Transition to IPv6

Thomas Jakab
Betreuer: Andreas Müller
Seminar Future Internet SS2010
Lehrstuhl Netzarchitekturen und Netzdienste
Fakultät für Informatik, Technische Universität München
Email: jakab@in.tum.de

KURZFASSUNG

Das Internet Protokoll IPv4 ist die Grundlage des heutigen Internets. Allerdings existiert es schon seit den 80er-Jahren und zeigt heutzutage eine Vielzahl an Schwachstellen, die für eine Ablösung drängen. Das größte Problem ist die zunehmende Adressenknappheit, die eine schnelle Ablösung von IPv4 erzwingt. Dafür wurde das „Internet Protocol version 6“ (IPv6) entwickelt, das auf IPv4 aufbaut und den Adressraum vergrößert. Zur Einführung von IPv6 werden verschiedene Techniken benötigt, die eine Koexistenz und Interoperabilität der beiden Protokolle ermöglichen, da nur eine schrittweise Einführung von IPv6 möglich ist. Der Schwerpunkt dieser Arbeit liegt auf der Vorstellung dieser Techniken. Desweiteren wird gezeigt, weshalb die Einführung von IPv6 noch immer sehr schleppend voranläuft, obwohl alle dafür gebrauchten Technologien gut funktionieren und eine Ablösung sinnvoll und von Tag zu Tag nötiger wird.

SCHLÜSSELWORTE

IPv6, Dual Stack Architecture, Tunneling, 6to4, Teredo, 6rd, Translation, SIIT, TRT

1. EINLEITUNG

Mithilfe von IPv6 kann das Problem des versiegenden IPv4-Adresspools gelöst werden. Die meisten Menschen, die mit Informatik zu tun haben, haben schon von IPv6 gehört, die wenigsten aber damit zu tun gehabt. Deshalb wird diese Thematik in diesem Paper genauer erläutert. Zunächst wird aufgezeigt, weshalb IPv4 in Zukunft nicht mehr benutzt werden kann/soll, danach die für das Verständnis wichtigsten Punkte von IPv6 vorgestellt, wie etwa der Aufbau des Headers. Tiefer ins Detail gehende Information werden hier nicht erläutert, aber referenziert. Der Hauptteil der Arbeit beschäftigt sich mit den verschiedenen Einführungsmethoden von IPv6 sowie den Problemen, die eine großflächige Einführung bis heute verhindert haben. Die Einführungsmethoden lassen sich dabei in drei verschiedene, grundlegende Techniken, nämlich Dual Stack Architecture, Tunneling und Translation, unterscheiden. Jede diese Techniken ist notwendig, um die Einführung von IPv6 problemlos zu ermöglichen, da jede bei unterschiedlichen Szenarien angewandt werden muss. Tunneling wird z.B. verwendet, wenn zwei IPv6 Knoten miteinander (teilweise) über ein IPv4-Netz kommunizieren. Translation hingegen wird benötigt, wenn ein reiner IPv4-Knoten Pakete an einen reinen IPv6-Knoten sendet. Bei diesem Fall bietet Tunneling keine Lösung. Trotz dieser Vielfältigen Möglichkeiten, IPv6 problemlos einzuführen wird das neue Protokoll noch so gut wie gar nicht verwendet. Im letzten Teil der Arbeit wird auf die Gründe dafür, wie z.B. das ständige Verbessern und Anpassen von IPv4 an neue Bedürfnisse, eingegangen und Stellung zu diesem Problem genommen.

2. ABLÖSUNGSGRÜNDE FÜR IPV4

IPv4, standardisiert in [2], wurde Anfang der 80er Jahre für eine geringe Anzahl von Nutzern wie Universitäten oder das Militär entwickelt. Damals konnte das rasante Wachstum des Internets nicht vorhergesehen werden. Aus diesem Grund besitzt IPv4 mehrere Schwachstellen, die seine Tauglichkeit als Protokoll für das heutige Internet einschränken:

Das größte Problem IPv4s ist die Anzahl der zu vergebenden Adressen, die theoretisch 2^{32} = 4.294.967.296 beträgt. Dies liegt an dem 32-Bit Format, das IPv4 Adressen verwenden. Eine Adresse besteht aus 4 Blöcken, die jeweils 8 Bit repräsentieren, z.B. 20.135.255.3. Die Spannweite der erlaubten Dezimalzahlen pro Block liegt daher zwischen 0 und 255. Dieser Adressraum von ca. 4 Milliarden Adressen ist mittlerweile fast vollkommen ausgeschöpft. Mithilfe mathematischer Schätzverfahren wird die Erschöpfung des IPv4-Adresspools auf ca. Ende August 2012 datiert[1].

Desweiteren hat IPv4 einige Sicherheitslücken, so ist die Verwendung von IPSec[7], einem Sicherheitsprotokoll zur Authentifizierung und Verschlüsselung, nur optional. Ein weiterer Punkt ist, dass das in Zukunft immer wichtiger werdende Mobile IP, das mobilen Geräten (auch mit statischen IP-Adressen) den Wechsel zwischen Rechnernetzen ermöglicht, sehr schlecht unterstützt wird.

Der Aufbau des IPv4-Headers führt, wie in Kapitel 3.2.3 beschrieben, zu wenig performantem Routing, genauso wie die Implementierung von Fragmentierung(dem Zerlegen eines Paketes in mehrere kleinere).

3. IPV6

Bereits Mitte der 90er Jahre wurden die oben genannten Schwächen von IPv4 bekannt und man begann an einem Nachfolger zu arbeiten: IPv6 (gelegentlich auch IPnG = IP next Generation).1998 wurde IPv6 als neuer Internet Protokoll Standard festgelegt[3]

Beim Design wurde das Hauptaugenmerk auf die Beseitigung der oben genannten Probleme von IPv4 gelegt. So wurde unter Anderem die Adressenlänge auf 128 Bit vergrößert, der Header Aufbau deutlich vereinfacht, die Sicherheitsfunktionen verbessert und die Autokonfiguration von Knoten eingeführt.

3.1 IPv6 Adressen

Die wichtigste Änderung in IPv6 ist das neue Adressenformat:
Zur Behebung der Knappheit von IPv4-Adressen wurden IPv6-Adressen auf 128 Bit vergrößert. Dadurch wird die Anzahl der theoretisch möglichen Adressen auf 2^{128} vergrößert, dies entspricht ca. $3{,}4*10^{38}$, eine Zahl, die das Problem der Adressenknappheit voraussichtlich für immer lösen wird. Veranschaulichen lässt sich diese Menge an Adressen mit folgendem Beispiel: „Je Quadratmillimeter der Erdoberfläche stellt IPv6 667 Billiarden Adressen bereit.“[9] Um diesen 4mal längeren Adressentyp übersichtlich darstellen zu können, werden

IPv6 –Adressen mithilfe von Hexadezimalzahlen dargestellt, die zu 8 Vierergruppen angeordnet werden(siehe Tabelle 2). Jede dieser Gruppen repräsentiert 16 Bit und wird mit einem Doppelpunkt von den anderen Gruppen getrennt. Desweiteren existieren mehrere Vereinfachungsregeln, um die immer noch sehr langen Adressen zu verkürzen. Tabelle 2 gibt einen Überblick über diese Regeln samt Beispiel.

Tabelle 1. IPv6-Adressen und Vereinfachungsregeln

Regel	**Adresse**
BeispielPv6-Adresse	2001:0db8:85a3:0000:0000:8a2e:0000:7344
Führende Nullen dürfen weggelassen werden	2001:db8:85a3:**0**:**0**:8a2e:0:7344 (dabei muss aber jede Gruppe aus min. einer Zahl bestehen)
Aufeinander folgende Nullerblöcke dürfen einmal weggelassen werden	2001:db8:85a3**::**8a2e:0:7344 (dabei werden die weggelassenen Gruppen durch 2 Doppelpunkte repräsentiert)

Ähnlich zu IPv4 ist eine IPv6 Adresse auch in einen globalen und einen lokalen Teil aufgegliedert. Der globale Teil (Netzwerkteil) ist weltweit einzigartig und verweist auf ein Netzwerk, während der lokale Teil (Interface Identifier/Geräteteil) nur innerhalb dieses Netzwerkes eindeutig sein muss und z.B. ein Interface eines PCs identifiziert. Allerdings wird die Aufteilung nicht mehr mithilfe der Subnetzmaske vorgenommen, sondern einfach die Präfixlänge, also die Länge des Netzwerkteils, in Bits angegeben, z.B. würden bei folgender Adresse „2001:db8:85a3::8a2e:0:7344/32“ die ersten 32 Bit das Netzwerk identifizieren, der globale Teil also „2001:db8::/32 “, der lokale Teil „85a3::8a2e:0:7344“ lauten.

Die aus IPv4 bekannten **Broadcast**-Adressen, mit deren Hilfe alle Interfaces in einem Zielnetz angesprochen werden, existieren in IPv6 nicht mehr, da durch Broadcasts häufig Interfaces angesprochen werden, die die Nachricht (IP-Paket) nicht benötigen und deshalb verwerfen. Gebraucht wurden diese in IPv4 für das **„Address Resolution Protocol“** (ARP), mit dessen Hilfe in IPv4 die Link-Layer-Address zu einer IP-Adresse ermittelt wird. Statt dieser nicht performanten Nachrichtenübermittlung mit Broadcasts über ARP wird in IPv6 das **„Neighbour Discovery Protocol“** (NPD) benutzt. Dieses verwendet **Multicasts**, bei denen eine Nachricht an die Gruppe der Interfaces geschickt wird, die an dieser Multicast-Adresse registriert sind. Eine weitere Neuerung ist die Verwendung von **Anycast**-Adressen, an denen ebenfalls jeweils eine Gruppe von Interfaces registriert ist. Wird eine Nachricht an eine Anycast-Adresse geschickt, wird diese jedoch nur an ein Mitglied der Gruppe weitergeleitet. Die Auswahl des Empfängers wird von den Routern anhand der Routing-Protokolle getroffen. Multi- sowie Anycast-Adressen können nur als Empfänger und nie als Absender auftreten. Dieser muss immer eine **Unicast**-Adresse, die einem Interface zugeordnet ist, sein. Natürlich können Pakete auch an Unicast-Adressen gesendet werden, wobei ein PC mehrere Interfaces und somit auch mehrere Adressen haben kann.

IPv6 bietet außerdem die Möglichkeit der Autokonfiguration von Knoten, mit deren Hilfe Knoten anhand ihrer physikalischen MAC-Adresse selbst internetfähige IP-Adressen ableiten können und so automatisch eine Netzwerkverbindung aufbauen können.

Während der Entwicklung von IPv6 wurde diskutiert, ob die neuen Adressen aus 64 oder 128 Bit bestehen sollen. Einen Ausschlag für die größere Variante gab die Möglichkeit nun wieder „verschwenderisch“ mit den Adressen umgehen zu können wie zu Anfang der IPv4 Zeiten; d.h. man kann bestimmte Präfixe für besondere Zwecke reservieren. Einige dieser Präfixe sind für Übergangsmechanismen von IPv4 zu IPv6 reserviert und werden in Kapitel 4 vorgestellt. Eine komplette Liste aller Eigenschaften von IPv6-Adressen findet sich in [10] und [11].

3.2 IPv6 Header

Ein IP Paket besteht immer aus einem Header, in dem für die Übertragung wichtige Daten gespeichert werden, sowie der Payload, also den zu verschickenden Daten, die an den Header angehängt werden. Für IPv6 wurde der IPv4-Header überarbeitet und deutlich vereinfacht. Im Folgenden wird der IPv6-Header vorgestellt und seine Vorteile gegenüber der IPv4 Version werden erläutert.

3.2.1 IPv6 Header Aufbau

Der Header hat eine fixe Länge von 320 Bit, davon entfallen 256 auf Ziel- und Quelladresse. Die genaue Aufteilung ist Abbildung 1 zu entnehmen, wobei die Zahl in Klammern die Anzahl an Bits repräsentiert.

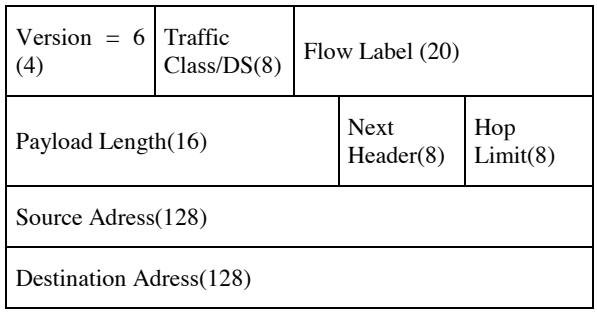

Abbildung 1. Aufbau des IPv6-Headers

Das **„Version“-**Feld ist immer mit der 6 besetzt und gibt an, dass es sich um ein IPv6 Paket handelt.

Das Feld **„Traffic Class/DS“** erlaubt es, Pakete beim Routing differenziert zu behandeln. Werte in diesem Feld werden von Anwendungen und Netzüberwachungsinstanzen gesetzt. Dieses Feld wurde aus dem IPv4 Header übernommen. Eine genaue Definition findet sich in [4].

Mithilfe des neu eingeführten **„Flow Label“** lassen sich Datenströme markieren und können so bevorzugt weitergeleitet werden, wodurch Streaming z.B. für Videokonferenzen deutlich effizienter funktioniert.

Die **„Payload Length“** gibt die Länge des Paketes abzüglich der Header-Länge an, also die Länge der „Nutzlast“. Die Länge eventuell verwendeter Erweiterungsheader(siehe 3.1.2) wird mitgezählt.

„Next Header“ enthält den Code, welcher den Typ des als nächstes folgenden Headers angibt. Dieser kann zum Beispiel ein Erweiterungsheader oder aber auch ein UDP oder TCP Header sein.

„Hop Limit“ bezeichnet die maximale Anzahl an Routern, die ein Paket durchlaufen darf, bis es sein Ziel erreicht. Pro durchlaufenen Router wird der Wert um 1 reduziert. Sollte der

Wert 0 werden und das Paket noch nicht am Ziel sein, wird es verworfen und eine Fehlernachricht an die **„Source Adress“** geschickt. Da es sich um ein 8 Bit Feld handelt, beträgt der maximale „Hop Wert“ 255.

3.2.2 Erweiterungs-Header

Ein zentrales Konzept in IPv6 sind die sogenannten „Erweiterungs-Header“. Mit ihrer Hilfe lassen sich zusätzliche Routing-Optionen, wie z.B. die Fragmentierung von Paketen oder Authentisierung leicht implementieren. Jeder Header-Typ besitzt einen eindeutigen Code. Wird ein Erweiterungs-Header verwendet, muss sein Code in dem „Next Header“-Feld des Basis-Headers eingetragen werden. Werden mehrere Erweiterungs-Header benutzt, dann steht der Code des als 2tes verwendeten Headers im „Next Header“-Feld des an erster Stelle stehenden usw. Dieses Konzept bietet zwei zentrale Vorteile:

Zum Einen müssen Router bei der Weiterleitung nicht alle Header auswerten, sondern können anhand der Codes die für sich Relevanten identifizieren. Dadurch läuft das Routing wesentlich effizienter ab als in IPv4, wo alle Router z.B. das „Options and optional Padding“-Feld auswerten müssen.

Zum Anderen können neue Option einfach mithilfe neuer Erweiterung-Header implementiert werden, wodurch IPv6 sehr flexibel und erweiterbar ist.

Die Reihenfolge, in der die verschiedenen Header verwendet werden, ist wie in Tabelle 2 festgelegt, um die Auswertung beim Routing weiter zu vereinfachen, aber auch aus Sicherheits-gründen. So müssen z.B. geheim zuhaltende Daten nach dem Verschlüsselungs-Header stehen, fürs Routing wichtige allerdings davor, da sie sonst eben verschlüsselt sind.

Tabelle 2. IPv6-Header Reihenfolge

Stelle	Header-Typ	Code
1	IPv6-Basis Header	-
2	Hop-by-Hop Options Header	0
3	Destination Options Header	60
4	Routing Header	43
5	Fragment Header	44
6	Authentication Header	51
7	Encapsulating Security Payload Header	50
8	Destination Options Header	60
9	Upper Layer Header	z.B. TCP = 6

Jeder Erweiterungs-Header besteht aus dem „Next-Header“-Feld und dem „Header Extension Length“-Feld sowie je nach Typ über weitere Felder. Die wichtigsten Header werden im Folgenden kurz beschrieben, eine ausführliche Beschreibung der Erweiterungs-Header findet sich in [5].

„Hop-by-Hop Options“ muss als einziger Erweiterungs-Header von jedem Router, den das Paket durchläuft, ausgewertet werden. Mit seiner Hilfe können Sonderbehandlungen des Paketes durch die Router festgelegt werden.

Der **„Destination Options Header“** kann als einziger Erweiterungs-Header zweimal vorkommen. Er enthält optionale Informationen, wie z.B. die Möglichkeit, extra große Daten zu verschicken. Der Header, der an dritter Stelle (siehe Tabelle 2) steht, wird von allen Knoten ausgewertet, die das Paket durchlaufen muss, der an achter Stelle nur vom Zielknoten.

Der **„Routing Header“** enthält eine Liste von Routern, die das Paket durchlaufen muss. Nur die in der Liste stehenden Router müssen den Header bearbeiten.

Fragmentierung soll in IPv6 aus Perfomance-Gründen vermieden werden. Ist sie dennoch vonnöten, werden die dazu erforderlichen Daten im **„Fragment Header“** gespeichert.

Mithilfe der **„Encapsulating Security Payload“** und **„Authentication“** Header können sichere Datenübertragungen realisiert werden. Mit Ersterem können Daten verschlüsselt werden, mit dem Anderen werden durch Authentisierung Angriffe unterbunden, bei denen sich der Angreifer als Empfänger oder Sender ausgibt. Weiterführende Informationen zu diesem Thema lassen sich in [6] finden.

3.2.3 Vorteile des IPv6 Headers

Um die Vorteile besser darstellen zu können, wird kurz der IPv4 Header vorgestellt. Eine ausführliche Beschreibung liefert [5] bzw. [2].

Version = 4	Header Length	Type of Service	IP Packet Length			
Packet Identification Number (ID)			0	DF	MF	Fragment Offset
Time to Live (TTL)		Protocol	Header Checksum			
Source Adress						
Destination Adress						
Options and optional Padding						

Abbildung 2. Aufbau des IPv4-Headers

Ein Problem von IPv4 ist der komplexe Aufbau des Headers, wodurch das Routing verlangsamt wird, sowie aufgrund des **„Options and optional Padding“**-Feldes die variable Länge des Headers, weshalb das Feld **„Header Length“** benötigt wird. In IPv6 ist mithilfe der Erweiterungs-Header die Header-Länge fix auf 320 Bit festgelegt – das Header Length Feld somit überflüssig.

Die gesamte 2. Zeile in Abbildung 2, die nur der Fragmentierung gilt, wurde ebenfalls entfernt und stattdessen der „Fragment Header“(siehe 3.2.2) eingeführt. Des Weiteren wurde das Fragmentierungs-Konzept aus Performancegründen in IPv6 stark überarbeitet: so kann nun, anders als in IPv4, nur noch der „Quell-Router“ Pakete fragmentieren, wodurch das Routing effizienter abläuft.

Das Feld **„Header Checksum“** wurde ebenfalls gestrichen, da „die Fehlererkennung der heutzutage verwendeten Link-Layer Protokolle so zuverlässig ist, dass man in der Netz-Schicht auf eine nochmalige Fehlersicherung durch Prüfsummen verzichten

kann“[8]. Die dadurch wegfallende Überprüfung der „Checksum“ führt natürlich auch zu einer Steigerung der Performance beim Routen.

4. EINFÜHRUNGSMETHODEN

Da es unmöglich ist das gesamte Internet auf einmal auf IPv6 umzustellen und es somit zu einer langsamen, schrittweisen Einführung kommt, werden die beiden Internetprotokolle für lange Zeit nebeneinander existieren und gezwungenermaßen auch miteinander kommunizieren müssen. So muss ein in einem IPv6-Netz befindlicher Knoten Pakete an ein IPv4-Netzknoten schicken können, oder zwei IPv6-Knoten müssen miteinander über ein IPv4-Netz Daten austauschen können und andersherum. Um diese Kompatibilität herzustellen wurden verschiedene Verfahren entwickelt, die sich in drei Kategorien einteilen lassen und deren Wichtigste im Folgenden vorgestellt werden(siehe auch [12]).

4.1 Dual Stack Architecture

Für Knoten, die häufig IPv4 sowie IPv6 Anfragen beantworten müssen, wie z.B. Server, ist die Implementierung der Dual Stack Architecture sinnvoll. Dank ihr kann sich der Knoten bei IPv4 Anfragen wie ein reiner IPv4 Knoten, bei IPv6 Anfragen wie ein reiner IPv6 Knoten verhalten. Um dies zu realisieren, muss der Knoten über eine IPv4- und eine IPv6-Adresse verfügen.

Sinnvoll ist dafür die Nutzung der **„IPv4-mapped-IPv6-Adresse“**(im Folgenden mit „mapped-Adresse“ abgekürzt). Für diesen Zweck wurde der Präfix „::ffff/96“ reserviert und die Mischschreibweise aus IPv6- und IPv4-Adressen eingeführt, z.B. „::ffff:128.2.33.200“. Natürlich kann diese Adresse auch als reine IPv6-Adresse geschrieben werden, in dem die Dezimalwerte ins Hexadezimalsystem umgerechnet und jeweils zwei Blöcke zusammengefasst werden: „::ffff:8002:21C8“. Mithilfe dieser Adresse können also IPv4-Adressen automatisch aus IPv6-Adressen abgeleitet werden und andersherum.

Diese Adressen werden benötigt, wenn z.B. ein IPv4-Client einen IPv6-Dienst an einem Dual Stack Server anfrägt. In diesem Fall muss seine IPv4-Adresse auf eine IPv6-Adresse abgebildet werden:

Empfängt ein Knoten mit Dual Stack Implementierung ein IPv4 Paket, bildet es die IPv4-Adresse auf eine mapped-Adresse ab und leitet es an die höher gelegene Protokollschicht weiter. Falls der Knoten eine Antwort an den Sender zurückschickt, erkennt er die mapped-Adresse und wandelt sie wieder in die IPv4-Adresse um.

Bei Empfang eines IPv6 Pakets verhält sich der Knoten wie ein reiner IPv6 Knoten.

Außerdem lassen sich die einzelnen Funktionen deaktivieren, so dass ein Dual Stack auch als reiner IPv4 bzw. IPv6 Knoten betrieben werden kann.

4.2 Tunneling

Es kann vorkommen, dass obwohl zwei IPv6-Knoten miteinander kommunizieren, ein IPv6-Paket über ein reines IPv4 Netz geroutet wird. In diesem Fall muss es in ein IPv4-Paket umgewandelt werden um ans Ziel zu kommen. Eine Möglichkeit dafür ist das „Tunneling“. Dabei wird das IPv6-Paket in ein IPv4-Paket eingewickelt und so über den „Tunnel“, das IPv4-Netz, übertragen. Wenn der Tunnel überbrückt wurde und das Paket sich wieder im IPv6-Netz befindet, wird der IPv4-Rahmen entfernt und das IPv6-Paket weitergeleitet. Dieser Vorgang geschieht für den Benutzer transparent, da beim Einpacken des Pakets das Hop-Limit Feld des IPv6-Headers um 1 verringert wird. Erst beim Weiterleiten nach dem Entpacken am Tunnelendpunkt wird das Hop-Limit wieder um 1 dekrementiert. Der Tunnel über beliebig viele IPv4-Knoten wird also als nur ein „Hop“ gezählt. Nachteile dieses Verfahrens sind der Performanceverlust durch das Ein- und Auspacken sowie die geringer werdende Kapazität für die Nutzlast, da der Overhead der Pakete vergrößert wird, wodurch es im ungünstigsten Fall zu Fragmentierung und einem weiteren Performanceverlust kommen kann. „Tunneling“ kann entweder manuell oder automatisch geschehen. Bei der manuellen Variante müssen der Tunnelstart- und Tunnelendpunkt fest konfiguriert werden, wodurch ein hoher Aufwand entsteht. Allerdings kann dies aus Sicherheitsaspekten sinnvoll sein. Als Endpunkt kann eine Anycast-Adresse benutzt werden, wodurch von mehreren Routern nur einer zum Empfang gewählt wird. Dadurch erhöht sich die Ausfallsicherheit. Um automatisches „Tunneling“ zu implementieren gibt es verschiedene Varianten, von denen die wichtigsten in diesem Abschnitt erläutert werden.

4.2.1 6to4

6to4 ist die weltweit am häufigsten benutzte Technik um IPv6 Konnektivität herzustellen[13]. Um 6to4 nutzen zu können, muss eine global eindeutige IPv4-Adresse vorhanden sein. Die Kommunikation erfolgt über spezielle 6to4-Router.

Für 6to4 wurde ähnlich zu „IPv4-mapped-IPv6“ ein Adress-Präfix reserviert. Dieser hat die Form „2002::/16“. Der globale Teil einer 6to4-Adresse setzt sich aus diesem Präfix und der Hexadezimalrepräsentation der IPv4-Adresse des 6to4 Routers zusammen; z.B. „2002:8002:21C8::/48“ bei der IPv4-Adresse „128.2.33.200“. Dadurch wird jeder IPv4 Adresse und damit auch jedem 6to4 Router ein eigenes /48 Netz zugeordnet.

Innerhalb eines 6to4-Netzes können Knoten problemlos miteinander kommunizieren(siehe Host A und Host B in Abbildung 3). Kommt es zu einer Kommunikation mit einem Knoten in einem anderen 6to4-Netz(Host A mit Host C), packt der sendende 6to4 Router(R1) das IPv6-Paket in ein IPv4 Paket ein und sendet es an den empfangenden 6to4 Router(R2). Die IPv4-Adresse des empfangenden Routers kann er dabei aus der IPv6-Adresse des Adressaten(Host C) ableiten. Zusätzlich wird die Protokollnummer im Header des IPv4-Paketes auf 41 gesetzt, woran der empfangende Router erkennt, dass es sich um ein getunneltes IPv6-Paket handelt, entpackt es und leitet es weiter. Auch hier kann an eine Anycast-Adresse gesendet werden, wodurch der nächstgelegene 6to4 Router ausgewählt wird und nicht ein Bestimmter adressiert werden muss. Sie ist in [14] festgelegt und lautet „2002:c058:6301::“.

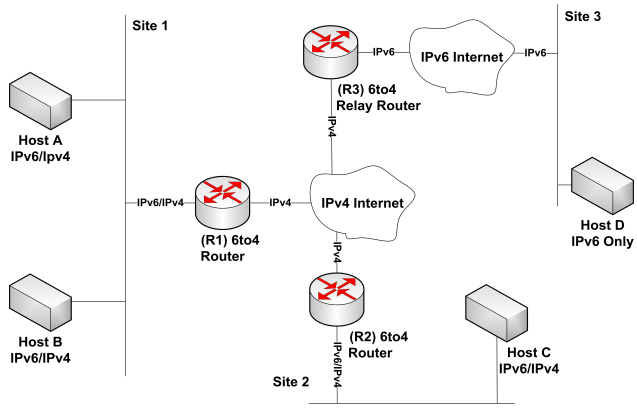

Abbildung 3. Möglichkeiten der Kommunikation von 6to4 Komponenten. In Anlehnung an [15]

Wird ein Knoten in einem reinen IPv6-Netz adressiert(Host A an Host D), muss ein 6to4 Relay Router dazwischengeschaltet

werden, da die IP-Adresse des Empfängers(Host D) nicht mehr wie vorher aus der Adresse des Routers abgeleitet wird.
Der Nachteil dieser Technik ist die Notwendigkeit der Benutzung spezieller Router, und dass 6to4 nicht kompatibel zu NAT (Network Address Translation) ist, da unter anderem die Protokollnummer 41 von NATs standardmäßig nicht verarbeitet werden kann und bei der Verwendung von NATs nicht immer eine global eindeutige IPv4-Adresse existiert.[16]

4.2.2 Teredo

Um bei der Verwendung von NAT nicht auf „Tunneling“ verzichten zu müssen entwickelte Microsoft Teredo[17]. Teredo ist als Übergangslösung bestimmt, bis 6to4 auch mit NAT funktioniert. Da Teredo einen großen Overhead erzeugt, nicht so performant ist wie andere Techniken und Sicherheitsrisiken aufweist, ist, falls möglich, eine andere Art zu tunneln zu verwenden.

Bei Teredo werden die IPv6-Pakete als UDP-Nutzlast in einem IPv4-Paket übertragen. Sie werden also in UDP-Pakete anstatt in IPv4-Pakete eingekapselt, damit der NAT Paketfilter sie nicht zurückweist.

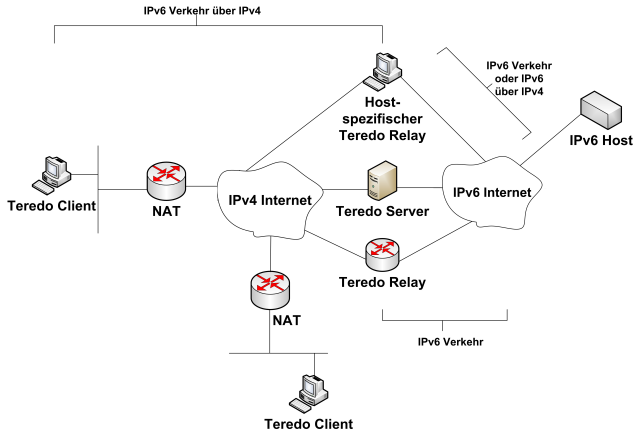

Abbildung 4. Teredo Komponenten und Zusammenspiel. In Anlehnung an [18]

Der **Teredo Client** (siehe Abbildung 4) ist ein IPv4-Knoten, der Zugang zu IPv6 Internet will. Dies geschieht mithilfe des **Teredo Servers** und des **Teredo Relays**. Der Server hat mittels einer globalen IPv4-Adresse Zugang zum IPv4-Internet und kann so dem Client eine spezielle IPv6-Adresse zuordnen, die **Teredo Address**. Sie setzt sich aus dem Teredopräfix „2001::/32“, der IPv4-Adresse des Servers und der des Clients sowie weiteren Feldern zusammen. Das **Teredo Relay** ist ein Router, der IPv6-Pakete an die Clients weiterleiten kann. Ein **Hostspezifischer-Teredo-Relay** kann über IPv4 und IPv6 kommunizieren und somit über IPv4 direkt mit dem Client in Verbindung treten – er benötigt deshalb kein zwischengeschaltetes Teredo-Relay wie der Server.

4.2.3 6rd(6 Rapid Deployment)

Eine relativ junge Technik ist die von Rémi Després entwickelte und im Januar 2010 in [19] veröffentlichte 6rd Tunnelmethode, die auf 6to4 aufbaut, aber versucht deren Schwächen zu beseitigen: So wird anstatt des festen 6to4 Präfixes ein pro ISP(Internet Service Provider) eindeutiges Präfix genutzt. Der Router des Benutzers kapselt bei Anschluss an ein IPv4-Netz die 6rd-Adresse dann in eine IPv4-Adresse und leitet das Paket an eine dem ISP gehörende Gateway, die das IPv6-Paket wieder entpackt. Dadurch ist sichergestellt, dass der ISP nur zu ihm gehörende Pakete weiterleiten muss, anders als bei 6to4, wo aufgrund des einheitlichen Präfixes nicht kontrolliert werden kann, woher der Traffic kommt. Es sind wie bei 6to4 auch Verbindungen von 6rd-Netz zu 6rd-Netz über ein IPv4 Netz möglich; dazu ist keine extra Gateway von Nöten.

Der globale Teil von 6rd-Adressen besteht aus dem maximal 32 Bit großen 6rd-Präfix des ISP sowie der IPv4-Adresse des Kunden. Der französische ISP „free“ hat als erstes 6rd seinen Kunden angeboten und dafür den Adresspräfix „2a01:0e00:::/26” erhalten[20].

4.3 Translation

Eine weitere Möglichkeit der Interoperabilität von IPv4 und IPv6 wird durch Übersetzen der Protokolle erreicht. Diese wird gebraucht, wenn Knoten, die sich in einer reinen IPv4-Umgebung befinden mit Knoten kommunizieren müssen, die in einer reinen IPv6-Umgebung stehen oder andersherum. Dann ist nämlich kein „Tunneling“ möglich. Zwei Beispiele dafür sind die hier vorgestellten SIIT(Stateless IP/ICMP Translation) und TRT (Transport Relay Translator).

4.3.1 SIIT

Bei SIIT-Kommunikation werden ausgehende IPv6-Pakete von einem speziellen SIIT-Router, über den der IPv6 Knoten mit dem IPv4-Netz kommuniziert, in IPv4, sowie eingehende Pakete aus dem IPv4-Netz in IPv6 übersetzt. Dazu wird eine spezielle IPv6-Adresse benötigt, die „**IPv4-translated IPv6-Address**“. Sie ist aus dem Präfix „0:0:0:0:FFFF::/96“ und einer IPv4-Adresse zusammengesetzt. Diese Adresse muss für die Dauer der Kommunikation dem IPv6-Knoten fest zugeteilt werden.

Die Felder der Header lassen sich ohne Schwierigkeiten automatisch erzeugen bzw. werden verworfen(wie die IPv4 Optionen), allerdings lassen sich viele IPv6-Erweiterungsheader nicht übersetzen(z.B. der Authentification-Header). Des Weiteren müssen noch zusätzlich Protokolle, wie ICMP, übersetzt werden. Eine genaue Beschreibung der Übersetzungsvorgänge findet sich in [21].

4.3.2 TRT

TRT wird eingesetzt um auf IPv6 basierende TCP/UDP-Anwendungen mit auf IPv4 basierenden zu verbinden. Dies funktioniert nur bei bidirektionaler Kommunikation unter Verwendung eines besonderen DNS-Servers.

Zum Aufbau der Kommunikation bekommt der IPv6-Knoten(C6 in Abbildung 5) über den DNS-Server eine von diesem speziell erstellte IPv6-Adresse, deren letzte 32 Bit aus der IPv4-Adresse des IPv4-Knotens(S4) bestehen, übermittelt. Die Kommunikation der beiden Knoten läuft nun über das TRT, welches aus den IPv6-Paketen IPv4-Pakete macht und an die Adresse des Knotens(S4) schickt. In die andere Richtung funktioniert die Kommunikation nach demselben Prinzip. TRT ist in [22] spezifiziert.

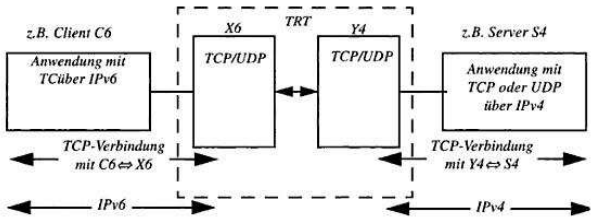

Abbildung 5. Funktionsweise von TRT[23]

5. STELLUNGNAHME ZU EINFÜHRUNGSPROBLEMEN

Trotz der in Kapitel 3 genannten Vorteile von IPv6, seiner schon relativ langen Existenz sowie der zunehmenden Knappheit von IPv4-Adressen wird das neue Protokoll noch so gut wie gar nicht genutzt. Laut eines Reports von Google[13] von 2008 haben gerade einmal 0,238% der Internetnutzer weltweit IPv6 Konnektivität und bevorzugen diese auch. Ein Grund dafür dürfte die typische Mentalität „wozu soll ich mein System ändern, wenn es doch funktioniert“ sein. Ein Weiterer, dass die meisten Benutzer wenig Ahnung haben, wie das Internet überhaupt funktioniert und sich deshalb keine Gedanken über diese Problematik machen. Desweiteren bieten die meisten ISPs keine IPv6-Konnektivität an, da sie keine Nachfrage verspüren und deshalb die Investitionen scheuen. Außerdem spielt noch eine Rolle, dass IPv4 ständig weiter verbessert wird, um das Ende des Protokolls möglichst lange hinauszuzögern: So wurde z.B. NAT eingeführt, um die Adressenknappheit hinauszuzögern, Sicherheitsbedenken wurden mit der nachträglichen Integration von IPSec umgangen und heutzutage wichtige Funktionen wie Mobilität wurden ebenfalls nachträglich (suboptimal) ermöglicht. Es wird also immer wieder repariert und erweitert, anstatt das neue Protokoll IPv6 zu verwenden, welches all diese Probleme vorerst lösen kann.

Dies führt wohl zu einer Art Teufelskreis, der erst durchbrochen werden kann, wenn die ISPs kurz davor sind, keine IPv4-Adressen mehr zu bekommen. Ob die Umstellung dann mithilfe von schnell zu implementierenden Einführungsmethoden wie 6rd(der französische ISP „free“ hat sein Netz innerhalb von 5 Wochen dank 6rd zu IPv6 erweitert) rechtzeitig zu schaffen ist, bleibt abzuwarten, scheint aber wahrscheinlich. Dennoch befinden sich die ISPs in einer Bringschuld: Umstellen werden sie sowieso irgendwann müssen und solange kein Angebot existiert wird die Nachfrage auch nicht steigen. „free“ und einige Andere sind hier schon mit gutem Beispiel vorangegangen.

6. ZUSAMMENFASSUNG

Bei der Entwicklung von IPv6 wurde darauf geachtet, aus den Problemen von IPv4 zu lernen und ein flexibles, leicht erweiterbares Protokoll zu schaffen. Erreicht wurde dies vor allem durch die enorme Vergrößerung des Adressraums auf 128 Bit und der damit möglichen Klassifizierung von IP-Adressen, der Fähigkeit zur Autokonfiguration und dem Ersetzen von Broadcast durch Multi und Anycast. Desweiteren spielt auch die Verbesserung des Headers eine große Rolle: So wurden Erweiterungs-Header eingeführt, überflüssige Felder entfernt und dadurch das Routing performanter gestaltet.

Da keine „Big-Bang“-Ablösung von IPv4 zu bewerkstelligen ist, wurden verschiedene Einführungsstrategien entwickelt, die sich in die drei Gruppen „Dual Stack Architecture“, „Tunneling“ und „Translation“ einteilen lassen. Jede dieser Gruppen ist für bestimmte Einsatzgebiete gut geeignet, wobei bei den „Tunneling“ Techniken vor allem der neue Ansatz 6rd sehr vielversprechend ist, da die auf 6to4 basierende Methode relativ leicht, schnell und günstig zu implementieren ist und deshalb die bis jetzt sehr langsam stattfindende Einführung von IPv6 beschleunigen könnte. Wichtige „Translation“-Techniken sind SIIT und TRT, bei denen die Protokolle automatisch in das jeweils andere Format übersetzt werden.

Gründe für die schleppende Einführung sind die noch geringe Nachfrage der Nutzer, aber auch das fehlende Anbieten von IPv6 durch die Internet Service Provider. Diese voneinander abhängigen Gründe werden die Einführung jedoch nicht aufhalten können, da bald keine IPv4-Adressen mehr vergeben werden können. Die Techniken und das Know-How sind vorhanden, jetzt muss nur noch gehandelt werden.

7. REFERENZEN

[1] Husten,Geoff, „IPv4 Adress Report“,11 May 2009. http://www.potaroo.net/tools/ipv4

[2] Information Science Institute USC, „Internet Protocol“, RFC 791, IETF, September 1981.

[3] S. Deering, R. Hinden, „IPv6“, rfc 2460, IETF, December 1998.

[4] K. Nichols, S. Blake, F. Baker, D. Black, „Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers“, RFC 2474, IETF, December 1998.

[5] Herbert Wiese, „Das neue Internetprotokoll IPv6“, 2002, Carl Hanser Verlag München Wien, ISBN 3-446-21685-5, Seiten 28-42.

[6] Silvia Hagen, „IPv6 Essentials“, 2006, O’Reilly Media, Inc., ISBN 978-0-596-10058-2, Chapter 2

[7] S. Kent, K. Seo, „Security Architecture for the Internet Protocol“, RFC 4301, IETF, December 2005

[8] Herbert Wiese, „Das neue Internetprotokoll IPv6“, 2002, Carl Hanser Verlag München Wien, ISBN 3-446-21685-5, Seite 27.

[9] Hans Peter Dittler, „IPv6-das neue Internet-Protokoll“, 1998, dpunkt-Verl., ISBN 3-932588-18-5,Seite 36 a

[10] Hans Peter Dittler, „IPv6-das neue Internet-Protokoll“, 1998, dpunkt-Verl., ISBN 3-932588-18-5,Kapitel 3

[11] R. Hinden, S. Deering, „IP Version 6 Addressing Architecture“,RFC 4291, IETF, February 2006

[12] R. Gilligan, E. Nordmark, „ Transition Mechanisms for IPv6 Hosts and Routers“, RFC 2893, IETF, August 2000

[13] Steinar H. Gunderson, „Global IPv6 statistics- Measuring the current state of IPv6 for ordinary users“,RIPE Meeting 57, October 2008

[14] C. Huitema, „An Anycast Prefix for 6to4 Relay Routers“, RFC 3068,IETF,June 2001

[15] Silvia Hagen, „IPv6 Essentials“, 2006, O’Reilly Media, Inc., ISBN 978-0-596-10058-2, Page 265

[16] B. Carpenter, K. Moore, „Connection of IPv6 Domains via IPv4 Clouds“, RFC 3056,IETF February 2001

[17] C. Huitema, „Teredo: Tunneling IPv6 over UDP through Network Address Translations (NATs)“, RFC 4380,IETF, February 2006

[18] Microsoft TechNet, „Überblick zu Teredo“, 05.März.2004, http://www.microsoft.com/germany/technet/datenbank/articles/600330.mspx

[19] R. Despres, „IPv6 Rapid Deployment on IPv4 Infrastructures (6rd)“, RFC 5569,IETF, January 2010

[20] A. Classen, „IPv6@free-native IPv6 to the User“, RIPE Meeting 58, 05.05.2009

[21] E. Nordmark, „Stateless IP/ICMP Translation Algorithm (SIIT)“, RFC 2765,IETF, February 2000

[22] J. Hagino, K. Yamamoto, „ An IPv6-to-IPv4 Transport Relay Translator“, RFC 3142,IETF, June 2001

[23] Herbert Wiese, „Das neue Internetprotokoll IPv6“, 2002, Carl Hanser Verlag München Wien, ISBN 3-446-21685-5, Seite 298