Zentrum für sichere Informationstechnologie – Austria
Secure Information Technology Center – Austria

A-1030 Wien, Seidlgasse 22 / 9
Tel.: (+43 1) 503 19 63–0
Fax: (+43 1) 503 19 63–66

A-8010 Graz, Inffeldgasse 16a
Tel.: (+43 316) 873-5514
Fax: (+43 316) 873-5520

http://www.a-sit.at
E-Mail: office@a-sit.at

DVR: 1035461 ZVR: 948166612 UID: ATU60778947

ERHEBUNG STATE-OF-THE-ART DIREKTER KOMMUNIKATIONSDIENSTE

Bernd Prünster – bernd.pruenster@a-sit.at

Zusammenfassung: Direkter Datenaustausch und direkte Kommunikationsdienste gewinnen durch nahezu immer verfügbare, große Bandbreiten zunehmend an Bedeutung. Gleichwohl führen einige Eigenschaften der dem Internet zu Grunde liegenden Infrastruktur diesbezüglich zu signifikanten Einschränkungen. Hierbei handelt es sich um „Altlasten“, welche noch auf absehbare Zeit die Art und Weise, wie direkte Verbindungen zwischen Endgeräten hergestellt werden, beeinflussen werden. Abhilfe ist mit IPv6 in Sicht, allerdings wird IPv4 nur schleppend durch IPv6 ersetzt. Im Rahmen dieses Dokuments wird auf diese und andere Umstände im Zusammenhang mit internetbasierten direkten Kommunikationsdiensten eingegangen. Zusätzlich werden weitere technische Grundlagen zum direkten Datenaustausch im Internet diskutiert. Anschließend wird eine Kategorisierung populärer Technologien zur direkten Kommunikation an Hand technischer Merkmale vorgenommen. Abschließend werden Sicherheitsaspekte direkter Kommunikationsdienste, wie z.B. Ende-zu-Ende-Verschlüsselung, beleuchtet.

Inhalt

1. Einleitung

Die Entwicklung des Internets, aber auch des World Wide Web im Speziellen, war einigen signifikanten Trendwenden bezüglich Nutzungsgewohnheiten unterworfen. Dies spiegelt sich vor allem auf Infrastrukturebene wider. Um dem raschen Zuwachs an Nutzerzahlen und internetverbundenen Endgeräten gerecht zu werden, wurde jedoch auch auf Protokollebene ein Paradigmenwechsel mit weitreichenden Folgen eingeleitet: War es ursprünglich prinzipiell möglich, auf jedem mit dem Internet verbundenen Gerät Dienste allen anderen Internetnutzerinnen und Internetnutzern zu Verfügung zu stellen und so direkten Kontakt zwischen beliebigen internetfähigen Geräten herzustellen, trifft dies heutzutage nur mehr sehr bedingt zu. Vor allem der großflächige Einsatz von *Network Address Translation* (NAT) hat dazu geführt, dass die meisten vernetzten Geräte lediglich ausgehende Verbindungen zu Serviceanbietern aufbauen können. Ohne selbst vorher einen ausgehenden Kommunikationskanal aufgebaut zu haben, können Teilnehmerinnen und Teilnehmer von anderen Internetnutzerinnen und Internetnutzern typischerweise nicht kontaktiert werden. In Zeiten als die Bandbreiten zu Normalverbraucherinnen und Normalverbrauchern hin relativ gering waren und interaktive Anwendungen ebenfalls spärlich vertreten waren, waren auch die Konsequenzen dieses Umstands überschaubar, bzw. meist irrelevant.

Damals wie heute wurde die meiste internetgestützte Mensch-zu-Mensch Kommunikation über zentrale Server abgewickelt. Frühe Dienste wie E-Mail, Webchats, IRC, aber auch aktuell populäre Instant-Messaging-Plattformen arbeiten nach demselben Prinzip: Nachrichten werden an einen Server übermittelt, zu dem die Empfängerin, bzw. der Empfänger verbunden sind, diese rufen die Nachrichten anschließend ab. Polling-Mechanismen und langlebige Verbindungen zwischen Client-Applikation und Server ermöglichen es, die Illusion aufrecht zu erhalten, dass Nachrichten aktiv vom Server an die empfangenden Parteien übermittelt werden. Selbst der Austausch großer Datenmengen wird (seit einigen Jahren zunehmend) über Anbieter von cloudbasierten Speicherdiensten abgewickelt. Subjektiv als direkte Kommunikation und direkte Informationsübertragung zwischen Endgeräten wahrgenommener Datenaustausch ist in den meisten Fällen ohne zentrale Server nicht umsetzbar.

Generell kann auch ein zunehmender (Re-)Zentralisierungstrend beobachtet werden: VoIP-Kommunikationsdienste wie Skype, welche ursprünglich als Peer-to-Peer-Netzwerk umgesetzt waren, werden zunehmend in Richtung serverzentrische Applikationen umstrukturiert. Das bewusst ohne zentrale Instanz konzipierte World Wide Web nimmt ebenfalls zunehmend zentralisierte Strukturen an. Beispielsweise hosten *Content Delivery Networks* (CDNs) mittlerweile einen signifikanten Anteil der Daten besonders populärer Webseiten. Viele webbasierte Diensten werden ebenfalls mit Hilfe einiger weniger Dienstleistungsanbieter, wie z.B. *Amazon Web Services* (AWS) realisiert. Störungen bei einem einzigen Anbieter derartiger Infrastruktur führen mittlerweile dazu, dass es zu globalen Service-Ausfällen kommt.[1]

Insgesamt nimmt die globale Vernetzung zu und auch das Nutzerverhalten verschiebt sich kontinuierlich zu immer schnelllebigerem Informationsaustausch, während die Infrastruktur, welche populäre Anwendungen wie Instant-Messaging und den Austausch von Fotos und Videos ermöglicht, zunehmend durch immer zentralisiertere Strukturen betrieben wird.

Gleichzeitig entwickeln sich jedoch auch Gegentrends: Dienste wie Filesharing sind im Wesentlichen nach wie vor als dezentral etabliert. Ausgehend von den Erfahrungen aus diesem Bereich wurden immer wieder Versuche unternommen, nicht nur große Dateien direkt zwischen Nutzerinnen und Nutzern zu verteilen, sondern auch Dienste wie Instant-Messaging oder *Voice over IP* (VoIP) als Peer-to-Peer-Anwendungen umzusetzen.

Im Rahmen dieses Dokuments wird zuerst ein Überblick über die historische Entwicklung des Internets aus Sicht der Endverbraucherinnen und Endverbraucher gegeben. Besonders die historische Entwicklung einiger grundlegender Technologien ist auch als Basis aktueller Kommunikationsdienste relevant. Anschließend werden Möglichkeiten zur direkten Kommunikation, bzw. zum direkten Informationsaustausch zwischen Endgeräten und aktuell verbreitete, konkrete Implementierungen dieser diskutiert. Abschließend werden Grundlegende Sicherheitsaspekte

[1] https://arstechnica.com/information-technology/2017/02/amazon-cloud-sputters-for-hours-and-a-boatload-of-websites-go-offline/

unterschiedlicher Kommunikationsdienste, bzw. verschiedene Herangehensweisen an diesbezügliche Sicherheitsfragen vorgestellt.

2. Limitierungen aktueller Internet-Infrastruktur

Das Internet in seiner aktuellen Form ist im Wesentlichen ein globales, heterogenes IP-Netzwerk. Allerdings sind nur wenige der angebundenen Geräte von außerhalb ihrer lokalen Subnetze direkt erreichbar. Teilweise wird dieser Zustand aus Sicherheitsgründen aktiv durch Firewalls herbeigeführt, meist haben Nutzerinnen und Nutzer jedoch wenig Kontrolle darüber, ob das eigene Endgerät von außen erreichbar ist, oder nicht. Abgesehen von Vertragsklauseln, welche es Privatpersonen verbieten können, in ihrem Heimnetzwerk Server zu betreiben, ist vor allem eine Kombination aus technischen Limitierungen und langsamem breitem Aufgreifen technischer Weiter-Entwicklungen der Grund für diesen vorherrschenden Zustand. Beispielsweise waren bereits im Jahr 2015 mehr als 16 Milliarden Geräte mit IP-Netzwerken verbunden, wovon viele auch an das Internet angebunden sind [1]. Version 4 des Internetprotokolls unterstützt jedoch nur insgesamt 4,3 Milliarden eindeutige IP-Adressen [2]. Zwar schafft IPv6 hier Abhilfe [3], allerdings zeigen von Google erhobene Statistiken, dass dessen Verbreitung nach wie vor gering ist.[2] Da die unweigerliche Adressknappheit bereits früh vorhersehbar war, wurde mit *Network Address Translation* (NAT) ein Mechanismus geschaffen, um dieser Knappheit entgegenzuwirken, bis Nachfolgeprotokolle wie IPv6 etabliert sind [4]. Dabei teilen sich alle Geräte innerhalb eines privaten Subnetzes eine einzige öffentliche IP-Adresse. Router, welche NAT umsetzen, werden als *Network Address Translator* bezeichnet (ebenfalls mit NAT abgekürzt). Die Verwendung einer einzigen öffentlich IP-Adresse für ein ganzes Netzwerk wird wie folgt umgesetzt: Beim Aufbau einer ausgehenden Verbindung wird vom NAT ein Port am öffentlichen Netzwerkinterface vergeben, welcher mit dieser Verbindung und dem verbindungaufbauenden Gerät im privaten Netzwerk assoziiert ist. Dadurch können über diese Verbindung eingehende Daten dem korrekten Gerät zugeordnet werden. Da jedoch Sender und Empfänger von Datenpaketen auf Protokollebene spezifiziert werden müssen, müssen alle Datenpakete vom NAT modifiziert werden, um die private Adresse gegen die öffentliche auszutauschen. Folglich handelt es sich hierbei um eine protokollspezifische Technik. Somit ist es aktuell de-facto unmöglich, Protokolle abseits von TCP, UDP und einiger weniger anderer (wie z.B. ICMP) für die Umsetzung von Internetbasierten Diensten einzusetzen.

Aus dieser ursprünglich interimistisch angesetzten Lösung wurde eine dauerhafte. Die Konsequenz daraus war, dass die ursprüngliche Intention, jedes mit dem Internet verbunden Gerät von überall aus anzusprechen zu können, nicht mehr aufrechterhalten werden konnte. Die Folge daraus ist der aktuell vorherrschende Zustand: Nur verhältnismäßig wenige Geräte, welche über statische IP-Adressen verfügen, sind von überall aus direkt erreichbar. Zusammengefasst wurde das Internet somit in zwei Klassen von Geräten eingeteilt und es hat sich ein Zustand der Partitionierung eingestellt.

In Zeiten geringer Bandbreiten, als Endverbraucherinnen und Endverbraucher hauptsächlich stunden- oder minutenweise über Einwahlmodems mit dem Internet verbunden waren, führte dies auch nur in den seltensten Fällen zu Problemen. Das Bereitstellen von Diensten gegenüber anderen Internetnutzerinnen und Internetnutzern, oder gar privater Serverbetrieb im klassischen Sinn, sind unter solchen Voraussetzungen auch nicht sinnvoll möglich. Allerdings unterscheidet sich die aktuelle Situation, in der durchschnittlich mehr als zwei internetfähige Geräte von einer Person benutzt werden[3], drastisch von dieser Zeit. Durch beinahe dauerhaft verfügbare hohe Bandbreiten und immer leistungsfähigere Endgeräte haben sich auch zunehmend Dienste entwickelt, welche diese Möglichkeiten nutzen, wodurch der Bedarf nach globaler Erreichbarkeit und direkter Vernetzung von Endgeräten signifikant gestiegen ist.
Im Folgenden werden Möglichkeiten beschreiben, welche versuchen, die durch NAT hervor-gerufenen Einschränkungen zu umgehen. Anschließend wird ein Überblick über die historische Entwicklung direkter Kommunikationsdienste gegeben.

[2] https://www.google.com/intl/en/ipv6/statistics.html
[3] https://www.statista.com/statistics/333861/connected-devices-per-person-in-selected-countries/

2.1. NAT-Traversal-Verfahren und deren Grenzen

Die Praxistauglichkeit und Zuverlässigkeit direkter Kommunikationsdienste ist vor allem davon abhängig, wie zuverlässig und stabil ein Verbindungsaufbau über NAT-Grenzen hinweg möglich ist. Die konzeptionell einfachste Möglichkeit, Informationen zwischen Endgeräten auszutauschen, bzw. eine Verbindung herzustellen, ist über Relays. Dabei wird eine Verbindung zu einem externen, direkt erreichbaren Knotenpunkt hergestellt, welcher anschließend den gesamten Datenverkehr weiterleitet. Zwar sind dadurch potentiell konstant hohe Bandbreiten seitens des Relays gefordert, allerdings bleibt dies jedoch die einzige Möglichkeit, Informationen zwischen Endgeräten auszutauschen, wenn keine direkte Verbindung hergestellt werden kann. Daher wurde diese Methode auch von der *Internet Engineering Task Force* (IETF) als TURN[4] [5] standardisiert. TURN beschreibt wohldefinierte Prozeduren, um eine scheinbar direkte Verbindung zwischen zwei Geräten über NAT-Grenzen hinweg aufzubauen. Auf Netzwerkebene kommen derartige Verfahren einer direkten Verbindung jedoch nicht näher.
Eine geläufige Methode, um tatsächlich eingehende Verbindungen zu Geräten hinter NATs aufzubauen, ist *Hole Punching*. Dieses Verfahren macht sich die Arbeitsweise von Routern, welche NAT umsetzen, zu Nutze. Beispielswiese wird bei verbindungslosen Protokollen wie UDP die Assoziation von privater IP-Adresse und Port zu öffentlicher IP-Adresse eine gewisse Zeit lang aufrechterhalten, selbst wenn keine Daten übertragen werden. Sind diese Informationen bekannt, kann ein Gerät trotz NAT direkt von außen erreicht werden. Verbindungorientierte Protokolle, wie beispielsweise TCP, erfordern aufwändigere Mechanismen, um alle beteiligten Instanzen zu „überlisten“, da Hole-Punching-Mechanismen teilweise entgegen der Intention der verwendeten Protokolle arbeiten. Der De-Facto-Standard im Bereich von Hole-Punching-Protokollen wurde unter den Namen *Session Traversal Utilities for NAT* (STUN) von der IETF zur Standardisierung vorgeschlagen [6]. Der Erfolg derartiger Methoden hängt jedoch stark von der tatsächlichen Netzwerktopologie ab und ist keinesfalls garantiert, weshalb im Bedarfsfall auf Relaying-Mechanismen zurückgegriffen wird.

Mit *Interactive Connectivity Establishment* (ICE) wurde ein Verfahren spezifiziert, welches eine Abstraktionsschicht anbietet, um unabhängig von der Netzwerkinfrastruktur direkte Verbindungen zwischen Endgeräten herzustellen. Dabei wird intern, falls notwendig, auf STUN und TURN zurückgegriffen um maximale Konnektivität zu gewährleisten [7].
Tatsächlich scheitert ICE aber konzeptbedingt an einem grundlegenden Anwendungsfall: Es ist trotz wohldefinierter Prozeduren und einem nahezu garantiert erfolgreichen Verbindungsaufbau unmöglich, hinter einem NAT beispielswiese einen Webserver zu betreiben, welcher auch von außerhalb des privaten Netzwerks erreichbar ist. Der Grund hierfür ist jedoch kein technischer, sondern rührt vom Anwendungsfall her, auf den ICE abzielt. Um maximale Konnektivität zu garantieren, schreiben ICE, STUN und TURN vor, dass beide Kommunikationsparteien gleichzeitig versuchen, eine Verbindung zueinander aufzubauen. Dadurch ist es einerseits irrelevant, in welche Richtung ein erfolgreicher Verbindungsaubau möglich ist, sofern nur eine Partei von außen erreichbar ist. Auf der anderen Seite wird damit prinzipiell ausgeschlossen, dass ein Service hinter einem NAT betrieben wird und lediglich auf eingehende Verbindungen wartet, ohne selbst aktiv zu werden. Diesem Problem wird im Rahmen der standardisierten Verbindungsaufbauverfahren schlichtweg keine Aufmerksamkeit gewidmet. ICE, STUN und TURN zielen vielmehr auf VoIP-Szenarien und Videochat-Anwendungsfälle ab. Dabei gibt es im Regelfall einen vom Dienstanbieter betriebenen, direkt erreichbaren Server, welcher den Verbindungsaufbau koordiniert. Somit ergibt sich das Problem einer passiven Kommunikationspartei nicht. Obwohl ein direkter Kommunikationskanal besonders für dezentral organisierte Anwendungsfälle elementar ist, wird ebendieser Anwendungsfall von standardisierten Verbindungsaufbauprozeduren konzeptionell außen vorgelassen.

Zusammenfassend handelt es sich auf Grund solcher Limitierungen bei tatsächlicher, auch auf Netzwerkebene als solche realisierte, direkter Kommunikation zwischen Endgeräten um ein ungelöstes Problem, welches je nach Anwendungsfall spezieller Lösungen bedarf. In der Vergangenheit gab es verschiedene Herangehensweisen an dieses Problem.

[4] TURN bezeichnet die Kurzform von RFC 5766; vollständiger Titel: *Traversal Using Relays around NAT (TURN): Relay Extensions to Session Traversal Utilities for NAT (STUN)*.

2.2. Entwicklung direkter Kommunikationsdienste

Ursprünglich war es möglich, jedes direkt mit dem Internet verbunden Gerät direkt zu erreichen. Auf Netzwerkebene war direkte Kommunikation auf globaler Ebene vorgesehen. Durch die bereits diskutierte Verknappung von IPv4-Adressen und die großflächige Einführung von NAT ist dies jedoch nicht mehr der Fall. Die zunehmende Verfügbarkeit hoher Bandbreiten und leistungsstarker Endgeräte lassen jedoch gleichzeitig das Verlangen nach direkter Vernetzung steigen. Voice-over-IP-Anwendungen, sowie Videokonferenzen sind nur bei direkter Vernetzung aller kommunizierenden Parteien flächendeckend umsetzbar. Doch bereits um die Jahrtausendwende gab es zunehmenden Bedarf, direkte Verbindungen zwischen Endgeräten aufzubauen. Zwar stand hier nicht Mensch-zu-Mensch Kommunikation im Vordergrund, jedoch sind die technologischen Grundlagen einiger Entwicklungen der vergangenen Jahre auch für aktuelle Kommunikationsdienste relevant. Daher wird in diesem Abschnitt auf die historische Entwicklung von Konzepten zum direkten Informationsaustausch unter besonderer Berücksichtigung des Dezentralisierungsaspekts eingegangen.

2.2.1. Instant-Messaging-Dienste und Internet-Chats

Mit den *Internet Relay Chat* (IRC) [8] wurde 1988 ein Protokoll für textbasierten Nachrichtenaustausch geschaffen, welches nach wie vor im Einsatz ist. IRC ist zentralisiert organisiert, wobei jeder IRC-Server betreiben kann. Abgesehen von Gruppenunterhaltungen (in so genannten *Channels*) werden auch private Konversationen zwischen einzelnen Benutzerinnen und Benutzern unterstützt. In jedem Fall werden Nachrichten jedoch unverschlüsselt übertragen und zumindest Nachrichten an einen Channel werden über den IRC-Server übertragen. Mit dem *Direct Client-to-Client*-Protokoll existiert jedoch eine Erweiterung, welche direkte Kommunikation zwischen Konversationsparteien unterstützt, ohne dass Nachrichten über IRC-Server geroutet werden [9]. Allerdings werden Nachrichten auch hier im Klartext übertragen.
Mitte der Neunzigerjahre gewannen Instant-Messenger in der heute bekannten Form zunehmend an Popularität. Beispiele hierfür sind *ICQ*[5], der *AOL Instant Messenger*[6] und der *MSN Messenger* [10]. Eine Gemeinsamkeit nahezu aller Instant-Messenger aus dieser Zeit ist, dass proprietäre Protokolle verwendet wurden. Die damals vorwiegend textbasierte Kommunikation wurde typischerweise serverzentrisch umgesetzt. Client-Programme von Drittanbieterinnen und Drittanbietern aber auch Analysen bezüglich der Sicherheit der verwendeten Protokolle waren auf Grund der proprietären Protokolle nur mittels Reverse-Engineering möglich. Teilweise wurden Verfahren auch patentiert[7], wodurch die Entwicklung von alternativen Clients, aber auch Veröffentlichungen von Protokollanalysen erschwert wurden.
Um die Jahrtausendwende wurde mit einem quellenoffenen Instant-Messaging Protokoll, *Jabber*, versucht, Abhilfe zu schaffen. Hier handelt es ich ebenfalls um ein serverzentrisches Protokoll, allerdings stand es jedem frei, eigene Jabber-Server zu betreiben. Ausgehend von Jabber hat sich in den folgenden Jahren das *Extensible Messaging and Presence Protocol* (XMPP) [11] entwickelt, welches auch nach wie vor im Rahmen eines öffentlichen Prozesses weiterentwickelt wird.
Mit dem *Session Initiation Protocol* (SIP) [12] und dem *Real-time Transport Protocol* (RTP) [13] gibt es auch offene Protokolle, welche besonders auf Echtzeitkommunikation in Form von *Voice over IP* (VoIP) und Videochat abzielen. Mit *WebRTC* [14] existieren auch entsprechende Protokolle für Webanwendungen. Aus Latenzgründen wird hierbei jedoch direkter Datenaustausch zwischen Parteien angestrebt. Integraler Bestandteil von Protokollsuiten sind auch der enge Einbezug von NAT-Traversal-Technologien. Vorherrschend auf diesem Gebiet ist jedoch der proprietäre, ursprünglich auf XMPP-basierende, Instant-Messaging-Dienst *WhatsApp*. Abseits von Informationsaustausch im Rahmen von Text- und Videonachrichten haben sich jedoch vor allem im Bereich von Filesharing-Diensten dezentrale Ansätze etabliert. Aktuell wird versucht diese Technologien auch für Kommunikationsdienste nutzbar zu machen.

[5] https://icq.com
[6] http://www.aim.com
[7] http://news.bbc.co.uk/2/hi/technology/2591723.stm

2.2.2. Filesharing-Dienste

Direkter Austausch von Daten wurde bereits von einigen Instant-Messengern direkt unterstützt. Dabei handelt es sich im Regelfall nicht nur scheinbar um direkte Verbindungen zwischen kommunizierenden Parteien, Daten werden auch tatsächlich ohne Umwege über zentrale Server übertragen. Das übelicherweise im IRC-Umfeld eingesetzte Direct-Channel-to-Channel-Protokoll wurde auch eingesetzt, um größere Dateien auszutauschen. Derartige Verfahren sind jedoch wenig geeignet, um Daten einer breiten Masse zur Verfügung zu stellen. Um solche Anwendungsfälle zu bedienen, wurden spezielle Filesharing-Protokolle entwickelt. Der erste breitenwirksame solche Dienst war *Napster*, welcher entwickelt wurde, um MP3-Dateien im großen Stil zu tauschen[8]. Zwar wurden Daten direkt zwischen Endgeräten ausgetauscht, allerdings oblag die Abarbeitung von Suchanfragen, sowie die Koordination von Verbindungsaufbau einem einzigen zentralen Server. Störungen oder Ausfälle dieses Servers führten zu einem Totalausfall des Dienstes.
Nachfolgende Filesharing-Dienste basierten auf dem (dezentralen) Peer-to-Peer-Ansatz und verzichteten auf zentrale Server. Ehemals populäre Beispiele hierfür sind das auf dem *FastTrack*-Protokoll aufbauende *KaZaA* [15]. Dabei wurden auch Suchanfragen und Verbindungsbau direkt über das Netzwerk, ohne Einbezug einer zentralen Instanz durchgeführt. Hervorzuheben ist in diesem Zusammenhang *eDonkey2000* [16], welches Dateien nicht im Ganzen, sondern Blockweise verteilte. Dadurch war es möglich, die Einzelteile einer Datei von mehreren Benutzerinnen und Benutzern gleichzeitig herunterzuladen. Systeme wie *eMule*[9] und später *BitTorrent*[10] kombinierten diesen Ansatz erfolgreich mit einer auf einer *Distributed Hash Table* (DHT) basierenden Netzwerkstruktur. Details zu Peer-to-Peer-Netzen und DHTs sind Abschnitt 3.1.2 zu entnehmen. Eine umfassendere Klassifikation aktuell populärer Kommunikationsdienste wird im nachfolgenden Abschnitt vorgenommen.

3. Aktuelle Trends direkter Kommunikationsdienste

Einer der aktuell populärsten Kommunikationsdienste ist der Instant-Messenger *WhatsApp*, welcher im Februar 2016 erstmals mehr als eine Milliarde aktiver Nutzerinnen und Nutzer verzeichnen konnte.[11] Nachrichten werden jedoch nicht direkt ausgetauscht, sondern im ersten Schritt an die zentralen WhatApp-Server übertragen und erst im Anschluss an die Empfängerin, bzw. den Empfänger weitergeleitet.[12] Laut ITU-T waren im selben Zeitraum rund die Hälfte der Weltbevölkerung aktive Internetnutzer. Folglich wird der Kurznachrichtenaustausch von über 28% aller Internetnutzerinnen und Internetnutzer über eine zentrale Instanz abgewickelt.
Im Rahmen des *Decentralized Web Summit*[13] wurde dem Web im gesamten ebenfalls ein zunehmender Zentralisierungsgrad attestiert. Allerdings bezieht sich diese Aussage vor allem auf das Nutzerverhalten: Einige wenige Dienste und Serviceanbieter verzeichnen Nutzerzahlen im Bereich mehrerer hundert Millionen. Grund dafür ist, dass Benutzerinnen und Benutzer stärker dazu tendieren, weniger unterschiedliche Dienste zu nutzen, diese aber dafür umso intensiver. Projekte wie *Solid*[14] versuchen der damit einhergehenden Konzentration von Nutzerdaten mit Hilfe von Linked-Data-Ansätzen entgegenzuwirken. Tatsächlich lässt die enorme Popularität von Diensten wie WhatsApp jedoch vermuten, dass (De-)Zentralisierung für die Akzeptanz eines Dienstes irrelevant ist. Die Betrachteten von dezentralen Filesharing-Diensten wie BitTorrent, welcher nach wie vor für einen signifikanten Teil des globalen Upstream-Traffic verantwortlich ist[15], bestärkt diese Annahme. In beiden Fällen werden Daten zwischen Parteien ausgetauscht – einmal eindeutig serverzentrisch und im anderen Fall völlig dezentral organisiert, direkt von Benutzerin, bzw. Benutzer zu Benutzerin, bzw. Benutzer. Ebenfalls insgesamt große Datenmengen, jedoch typischerweise kleinere Einzeldateien, werden von vielen Nutzerinnen und Nutzern hingegen häufig über Cloud-Speicherdienste verteilt. Tatsächlich ist es oftmals eine Standpunktfrage, ob ein System, eine Architektur, Konzepte, oder Technologien als dezentral bezeichnet werden können und an Hand

[8] http://www.pcworld.idg.com.au/article/22380/requiem_napster/
[9] http://emule-project.net/
[10] http://www.bittorrent.org/
[11] https://blog.whatsapp.com/616/One-billion
[12] http://digitalperiod.com/explore-whatsapp-clock-sign-and-tick/
[13] https://www.decentralizedweb.net/
[14] https://solid.mit.edu/
[15] https://torrentfreak.com/bittorrent-still-dominates-internets-upstream-traffic-151208/

welcher Kriterien ein Datenaustausch zwischen Knoten eines Netzwerks als direkt einzustufen ist. Im Rahmen dieses Abschnitts wird eine Klassifizierung definiert, welche die aktuell populärsten Ansätze zur direkten Kommunikation und Dezentralisierung an Hand technischer Merkmale kategorisiert.

3.1. Unterscheidung basierend auf Netzwerkinfrastruktur

Ein oft herangezogenes Merkmal, an Hand dessen der Dezentralisierungsgrad eines Netzwerks, bzw. der darüber angebotenen Dienste festgelegt werden kann, ist dessen Abhängigkeit von zentralen Instanzen. Hier reicht die Spannweite von vollständig zentralisiert organisierten, serverbasierten Diensten bis hin zu vollständig dezentral strukturierten Peer-to-Peer-Netzen. Innerhalb dieser Grenzen gibt es unterschiedliche Systemeigenschaften, welche adaptiert werden können, um mehr oder weniger direkte Kommunikation ermöglichen. Primär muss jedoch zwischen serverbasierten Systemen und Peer-to-Peer-Netzwerken unterschieden werden. Grund dafür ist die Eigenschaft von P2P-Netzen, Daten tatsächlich direkt zwischen Nutzern, bzw. Nutzerinnen auszutauschen, selbst wenn für den Verbindungsaufbau zwischen kommunizierenden Parteien eine zentrale Instanz verantwortlich ist. Demgegenüber findet Datenübertragung bei serverbasierten Kommunikationsdiensten unter Einbezug zentraler Instanzen statt, selbst wenn dies für Benutzerinnen und Benutzer nicht direkt ersichtlich ist. Nachdem jedoch auch derartige Architekturen verwendet werden um internetbasierte Mensch-zu-Mensch-Kommunikation zu realisieren, wird auch dieser Ansatz genauer beleuchtet.

3.1.1. Traditionelle serverbasierte Kommunikationsdienste

Im einfachsten Fall wird Benutzerinnen und Benutzern eines Dienstes der Eindruck vermittelt, dass direkte Kommunikation zwischen Teilnehmerinnen bzw. Teilnehmern ermöglicht wird. Im Hintergrund wird jedoch der gesamte Datenverkehr über die Server des Dienstanbieters geleitet. Das offensichtlichste technische Argument für eine derartige Architektur liegt in den zuvor beschriebenen Limitierungen der internet-Infrastruktur begründet. Der Preis für ein rein serverzentrische Umsetzungen ist ein zentraler Angriffspunkt, dessen Ausfall das gesamte System zum Erliegen bringt. Typische Beispiele für Serverbasierte Kommunikationsdienste sind Web-basierte Chats oder Instant-Messaging-Dienste, wie das zuvor erwähnte WhatsApp oder *Nimbuzz*[16], welche auf dem XMPP-Standard basieren [17]. XMPP ist dabei selbst im Kern als serverbasiertes Kommunikationsprotokoll konzipiert [11]. Unter diesem Gesichtspunkt handelt es sich eindeutig nicht um direkte Kommunikation, sondern um ein klar zentralisiertes System.
Nachdem XMPP-Identifier ebenso wie E-Mail-Adressen global eindeutig sind, das Protokoll jedoch keine konkreten Server definiert, kann theoretisch auf jedem öffentlich erreichbaren internetfähigen Gerät ein XMPP-Server betrieben werden. Die Protokollspezifikation definiert (ebenfalls analog zu E-Mail) abseits Benutzer-zu-Server-Kommunikation auch Server-zu-Server-Kommunikation, wodurch sich ein globales XMPP-Netz aufbauen lässt, ohne eine zentrale Instanz. An Stelle eines einzelnen zentralen Knotenpunkts treten somit viele Cluster bestehend aus einem XMPP-Server und dessen Nutzerinnen und Nutzern. Eine Voraussetzung dafür ist Protokollkonformität aller teilnehmenden Instanzen. Der populärste XMPP-basierte Kommunikationsdienst, WhatsApp, verletzt die Spezifikation jedoch in einigen Punkten, weshalb Interoperabilität zwischen den Netzwerken um XMPP-Server in der Realität nicht gewährleistet ist. Zusammenfassend lässt sich auf diesem Wege (zumindest theoretisch) ein gewisser Grad an Dezentralisierung erreichen. Allerdings ist eine derartige Struktur noch weiter von direkter Kommunikation entfernt als z.B. Webchats, da mehr als ein einzelner, zentraler Server die Nachrichten zwischen Benutzerinnen, bzw. Benutzern überträgt. Im Idealfall bleiben diese indirekten Kommunikationswege der Benutzerin, bzw. dem Benutzer jedoch verborgen. Möglichkeiten zu tatsächlicher direkter Kommunikation bieten die im Folgenden beschreiben Peer-to-Peer-Netzwerke.

3.1.2. Peer-to-Peer-Netzwerke

Das Peer-to-Peer-Konzept beschreibt einen direkten Datenaustausch zwischen Teilnehmerinnen und Teilnehmern eines Netzwerks [18]. Um tatsächliche Konnektivität zu garantieren, müssen Techniken eingesetzt werden, welche z.B. NATs umgehen können und es somit ermöglichen,

[16] http://nimbuzz.com/

beliebige Netzwerkteilnehmerinnen und Netzwerkteilnehmer direkt zu kontaktieren. Jeder Nutzer und jede Nutzerin sind somit Client und Server gleichzeitig und bietet einen Teil ihrer Ressourcen dem Netzwerk an, um die Funktionalität des Gesamtsystems zu garantieren, bzw. zu unterstützen. Hierdurch ergeben sich jedoch neue Sicherheitsanforderungen und Herausforderungen. Beispielsweise können Verbindungen innerhalb klassischer Client-Server-Architekturen mittels von Zertifizierungsdiensten ausgestellter Zertifikate und TLS abgesichert werden. Serverbasierte Dienste können über eine zentrale Instanz direkt kontrollieren, welche Teilnehmerinnen und Teilnehmer den Dienst nutzen dürfen. Eine weitere grundlegende Eigenschaft von P2P-Netzen ist die nicht garantiere Verfügbarkeit von Daten. Sollen Daten von A nach B übertragen werden, müssen im einfachsten Fall beide Kommunikationsteilnehmer gleichzeitig online sein. Mangels zentraler Server ist eine Zwischenspeicherung nicht ohne Weiteres umsetzbar. Caching-Mechanismen und Replikationsstrategien, welche Daten im Netzwerk verteilen, können hier Abhilfe schaffen. Details hierzu sind jedoch Implementierungsabhängig und auch je nach Anforderung variabel.
Unabhängig davon kann durch den Einsatz von Peer-to-Peer-Netzen eine Abstraktionsschicht geschaffen werden, welche die physische Netzwerktopologie unter einem homogenen Overlay-Netzwerk verschleiert. Somit ist es nicht mehr notwendig, die IP-Adresse eines Netzwerkteilnehmers bzw. einer Netzwerkteilnehmerin zu kennen, um eine Verbindung aufbauen zu können. Stattdessen genügt es, den eindeutigen Identifikator dieses Teilnehmers, bzw. dieser Teilnehmerin zu kennen. Der Verbindungsaufbau auf Netzwerkebene wird vom Netzwerk übernommen. Dadurch wird eine Basis geschaffen, um direkte Kommunikation und direkten Datenaustausch unabhängig von der tatsächlichen Netzwerktopologie zu ermöglichen. Die interne Organisation eines solchen Peer-to-Peer-Netzwerks kann auf unterschiedliche Arten erfolgen, welche nachfolgend basierend auf der im *Handbook of Peer-to-Peer Networking* [19] definierten Terminologie beschrieben werden.

Zentralisierte Peer-to-Peer-Netze

Besonders frühe P2P-Netze waren serverzentrisch umgesetzt und werden als *zentralisierte Peer-to-Peer Netze* bezeichnet. Dabei handelt es sich jedoch nur scheinbar um einen Widerspruch. Tatsächlich üben zentrale Instanzen in derartigen P2P-Netzen eine koordinierende Funktion aus, wie beispielsweise den Verbindungsaufbau zwischen Netzwerkteilnehmerinnen, bzw. Netzwerkteilnehmern zu vermitteln, oder unterstützen das Auffinden von Ressourcen im Netzwerk. Besonders Zugangskontrollen lassen sich über serverzentrische Modelle vergleichsweise einfach umsetzen. Der eigentliche Datenaustausch wird im Sinne des P2P-Prinzips direkt zwischen Teilnehmerinnen und Teilnehmern durchgeführt. Ähnliche Strategien werden bei VoIP-Lösungen verfolgt: Aus Latenzgründen ist es erstrebenswert, dass die Benutzer-zu-Benutzer-Kommunikation direkt stattfinden. Der Verbindungsaufbau wird jedoch typischerweise von einem zentralen Server koordiniert. Jedoch hat eine Kompromittierung einer solchen zentralen Instanz ähnlich schwerwiegende Konsequenzen, wie im Falle serverbasierter Anwendungen. Beispielsweise brauchen Denial-of-Service-Angriffe auf das gesamte Netzwerk lediglich auf die zentrale Instanz abzuzielen, um den Informationsfluss im Netzwerk zum Erliegen zu bringen.

Hybride Peer-to-Peer-Netze

Auch ohne zentrale Instanzen ergeben sich auf Grund der typischerweise merklichen Heterogenität der Geräte eines P2P-Netzes Asymmetrien innerhalb des Netzwerks. Im Fall von Filesharing-Systemen können punktuell hohe Lasten entstehen, wenn populäre Daten nur von sehr wenigen Teilnehmerinnen und Teilnehmern zur Verfügung gestellt werden. So genannte *hybride Peer-to-Peer-Netze* berücksichtigen diesen Umstand und bestehen aus Teilnehmerinnen und Teilnehmern, welche, ähnlich wie zentrale Instanzen, mehr Verantwortung übernehmen als andere und koordinierende Aufgaben übernehmen. Diese höherrangingen Knoten werden auch als *Super Peers* oder *Supernodes* bezeichnet. Hierbei handelt es sich jedoch je nach konkreter Umsetzung um eine dynamische Struktur. Fallen einige Supernodes aus, können deren Aufgaben von anderen Supernodes übernommen werden. Änderungen im Nutzerverhalten und in der Netzwerkstruktur können auch bewirken, dass reguläre Teilnehmerinnen und Teilnehmer mehr Verantwortung übernehmen und so zu neuen Supernodes werden. Insgesamt ergibt sich dadurch eine erhöhte Ausfallsicherheit, da es nicht mehr nur eine zentrale Instanz gibt, welche für den korrekten Betrieb eines Netzwerks verantwortlich ist. Der Messaging-, VoIP- und Videochat-Dienst *Skype* entwickelt sich zwar zunehmend in Richtung eines zentralisierten Service, basierte jedoch Jahrelang auf einer hybriden P2P-Struktur [20].

Dezentrale Peer-to-Peer-Netze

Vollkommen homogen organisierte P2P-Netze werden als dezentrale *Peer-to-Peer-Netze* bezeichnet. Tatsächlich sind in solchen Netzwerken alle Knoten gleichberechtigt und auch gleich wichtig. Ein besonders erfolgreiches Beispiel für ein vollkommen dezentral organisiertes System basierend auf einer solchen Struktur ist globales Filesharing basierend auf dem *BitTorrent*-Protokoll. Von Benutzerinnen und Benutzern ausgehende Suchanfragen werden völlig vom eigentlichen Netzwerk entkoppelt durchgeführt, der Datenaustausch selbst wird jedoch ohne Supernodes oder zentrale Instanzen durchgeführt. Die dem System zu Grunde liegende Netzwerkstruktur ist eine *Distributed Hashs Table* (DHT) basierend auf dem *Kademlia*-Design [21]. Bei einer Distributed Hash Table handelt es sich um einen verteilten Key-Value-Store. Jeder zu speichernde Wert wird an Hand seines Hashwertes identifiziert. Hierfür kommen kryptografische Hashfunktionen zum Einsatz. Ebenso handelt es sich bei den Identifikatoren jedes Netzwerkteilnehmers und jeder Netzwerkteilnehmerin um einen Wert eines kryptografischen Hashes. Soll ein Datum innerhalb einer DHT gespeichert werden, wird dessen Hashwert berechnet und das Datum an dem Knoten gespeichert, dessen Identifikator diesem Hashwert am nächsten kommt. Die Definition einer Distanzmetrik hängt von der konkreten Implementierung einer DHT ab, ebenso wie die verwendete Hashfunktion. BitTorrent, bzw. Kademlia verwenden beispielsweise SHA-1 und berechnen die Distanz zwischen zwei Hashwerten durch die Anwendung des Exklusiv-Oder-Operators auf beide Werte. Tatsächlich sind unter dem Identifikator eines Knotens dessen IP-Adresse abgelegt. Die Assoziation von Identifikator zu IP-Adresse wird von benachbarten Knoten zwischengespeichert, und in Routingtabellen bereitgehalten, um Routing auf Basis von Identifikatoren überhaupt erst zu ermöglichen. Zusätzlich zum Speichern und Abfragen von Daten werden von einigen DHTs auch Möglichkeiten zur Verfügung gestellt, einen Kommunikationskanal zu anderen Teilnehmern und Teilnehmerinnen aufzubauen. Hierbei kommen dieselben Routingverfahren zum Einsatz, wie bei regulären Datenabfragen. Das Ergebnis einer solchen Abfrage ist jedoch kein Datum, sondern ein direkter Verbindungsaufbau. Die Details hierzu unterscheiden sich von Implementierung zu Implementierung.

Obwohl diese Technologien primär dafür geschaffen wurden, um Daten in einem heterogenen Netzwerk zu verteilen, hat es wiederholt Versuche gegeben, besonders dezentrale P2P-Netze auch für Instant-Messaging einzusetzen. Hierbei liegt der Fokus auf den Overlay-Netzen, welche die Basis für alle Kommunikation und jeden Datenaustausch bilden. Der auf einer DHT basierende Messenger *Bleep*[17] ist als prominentes Beispiel dieser neuen Generation von Kommunikationsdiensten anzuführen.

Strukturierte und unstrukturierte Peer-to-Peer-Netze

Eine weitere Eigenschaft an Hand derer sich P2P-Netze unterscheiden lassen, ist, wie Anfragen nach Daten im Netzwerk durchgeführt werden und wie strukturiert oder unstrukturiert die verwendeten Routingverfahren arbeiten. Im Wesentlichen wird zwischen *strukturierten* und *unstrukturierten* P2P-Netzen differenziert. Unstrukturierte Peer-to-Peer-Netze bauen im einfachsten Fall auf dem Flooding-Ansatz auf: Anfragen nach Daten, aber auch Routinganfragen, um eine Verbindung zu einem bestimmten Knoten herzustellen, werden schlichtweg an bereits bekannte Knoten weitergeleitet, welche diese wiederum an die ihnen bekannten Teilnehmerinnen und Teilnehmer weiterleiten. Es gibt keine vordefinierte Strategie, welche derartige Anfragen nur an bestimmte Knoten weiterleitet, oder vorhersehbare Umlaufzeiten garantiert. Außerdem können dadurch hohe Lastspitzen im Netzwerk entstehen, da Anfragen nicht gezielt weitergeleitet werden, sondern, weite Teile des Netzwerks belasten. Der große Vorteil unstrukturierter Routingmechanismen, ist jedoch deren Unabhängigkeit von der Anfrage: Egal, ob gezielt nach einem bereits bekannten Datum oder einem Teilnehmer gesucht wird, oder komplexere Anfragen betreffend den Inhalt mehrerer im Netzwerk vorhandener Daten abgesetzt werden, jeder erreichte Knoten kann eingehende Anfragen auswerten. Dieser Vorgang kann in unstrukturierten P2P-Netzen tatsächlich unabhängig vom Inhalt einer Anfrage durchgeführt werden.
Strukturierte P2P-Netze können hingegen nur einige wohldefinierte Anfragen abarbeiten, bieten dafür aber effiziente Routingmechanismen, welche minimale Netzwerklast verursachen und garantierte Umlaufzeiten ermöglichen. Distributed Hash Tables sind ein Beispiel für strukturierte Peer-to-Peer-Netze. Im Rahmen von Kommunikationsdiensten ist der von DHTs bereitgestellte

[17] http://www.bleep.pm/

Funktionsumfang typischerweise ausreichend, weshalb die Technologie auch für dezentrale Instant-Messaging-Dienste interessant ist.

Besonders auf Grund aktueller Entwicklungen sind jedoch auch Anonymisierungsnetzwerke wie Tor und I2P, aber auch die Blockchain-Technologie für eine umfassende Klassifizierung aktueller Kommunikationsdienste relevant. Diese Konzepte unterscheiden sich jedoch vorrangig durch ihre primären Ziele und die bereitgestellte Fiktionalität. Die darunterliegende Netzwerkorganisation ist unter diesem Gesichtspunkt zweitranging, daher werden diese Technologien im Rahmen des folgenden Abschnittes an Hand ihrer Funktionalität klassifiziert.

3.2. Unterscheidung an Hand der bereitgestellten Funktionalität

Aus dem aktuellen Blockchain-Hype sind auch Entwürfe für Kommunikationsdienste auf Blockchain-Basis entstanden. Daneben gibt es auch Bestrebungen, Kommunikationsplattformen direkt auf Basis von Anonymisierungsnetzwerken zu entwickeln. Die Organisation und Struktur dieser Netzwerke bezüglich Zentralisierungsgrad, oder eine Klassifizierung in strukturierte und unstrukturierte Netze ist dabei zweitrangig. Vielmehr hängt die Funktionalität von Anwendungen, welche auf solchen Technologien aufbauen, von ein Haupteigenschaften ebendieser Technologien ab.

3.2.1. Blockchainbasierte Kommunikationsdienste

Die Blockchain-Technologie [22] wird in vielen Bereichen als Heilsbringer propagiert, teilweise jedoch ohne konkrete Konzepte. Der Einsatz als Grundpfeiler eines Kommunikationsdienstes ist hingegen unter bestimmten Voraussetzungen direkt mit den Eigenschaften der Blockchain als zentrales, inkrementelles (Transaktions-)Register begründbar: Die Tatsache, dass eine einmalig bestätigte Transaktion nicht mehr rückgängig gemacht, gelöscht, oder modifiziert werden kann, lässt sich auch auf den Nachrichtenaustausch im Rahmen von Mensch-zu-Mensch-Kommunikation umlegen. Ein Wissen um die prinzipielle Funktionsweise der Blockchain-Technologie wird an dieser Stelle vorausgesetzt.

Ausgehend von einem regulären Transaktionsablauf können die Blockchain-Operationen direkt auf den Nachrichtenaustausch zwischen zwei Kommunikationsteilnehmerinnen, bzw. Kommunikationsteilnehmern angewandt werden[18]: Eine Nachricht wird in Form zusätzlicher Nutzdaten in eine Transaktion kodiert. Nachdem der Empfänger, bzw. die Empfängerin und damit auch deren öffentliche Schlüssel bekannt sind, kann diese Information benutzt werden, um den Nachrichteninhalt zu verschlüsseln. Als Freigabebedingung dieser „Nachrichtentransaktion“ wird der Besitz des zugehörigen Schlüssels festgelegt. Dadurch kann die Nachricht öffentlich in der Blockchain abgelegt werden, ohne dass der Nachrichteninhalt preisgegeben wird. Gleichzeitig wird sichergestellt, dass nur der Empfänger, bzw. die Empfängerin die Nachricht entschlüsseln kann. Wird diese Nachricht in einen Block aufgenommen, ist dadurch öffentlich, und für alle Parteien ersichtlich und nachprüfbar, wer eine Nachricht an wen gesendet hat. Wenn der Empfänger bzw. die Empfängerin beim Empfang einer Nachricht eine „Bestätigungstransaktion“ erstellt, welche die ursprüngliche Nachrichtentransaktion referenziert, lassen sich ebenfalls öffentliche, beweisbare Empfangsbestätigungen implementieren. Auf diesem Weg lässt sich nachvollziehbarer Nachrichtenverkehr umsetzen. Da unter Umständen jedoch bereits Metadaten als sensibel klassifiziert werden müssen, ist ein Kommunikationsdienst nach diesem Schema nicht für jede Art sensibler Kommunikation geeignet. Echtzeitkommunikation, wie VoIP, hingegen kann ebenfalls nicht auf diese Weise abgebildet werden. Fragen der Skalierbarkeit, nach geeigneten Anreizsystemen und adäquatem Konsensmechanismus müssen je nach Anwendungsfall beantwortet werden und werden daher hier nicht näher beleuchtet. Vollständige Anonymität und durchwegs private Kommunikation versprechen die im nachfolgenden Abschnitt beschriebenen Kommunikationsdienste auf Basis von Anonymisierungsnetzwerken.

[18] Dieser Abschnitt basiert auf dem im Rahmen des *VooMessenge*r unter https://faizod.com/blockchain-solutions/voomessenger/ vorgestellten Konzepts.

3.2.2. Anonymisierungsnetzwerkbasierte Kommunikationsdienste

Anonymisierungsnetzwerke wie *Tor* [23] oder *I2P*[19] versuchen vorrangig zu verschleiern, wer mit wem kommuniziert. Auf technischer Ebene wird mit *Onion-Routing*-Verfahren versucht, Datenverkehr auf eine Art und Weise durch das Netzwerk zu leiten, dass zu keinem Zeitpunkt nachvollzogen werden kann, wer Daten an wen überträgt. Die dafür eingesetzten kryptografischen Verfahren garantieren notwendigerweise, dass alle übertragenen Daten verschlüsselt werden. Des Weiteren bieten Anonymisierungsnetzwerke auch Möglichkeiten, dem Netzwerk Services, wie z.B. Webserver oder E-Mail-Server, von Endgeräten aus zur Verfügung zu stellen. Diese werden als *Hidden Service* bezeichnet, da es nicht möglich sein soll, diese zu lokalisieren und zumindest auf Netzwerkebene mit dem Endgerät auf dem sie betrieben werden, in Verbindung zu bringen. NAT-Traversal ist auf Grund der Arbeitsweise solcher Netzwerke, welche langlebige Verbindungen zu anderen Teilnehmern und Teilnehmerinnen zwingend erfordern, auch unproblematisch. Anonymitätsnetzwerke bilden somit eine Abstraktionsschicht, welche es ermöglicht, direkte Verbindungen zwischen Netzwerkteilnehmerinnen, bzw, Netzwerkteilnehmern herzustellen.

Innerhalb des I2P-Netzes werden beispielsweise IRC-Server als Hidden Service betrieben und es werden eigene IRC-Proxy zur Verfügung gestellt um anonyme Diskussionsrunden zu ermöglichen. Der auf dem Tor-Netzwerk basierende Messenger *Ricochet*[20] ist hingegen als Peer-to-Peer-System mittels Hidden Services umgesetzt, ohne dass eine zentrale Instanz involviert ist. Folglich handelt es sich dabei um dezentral organisierte Kommunikationskanäle, welche nicht über zentrale Server aufgebaut werden. Bereits 2007 wurden mit *TorChat*[21] ähnliche Versuche unternommen. Bei all diesen Ansätzen hängt die Sicherheit einer Datenübertragung jedoch direkt von der Sicherheit des darunterliegenden Netzwerks ab. Eine diesbezügliche Weiterentwicklung eines einzelnen Dienstes unabhängig vom Netzwerk ist daher entweder nur eingeschränkt oder mit erheblichem Aufwand auf Ebenen oberhalb des Netzwerks möglich. Auch wenn man sich als Entwicklerin oder Entwickler derartiger Kommunikationsdienste auf die vom Netzwerk bereitgestellten Sicherheitsfunktionen verlässt, müssen nach wie vor Sicherheitskonzepte auf Applikationsebene entwickelt werden. In den letzten Jahren wurden zunehmend Kommunikationsdienste entwickelt, deren primäres Ziel es war, sichere, Ende-zu-Ende verschlüsselte Kommunikation unabhängig vom darunterliegenden Netzwerk zu ermöglichen. Dadurch ergibt sich einerseits höhere Flexibilität, was Verbesserungen und Reaktion auf eventuell entdeckte Sicherheitslücken angeht, andererseits sind die Sicherheitskonzepte selbst auf den jeweiligen Anwendungsfall und das konkrete Bedrohungsszenario maßgeschneidert. Im Folgenden wird ausführlicher auf Sicherheitsaspekte direkter Kommunikationsdienste eingegangen, sowie an Hand konkreter Beispiele illustriert, welche Lösungsansätze für bestimmte Klassen von Sicherheitsfragen existieren. Dabei steht direkte Kommunikation im Sinne von Mensch-zu-Mensch-Kommunikation im Vordergrund. Daher werden auch auf XMPP basierende und andere serverzentrische Dienste näher beleuchtet.

4. Sicherheitsaspekte direkter Kommunikationsdienste

Im Rahmen traditioneller Client-Server-Szenarien gibt es mit *Transport Layer Security* (TLS) in Kombination mit *Public Key Infrastructures* (PKIs) etablierte Verfahren, welche Authentifizierung Transportsicherheit und, je nach Einsatz kryptografischer Primitive, auch Forward Secrecy garantieren. Auch für E-Mails finden besonders die zertifikatsbasierenden Authentifizierungs-konzepte im Rahmen von *S/MIME* [24] Anwendung. Bei serverbasierter Mensch-zu-Mensch-Kommunikation stellt sich jedoch bei verschlüsseltem Datenverkehr die Frage, ob dieser Ende-zu-Ende verschlüsselt ist, oder ob der involvierte Server alle Daten beim Empfang entschlüsselt, für den Empfänger oder die Empfängerin neu verschlüsselt und somit jeglichen Datenverkehr einsehen und potentiell modifizieren kann.
End-to-End-Encryption (E2EE) ist jedoch nur zielführend, wenn zweifelsfrei festgestellt werden kann, mit wem kommuniziert werden soll. Zuverlässige Identitätsnachweise und Bindung einer Identität an kryptografisches Material müssen daher vorab gewährleistet werden. Im PKI-Kontext wird die Identitätsüberprüfung an Zertifizierungsdienste ausgelagert, welche dafür teilweise auch haftbar sind [25]. Ebenso wie bei Vertrauensfragen handelt es sich hierbei nicht mehr um ein rein

[19] https://geti2p.net/
[20] https://ricochet.im/
[21] https://github.com/prof7bit/TorChat

technisches Problem. Besonders im Rahmen von Peer-to-Peer-Netzen sind einzelne Parteien, welche mehrere unabhängige Identitäten vortäuschen - so genannte *Sybil*-Attacken [26] - problematisch. Insbesondere dann, wenn die Funktionalität oder Sicherheit eines P2P-Netzwerks davon abhängt, dass alle Parteien unabhängig voneinander agieren und sich nicht gegen andere Teilnehmerinnen und Teilnehmer verschwören. Zwar ist dieser Aspekt vor allem auch im Blockchain-Kontext relevant, jedoch spielt er für typische Kommunikationsdienste nur eine untergeordnete Rolle. Daher werden in diesem Abschnitt, ausgehend von OTR, Verfahren vorgestellt, welche vorrangig im Rahmen direkter Kommunikationsdienste eingesetzt werden können, um die Identität von Kommunikationsparteien zu bestätigen und infolgedessen sichere Kommunikationskanäle aufzubauen.

4.1. Off-the-Record Messaging

Off-the-Record Mesaging (OTR) [27] ermöglicht unabhängig vom eingesetzten Kommunikationsprotokoll Ende-zu-Ende verschlüsselten Nachrichtenaustausch, bietet unter anderem *Forward Secrecy*, *Deniable Authentication* (bestreitbare Authentizität der ausgetauschten Nachrichten) und *Plausible Deniability* (glaubhafte Abstreitbarkeit) und ist außerdem nicht für Man-in-the-Middle-Angriffe anfällig. Dem Key-Agreement-Prozess liegt ein Diffie-Hellman-Schlüsselaustausch zu Grunde. Aus der Unabhängigkeit vom darunterliegenden Kommunikationsprotokoll ergibt sich auch eine Unabhängigkeit von zentralen Instanzen. Durch die Kombination dieser und anderer Eigenschaften des Protokolls ergeben sich einige spezielle Eigenschaften von OTR-geschütztem Nachrichtenverkehr. Die Immunität gegenüber MITM-Angriffen wird entweder durch den Abgleich von Public-Key-Fingerprints oder einem p*re-shared Secret* ermöglicht. Ein Fingerprint-Abgleich wird dabei üblicherweise manuell über einen separaten, sicheren Kommunikationskanal durchgeführt. Die Authentifizierung mit einem pre-shared Secret erfolgt hingegen direkt über den Kommunikationskanal selbst. Eine Besonderheit von OTR ist in diesem Zusammenhang, dass dieses Geheimnis selbst im Zuge des Aufbaus einer gesicherten Verbindung vor Dritten geschützt bleibt. Weiters können sich die Kommunikationsparteien zu jedem Zeitpunkt sicher sein, mit wem sie worüber kommunizieren. Gleichzeitig kann zu einem späteren Zeitpunkt nicht mehr nachgewiesen werden, wer mit wem worüber kommuniziert hat. Derartige Eigenschaften ermöglichen unter anderem geschützte Kommunikationskanäle beispielsweise für Whistleblower. Seit einigen Jahren gewinnen Kommunikationsdienste, welche ähnliche, teilweise auch erweiterte Funktionalitäten bereitstellen unter Normalverbraucherinnen und Normalverbrauchern zunehmend an Popularität. Das auch im Rahmen von WhatsApp eingesetzte *Signal*-Protokoll und damit populärste OTR-inspirierte Protokoll ist ein Beispiel dafür. Überdies können einige der Techniken auch abseits textbasierter Mensch-zu-Mensch-Kommunikation für andere Formen der Datenübertragung eingesetzt werden.

4.2. Signal

Ursprünglich als Protokoll für einen eigenständigen textbasierten Kommunikationsdienst entwickelt, verbreitete sich das *Signal*-Protokoll [28] auch abseits von quellenoffenen Diensten. Wie OTR ermöglich es authentifizierte, Ende-zu-Ende-verschlüsselte Kommunikation. Authentifizierung kann genau wie im Rahmen OTR-geschützter Kommunikation über Fingerprint-Abgleiche erfolgen. Es wird jedoch auch das Konzept der Wiederholungsidentität unterstützt. Signal garantiert, dass eine einmal bestätigte Identität (ohne Fingerprint-Abgleich, oder sonstiger Benutzerinteraktion) nicht gestohlen werden kann: Sobald A und B die erste Nachricht ausgetauscht haben, können sich beide Parteien somit sicher sein, dass auch alle nachfolgenden Nachrichten tatsächlich ebenfalls vom selben Gegenüber stammen. Im Gegensatz zu OTR kann Signal jedoch nicht auf jedes beliebige Kommunikationsprotokoll, bzw. jeden beliebigen Dienst angewandt werden. Signal benötigt zwingend einen Server, um Nachrichten zu übermitteln und kann somit nicht ohne Weiteres im Rahmen dezentraler Dienste eingesetzt werden. Damit schließt Signal direkte Kommunikation auf Netzwerkebene aus. Besonders die Verwaltung von öffentlichem Schlüsselmaterial und damit auch der Authentifizierungsprozess erfordert zentrale Instanzen. Diese Einschränkungen, sowie einige hautsächlich dadurch ermöglichte Erweiterungen gegenüber OTR ermöglichen jedoch neue Funktionalität, wie z.B. sichere Gruppenunterhaltungen. Zusammenfassend kann Signal als eine Weiterentwicklung von OTR-Konzepten mit besonderer Berücksichtigung der Anforderungen

moderner Kommunikationsgewohnheiten angesehen werden. Für sichere VoIP-Anwendungen und Videochat kommt üblicherweise die Kombination von *ZRTP* und *SRTP* zum Einsatz.[22]

4.3. ZRTP und SRTP

Das *Secure Real-time Transport Protocol* (SRTP) [29] erweitert das auf VoIP- und Videochat-Anwendungen ausgelegte *Real-time Transport Protocol* (RTP) um Transportsicherheit. Dabei ist auch die Integrität verschlüsselt übertragener Daten garantiert. Ebenso bietet es Schutz vor Replay-Attacken. Ob Forward Secrecy gewährleistet werden kann oder nicht, hängt ähnlich wie bei TLS vom eingesetzten Key-Agreement-Verfahren ab. Hierfür wird üblicherweise das *Z Real-time Transport Protocol* (ZRTP) [30] eingesetzt. Wie bei OTR und Signal kommt auch im Rahmen von ZRTP ein Diffie-Hellman-basiertes Key-Agreement-Verfahren zum Einsatz. Üblicherweise werden Einweg-Schlüssel verwendet (vgl. *Diffie-Hellman Ephemeral*) um Forward Secrecy zu garantieren. ZRTP macht sich die Eigenschaften von sprachbasierter Kommunikation zu Nutze, um Schutz vor Man-in-the-Middle-Angriffen zu bieten. Dadurch ist es nicht notwendig, Schlüsselmaterial oder Fingerprints auszutauschen. Stattdessen wird nach dem Aufbau einer verschlüsselten Verbindung eine kompakte Repräsentation des für diese Verbindung verwendeten Schlüsselmaterials angezeigt. Die Kommunikationsteilnehmerinnen und Kommunikationsteilnehmer werden im Zuge dessen aufgefordert, einander diesen Wert anzusagen. Somit ist zwar zusätzlicher Aufwand notwendig, allerdings kann innerhalb der ersten Sekunden einer VoIP-Übertragung sichergestellt werden, dass tatsächlich eine direkte Verbindung (ohne Man-in-the-Middle) besteht, ohne dass hierfür ein zusätzlicher Kommunikationskanal benötigt wird. Allerdings beruht die Sicherheit dieses Verfahrens auf der Annahme, dass es Angreiferinnen und Angreifern nicht möglich ist, in (nahezu) Echtzeit für den Menschen glaubhafte Fälschungen menschlicher Stimme zu produzieren. Unabhängig davon ist diese Art der MITM-Abwehr von verbaler Kommunikation abhängig und daher lediglich für audio(-visuelle) Mensch-zu-Mensch Kommunikation geeignet.

5. Fazit

Im Rahmen dieses Dokuments wurden direkte Kommunikationsdienste diskutiert. Ausgehend von technischen Grundlagen auf Netzwerkebene wurde gezeigt, warum die dem Internet zu Grunde liegende Infrastruktur es aktuell erschwert, tatsächlich direkte Kommunikationskanäle zwischen beliebigen Endgeräten herzustellen. Es wurden auch Lösungen für diese Problematik vorgestellt und deren Limitierungen dargelegt. Des Weiteren wurden einige Meilensteine direkter Kommunikationsdienste und die Evolution von Möglichkeiten zum direkten Datenaustausch beschrieben, welche nachhaltige Wirkung auf die Entwicklung aktuell populärer Kommunikationsdienste hatten. Gefolgt von technischen Grundlagen aktueller Trends internetgestützter Mensch-zu-Mensch Kommunikation und direktem Informationsaustausch wurde im Besonderen auf die diesbezüglichen Sicherheitsaspekte näher eingegangen. Insgesamt wurde somit ein Überblick über aktuell populäre Technologien, deren Funktionsumfang und Limitierungen gegeben.
Die aktive Weiterentwicklung und Pflege von Kommunikationsprotokollen, welche Sicherheit als Kernfunktionalität betrachten, kann auch als Indikator für erhöhtes Sicherheitsbewusstsein, vor allem Seitens der Anbieterinnen und Anbieter solcher Dienste gewertet werden. Zunehmend nicht nur optionale, sondern standardmäßig aktivierte Ende-zu-Ende-Verschlüsselung, beispielsweise von WhatsApp, ermöglicht über einer Milliarde Menschen sicheren Informationsaustausch. Allerdings gilt es hierbei zu beachten, dass die zu Grunde liegende serverzentrische Architektur solcher Dienste Metadaten, wie beispielsweise wer mit wem wann kommuniziert, nicht schützt. Gleichzeitig gibt es jedoch auch Gegentrends: Der ursprünglich als P2P-Netzwerk organisierte VoIP-Dienst Skype wird aktuell in einen vollständig serverzentrischen Dienst umstrukturiert. Nicht nur dass Metadaten damit an einer zentralen Stelle zusammenlaufen, auch Ende-zu-Ende-Verschlüsselung wird in diesem Fall nicht eingesetzt. Zumindest für die serverzentrische Umsetzung eines solchen Dienstes gibt es jedoch auf Grund der Schwierigkeiten beim Aufbau direkter Verbindungen zwischen Endgeräten Argumente. Mittelfristig ist diesbezüglich vor allem mangels Verbreitung von IPv6 kaum Änderung in Sicht.

[22] Seit März 2017 wurde ZRTP im Rahmen der *Signal-App* gegen das Signal-Protokoll ersetzt. Die Signal-App ist ein sicherer Kommunikationsdienst, welcher von denselben Entwicklern wie das Signal-Protokoll angeboten wird. Die Weiterentwicklung der App und des Protokolls richten sich daher in einigen Punkten aneinander aus.

6. Literaturverzeichnis

[1] Cisco Systems, Inc., „The Zettabyte Era---Trends and Analysis," 06 2016. [Online]. Available: http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/VNI_Hyperconnectivity_WP.html. [Zugriff am 01 05 2017].

[2] J. Postel, „Internet Protocol," Internet Engineering Task Force Request for Comments, 09 1981. [Online]. Available: http://www.rfc-editor.org/rfc/rfc791.txt. [Zugriff am 15 03 2017].

[3] S. Deering und R. Hinden, „Internet Protocol," Internet Engineering Task Force Request for Comments, 12 1998. [Online]. Available: https://www.ietf.org/rfc/rfc2460.txt. [Zugriff am 16 03 2017].

[4] K. Egevang und P. Francis, „The IP Network Address Translator (NAT)," Internet Engineering Task Force Request for Comments, 05 1994. [Online]. Available: http://www.rfc-editor.org/rfc/rfc1631.txt. [Zugriff am 17 03 2017].

[5] R. Mahy, P. Matthews und J. Rosenberg, „Traversal Using Relays around NAT (TURN): Relay Extensions to Session Traversal Utilities for NAT (STUN)," Internet Engineering Task Force Request for Comments, 04 2010. [Online]. Available: http://www.rfc-editor.org/rfc/rfc5766.txt. [Zugriff am 17 03 2017].

[6] J. Rosenberg, R. Mahy, P. Matthews und D. Wing, „Session Traversal Utilities for NAT (STUN)," Internet Engineering Task Force Request for Comments, 10 2008. [Online]. Available: http://www.rfc-editor.org/rfc/rfc5389.txt. [Zugriff am 17 03 2017].

[7] J. Rosenberg, „Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal for Offer/Answer Protocols," Internet Engineering Task Force Request for Comments, 04 2010. [Online]. Available: http://www.rfc-editor.org/rfc/rfc5245.txt. [Zugriff am 17 03 2017].

[8] J. Oikarinen und D. Reed, „Internet Relay Chat Protocol," Internet Engineering Task Force Request for Comments, 05 1993. [Online]. Available: http://www.rfc-editor.org/rfc/rfc1459.txt. [Zugriff am 01 05 2017].

[9] P. Piccard, B. Baskin, G. Spillman und M. Sachs, „IRC Networks and Security," in *Securing IM and P2P Applications for the Enterprise*, Syngress, 2005.

[10] A. Majid, „RIP: MSN Web Messenger, July 22, 1999-June 30, 2009," 29 06 2009. [Online]. Available: http://www.merinews.com/article/rip-msn-web-messenger-july-22-1999-june-30-2009/15774411.shtml. [Zugriff am 01 05 2017].

[11] P. Saint-Andre, „Extensible Messaging and Presence Protocol (XMPP): Core - Architecture," Internet Engineering Task Force (IETF), 03 2011. [Online]. Available: https://tools.ietf.org/html/rfc6120#section-2. [Zugriff am 28 03 2017].

[12] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley und E. Schooler, „SIP: Session Initiation Protocol," Internet Engineering Task Force Request for Comments, 06 2002. [Online]. Available: http://www.rfc-editor.org/rfc/rfc3261.txt. [Zugriff am 01 05 2017].

[13] H. Schulzrinne, S. Casner, R. Frederick und V. Jacobson, „RTP: A Transport Protocol for Real-Time Applications," Internet Engineering Task Force Request for Comments, 07 2003. [Online]. Available: http://www.rfc-editor.org/rfc/rfc3550.txt. [Zugriff am 01 05 2017].

[14] A. Bergkvist, D. C. Burnett, C. Jennings, A. Narayanan und B. Aboba, „WebRTC 1.0: Real-time Communication Between Browsers," 17 03 2017. [Online]. Available: https://www.w3.org/TR/webrtc/. [Zugriff am 01 05 2017].

[15] X. Jin und S.-H. G. Chan, „Unstructured Peer-to-Peer Network Architectures," in *Handbook of Peer-to-Peer Networking*, New York, NY, USA, Springer Science+Business Media, LLC, 2010, p. 117.142.

[16] P. Piccard, B. Baskin, G. Spillman und M. Sachs, „Introduction to P2P," in *Securing IM and P2P Applications for the Enterprise*, Syngress, 2005, pp. 219-238.

[17] XMPP Standards Foundation (XSF), „Instant Messaging," [Online]. Available: https://xmpp.org/uses/instant-messaging.html. [Zugriff am 28 03 2017].

[18] J. F. Buford und H. Yu, „Peer-to-Peer Networking and Applications: Synopsis and Research Directions,“ in *Handbook of Peer-to-Peer Networking*, New York, NY, USA, Springer Science+Business Media, LLC, 2010, pp. 3-45.

[19] L. Liu und N. Antonopoulosu, Handbook of Peer-to-Peer Networking, New York, NY, USA: Springer Science+Business Media, LLC, 2010.

[20] D. Goodin, „Skype replaces P2P supernodes with Linux boxes hosted by Microsoft (updated),“ 01 05 2012. [Online]. Available: https://arstechnica.com/business/2012/05/skype-replaces-p2p-supernodes-with-linux-boxes-hosted-by-microsoft/. [Zugriff am 02 05 2017].

[21] P. Maymounkov und D. Mazières, „Kademlia: A Peer-to-Peer Information System Based on the XOR Metric,“ *Revised Papers from the First International Workshop on Peer-to-Peer Systems,* pp. 53-65, 2002.

[22] S. Nakamoto, „Bitcoin: A Peer-to-Peer Electronic Cash System,“ 2008. [Online]. Available: https://bitcoin.org/bitcoin.pdf. [Zugriff am 10 01 2015].

[23] R. Dingledine, N. Mathewson und P. Syverson, „Tor: The Second-Generation Onion Router,“ *Proceedings of the 13th USENIX Security Symposium,* 08 2004.

[24] B. Ramsdell und S. Turner, „Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 3.2 Message Specification,“ Internet Engineering Task Force Request for Comments, 01 2010. [Online]. Available: https://tools.ietf.org/html/rfc5751. [Zugriff am 02 05 2017].

[25] J. A. Buchmann, E. Karatsiolis und A. Wiesmaier, Introduction to Public Key Infrastructures, Springer, 2013.

[26] J. R. Douceur, „The Sybil Attack,“ in *Peer-to-Peer Systems*, Springer, 2002, pp. 251-260.

[27] N. Borisov, I. Goldberg und E. Brewer, „Off-the-record Communication, or, Why Not to Use PGP,“ *Proceedings of the 2004 ACM Workshop on Privacy in the Electronic Society,* pp. 77-84, 2004.

[28] K. Ermoshina, F. Musiani und H. Halpin, „End-to-End Encrypted Messaging Protocols: An Overview,“ *Internet Science: Third International Conference,* pp. 244-254, 09 2016.

[29] M. Baugher, D. McGrew, M. Naslund, E. Carrara und K. Norrman, „The Secure Real-time Transport Protocol (SRTP),“ Internet Engineering Task Force Request for Comments, 03 2004. [Online]. Available: https://tools.ietf.org/html/rfc3711. [Zugriff am 01 05 2017].

[30] P. Zimmermann und J. Callas, „ZRTP: Media Path Key Agreement for Unicast Secure RTP,“ Internet Engineering Task Force Request for Comments, 04 2011. [Online]. Available: https://tools.ietf.org/html/rfc6189. [Zugriff am 01 05 2017].