Skype zu verwenden. Der Nachrichtenaustausch zwischen Skype und den Applikation findet hierbei durch den Austausch von Textnachrichten statt. Die Skype Prgrammierschnittstelle besteht aus zwei Hauptkomponenten, der *Skype phone API* und der *Skype access API*. Hierbei stellt die phone API eine Schnitt-

stelle zu Verfügung, die es Geräten wie zum Beispiel USB Telefonen ermöglicht sich mit Skype zu verbinden. Der Skypeclient kontrolliert die phone API und sendet Events an den Gerätetreiber. Die Skype access API ermöglicht externen Geräten hingegen den Zugriff auf einige Skypefunktionen, wie etwa das Tätigen von Anrufen.

4 Zusammenfassung

Seit seiner Veröffentlichung vor drei Jahren ist Skype heutzutage eines der meistbenutzten VoIP Programme. Die Hauptgründe hierfür liegen wohl in der intuitiven Benutzeroberfläche und der Einfachheit der Benutzung. Auch wenn Skype Ende 2005 laut dem IT-Analyseunternehmen Sandvine starke Marktanteileinbußen hinnehmen musste [15] sind die Zukunfsprognosen für das Unternehmen sehr gut. Der sinkende Marktanteil liegt nicht an fallenden Kundenzahlen, täglich kommen 150000 hinzu, sondern an der Verbreitung von VoIP Diensten die an DSL Anschlüsse gekoppelt sind. Skype wurde zudem 2005 vom Online Auktionsportal Ebay für 2,6 Milliarden Euro übernommen und erhält somit neue Möglichkeiten der Verbreitung. So ist vorgesehen Skype als leistungfähiges Kommunikationstool in die weltweite Ebay-Plattform einzubinden und somit eine unkompliziertere und schnellere Abwicklung von Geschäften zu ermöglichen [16].

Literatur

[1] *Sprechen übers Netz.* C't 2001.
<http://www.heise.de/ct/01/16/154/>

[2] *Skype.* Wikipedia über Skype, November 2006.
<http://de.wikipedia.org/wiki/Skype>

[3] *Skype mit eigener Repräsentanz in Deutschland.* Pressemitteilung von Skype, September 2005.
<http://openpr.de/pdf/64946/Skype-mit-eigener-Repraesentanz-in-Deutschland.pdf>

[4] *FastTrack (protocol).* Wikipedia über Fasttrack, Oktober 2006.
<http://en.wikipedia.org/wiki/Fasttrack >

[5] Jian Liang, Rakesh Kumar, Keith W. Ross: *Understanding KaZaA*. University Brooklyn, 2004.

[6] Salman A. Baset, Henning Schulzrinne: *An Analysis of the Skype Peer-to-Peer Internet Telephony Protocol.* Department of Computer Science Columbia University, September 15, 2004.

[7] *Network Address Translation (NAT).* FLorian Messner: Einführung in VoIP, August 2004.
<http://www.florianmessner.com/support/themen/voip/firewall-nat-wlan/voip-nat-1-2-p2.htm>

[8] *Skype als Sicherheitsrisiko.* IT im Unternehmen, Dezember 2005.
<http://www.it-im-unternehmen.de/strategie/article20051229020.aspx>

[9] *Skype Technische FAQ.* Skype Homepage, 2006.
<http://www.skype.com/intl/de/help/faq/technical.html>

[10] *ILBC codec.* Global IP Sound, Oktober 2004.
<http://www.globalipsound.com/solutions/solutions_whiteprs.php?newsID=13&tot=12 >

[11] *iSAC codec.* Global IP Sound, 2005.
<http://www.globalipsound.com/datasheets/iSAC.pdf>

[12] J. Rosenberg, J. Weinberger, C. Huitema, and R. Mahy: STUN: Simple Traversal of User Datagram Protocol (UDP) Through Network Address Translators (NATs). RFC 3489, IETF, März 2003.

[13] J. Rosenberg, R. Mahy, C. Huitema: *TURN: traversal using relay NAT.* Internet draft, Internet Engineering Task Force, Juli 2004.

[14] *Challenge-Response Authentifizierung.* Wikipedia, Oktober 2006.
<http://de.wikipedia.org/wiki/Challenge-Response_Authentifizierung>

[15] *Skype verliert Marktanteile.* Zdnet, Februar 2006.
<http://www.zdnet.de/news/tkomm/0,39023151,39140645,00.htm>

[16] *Skype: Geschichte und Zukunft.* 2006.
<http://www.voip-information.de/skype-geschichte.html>