

Studienarbeit zu IPv6

Arne P. Böttger
Fachhochschule Wedel (<http://www.fh-wedel.de>)
mi4053@stud.fh-wedel.de

Studienarbeit zu IPv6

von Arne P. Böttger

Inhaltsverzeichnis

Vorwort	i
1. Grundlegende Arbeitsweise	1
1.1. Addressierung	1
1.1.1. Schreibweise	1
1.1.2. Typen	2
1.1.3. Strukturierung des Adressraums (Adresstypinformation)	2
1.1.4. Innere Struktur von Unicast-Adressen	??
1.1.5. Anycast-Adressen	??
1.1.6. Multicast-Adressen	??
1.2. Headerformat	??
1.2.1. Standard-Header	??
1.2.2. Erweiterungs-Header	??
1.3. Autoconfiguration	??
1.4. Namensauflösung	??
1.5. Routing	9
1.6. Unterstützung mobiler Hosts	??
2. Bestehende IPv6-Testnetze und Anbindungsmöglichkeiten	??
2.1. 6bone	??
2.2. freenet6	??
3. Offizielle Unterstützung auf Betriebssystem-Seite	??
3.1. Linux	??
3.2. Microsoft Windows	??
3.3. MacOS 9/X	12
3.4. OpenBSD	??
4. IPv6-Unterstützung durch Server-Programme	13
4.1. Name-Server	14
4.1.1. ISC Bind	??
4.1.2. Microsoft Nameserver	14
4.2. Web-Server	14
4.2.1. Apache Webserver	??
4.2.2. Microsoft IIS	14
4.3. Datei-Dienste	15
4.3.1. Samba	??
4.3.2. Microsoft CIFS	15
4.3.3. NFS	15
4.4. SMTP (eMail)	15
4.4.1. Sendmail	??
4.4.2. Exim	15
4.4.3. Postfix	15
4.5. FTP	16
4.5.1. ProFTPd	??
4.5.2. WuFTP	16
4.5.3. vsFTP	16

5. Test der Fähigkeiten in einem Testaufbau	16
5.1. Debian GNU/Linux 3.0 (Woody).....	??
5.1.1. Zustand nach der Installation.....	??
5.1.2. Schritte zu einem IPv6-fähigen Netzwerkstack	17
5.1.3. Konfiguration von statischen Site-Local Adressen.....	18
5.1.4. Bereits vorhandene IPv6-fähige Dienste	20
5.1.5. Bereits vorhandene IPv6-fähige Clients	21
5.1.6. Schritte zu IPv6-fähigen Diensten.....	21
5.1.7. Schritte zu IPv6-fähigen Clients.....	23
5.2. Microsoft Windows 2000	??
5.2.1. Zustand nach der Installation.....	??
5.2.2. Schritte zu einem IPv6-fähigen Netzwerkstack	24
5.2.3. Konfiguration von statischen Site-Local Adressen.....	??
5.2.4. Bereits vorhandene IPv6-fähige Dienste	??
5.2.5. Bereits vorhandene IPv6-fähige Clients	28
5.2.6. Schritte zu weiteren IPv6-fähigen Diensten und Clients.....	??
5.3. Microsoft Windows XP.....	29
5.3.1. Zustand nach der Installation.....	??
5.3.2. Schritte zu einem IPv6-fähigen Netzwerkstack	29
5.3.3. Konfiguration von statischen Site-Local Adressen.....	??
5.3.4. Bereits vorhandene IPv6-fähige Dienste	??
5.3.5. Bereits vorhandene IPv6-fähige Clients	32
5.3.6. Schritte zu weiteren IPv6-fähigen Diensten und Clients.....	??
5.4. Unterstützung durch Cisco-Router.....	33
5.4.1. Voruntersuchung	??
5.4.2. Testaufbau.....	33
6. Integration von IPv6 in bestehende Infrastrukturen	36
6.1. 6to4 nach rfc3056.....	??
6.2. IPv6-to-IPv4 nach rfc3142.....	??
6.3. NAT-PT nach rfc2766	42
6.4. Kommunikation zwischen IPv6-Rechnern über IPv4 (IPv4-compatible IPv6)	42
6.5. Verbindung zwischen IPv4-Rechnern und IPv6-Rechnern (IPv4-mapped IPv6).....	??
7. Fazit.....	??
A. Quellen	45

Tabellenverzeichnis

1-1. Beispiele für IPv6-Adressen.....	1
1-2. Beispiele gemischte Schreibweise.....	1
1-3. Präfix-Notation	1
1-4. IPv6-Adresse kombiniert mit Präfix.....	2
1-5. Format-Präfix-Allokation	2
1-6. Aggregierbare globale Unicast-Adressen.....	??
1-7. Aufbau von Multicast-Adressen.....	5
1-8. Gültigkeitsbereiche von Multicast-Adressen	??
1-9. Beispiel für globale Multicast-Adressen	??
A-1. RFCs zu IPv6.....	??

Abbildungsverzeichnis

1-1. IPv6 Standard-Header	7
5-1. Interface-Konfiguration mit IPv6-fähigem Kernel	??
5-2. ping6 per Link-Local-Adresse.....	??
5-3. Site-Local Adresse in /etc/network/interfaces.....	??
5-4. Ausgabe von ifconfig nach manueller Vergabe der Site-Local-Adresse	??
5-5. ping6 auf benachbarte Rechner per Site-Local-Adresse	??
5-6. Auszug aus /etc/inetd.conf für IPv6 echo und telnet.....	??
5-7. Einträge in named.conf zur Auflösung von Site-Local-Adressen	??
5-8. Zonen-Datei db.ipv6 für die Domain ipv6.local	21
5-9. Zonen-Datei db.int für die Auflösung von Adressen zu Namen	22
5-10. Ausgabe von netstat -an (gefiltert)	??
5-11. Zugriff mit Mozilla auf den lokalen Webserver	23
5-12. Hinzufügen des IPv6-Protokolls zu einer Netzwerkkarte	24
5-13. Ausgabe von "ipv6 if" nach Installation.....	??
5-14. Test der Installation per ping6	??
5-15. Konfiguration und Test von Site-Local-Adressen	??
5-16. Zugriff auf Apache-Webserver per Link-Local-Adresse.....	??
5-17. Zugriff auf Apache-Webserver per Site-Local-Adresse	??
5-18. Ausgabe von "ipv6 if" nach Installation.....	??
5-19. Test der Installation per ping6	??
5-20. Konfiguration von Site-Local-Adressen.....	31
5-21. Zugriff auf Apache-Webserver per Site-Local-Adresse	??
5-22. Testaufbau IPv6-Routing.....	33
5-23. Cisco-Konfiguration für Site-Local-Adressen.....	??
5-24. Kontrolle der IPv6-Verbindung über den Cisco-Router - WinXP-Seite	??
5-25. Kontrolle der IPv6-Verbindung über den Cisco-Router - Linux-Seite.....	??
6-1. Testaufbau 6to4	??
6-2. 6to4-Konfiguration c1605a.....	??
6-3. 6to4-Konfiguration c2611b	??
6-4. Interface-Konfiguration NWS-04	??
6-5. Interface-Konfiguration NWS-03	41
6-6. Verbindungstest von NWS-04 zu NWS-03	41

6-7. Verbindungstest von NWS-03 zu NWS-04	41
6-8. Aktivierung des Tunnel-Devices sit0	43

Vorwort

In dieser Studienarbeit werde ich untersuchen, in wie weit das neue Internet-Protokoll IPv6 bereits einsatzfähig ist. Da es sich noch um ein relativ junges Protokoll handelt, kann ich kein tiefergehendes Wissen voraussetzen und beginne daher mit einer Einführung, was IPv6 ist und wie es funktionieren soll.

Im Anschluss daran prüfe ich einige verbreitete Betriebssysteme auf ihre Unterstützung. Ausserdem prüfe ich mit Hilfe von Cisco-Routern, welche Möglichkeiten es gibt, um den Übergang von IPv4 zu IPv6 zu schaffen. Denn im Gegensatz zu früheren Protokolländerungen in den Anfangstagen des Internet (z.B. vom ICP zu TCP) wird es eine langsame Umstellung sein, die sich über Jahre, wenn nicht Jahrzehnte hinziehen wird.

Kapitel 1. Grundlegende Arbeitsweise

1.1. Adressierung

1.1.1. Schreibweise

Eine IPv6-Adresse besteht aus einer 128 Bit grossen Zahl. Diese Zahl wird in acht Gruppen zu je 16 Bit aufgeteilt, die dann in hexadezimaler Zahldarstellung durch Doppelpunkte getrennt aufgeschrieben werden. Führende Nullen können hierbei weggelassen werden. Beispiele sind hier die Adressen FEDC:BA98:7654:3210:FEDC:BA98:7654:3210 oder 1080:0:0:0:8:800:200C:417A.

Da es aufgrund der Adressverteilung zu langen Folgen von Nullbits kommen kann, ist es erlaubt, einmal in einer Adresse eine beliebig lange Folge von Nullen durch zwei Doppelpunkte abzukürzen. Dies kann auch am Anfang oder Ende einer Adresse sein.

Tabelle 1-1. Beispiele für IPv6-Adressen

Beschreibung	Langform	Kurzform
eine Unicast-Adresse	1080:0:0:0:8:800:200C:417A	1080::8:800:200C:417A
eine Multicast-Adresse	FF01:0:0:0:0:0:101	FF01::101
die Loopback-Adresse	0:0:0:0:0:0:1	::1
die un spezifizierte Adresse	0:0:0:0:0:0:0	::

Für den Fall, dass IPv4- und IPv6-Netze gemischt eingesetzt werden, kann es zweckmässig sein, die zwei niederwertigsten 16-Bit-Teile in vier 8-Bit-Teile aufzuspalten und in IPv4-Schreibweise an die höherwertigen sechs Teile anzuhängen. So muss nicht jede verwendete IPv4-Adresse in ihre hexadezimale Darstellung umgerechnet werden.

Tabelle 1-2. Beispiele gemischte Schreibweise

Langform	Kurzform
0:0:0:0:0:0:13.1.68.3	::13.1.68.3
0:0:0:0:FFFF:129.144.52.38	::FFFF:129.144.52.38

Um ganze Netzbereiche zu benennen wird eine Schreibweise verwendet, die der Classless Inter-Domain Routing Notation (CIDR) von IPv4 entspricht, indem die Adresse von einem Schrägstrich und der Prefix-Länge in Bits gefolgt wird.

Tabelle 1-3. Präfix-Notation

Präfix 12AB00000000CD30
12AB:0000:0000:CD30:0000:0000:0000:0000/60
12AB::CD30:0:0:0/60
12AB:0:0:CD30::/60

Soll sowohl eine IPv6-Adresse als auch der zugehörige Präfix notiert werden, so lassen sich die Schreibweisen kombinieren.

Tabelle 1-4. IPv6-Adresse kombiniert mit Präfix

getrennt	kombiniert
12AB:0:0:CD30:123:4567:89AB:CDEF	12AB:0:0:CD30:123:4567:89AB:CDEF/60
12AB:0:0:CD30::/60	

1.1.2. Typen

Die Adressen in IPv6 sind in unterschiedliche Typklassen aufgeteilt, die sich nicht in der Form, sondern nur in der vorgesehenen Verwendung unterscheiden. Das sind Unicast-Adressen, die jeweils einem einzelnen IPv6-Host zugeordnet sind, Anycast-Adressen, mit denen jeweils der am nächsten gelegene Host einer bestimmten Gruppe angesprochen wird und Multicast-Adressen, mit deren Hilfe Pakete an alle Mitglieder einer Gruppe versandt werden können.

Es gibt keine Broadcast-Adressen mehr. Diese Funktion erfüllen für IPv6 die Multicast-Adressen. Eine Adresse ist immer einem Netzwerkknoten zugeordnet, nicht einem Netzwerkknoten (Node). Hat ein Node mehrere Interfaces, so kann er über die Adresse eines beliebigen Interfaces angesprochen werden.

1.1.3. Strukturierung des Adressraums (Adresstypinformation)

Der Typ einer Adresse ist an den führenden Bits zu erkennen. Diese Bits, deren Anzahl bei den unterschiedlichen Typen variieren kann, heissen Format-Präfix.

Tabelle 1-5. Format-Präfix-Allokation

Verwendung	Präfix (binaer)	Anteil am Adressraum
reserviert	0000 0000	1/256
nicht zugeteilt	0000 0001	1/256

Verwendung	Präfix (binaer)	Anteil am Adressraum
reserviert für NSAP	0000 001	1/128
reserviert für IPX	0000 010	1/128
nicht zugeteilt	0000 011	1/128
nicht zugeteilt	0000 1	1/32
nicht zugeteilt	0001	1/16
aggregierbare globale Unicast-Adressen	001	1/8
nicht zugeteilt	010	1/8
nicht zugeteilt	011	1/8
nicht zugeteilt	100	1/8
nicht zugeteilt	101	1/8
nicht zugeteilt	110	1/8
nicht zugeteilt	1110	1/16
nicht zugeteilt	1111 0	1/32
nicht zugeteilt	1111 10	1/64
nicht zugeteilt	1111 110	1/128
nicht zugeteilt	1111 1110 0	1/512
Link-Lokale Unicast-Adressen	1111 1110 10	1/1024
Site-Lokale Unicast-Adressen	1111 1110 11	1/1024
Multicast-Adressen	1111 1111	1/512

Die "unspezifizierte Adresse", die Loopback-Adresse und die Einbettung von IPv4 in IPv6 befinden sich unterhalb des reservierten Adressraums 0000 0000.

Diese Allokation erlaubt es, sofort Adressen für globale, lokale und Multicast-Zwecke zu reservieren. Auch für den Übergang von NSAP (Network Service Access Point, der Adressierung der ISO-Protokollsuite) und IPX sind Adressbereiche reserviert. Dadurch, dass grosse Bereiche noch keiner Verwendung zugewiesen sind, ist Raum für zukünftige Erweiterungen vorhanden.

Multicast-Adressen lassen sich durch den Format-Präfix FF, also 1111 1111, von allen anderen Adressen unterscheiden. Anycast-Adressen stellen eine Untermenge der Unicast-Adressen dar.

1.1.4. Innere Struktur von Unicast-Adressen

IPv6-Unicast-Adressen lassen sich ähnlich dem CIDR bei IPv4 zusammenfassen. Dies gilt für die unterschiedlichen Kategorien von Unicast-Adressen ebenso wie für den NSAP- und den IPX-Adressraum. Ein einzelner IPv6-Host muss keinerlei Kenntnis über den Aufbau der verwendeten Unicast-Adresse haben, es ist allerdings denkbar, dass komplexere Implementierungen den Subnet-Präfix

verwenden. Router werden generell ein tieferes Verständnis für den Aufbau der Adressen haben, da ansonsten keine komplexen Routing-Entscheidungen möglich sind.

Unicast-Adressen können einen Interface-Identifizierer enthalten, der aus einer Adresse des einzelnen Netzwerkinterfaces gebildet wird. Zwingend notwendig ist es, dass solche Identifizierer, bezogen auf die lokale Verbindung, eindeutig sind. Denkbar sind auch global eindeutige Identifizierer, wie z.B. im Fall der 48bit umfassenden MAC-Adresse nach IEEE im Ethernet.

Die Unspezifizierte Adresse 0:0:0:0:0:0:0:0, oder ::, wird als Absender-Adresse verwendet, solange ein Rechner noch keine eigene Adresse ermitteln konnte. Sie darf niemals für ein Interface konfiguriert oder als Empfängeradresse einer Verbindung verwendet werden.

Die Adresse 0:0:0:0:0:0:0:1, oder ::1, wird als loopback-Adresse bezeichnet. Sie kann von einzelnen Knoten verwendet werden, um Pakete an sich selber zu senden. Hierfür kann sie einem virtuellen Interface, z.B. dem Loopback-Device, zugewiesen werden. Es ist nicht erlaubt einem physikalischen Interface diese Adresse zu geben.

Um den Übergang von IPv4 auf IPv6 zu vereinfachen wurden zwei Adressräume reserviert, die jeweils den kompletten Adressraum von IPv4 in sich aufnehmen können. Der eine Bereich 0:0:0:0:0:xxx:xxx wird als IPv4-kompatible Adresse bezeichnet. Ein Rechner mit einer solchen Adresse ist tatsächlich in der Lage, an ihn adressierte IPv6-Pakete zu verarbeiten. Rechner, die nur IPv4 unterstützen, können über die sogenannte IPv4-Abbildbare IPv6-Adresse (IPv4 mappable IPv6 address) angesprochen werden. Dies sind die Adressen im Bereich 0:0:0:0:FFFF:xxx:xxx. Die letzten 32 Bit dieser IPv6-Adressen sind jeweils die 32 Bit der IPv4-Adresse in hexadezimaler Zahldarstellung.

Aggregierbare globale Unicast-Adressen nach rfc2374 dienen der globalen Adressierung einzelner Rechner. Die Aggregierbarkeit ist von grosser Bedeutung, weil durch die Abbildung von Netzwerkhierarchien in der Adresszuteilung die Routing-Tabellen von default-freien Routern eine klar begrenzte Grösse erreichen. Ein default-freier Router ist ein Router, der für alle weltweit möglichen Adressen eine exakte Route kennt und nicht in einigen Fällen das Routing an einen Default-Router mit umfangreicheren Informationen delegiert.

Tabelle 1-6. Aggregierbare globale Unicast-Adressen

3	13	8	24	16	64 bits
FP	TLA ID	RES	NLA ID	SLA ID	Interface ID

Der Format-Präfix FP ist bisher auf 001 festgelegt. Die Toplevel-Identifikation TLA ID ist die oberste Ebene in der Routinghierarchie. Ein defaultfreier Router muss mindestens die Routen zu diesen 8192 unterschiedlichen Netzen kennen. Das reservierte Feld RES, das bis zu seiner Verwendung immer auf 0 gesetzt werden muss, dient der eventuellen Erweiterung. Die Nextlevel-Identifikation NLA ID ist eine Möglichkeit für Backbone-Provider, den verfügbaren Adressraum in Teile zu zerlegen, die dann entweder als Abbild der Netzwerkstruktur oder als Delegation eines Bereiches an einen Kunden betrachtet werden kann. Die Sitelevel-Identifikation ist dann wiederum der Bereich, innerhalb dessen eine Organisation, die das Internet nutzt, ihre interne Struktur abbilden kann.

Link-Lokale Unicast-Adressen verwenden den Präfix FE80::/64, gefolgt von dem 64 Bit grossen Interface-Identifizier. Diese Adressen werden von den einzelnen Rechnern automatisch ermittelt und dienen dem Datenaustausch ohne lokale Infrastruktur und zur Autokonfiguration. Site-Lokale Adressen werden aus dem Präfix FEC0::/48 gebildet, gefolgt von einer 16 Bit langen Subnet-ID und dem 64 Bit langen Interface-Identifizier. Link-Lokale Adressen dürfen auf keinen Fall von Routern weitergeleitet werden, Site-Lokale Adressen sollten nicht ins Internet geroutet werden, ähnlich den aktuellen IPv4-Adressen nach rfc1918.

1.1.5. Anycast-Adressen

Eine Adresse, die mehr als einem Interface zugewiesen wurde, wird als Anycast-Adresse bezeichnet. Da Anycast-Adressen aus dem Adressraum der Unicast-Adressen entnommen werden, sind diese nicht voneinander unterscheidbar. Daher muss jeder einzelne Rechner, der für eines seiner Interfaces eine Anycast-Adresse verwenden soll, explizit dafür konfiguriert werden. Zu jeder Anycast-Adresse wird ein längster Adresspräfix konfiguriert, der besagt, wie weit die einzelnen Mitglieder einer Anycast-Gruppe als Routing-Einträge bekanntgegeben werden und ab welcher Entfernung diese zu einem Routing-Eintrag zusammengefasst werden dürfen. Da im schlimmsten Fall bei einem Präfix von 0 jeder Anycast-Rechner einzeln global bekanntgemacht wird, ist es notwendig, die Verwendung einzuschränken.

Um erstmal Erfahrungen mit dem neuen Konzept von Anycast-Adressen sammeln zu können wird die Verwendung erstmal dahingehend eingeschränkt, dass es nicht erlaubt ist, eine Anycast-Adresse als Absender zu verwenden. Ausserdem dürfen bisher nur IPv6-Router Anycast-Adressen verwenden. Die einzige definierte Verwendung ist die Anycast-Adresse für alle Router eines Subnetzes. Hierbei wird der Präfix des betreffenden Subnetzes verwendet und der restliche Teil auf 0 gesetzt. Es wird von allen Routern erwartet, dass sie diese Anycast-Adresse unterstützen.

1.1.6. Multicast-Adressen

Eine IPv6 Multicast-Adresse dient dazu, eine Gruppe von Rechnern anzusprechen. Ein Rechner kann Mitglied beliebig vieler dieser Gruppen sein.

Tabelle 1-7. Aufbau von Multicast-Adressen

8	4	4	112 bits
11111111	Flags	Gültigkeit	Gruppen-ID

Das Flags-Feld kann bisher zwei Werte annehmen, 0000 für permanent zugewiesene Adressen, 0001 bezeichnet nicht-permanent zugewiesene Adressen. Die Gültigkeit bezeichnet, in welchem Umfang diese Multicast-Gruppe verwendet wird.

Tabelle 1-8. Gültigkeitsbereiche von Multicast-Adressen

Wert	Bedeutung
0	reserviert
1	Rechner-Lokal
2	Link-Lokal
3	(nicht zugewiesen)
4	(nicht zugewiesen)
5	Site-Lokal
6	(nicht zugewiesen)
7	(nicht zugewiesen)
8	Organisations-Lokal
9	(nicht zugewiesen)
A	(nicht zugewiesen)
B	(nicht zugewiesen)
C	(nicht zugewiesen)
D	(nicht zugewiesen)
E	Global
F	reserviert

Die Gruppen-ID dient dazu, die erwünschte Gruppe von Rechnern zu bezeichnen. Permanent zugewiesene Multicast-Adressen sind unabhängig von der Gültigkeit. Nicht-permanente Gruppen-IDs können durchaus in unterschiedlichen Gültigkeitsbereichen unterschiedliche Bedeutungen erfüllen. Multicast-Adressen dürfen nie als Absender verwendet werden.

Tabelle 1-9. Beispiel für globale Multicast-Adressen

Adresse	Bedeutung
FF01::101	alle NTP-Server auf dem gleichen Rechner wie der Absender
FF02::101	alle NTP-Server an der gleichen Verbindung wie der Absender
FF05::101	alle NTP-Server am gleichen Standort wie der Absender
FF0E::101	alle NTP-Server im Internet

Multicast-Pakete werden auch verwendet, um die Link-Layer-Adresse eines anderen Rechners zu ermitteln. Dadurch, dass keine Broadcasts wie bei dem für IPv4 verwendeten Address Resolution Protocol ARP mehr verwendet werden, sinkt die Netzwerklast für Rechner, die nicht unmittelbar von der Anfrage betroffen sind. Diese Multicast-Pakete heißen Neighbour Solicitation und Advertisement Message und entsprechen dem Standard für ICMPv6.

1.2. Headerformat

1.2.1. Standard-Header

Der Standard-Header ist der Header, den jedes IPv6 Paket mindestens mitführen muss.

Abbildung 1-1. IPv6 Standard-Header

Byte	0				1				2				3																			
Bit	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
Word 0	Version				Traffic Class				Flow Label																							
Word 1	Payload Length								Next Header				Hop Limit																			
Word 2	Source Address																															
Word 3																																
Word 4																																
Word 5																																
Word 6	Destination Address																															
Word 7																																
Word 8																																
Word 9																																

Die Version ist festgesetzt auf 6. Die Verkehrsklasse, oder Traffic Class, dient der Klassifizierung verschiedener Datenströme mit der Perspektive, bestimmte Klassen bevorzugt zu behandeln. Die Flusskennung, oder Flow-Label, kann von einem Rechner, der Pakete aussendet, genutzt werden, um damit zusammengehörige Pakete zu kennzeichnen. Pakete mit gleichen Quell- und Zieladressen und demselben Flow Label dürfen sich nicht in den Headern für Zwischenstationen unterscheiden, so dass Router nicht die Header jedes Paketes auswerten müssen, sondern diese Berechnungen nur einmal stattfinden. Die Nutzlast-Länge, oder Payload Length, ist die Anzahl an Bytes, die auf den Standard-Header folgen, inklusive aller Erweiterungsheader. Das Next Header Feld bezeichnet den Typ des Headers, der direkt auf den Standard-Header folgt. Hat dieses Feld den Wert 59, so bedeutet das, dass kein weiterer Header auf diesen folgt. Das Hop Limit bezeichnet die maximale Anzahl verbleibender Netzwerkübergänge. Dieses Feld wird von jedem Router, den das Paket passiert, um eins reduziert. Nimmt das Feld den Wert 0 an, so wird das Paket vom Router verworfen. Die Quelladresse bezeichnet den Absender des Paketes, die Zieladresse den vorgesehenen Empfänger des Paketes. Dies muss allerdings nicht der endgültige Empfänger sein, wenn ein Routing-Header vorhanden ist.

1.2.2. Erweiterungs-Header

Für den Fall, dass weitere Paketinformationen übertragen werden müssen, werden diese als

Erweiterungs-Header in Form einer verketteten Liste an den Standard-Header angehängt. Diese Erweiterungs-Header werden ausschliesslich von dem Rechner ausgewertet, dessen Adresse als Empfänger eingetragen ist. Alle Zwischenstationen werten ausschliesslich den Hop-By-Hop Options Header aus.

Um die Verarbeitung zu beschleunigen ist die Reihenfolge der Header im Paket fest vorgeschrieben. Daher muss ein Router immer nur das Next Header Feld des Standard-Headers dahingehend überprüfen, ob ein Hop-By-Hop Options Header folgt. Ist das nicht der Fall so befinden sich in dem Paket keine Informationen für den Router.

Ist es notwendig, dass der absendende Rechner bestimmt, welchen Weg das Paket zum Ziel nimmt, so kann der erste vorgesehene Router als Ziel des Paketes angegeben werden und alle weiteren Schritte zum endgültigen Ziel in einem Routing-Header abgelegt werden. Ein Router verarbeitet solche Pakete, indem er aus dem Routing-Header die nächste Zwischenstation ausliest und als neue Zieladresse verwendet.

Ein weiterer wichtiger Erweiterungs-Header ist der Fragmentation Header. Anders als bei IPv4 dürfen IPv6-Pakete niemals von Zwischenstationen verändert werden. Ist ein Paket zu gross um weitergeleitet zu werden muss eine ICMPv6-Nachricht an den Absender erzeugt werden. Idealerweise sollten Anwendungen vor Verbindungsaufbau eine Path-MTU-Discovery durchführen, bei der die maximale Paketgrösse auf dem Weg zum Ziel ermittelt wird. Es ist vorgeschrieben, dass jede IPv6-Verbindung mindestens 1280 Bytes grosse Pakete übertragen kann.

Die grosse Zahl weiterer Header, die bei Bedarf verwendet werden können, sind den entsprechenden RFCs zu entnehmen.

1.3. Autoconfiguration

Die automatische Konfiguration von IP-Adressen ist eine Grundlage für IPv6, da die automatisch aus der Link-Layer-Adresse (MAC-Adresse) erzeugten Link-Local-Adressen als Grundlage für jegliche Kommunikation dienen. Ausserdem ist es wünschenswert, dass Benutzer nicht mehr direkt IP-Adressen konfigurieren müssen, weil durch die Adressgrösse von 128 Bit die Fehlerwahrscheinlichkeit deutlich steigt.

Ausserdem zielt die Adressvergabe darauf ab, dass sich bei einem Providerwechsel der Präfix der Adressen ändert. Damit nicht jedes Mal sämtliche Rechner neu konfiguriert werden müssen ist es besser, von vornherein eine automatische Konfiguration vorzunehmen, die jederzeit von zentraler Stelle aus geändert werden kann.

Wird ein Rechner mit einem IPv6-Netz verbunden, erzeugt er zunächst eine Link-Local-Adresse. Mit dieser wendet er sich dann über eine Router Solicitation Message an alle Router des lokalen Netzwerks und erfragt den Präfix des Netzes. Dieser Präfix wird dann mit der MAC-Adresse kombiniert als

IPv6-Adresse verwendet. Alternativ kann der Router eine Nachricht versenden, die statische Autokonfiguration festlegt. Dann muss der Rechner seine Konfiguration per DHCPv6 ermitteln.

1.4. Namensauflösung

Um die Auflösung von Namen zu IPv6-Adressen zu ermöglichen muss das Domain Name System (DNS) erweitert werden. Hierzu wurde ein neuer Anfrage-Typ eingeführt, der AAAA Record. Dieser liefert zu einem Namen die zugehörige IPv6-Adresse. Um zu einer bekannten IPv6-Adresse den zugehörigen Namen zu ermitteln gibt es zwei Standards. Der erste, der mittlerweile nicht mehr verwendet werden sollte, verwendet die Domain ip6.int. Um die Auflösung vorzunehmen wird die IPv6-Adresse in Hexadezimal-Schreibweise rückwärts geschrieben und jede Ziffer als Subdomain betrachtet. Die Adresse 2001:200:0:4819:203:47ff:fea5:3085 kann also durch eine Anfrage nach 5.8.0.3.5.a.e.f.f.f.7.4.3.0.2.0.9.1.8.4.0.0.0.0.0.2.0.1.0.0.2.ip6.int. aufgelöst werden. Der aktuellere Standard, der auch dauerhaft verwendet werden soll, nutzt die Domain ip6.arpa, analog zu IPv4. Hierbei wird die Adresse in der sogenannten Bitstring-Notation verwendet. Diese Schreibweise erlaubt die Delegation von Zonen an beliebigen Stellen in der Bitweisen Schreibweise. Das obige Beispiel würde zu einer Abfrage wie `\[x2001020000004819020347fffea53085/128].IP6.ARPA.` führen, wobei diese auch in mehrere hierarchische Teile unterteilt werden kann, also `\[020347fffea53085/64].\[x4819/16].\[x200102000000/48].IP6.ARPA.`

1.5. Routing

Das Routing von IPv6 unterscheidet sich nicht signifikant vom Routing von IPv4 mit CIDR. Ein Router kennt immer die direkt angrenzenden IPv6-Netze, und hat entweder Routing-Informationen für alle 8192 Top-Level-Aggregation-IDs oder besitzt einen Default-Router-Eintrag. Diese Einträge können statisch vorgenommen werden, in der Regel werden hierfür jedoch Routing-Protokolle eingesetzt werden. Um die Autoconfiguration zu unterstützen muss ein Router sich an die entsprechenden Multicast- und Anycast-Adressen binden und sowohl auf Router-Solicitations antworten als auch Router Advertisements senden.

Allerdings fordern einige Funktionen, die direkt in IPv6 implementiert sind, dass ein Router gewisse Zusatzleistungen erfüllt. Welche dies sind, und was für Hardwareanforderungen diese an verwendete Router stellen hängt sehr stark von den gewünschten Funktionen ab.

1.6. Unterstützung mobiler Hosts

IPv6 soll von Anfang an eine Unterstützung für Rechner mitbringen, die sich nicht immer in ihrem Heimatnetzwerk befinden, die allerdings trotzdem über ihre Heimatadresse erreicht werden sollen. Dies entspricht MobileIP für IPv4, das in rfc3344 definiert ist.

Hierzu sendet ein Rechner, der für mobile Verwendung konfiguriert ist, eine Nachricht an die Subnetz-Router-Anycastadresse seines Heimatnetzes. Dadurch wird einem Router des Heimatnetzes die aktuelle Adresse der mobilen Station mitgeteilt. Dieser kann dann im Auftrag des mobilen Rechners auf Neighbour Discovery Anfragen antworten und somit allen Rechnern, die versuchen, mit dem mobilen Rechner eine Verbindung aufzubauen, die momentane Adresse mitteilen, so dass die folgende Kommunikation direkt erfolgen kann. Dies ist ein grosser Vorteil gegenüber IPv4, da diese Funktion nicht im ursprünglichen Standard vorgesehen war. Mobile IPv4-Rechner müssen alle Kommunikation, die mit der Adresse im Heimatnetzwerk stattfinden soll, durch einen Tunnel zum Heimatrouter abwickeln, was zu unnötig komplexen Datenwegen führen kann.

Kapitel 2. Bestehende IPv6-Testnetze und Anbindungsmöglichkeiten

2.1. 6bone

Der 6bone stellt ein erstes Testbett dar, das dazu dient, Implementierungen von IPv6 zu erproben und allgemein Erfahrungen mit der Verwendung und Migration zu IPv6 zu sammeln. In der Anfangsphase bestanden die Verbindungen des 6bone aus Tunneln über IPv4, mittlerweile existieren allerdings auch schon einige native Verbindungen.

Um am 6bone teilzunehmen benötigt man einen Rechner, der über eine statische IPv4-Adresse verfügt, und der somit als Endpunkt des Tunnels dienen kann.

Der 6bone ist seit März 1996 in Betrieb und verwendet Adressen aus dem experimentellen Bereich nach rfc2471, unterhalb des Präfix 3FFE::/16. Die Planung sah von vornherein vor, dass dieser Adressraum nicht dauerhaft hierfür verwendet wird, sondern zu einem Zeitpunkt wieder freigegeben wird, zu dem der 6bone seine Berechtigung verloren hat. Mittlerweile ist IPv6 über die Erprobungsphase hinaus und soweit entwickelt, dass der produktive Einsatz absehbar ist. Daher ist es geplant, diese Abschaltung nach einer Übergangsphase, die am 1. Juli 2006 enden wird, vorzunehmen. Wer heutzutage ein IPv6-Netzwerk einrichten möchte, das auch von ausserhalb erreicht werden kann, sollte hierfür andere Möglichkeiten nutzen.

2.2. freenet6

Das freenet6 ist eine Möglichkeit, auch mit dynamisch vergebener IPv4-Adresse eine Verbindung zum 6bone herzustellen. Hierbei gibt es zwei Möglichkeiten. Handelt es sich um einen einzelnen Rechner, so kann man über einen anonymen Zugang einen direkten Tunnel zu einem freenet6 Router errichten.

Soll ein ganzes Netzwerk angebunden werden, und ist eventuell vorgesehen, dass auch Zugriffe aus dem 6bone auf diese Rechner erfolgen, muss man sich beim freenet6 registrieren und bekommt dann einen eigenen, festen Präfix unterhalb des freenet6-Präfixes mit einer Länge von 48bit. Diesen kann man dann mit Hilfe von Site-Level-Aggregation-IDs auf mehrere lokale Subnetze verteilen, so dass alle Rechner feste IPv6-Adressen im Adressraum des 6bone haben können.

Kapitel 3. Offizielle Unterstützung auf Betriebssystem-Seite

3.1. Linux

Der Linux-Kernel enthält eine experimentelle Implementierung eines IPv6 Stacks, der parallel zu IPv4 genutzt werden kann. Der Umfang der Unterstützung, was Server-Dienste und Client-Anwendungen angeht, ist natürlich Distributionsabhängig. Der Paketfilter (Netfilter/IPTables) enthält ebenfalls experimentellen Support für IPv6.

Debian GNU/Linux wird aktuell noch nicht mit einem IPv6-fähigen Kernel ausgeliefert, grundlegende Unterstützung ist allerdings für jemanden, der den Umgang mit Linux-Systemen gewohnt ist, leicht durch kompilieren eines angepassten Kernels zu erreichen. Demzufolge sind auch noch keine der mitgelieferten Serveranwendungen von sich aus IPv6-fähig, es werden jedoch von Entwicklern IPv6-fähige Versionen als Debian-Pakete angeboten.

SuSE Linux enthält IPv6-Unterstützung mit einer steigenden Anzahl von Diensten. Detailliertere Informationen waren leider nicht zu bekommen.

Auch Redhat Linux unterstützt ab Installation IPv6, das auch von diversen Server-Diensten genutzt wird.

3.2. Microsoft Windows

Es existiert eine experimentelle Version eines IPv6-Stacks für Windows 2000 Service Pack 1 und Windows XP. Windows XP SP1 enthält die erste Version, die von Microsoft offiziell für den produktiven Einsatz freigegeben ist. Die nächste erwartete Windows-Version, momentan unter dem Namen Windows 2003 gehandelt, wird ab Auslieferung IPv6 unterstützen.

3.3. MacOS 9/X

Für MacOS 9 existiert keine IPv6-Implementierung.

MacOS X enthält mindestens ab Version 10.2 (Jaguar) einen per default aktivierten IPv6-Stack. Frühere Versionen mussten noch mit Hilfe eines Kernelpatches vom Kame-Projekt IPv6-fähig gemacht werden.

3.4. OpenBSD

Enthält offizielle IPv6-Unterstützung seit Version 2.7, in der aktuellen Version 3.2 sind einige Dienste enthalten mit direkter Unterstützung für IPv6, ausserdem beherrscht der enthaltene Paketfilter pf IPv6.

Kapitel 4. IPv6-Unterstützung durch Server-Programme

Da sich TCP/IP, im Gegensatz zu den ISO-Protokollen, nicht sauber in ein Schichtenmodell einfügt, ist es nicht ausreichend, wenn das jeweilige Betriebssystem über einen IPv6-Stack verfügt. Alle Anwendungen müssen an das neue Socket-Interface angepasst werden um IPv6 verwenden zu können.

Das Socket-Interface ist die standardisierte Programmierschnittstelle für netzwerkbasierende Programme.

Die folgende Übersicht soll den aktuellen Entwicklungsstand einiger gern verwendeter Programme zeigen.

4.1. Name-Server

4.1.1. ISC Bind

Dieser weit verbreitete Nameserver enthält rudimentäre Unterstützung bereits seit der Bind8-Serie, vollständige Unterstützung ist allerdings erst seit Bind9 vorhanden. Unterstützt werden alle nötigen Resource-Records, also AAAA, A6, DNAME, usw.

4.1.2. Microsoft Nameserver

Der in Windows 2000 enthaltene Nameserver unterstützt zwar bereits die neuen AAAA-Records, um Namen auf IPv6-Adressen abzubilden, kann allerdings noch nicht auf Anfragen via IPv6 antworten.

4.2. Web-Server

4.2.1. Apache Webserver

Vom Apache Webserver existieren zwei unterschiedliche Versionen parallel, die 1.3er Serie und die neue 2.0er Serie. Für die ältere 1.3er Serie gibt es einen Patch, der die fehlende IPv6-Unterstützung hinzufügt, was sie für Produktionssysteme uninteressant macht. Anders hingegen die neuere Version 2.0, dort ist volle Unterstützung für IPv6 enthalten. Allerdings sind viele Module, die auf einem durchschnittlichen Apache Webserver benötigt werden, noch nicht auf diese Version portiert, so dass ein produktiver Einsatz hier noch nicht sinnvoll ist.

4.2.2. Microsoft IIS

Gegenwärtig unterstützt der Microsoft Internet Information Server noch kein IPv6.

4.3. Datei-Dienste

4.3.1. Samba

Die freie Implementierung des Common Internet Filesystem (CIFS) enthält ebenfalls noch keine IPv6-Unterstützung und wird diese wahrscheinlich auch erst bekommen, sobald es Clients gibt, die darauf zugreifen können. Bis zu dem Zeitpunkt wird ein externer Patch gepflegt, der bei Bedarf eingespielt werden kann.

4.3.2. Microsoft CIFS

In der zukünftigen Version, bisher als Windows 2003 angekündigt, wird eine Unterstützung für Datei- und Druckdienste über IPv6 enthalten sein.

4.3.3. NFS

Bisher sind nur Implementierung unter Solaris und NetBSD bekannt, da zunächst der Portmapper, auf den nfs angewiesen ist, IPv6 unterstützen muss.

4.4. SMTP (eMail)

4.4.1. Sendmail

Unterstützt in der offiziellen Version IPv6, die bei OpenBSD enthaltene Version ist bereits per default auch für IPv6 konfiguriert.

4.4.2. Exim

Auch dieser mittlerweile weit verbreitete Mail Transfer Agent (MTA) enthält bereits in der offiziellen Distribution vollständige Unterstützung für IPv6.

4.4.3. Postfix

Für diesen MTA ist eine IPv6-Unterstützung in der aktuellen Entwicklerversion enthalten, eine stabile Version ist also absehbar.

4.5. FTP

4.5.1. ProFTPd

Dieser weit verbreitete FTP-Daemon unterstützt noch kein IPv6. Allerdings existiert ein Patch für die Version 1.2.5, mit dessen Hilfe IPv6-Fähigkeit erreicht werden kann. Ausserdem ist die feste Integration bereits in der Roadmap für die Weiterentwicklung enthalten.

4.5.2. WuFTP

Dieser ebenfalls viel verwendete FTP-Daemon ist ebenfalls nicht IPv6-fähig. Ein Patch ist allerdings vom Kame-Projekt entwickelt worden. Eine feste Integration ist allerdings noch nicht geplant.

4.5.3. vsFTP

Dieser FTP-Daemon, dessen Name "Very Secure FTP Daemon" bedeutet, ist der einzige von dem ein Paket im Debian-IPv6-Repository vorhanden ist. Die IPv6-Unterstützung ist allerdings auch hier nur durch einen Patch möglich. Leider war es nicht möglich, mit diesem FTP-Daemon Dateien zu übertragen, eine Eingrenzung des Fehlers auf den Server oder den Client war leider nicht möglich.

Kapitel 5. Test der Fähigkeiten in einem Testaufbau

5.1. Debian GNU/Linux 3.0 (Woody)

5.1.1. Zustand nach der Installation

Unmittelbar nach der Installation, auch mit dem Bootparameter bf24, um mit einem 2.4er Kernel zu installieren, hat das System keinerlei Unterstützung für IPv6.

5.1.2. Schritte zu einem IPv6-fähigen Netzwerkstack

Um einen Kernel zu kompilieren, der einen IPv6 Stack enthält, müssen zunächst die Kernel-Sourcen per "apt-get install kernel-sources-2.4.18" nachinstalliert werden. Ausserdem werden ein paar zusätzliche Pakete zum späteren konfigurieren und kompilieren benötigt. Dies sind kernel-package, bin86 und libncurses-dev. Die als .tar.bz2 vorliegenden Kernelsourcen werden dann in /usr/src entpackt, die Standardkonfiguration von /boot/config-2.4.18-bf24 als .config in das Kernelverzeichnis kopiert. Ein Aufruf von "make menuconfig" bringt das übliche Konfigurationsmenü, in dem dann unter "Networking Options" der Punkt "The IPv6 Protocol" ausgewählt wird. Da es sich um experimentellen Code handelt muss auf jeden Fall unter "Code maturity options" die Option "Prompt for development and/or incomplete drivers" aktiviert werden. Der neue Kernel wird dann mit dem Aufruf "make-kpkg --revision ipv6.01 kernel-image" kompiliert, so dass nach Vollendung ein .deb-Paket in /usr/src zu finden ist. Dieses Paket wird mittels dpkg installiert. Nach einem Reboot mit dem neuen Kernel kann dann per modconf das neue Modul ausgewählt werden. Nach Aufruf von "/etc/init.d/networking restart" sind dann auch die Interfaces auf IPv6 Link-Local Adressen konfiguriert.

Um die Konfiguration zu testen wurde dann noch das Paket iputils-ping installiert, das den ping6 Befehl enthält.

Abbildung 5-1. Interface-Konfiguration mit IPv6-fähigem Kernel

```
nws-04:~# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:10:5A:6B:E3:71
          inet addr:172.17.17.4  Bcast:172.17.17.255  Mask:255.255.255.0
          inet6 addr: fe80::210:5aff:fe6b:e371/10 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:307 errors:0 dropped:0 overruns:0 frame:0
          TX packets:27 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:100
          RX bytes:42892 (41.8 KiB)  TX bytes:2284 (2.2 KiB)
          Interrupt:10 Base address:0xb400
```

```
lo          Link encap:Local Loopback
           inet addr:127.0.0.1  Mask:255.0.0.0
           inet6 addr: ::1/128 Scope:Host
           UP LOOPBACK RUNNING  MTU:16436  Metric:1
           RX packets:8 errors:0 dropped:0 overruns:0 frame:0
           TX packets:8 errors:0 dropped:0 overruns:0 carrier:0
           collisions:0 txqueuelen:0
           RX bytes:560 (560.0 b)  TX bytes:560 (560.0 b)
```

Hier ist zu sehen, dass sobald der Kernel IPv6 unterstützt, auch automatisch eine Link-Local-Adresse generiert wird, in der die MAC-Adresse enthalten ist. Ausserdem wurde das loopback-Interface automatisch mit der entsprechenden IPv6-Adresse initialisiert.

Hierbei muss erwähnt werden, dass die MAC-Adresse der Netzwerkkarte nicht unverändert in die Link-Local-Adresse einfließt. Im ersten Oktett wurde das vorletzte Bit auf 1. Auf diese Weise wird die Information kodiert, dass es sich um eine weltweit eindeutige Adresse handelt.

Abbildung 5-2. ping6 per Link-Local-Adresse

```
nws-04:~# ping6 -c 4 -I eth0 fe80::210:5aff:fe6b:e371 etl
PING fe80::210:5aff:fe6b:e371 (fe80::210:5aff:fe6b:e371) from fe80::210:5aff:fe6b:e371 etl
64 bytes from fe80::210:5aff:fe6b:e371: icmp_seq=1 ttl=128 time=0.242 ms
64 bytes from fe80::210:5aff:fe6b:e371: icmp_seq=2 ttl=128 time=0.237 ms
64 bytes from fe80::210:5aff:fe6b:e371: icmp_seq=3 ttl=128 time=0.128 ms
64 bytes from fe80::210:5aff:fe6b:e371: icmp_seq=4 ttl=128 time=0.176 ms

--- fe80::210:5aff:fe6b:e371 ping statistics ---
4 packets transmitted, 4 received, 0% loss, time 2998ms
rtt min/avg/max/mdev = 0.128/0.195/0.242/0.049 ms

nws-04:~# ping6 -c 4 -I eth0 fe80::210:5aff:fe6b:e3e0
PING fe80::210:5aff:fe6b:e3e0 (fe80::210:5aff:fe6b:e3e0) from fe80::210:5aff:fe6b:e371 etl
64 bytes from fe80::210:5aff:fe6b:e3e0: icmp_seq=1 ttl=128 time=0.271 ms
64 bytes from fe80::210:5aff:fe6b:e3e0: icmp_seq=2 ttl=128 time=0.131 ms
64 bytes from fe80::210:5aff:fe6b:e3e0: icmp_seq=3 ttl=128 time=0.159 ms
64 bytes from fe80::210:5aff:fe6b:e3e0: icmp_seq=4 ttl=128 time=0.166 ms

--- fe80::210:5aff:fe6b:e3e0 ping statistics ---
4 packets transmitted, 4 received, 0% loss, time 2997ms
rtt min/avg/max/mdev = 0.131/0.181/0.271/0.055 ms
```

Bei der direkten Verwendung von Link-Lokalen Adressen ist es unbedingt notwendig, das Interface mit anzugeben. Der IPv6-Stack hat nämlich keine Möglichkeit, allein aus der angegebenen Link-Lokalen IPv6-Adresse die Information zu ermitteln, über welches Netzwerkinterface dieses Paket gesendet werden muss. Es ist nämlich durchaus denkbar, dass ein Rechner mit zwei Netzwerken direkt verbunden ist, und dieselbe Link-Lokale Adresse von zwei verschiedenen Stationen in beiden Netzen verwendet wird. Diese Einschränkung ist auch der Grund, aus dem es nicht praktikabel ist, nur mit Link-Lokalen

Adressen zu arbeiten, selbst wenn eine flache Netzwerkstruktur, d.h. ein Routerloses Netzwerk, verwendet wird.

5.1.3. Konfiguration von statischen Site-Local Adressen

Der nächste Schritt zu einem komplexeren Netzwerk ist die Verwendung von Site-Local Adressen. Da zu einem späteren Zeitpunkt noch eine Aufteilung in mehrere Netze stattfinden soll, werden hier zunächst Adressen mit dem Präfix fec0:1::/64 vergeben, und zwar die Adresse 4 für den Linux-Rechner.

Unter Debian GNU/Linux erfolgt die Konfiguration aller Netzwerkinterfaces zentral in der Datei /etc/network/interfaces. Nach der Installation enthält sie bereits die Einstellungen für IPv4, es muss nur noch ein Eintrag für IPv6 vorgenommen werden.

Abbildung 5-3. Site-Local Adresse in /etc/network/interfaces

```
# /etc/network/interfaces -- configuration file for ifup(8), ifdown(8)

# The loopback interface
auto lo
iface lo inet loopback

# The first network card - this entry was created during the Debian installation
# (network, broadcast and gateway are optional)
auto eth0
iface eth0 inet static
address 172.17.17.4
netmask 255.255.255.0
network 172.17.17.0
broadcast 172.17.17.255
gateway 172.17.17.254

iface eth0 inet6 static
address fec0:1::4
netmask 64
```

Um diese Einstellung wirksam werden zu lassen wird einmal "/etc/init.d/networking restart" aufgerufen. Der Erfolg lässt sich per ifconfig überprüfen.

Abbildung 5-4. Ausgabe von ifconfig nach manueller Vergabe der Site-Local-Adresse

```
nws-04:~# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:10:5A:6B:E3:71
          inet addr:172.17.17.4  Bcast:172.17.17.255  Mask:255.255.255.0
          inet6 addr: fec0:1::4/64 Scope:Site
          inet6 addr: fe80::210:5aff:fe6b:e371/10 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
```

```
RX packets:66 errors:0 dropped:0 overruns:0 frame:0
TX packets:26 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:100
RX bytes:7466 (7.2 KiB) TX bytes:1890 (1.8 KiB)
Interrupt:10 Base address:0xb400

lo      Link encap:Local Loopback
        inet addr:127.0.0.1  Mask:255.0.0.0
        inet6 addr: ::1/128 Scope:Host
        UP LOOPBACK RUNNING  MTU:16436  Metric:1
        RX packets:8 errors:0 dropped:0 overruns:0 frame:0
        TX packets:8 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:0
        RX bytes:560 (560.0 b) TX bytes:560 (560.0 b)
```

Ausserdem ist es jetzt möglich, die benachbarten Rechner auch per ping6 an die Site-Local-Adressen anzusprechen. Auf die Konfiguration der hier getesteten Gegenstellen wird noch näher eingegangen. Es ist sehr hilfreich, dass der ifconfig Befehl nicht einfach nur die IPv6-Adressen ausgibt, sondern automatisch darüber informiert, aus welchem Adressraum diese Adresse stammt, so wie hier die beiden Angaben "Scope: Site" und "Scope: Link".

Abbildung 5-5. ping6 auf benachbarte Rechner per Site-Local-Adresse

```
nws-04:~# ping6 -c 4 fec0:1::6
PING fec0:1::6(fec0:1::6) from fec0:1::4 : 56 data bytes
64 bytes from fec0:1::6: icmp_seq=1 ttl=128 time=0.183 ms
64 bytes from fec0:1::6: icmp_seq=2 ttl=128 time=0.142 ms
64 bytes from fec0:1::6: icmp_seq=3 ttl=128 time=0.129 ms
64 bytes from fec0:1::6: icmp_seq=4 ttl=128 time=0.136 ms

--- fec0:1::6 ping statistics ---
4 packets transmitted, 4 received, 0% loss, time 2997ms
rtt min/avg/max/mdev = 0.129/0.147/0.183/0.024 ms

nws-04:~# ping6 -c 4 fec0:1::5
PING fec0:1::5(fec0:1::5) from fec0:1::4 : 56 data bytes
64 bytes from fec0:1::5: icmp_seq=1 ttl=128 time=0.198 ms
64 bytes from fec0:1::5: icmp_seq=2 ttl=128 time=0.170 ms
64 bytes from fec0:1::5: icmp_seq=3 ttl=128 time=0.187 ms
64 bytes from fec0:1::5: icmp_seq=4 ttl=128 time=0.167 ms

--- fec0:1::5 ping statistics ---
4 packets transmitted, 4 received, 0% loss, time 2997ms
rtt min/avg/max/mdev = 0.167/0.180/0.198/0.018 ms
```

5.1.4. Bereits vorhandene IPv6-fähige Dienste

Es werden standardmässig keine IPv6-Fähigen Dienste installiert.

5.1.5. Bereits vorhandene IPv6-fähige Clients

Der per default installierte Telnet-Client ist in der Lage, Verbindungen zu IPv6 Telnet Servern aufzubauen.

5.1.6. Schritte zu IPv6-fähigen Diensten

Um experimentelle Debian-Pakete installieren zu können, die IPv6 unterstützen, muss in der Datei `/etc/apt/sources.list` die Zeile

```
deb http://ftp.bononia.it/debian-ipv6 woody ipv6
```

eingefügt werden. Wenn danach die Pakete `openbsd-inetd` und `apache` installiert werden sowie das `ssh`-Paket aktualisiert wird, ist bereits die Standard-Website und der SSH-Daemon erreichbar. Ausserdem habe ich in der `/etc/inetd.conf` einen Eintrag für den Echo-Server eingefügt und den Telnet-Server aktiviert. Dadurch lässt sich die Erreichbarkeit des Rechners von jedem anderen Betriebssystem prüfen, weil auf allen getesteten Plattformen zumindest ein Telnet-Client vorhanden ist.

Abbildung 5-6. Auszug aus `/etc/inetd.conf` für IPv6 echo und telnet

```
echo stream tcp46 nowait root internal
telnet stream tcp46 nowait telnetd.telnetd /usr/sbin/tcpd /usr/sbin/in.telnetd
```

Da die reine Verwendung von Link-Lokalen Adressen nicht praktikabel ist, und von einigen Anwendungen die notwendige Angabe des Interfaces nicht verarbeitet werden kann, mussten Site-Lokale Adressen verwendet werden. Allerdings bestehen manche Programme darauf, dass gar keine IPv6-Adressen explizit angegeben werden, sondern dass immer symbolische Namen verwendet werden, die dann vom Resolver der lokalen Station aufgelöst werden. Daher war es notwendig, auf dem Linux-Rechner ausserdem einen Nameserver einzurichten, der von Namen auf Site-Lokale IP-Adressen und umgekehrt abbilden kann. Hierbei handelt es sich um den Bind9, allerdings in der Debian-Version, die keine Anfragen über IPv6, aber AAAA-Records unterstützt.

Abbildung 5-7. Einträge in `named.conf` zur Auflösung von Site-Local-Adressen

```
...
zone "ipv6.local" {
type master;
file "/etc/bind/db.ipv6";
```


Obwohl der FTP-Server vsftp IPv6 unterstützen soll, war es nach der Installation nicht möglich, nach dem erfolgreichen Login Daten zu übertragen, weder für Verzeichnislistings noch für Dateien.

Um eine Liste der Dienste zu bekommen, die auf IPv6-Verbindungen warten, kann die Ausgabe des Befehls "netstat -an" dienen. Aus dem folgenden Beispiel ist zu erkennen, dass es einen echo-, einen http-, einen ssh- und einen telnet-Server gibt, die auf den jeweiligen Well-Known-Ports lauschen.

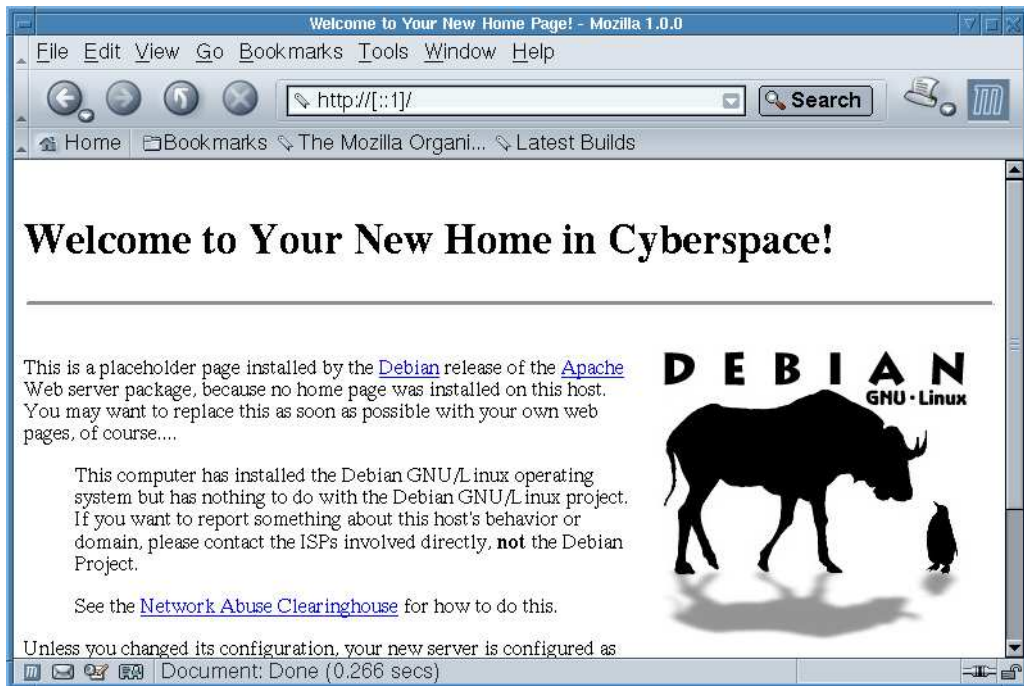
Abbildung 5-10. Ausgabe von netstat -an (gefiltert)

```
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp      0      0 :::7                    :::*                    LISTEN
tcp      0      0 :::80                   :::*                    LISTEN
tcp      0      0 :::22                   :::*                    LISTEN
tcp      0      0 :::23                   :::*                    LISTEN
```

5.1.7. Schritte zu IPv6-fähigen Clients

Um einen IPv6-fähigen Browser zu bekommen wurde einfach Mozilla in der Version 1.0.0 aus dem Stable-Zweig der Debian-Distribution installiert. Dieser unterstützt sowohl die direkte Angabe von IPv6-Adressen in eckigen Klammern, für den lokal installierten Apache also "http://[::1]/", als auch die Namensauflösung über AAAA-Records per IPv4.

Abbildung 5-11. Zugriff mit Mozilla auf den lokalen Webserver



5.2. Microsoft Windows 2000

5.2.1. Zustand nach der Installation

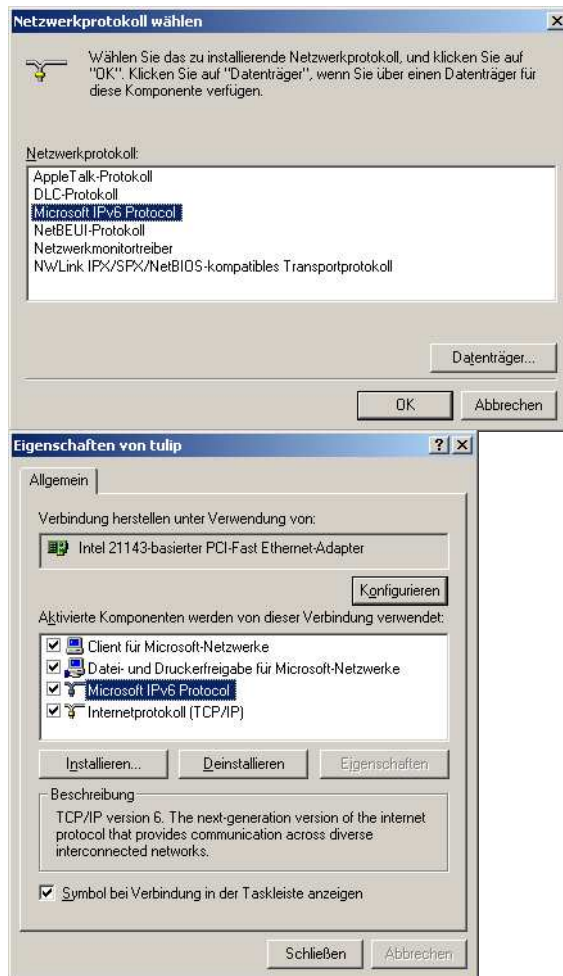
Unmittelbar nach der Installation ist keine Unterstützung für IPv6 vorhanden.

5.2.2. Schritte zu einem IPv6-fähigen Netzwerkstack

Um einen IPv6-Stack unter Windows 2000 zu installieren wird zunächst das IPv6kit von Microsoft benötigt. Ist eine aktuellere Version als Windows 2000 Service Pack 1 installiert, so muss eine kleine Änderung entsprechend den FAQ an einer .inf-Datei vorgenommen werden. Dann kann das Paket, das in der gleichen Form zur Verfügung steht wie die bekannten Microsoft Hotfixes, installiert werden. Nach einem obligatorischen Reboot kann dann unter den Netzwerkeinstellungen das "Microsoft IPv6

Protocol" installiert werden.

Abbildung 5-12. Hinzufügen des IPv6-Protokolls zu einer Netzwerkkarte



Danach ist der Rechner für den Einsatz von IPv6 bereit, was sich leicht durch Eingabe von "ipv6 if" auf der Kommandozeile prüfen lässt.

Abbildung 5-13. Ausgabe von "ipv6 if" nach Installation

```

Eingabeaufforderung
Microsoft Windows 2000 [Version 5.00.2195]
(C) Copyright 1985-2000 Microsoft Corp.

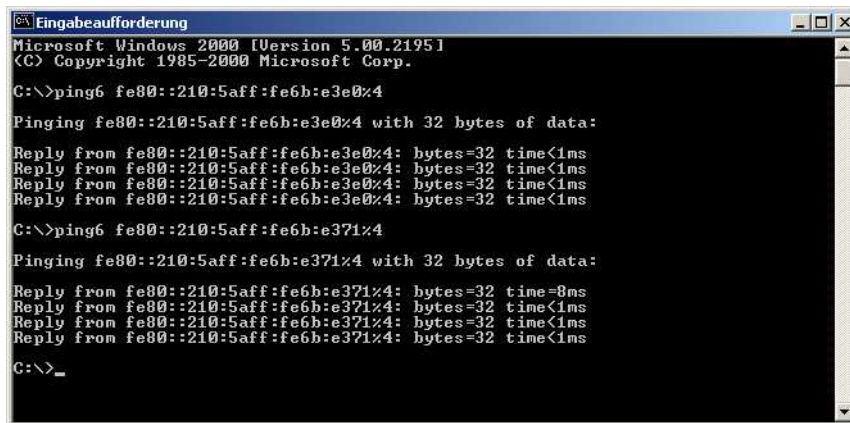
C:\>ipv6 if
Interface 4 (site 1): LAN-Verbindung
uses Neighbor Discovery
link-level address: 00-10-5a-cf-65-68
  preferred address fe80::210:5aff:fcf:6568, infinite/infinite
  multicast address ff02::1, 1 refs, not reportable
  multicast address ff02::1:ffcf:6568, 1 refs, last reporter
link MTU 1500 (true link MTU 1500)
current hop limit 128
reachable time 19500ms (base 30000ms)
retransmission interval 1000ms
DAD transmits 1
Interface 3 (site 1): 6-over-4 Virtual Interface
uses Neighbor Discovery
link-level address: 172.17.17.6
  preferred address fe80::ac11:1106, infinite/infinite
  multicast address ff02::1, 1 refs, not reportable
  multicast address ff02::1:ff11:1106, 1 refs, last reporter
link MTU 1280 (true link MTU 65515)
current hop limit 128
reachable time 44500ms (base 30000ms)
retransmission interval 1000ms
DAD transmits 1
Interface 2 (site 0): Tunnel Pseudo-Interface
does not use Neighbor Discovery
link-level address: 0.0.0.0
  preferred address ::172.17.17.6, infinite/infinite
link MTU 1280 (true link MTU 65515)
current hop limit 128
reachable time 0ms (base 0ms)
retransmission interval 0ms
DAD transmits 0
Interface 1 (site 0): Loopback Pseudo-Interface
does not use Neighbor Discovery
link-level address:
  preferred address ::1, infinite/infinite
link MTU 1500 (true link MTU 1500)
current hop limit 1
reachable time 0ms (base 0ms)
retransmission interval 0ms
DAD transmits 0

C:\>_

```

Hier ist zu erkennen, dass das Ethernet-Interface mit der Nummer 4 automatisch eine Link-Lokale Adresse zugewiesen bekommen hat. Hierbei handelt es sich um die MAC-Adresse, die durch Einfügen der Werte "FF:FE" auf 64 Bit erweitert wurde. Ausserdem wurden zwei Multicast-Adressen konfiguriert, nämlich ff02::1 als Adresse für alle Rechner an der lokalen Verbindung und ff02::1:ffcf:6568 als Multicast-Adresse zur Stationsfindung, also um eine Abbildung von IPv6-Adressen auf MAC-Adressen möglich zu machen.

Abbildung 5-14. Test der Installation per ping6



```

C:\>ping6 fe80::210:5aff:fe6b:e3e0%4
Pinging fe80::210:5aff:fe6b:e3e0%4 with 32 bytes of data:
Reply from fe80::210:5aff:fe6b:e3e0%4: bytes=32 time<1ms
Reply from fe80::210:5aff:fe6b:e3e0%4: bytes=32 time<1ms
Reply from fe80::210:5aff:fe6b:e3e0%4: bytes=32 time<1ms
Reply from fe80::210:5aff:fe6b:e3e0%4: bytes=32 time<1ms
C:\>ping6 fe80::210:5aff:fe6b:e371%4
Pinging fe80::210:5aff:fe6b:e371%4 with 32 bytes of data:
Reply from fe80::210:5aff:fe6b:e371%4: bytes=32 time=0ms
Reply from fe80::210:5aff:fe6b:e371%4: bytes=32 time<1ms
Reply from fe80::210:5aff:fe6b:e371%4: bytes=32 time<1ms
Reply from fe80::210:5aff:fe6b:e371%4: bytes=32 time<1ms
C:\>_

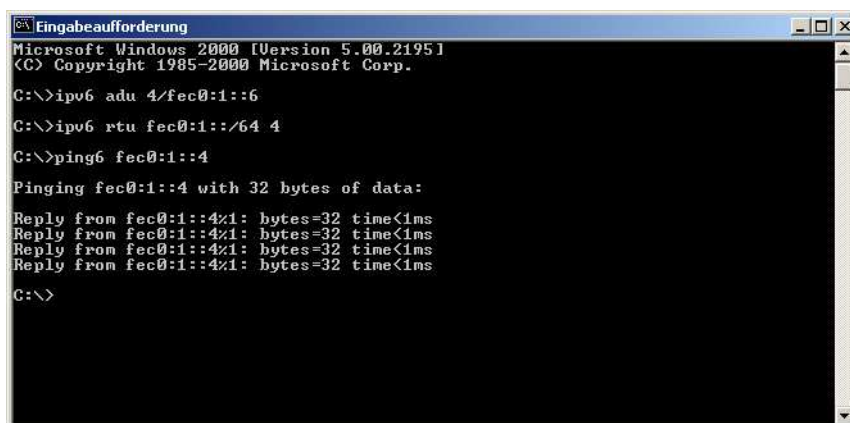
```

Die Angabe des Interfaces, über das eine Link-Lokale Adresse erreicht werden kann, erfolgt unter Windows, indem die Interface-Nummer mit einem Prozentzeichen an die Adresse angehängt wird.

5.2.3. Konfiguration von statischen Site-Local Adressen

Unter Windows müssen statische Adressen über die Kommandozeile konfiguriert werden. Hierzu wird der `ipv6` Befehl mit zwei unterschiedlichen Parametern aufgerufen, zuerst "`ipv6 adu 4/fec0:1::6`", um die Unicast-Adresse dem Interface hinzuzufügen (`adu` = add unicast), und danach "`ipv6 rtu fec0:1::/64 4`" um eine Unicast-Route zu konfigurieren (`rtu` = route unicast). Bei einem dauerhaften Aufbau sollte dies in einem Startskript eingetragen werden.

Abbildung 5-15. Konfiguration und Test von Site-Local-Adressen



```

C:\>ipv6 adu 4/fec0:1::6
C:\>ipv6 rtu fec0:1::/64 4
C:\>ping6 fec0:1::4
Pinging fec0:1::4 with 32 bytes of data:
Reply from fec0:1::4%1: bytes=32 time<1ms
Reply from fec0:1::4%1: bytes=32 time<1ms
Reply from fec0:1::4%1: bytes=32 time<1ms
Reply from fec0:1::4%1: bytes=32 time<1ms
C:\>

```

5.2.4. Bereits vorhandene IPv6-fähige Dienste

Laut FAQ soll nach Installation des IPv6-Paketes der in Windows enthaltene Telnet Server IPv6 unterstützen. Diese Aussage konnte leider nicht bestätigt werden, da der Server auf keinerlei Anfragen reagierte und in dem Paket keine IPv6-fähige Version von netstat enthalten ist, und dadurch leider auch keine Kontrollmöglichkeit besteht.

5.2.5. Bereits vorhandene IPv6-fähige Clients

Nach Aktivierung des Echo-Servers auf dem Linux-Rechner war es möglich, sich mit dem Dienst zu verbinden. In Ermangelung eines funktionierenden FTP-Servers konnte die Aussage, dass der mitgelieferte FTP-Client IPv6 unterstützt, leider nicht bestätigt werden.

Der Zugriff aus dem Internet-Explorer auf den Apache unter Linux stellte auch kein Problem dar, sowohl mit explizit angegebenen Adressen als auch über den im DNS eingetragenen Namen war der Webserver erreichbar.

Abbildung 5-16. Zugriff auf Apache-Webserver per Link-Local-Adresse



Abbildung 5-17. Zugriff auf Apache-Webserver per Site-Local-Adresse



5.2.6. Schritte zu weiteren IPv6-fähigen Diensten und Clients

Da die vorhandenen Dienste und Clients ausreichend waren um die grundlegende Funktion zu prüfen wurden keine weiteren Programme installiert.

5.3. Microsoft Windows XP

5.3.1. Zustand nach der Installation

Unmittelbar nach der Installation ist die IPv6-Unterstützung von Windows XP nicht aktiviert, aber bereits installationsbereit.

5.3.2. Schritte zu einem IPv6-fähigen Netzwerkstack

Um die nach der Installation vorhandene Unterstützung für IPv6 zu aktivieren muss nur einmal der Befehl "ipv6 install" aufgerufen werden. Nach einem Neustart ist dann der IPv6-Stack aktiviert, was durch den Aufruf von "ipv6 if" überprüft werden kann.

Abbildung 5-18. Ausgabe von "ipv6 if" nach Installation

```

C:\Dokumente und Einstellungen\Administrator>ipv6 if
Schnittstelle 4: Ethernet: LAN-Verbindung
verwendet Umgebungsentdeckung
verwendet Routersuche
Verbindungsschichtadresse: 00-10-5a-6b-e3-e0
  preferred link-local fe80::210:5aff:fe6b:e3e0, Gültigkeitsdauer infinite
  Multicast interface-local ff01::1, 1 refs, kann nicht berichtet werden
  Multicast link-local ff02::1, 1 refs, kann nicht berichtet werden
  Multicast link-local ff02::1:ff6b:e3e0, 1 refs, letzter Bericht
Verbindungs-MTU 1500 (Wahrer Verbindungs-MTU 1500)
  aktuelles Abschnittsmaximum 128
  erreichbare Zeit 41500ms (Basis 30000ms)
  Intervall für erneute Übertragung 1000ms
  DAD-Übertragungen 1
Schnittstelle 3: Pseudoschnittstelle für Ipv6-nach-Ipv4-Tunneling
verwendet keine Umgebungsentdeckung
verwendet nicht Routersuche
Verbindungs-MTU 1280 (Wahrer Verbindungs-MTU 65515)
  aktuelles Abschnittsmaximum 128
  erreichbare Zeit 32000ms (Basis 30000ms)
  Intervall für erneute Übertragung 1000ms
  DAD-Übertragungen 0
Schnittstelle 2: Automatische Tunneling-Pseudoschnittstelle
verwendet keine Umgebungsentdeckung
verwendet nicht Routersuche
Router-Verbindungsschichtadresse: 0.0.0.0
EUI-64 eingebettete IPv4-Adresse: 0.0.0.0
  preferred link-local fe80::5efe:172.17.17.5, Gültigkeitsdauer infinite
Verbindungs-MTU 1280 (Wahrer Verbindungs-MTU 65515)
  aktuelles Abschnittsmaximum 128
  erreichbare Zeit 33000ms (Basis 30000ms)
  Intervall für erneute Übertragung 1000ms
  DAD-Übertragungen 0
Schnittstelle 1: Pseudoschnittstelle für Loopback
verwendet keine Umgebungsentdeckung
verwendet nicht Routersuche
Verbindungsschichtadresse:
  preferred link-local ::1, Gültigkeitsdauer infinite
  preferred link-local fe80::1, Gültigkeitsdauer infinite
Verbindungs-MTU 1500 (Wahrer Verbindungs-MTU 4294967295)
  aktuelles Abschnittsmaximum 128
  erreichbare Zeit 38000ms (Basis 30000ms)
  Intervall für erneute Übertragung 1000ms
  DAD-Übertragungen 0

C:\Dokumente und Einstellungen\Administrator>

```

Ähnlich wie unter Windows 2000 wurden bestimmte IPv6-Adressen automatisch zugewiesen. Neben der Link-Lokalen Adresse für das Ethernet-Interface und den bereits bei Windwos 2000 besprochenen Multicast-Adressen wurde ausserdem die Multicast-Adresse ff01::1 konfiguriert, die alle Interfaces auf dem lokalen Rechner anspricht.

Abbildung 5-19. Test der Installation per ping6

```

Eingabeaufforderung
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

c:\>ping6 fe80::210:5aff:fe6b:e3e0%4

Pinging fe80::210:5aff:fe6b:e3e0%4 wird angepingt
von fe80::210:5aff:fe6b:e3e0%4 mit 32 Bytes Daten:

Antwort von fe80::210:5aff:fe6b:e3e0%4: Bytes=32 Zeit<1ms
Antwort von fe80::210:5aff:fe6b:e3e0%4: Bytes=32 Zeit<1ms
Antwort von fe80::210:5aff:fe6b:e3e0%4: Bytes=32 Zeit<1ms
Antwort von fe80::210:5aff:fe6b:e3e0%4: Bytes=32 Zeit<1ms

Ping-Statistik für fe80::210:5aff:fe6b:e3e0%4
    Pakete: Gesendet = 4, Empfangen = 4, Verloren = 0 (0% Verlust),
    Ungefähre Zeitangaben in Millisekunden:
        Minimum = 0ms, Maximum = 0ms, Mittelwert = 0ms

c:\>ping6 fe80::210:5aff:fe6b:e371%4

Pinging fe80::210:5aff:fe6b:e371%4 wird angepingt
von fe80::210:5aff:fe6b:e371%4 mit 32 Bytes Daten:

Antwort von fe80::210:5aff:fe6b:e371%4: Bytes=32 Zeit<1ms
Antwort von fe80::210:5aff:fe6b:e371%4: Bytes=32 Zeit<1ms
Antwort von fe80::210:5aff:fe6b:e371%4: Bytes=32 Zeit<1ms
Antwort von fe80::210:5aff:fe6b:e371%4: Bytes=32 Zeit<1ms

Ping-Statistik für fe80::210:5aff:fe6b:e371%4
    Pakete: Gesendet = 4, Empfangen = 4, Verloren = 0 (0% Verlust),
    Ungefähre Zeitangaben in Millisekunden:
        Minimum = 0ms, Maximum = 0ms, Mittelwert = 0ms

c:\>
    
```

5.3.3. Konfiguration von statischen Site-Local Adressen

Die Konfiguration unter Windows XP funktioniert genauso wie unter Windows 2000. Für diesen Rechner habe ich allerdings die Adresse fec0:1::6 verwendet.

Abbildung 5-20. Konfiguration von Site-Local-Adressen

```

Eingabeaufforderung
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

c:\>ipv6 adu 4/fec0:1::5
c:\>ipv6 rtu fec0:1::/64 4
c:\>_
    
```

5.3.4. Bereits vorhandene IPv6-fähige Dienste

Nach offiziellen Angaben soll auch der Telnet-Server unter Windows XP eine IPv6-Unterstützung aufweisen. Leider liess sich auch dies nicht überprüfen, da er das gleiche Verhalten wie der Windows 2000 Telnet Server zeigte.

5.3.5. Bereits vorhandene IPv6-fähige Clients

Der bei Windows XP enthaltene Telnet-Client ist voll IPv6-Fähig, und somit als Diagnosewerkzeug gut zu gebrauchen. Auch hier beherrscht der FTP-Client offiziell das IPv6-Protokoll, was sich allerdings leider aufgrund des fehlenden Servers nicht überprüfen lässt.

Solange kein Proxy Server eingestellt ist kann man auch problemlos mit dem Internet Explorer 6.0 auf IPv6-Webserver zugreifen. Allerdings wird die direkte Eingabe von IPv6-Adressen nicht mehr unterstützt, so dass eine Namensauflösung zwingend notwendig ist. Diese kann wahlweise über die hosts-Datei oder über einen DNS-Server erfolgen, der die entsprechenden AAAA-Anfragen beantworten kann.

Abbildung 5-21. Zugriff auf Apache-Webserver per Site-Local-Adresse



5.3.6. Schritte zu weiteren IPv6-fähigen Diensten und Clients

Da die vorhandenen Dienste und Clients ausreichend waren um die grundlegende Funktion zu prüfen wurden keine weiteren Programme installiert.

5.4. Unterstützung durch Cisco-Router

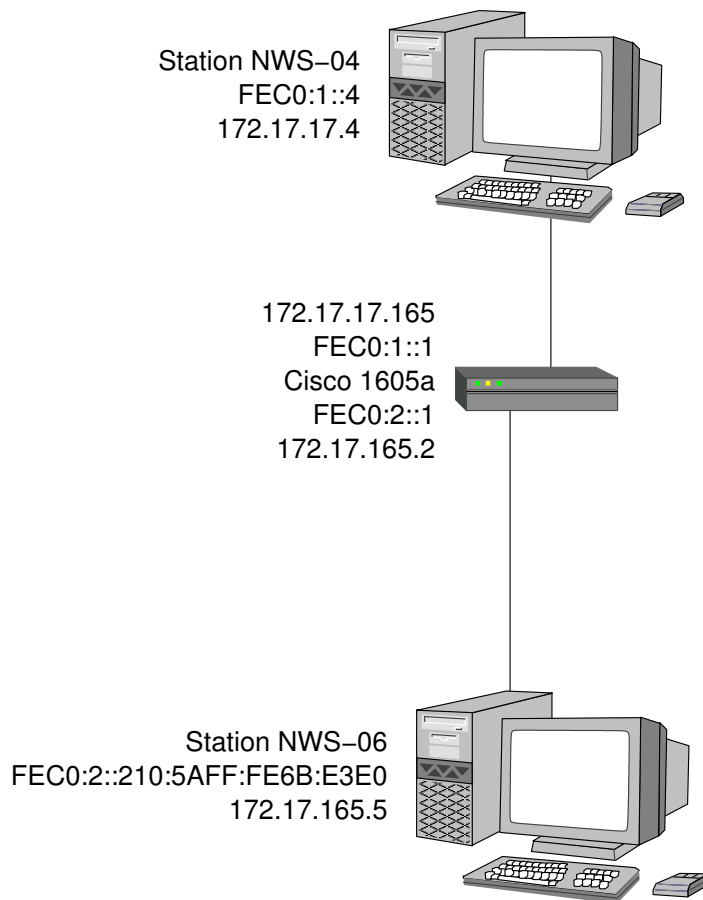
5.4.1. Voruntersuchung

Cisco als führendem Router-Hersteller kommt eine tragende Rolle bei der Umsetzung und Verbreitung von IPv6 zu. Daher ist es ein Zeichen für die fortschreitende Vorbereitung auf eine Umstellung, dass Ciscos IOS (Internetworking Operating System) seit Version 12.2(2)T offizielle Unterstützung für IPv6 bietet. Ansonsten würde es zu einem Henne-Ei-Problem kommen, dass kein Internet Provider IPv6 anbietet, weil seine Router dies nicht unterstützen, und kein Router es unterstützt, weil es niemand einsetzen würde.

5.4.2. Testaufbau

Für den ersten Testaufbau mit einem Cisco Router als Testobjekt ergab sich die im folgenden abgebildete Struktur. Im Rechnernetze-Labor der FH Wedel wird generell das Netzwerk 172.17.17.0/24 verwendet, was sich in der Adressvergabe widerspiegelt. Ausserdem ist der Default-Router mit einer statischen Route auf das Netzwerk 172.17.165.0/24 über den für diesen Aufbau verwendeten Router mit der Adresse 172.17.17.165 konfiguriert.

Abbildung 5-22. Testaufbau IPv6-Routing



Um als erstes die grundlegende Funktion von IPv6 sicherzustellen, wurde der Test-Router, bei dem es sich um das Modell Cisco 1605 handelt, mit statischen Site-Local-Adressen konfiguriert. Dabei wurde das eine Interface, Ethernet0, auf eine Site-Local-Adresse des Netzwerks fec0:1::/64 konfiguriert, das bereits für die anderen Tests verwendet wurde. Das andere Interface, Ethernet1, bekam die feste Adresse fec0:2::1/64, und wurde ausserdem dahingehend konfiguriert, dass der Netzwerk-Präfix per Advertisement bekanntgegeben wird, so dass Rechner, die sich in dem Netzwerk befinden, das über dieses Interface angebunden ist, keiner manuellen Konfiguration bedürfen. Ausserdem wurde das Routing von IPv6-Paketen zwischen den Interfaces des Routers aktiviert.

Abbildung 5-23. Cisco-Konfiguration für Site-Local-Adressen

```
nws-04:~# telnet 172.17.17.165
Trying 172.17.17.165...
Connected to 172.17.17.165.
Escape character is '^['.
```

```
User Access Verification
```

```
Password:
1605a>enable
Password:
1605a#conf t
Enter configuration commands, one per line. End with CNTL/Z.
1605a(config)#interface ethernet1
1605a(config-if)#ip address 172.17.165.1 255.255.255.0
1605a(config-if)#no down
1605a(config)#ipv6 unicast-routing
1605a(config)#interface ethernet0
1605a(config-if)#ipv6 address fec0:1::1/64
1605a(config-if)#interface ethernet 1
1605a(config-if)#ipv6 address fec0:2::1/64
1605a(config-if)#ipv6 nd prefix fec0:2::/64 autoconfig
1605a(config-if)#end
1605a#show ipv6 interface
Ethernet0 is up, line protocol is up
  IPv6 is enabled, link-local address is FE80::202:4BFF:FE5B:3F5A
  Global unicast address(es):
    FEC0:1::1, subnet is FEC0:1::/64
  Joined group address(es):
    FF02::1
    FF02::2
    FF02::1:FF00:1
    FF02::1:FF5B:3F5A
  MTU is 1500 bytes
  ICMP error messages limited to one every 100 milliseconds
  ICMP redirects are enabled
  ND DAD is enabled, number of DAD attempts: 1
  ND reachable time is 30000 milliseconds
  ND advertised reachable time is 0 milliseconds
  ND advertised retransmit interval is 0 milliseconds
  ND router advertisements are sent every 200 seconds
  ND router advertisements live for 1800 seconds
  Hosts use stateless autoconfig for addresses.
Ethernet1 is up, line protocol is down
  IPv6 is enabled, link-local address is FE80::202:4BFF:FE5B:3F5B [TENTATIVE]
  Global unicast address(es):
    FEC0:2::1, subnet is FEC0:2::/64 [TENTATIVE]
  Joined group address(es):
    FF02::1
    FF02::2
    FF02::1:FF00:1
    FF02::1:FF5B:3F5B
  MTU is 1500 bytes
  ICMP error messages limited to one every 100 milliseconds
  ICMP redirects are enabled
  ND DAD is enabled, number of DAD attempts: 1
  ND reachable time is 30000 milliseconds
  ND advertised reachable time is 0 milliseconds
  ND advertised retransmit interval is 0 milliseconds
  ND router advertisements are sent every 200 seconds
  ND router advertisements live for 1800 seconds
```

```
Hosts use stateless autoconfig for addresses.  
1605a#exit
```

Wird nun eine Arbeitsstation an dieses Interface angeschlossen - ich habe hierfür den Windows XP Testrechner ausgewählt - konfiguriert sich dieser nach einer gewissen Zeit automatisch. Zur Kontrolle wurde dann zuerst ein ping6 in das andere Netzwerk geschickt, und, nachdem die grundlegende Funktion sichergestellt war, auch die Homepage des Linux-Servers per IPv6 aufgerufen. Um zu überprüfen, dass die Verbindung auf dem erwarteten Weg zustande gekommen ist, wurde danach die Ausgabe des netstat-Befehls kontrolliert. Hierbei ist es wichtig festzustellen, dass die Ausgabe von netstat noch nicht optimal ist. Dadurch, dass die Portnummer von der IP-Adresse durch einen Doppelpunkt getrennt wird, erscheint es so, als sei die Portnummer noch ein Teil der IPv6-Adresse, was vor allem dann zu Missverständnissen führen kann, wenn die IPv6-Adresse zu lang für das Ausgabefeld ist und verkürzt werden muss.

Abbildung 5-24. Kontrolle der IPv6-Verbindung über den Cisco-Router - WinXP-Seite

Abbildung 5-25. Kontrolle der IPv6-Verbindung über den Cisco-Router - Linux-Seite

```
nws-04:~# netstat -n  
Active Internet connections (w/o servers)  
Proto Recv-Q Send-Q Local Address           Foreign Address         State  
tcp          0      0 fec0:1::4:80           fec0:2::210:5aff:f:1029 ESTABLISHED
```

Kapitel 6. Integration von IPv6 in bestehende Infrastrukturen

6.1. 6to4 nach rfc3056

Besitzt ein Standort mindestens eine global gültige IPv4-Adresse, so kann dieser nach rfc3056 angebunden werden. Dieser Standard beschreibt, wie aus einer vorhandenen IPv4-Adresse, idealerweise der des äussersten Routers, ein IPv6-Präfix gebildet werden kann. Hierbei wird die Top-Level-Aggregation-ID 0x2002 verwendet, mit der IP-Adresse des Routers für die folgenden 32 Bit, also in den Feldern RES und NLA-ID der Unicast-Adressdefinition. Aus der IPv4-Adresse 172.17.17.165 wird dann also der IPv6-Präfix 2002:ac11:11a5::/48.

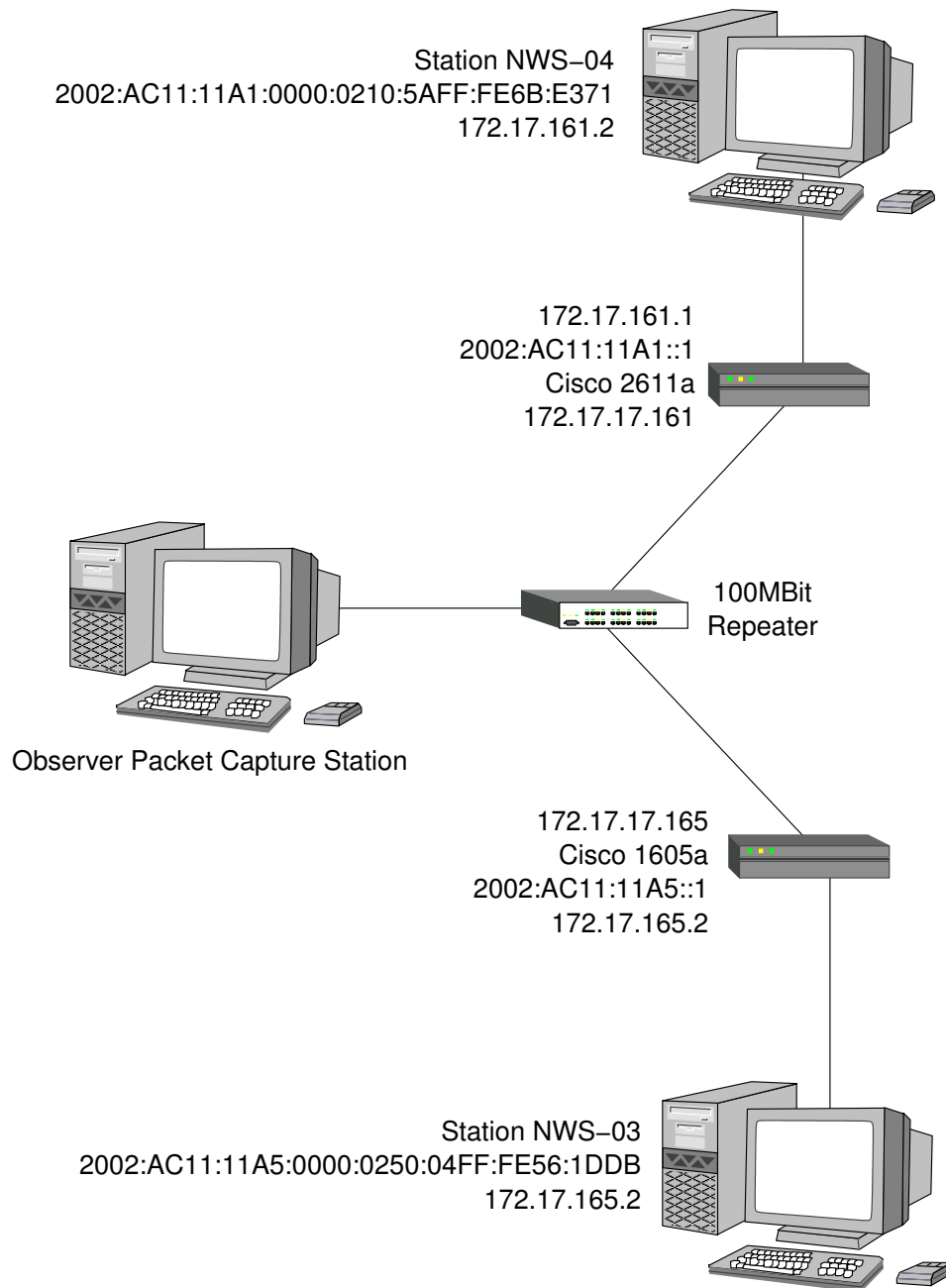
Ziel hierbei ist, dass ein Rechner in einem beliebigen IPv6-Netzwerk dieses Netz über einen automatisch konfigurierten Tunnel erreichen kann. Die Vorgehensweise hierbei ist es, dass jeder Router, der sowohl über IPv6 als auch über IPv4 adressierbar ist, ein IPv6 Router Advertisement für das hierfür vorgesehene Präfix 2002 mit maximaler Gültigkeit versendet. Durch Filterung dieser Broadcasts an den Standortgrenzen wird sichergestellt, dass hierbei keine extremen Routing-Umwege entstehen.

Versendet nun ein Rechner ein IPv6-Paket zu einer IPv6-Adresse mit 6to4-Präfix, wird dieses zu dem nächstgelegenen 6to4-Router geleitet. Dieser verpackt das IPv6-Paket in ein IPv4-Paket, dessen Quelladresse seine externe IPv4-Adresse ist, und dessen Zieladresse die aus der 6to4-Adresse gewonnene IPv4-Adresse des Tunnelendpunktes ist. Diese Pakete werden regulär via IPv4 zum Zielrouter geleitet. Dieser packt das enthaltene IPv6-Paket wieder aus und leitet es entsprechend seiner IPv6-Routingtabelle weiter zu dem Zielrechner.

Diese Art der Anbindung bietet sich an, wenn der genutzte Provider noch keine IPv6-Anbindung anbietet, aber bereits IPv6-Server eingerichtet werden sollen, die auch global erreichbar sein müssen.

Der Testaufbau hierfür stellt sich so dar, dass ein zweiter Cisco-Router zum Einsatz kommt. Zwischen den beiden Routern befindet sich das reine IPv4-Netzwerk, also das Netzwerk des simulierten Providers. Zur Kontrolle der Kommunikation zwischen den beiden Routern kommt eine Arbeitsstation unter Windows XP zum Einsatz, auf der die Software "Observer" von Network Instruments installiert ist. Hiermit lassen sich die Pakete, die im reinen IPv4-Transfernetz ausgetauscht werden, protokollieren. Die Router sind über das Netzwerk 172.17.17.0/24 miteinander verbunden.

Abbildung 6-1. Testaufbau 6to4



Der aus dem vorigen Aufbau bereits bekannte Router 1605a ist, was die IPv4-Konfiguration betrifft, vollkommen unverändert. Die IPv6-Konfiguration stellt sich so dar, dass das Interface Ethernet1, also das interne, mit dem IPv6-Netz verbundene Interface, mit der Adresse 2002:AC11:11A5::1/64 konfiguriert wird. Dieser Präfix wird auch im Netzwerk bekanntgegeben. Des weiteren wird ein 6to4-Tunnel an das mit dem "Internet" verbundene Interface Ethernet0 gebunden.

Abbildung 6-2. 6to4-Konfiguration c1605a

```

!
version 12.2
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname 1605a
!
enable secret 5 *****
enable password *****
!
ip subnet-zero
!
ipv6 unicast-routing
!
!
!
!
interface Tunnel0
 no ip address
 no ip redirects
 ipv6 unnumbered Ethernet0
 tunnel source Ethernet0
 tunnel mode ipv6ip 6to4
 tunnel path-mtu-discovery
!
interface Ethernet0
ip address 172.17.17.165 255.255.255.0
!
interface Ethernet1
 ip address 172.17.165.1 255.255.255.0
 ipv6 address 2002:AC11:11A5::1/64
 ipv6 nd prefix 2002:AC11:11A5::/64
!
ip default-gateway 172.17.17.254
ip classless
ip route 0.0.0.0 0.0.0.0 172.17.17.254
ip route 172.17.161.0 255.255.255.0 172.17.17.161
!
!
ipv6 route 2002::/16 Tunnel0

```

Neu hinzugekommen ist ein Cisco-Router vom Typ 2611. Dieser hat in Richtung "Internet" das Interface Ethernet0/0 mit der IP-Adresse 172.17.17.161, und wird dementsprechend auf dem zweiten Interface Ethernet0/1 auf die Adresse 2002:AC11:11A1::1/64 konfiguriert. Auch hier wird natürlich das Präfix Announcement aktiviert. Analog zum Cisco 1605a bekommt das Interface Ethernet0/1 die IPv4-Adresse 172.17.161.1.

Abbildung 6-3. 6to4-Konfiguration c2611b

```
!  
version 12.2  
service timestamps debug uptime  
service timestamps log uptime  
no service password-encryption  
!  
hostname c2611b  
!  
enable secret 5 *****  
enable password *****  
!  
ip subnet-zero  
!  
!  
ip audit notify log  
ip audit po max-events 100  
ipv6 unicast-routing  
!  
!  
!  
interface Tunnel0  
no ip address  
no ip redirects  
ipv6 unnumbered Ethernet0/0  
tunnel source Ethernet0/0  
tunnel mode ipv6ip 6to4  
tunnel path-mtu-discovery  
!  
interface Ethernet0/0  
ip address 172.17.17.161 255.255.255.0  
no ip mroute-cache  
!  
interface Ethernet0/1  
ip address 172.17.161.1 255.255.255.0  
ipv6 address 2002:AC11:11A1::1/64  
ipv6 nd prefix 2002:AC11:11A1::/64  
!  
ip default-gateway 172.17.17.254  
ip classless  
ip route 172.17.165.0 255.255.255.0 172.17.17.165  
!  
!  
ipv6 route 2002::/16 Tunnel0
```

Für die Endstationen wurde als Betriebssystem Debian GNU/Linux gewählt. Hier kamen die Arbeitsstationen nws-04 und nws-03 zum Einsatz. Beide Rechner haben sich innerhalb einer gewissen Zeitspanne selber konfiguriert.

Abbildung 6-4. Interface-Konfiguration NWS-04

```
nws-04:~# ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 00:10:5A:6B:E3:71
          inet addr:172.17.161.2  Bcast:172.17.161.255  Mask:255.255.255.0
          inet6 addr: 2002:ac11:11a1:0:210:5aff:fe6b:e371/64 Scope:Global
          inet6 addr: fe80::210:5aff:fe6b:e371/10 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:4105 errors:0 dropped:0 overruns:0 frame:0
          TX packets:4207 errors:0 dropped:0 overruns:0 carrier:144
          collisions:0 txqueuelen:100
          RX bytes:479531 (468.2 KiB)  TX bytes:313256 (305.9 KiB)
          Interrupt:10 Base address:0xb400
```

Abbildung 6-5. Interface-Konfiguration NWS-03

```
nws-03:~# ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 00:50:04:56:1D:DB
          inet addr:172.17.165.2  Bcast:172.17.165.255  Mask:255.255.255.0
          inet6 addr: fe80::250:4ff:fe56:1ddb/10 Scope:Link
          inet6 addr: 2002:ac11:11a5:0:250:4ff:fe56:1ddb/64 Scope:Global
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:4558 errors:0 dropped:0 overruns:0 frame:0
          TX packets:5354 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:100
          RX bytes:529991 (517.5 KiB)  TX bytes:388478 (379.3 KiB)
          Interrupt:10 Base address:0xb400
```

Abbildung 6-6. Verbindungstest von NWS-04 zu NWS-03

```
nws-04:~# ping6 -c4 2002:ac11:11a5:0:250:4ff:fe56:1ddb
PING 2002:ac11:11a5:0:250:4ff:fe56:1ddb(2002:ac11:11a5:0:250:4ff:fe56:1ddb)
from 2002:ac11:11a1:0:210:5aff:fe6b:e371 : 56 data bytes
64 bytes from 2002:ac11:11a5:0:250:4ff:fe56:1ddb: icmp_seq=1 ttl=62 time=7.15 ms
64 bytes from 2002:ac11:11a5:0:250:4ff:fe56:1ddb: icmp_seq=2 ttl=62 time=7.09 ms
64 bytes from 2002:ac11:11a5:0:250:4ff:fe56:1ddb: icmp_seq=3 ttl=62 time=7.14 ms
64 bytes from 2002:ac11:11a5:0:250:4ff:fe56:1ddb: icmp_seq=4 ttl=62 time=7.14 ms

--- 2002:ac11:11a5:0:250:4ff:fe56:1ddb ping statistics ---
4 packets transmitted, 4 received, 0% loss, time 3031ms
rtt min/avg/max/mdev = 7.097/7.134/7.155/0.105 ms
```

Abbildung 6-7. Verbindungstest von NWS-03 zu NWS-04

```
nws-03:~# ping6 -c 4 2002:ac11:11a1:0:210:5aff:fe6b:e371
PING 2002:ac11:11a1:0:210:5aff:fe6b:e371(2002:ac11:11a1:0:210:5aff:fe6b:e371)
from 2002:ac11:11a5:0:250:4ff:fe56:1ddb : 56 data bytes
64 bytes from 2002:ac11:11a1:0:210:5aff:fe6b:e371: icmp_seq=1 ttl=62 time=6.94 ms
64 bytes from 2002:ac11:11a1:0:210:5aff:fe6b:e371: icmp_seq=2 ttl=62 time=6.98 ms
64 bytes from 2002:ac11:11a1:0:210:5aff:fe6b:e371: icmp_seq=3 ttl=62 time=7.01 ms
64 bytes from 2002:ac11:11a1:0:210:5aff:fe6b:e371: icmp_seq=4 ttl=62 time=6.81 ms
```

```
--- 2002:ac11:11a1:0:210:5aff:fe6b:e371 ping statistics ---  
4 packets transmitted, 4 received, 0% loss, time 3031ms  
rtt min/avg/max/mdev = 6.819/6.942/7.019/0.112 ms
```

Um sicherzustellen, dass zwischen den beiden Routern wirklich nur IPv4-Pakete übertragen werden, wurde deren Kommunikation mit der besagten Observer Arbeitsstation kontrolliert. Hieraus war eindeutig herauszulesen, dass alles genau so verlief wie es die theoretischen Vorüberlegungen erwarten liessen.

6.2. IPv6-to-IPv4 nach rfc3142

Soll eine Gruppe von Client-Rechnern, die nur IPv6 unterstützen, auch auf IPv4-Server zugreifen können, so wird eine Protokollumsetzung benötigt. Diese ist im rfc3142 definiert. Hierbei wird ein IPv6-Präfix reserviert, unterhalb dessen die IPv4-Adressen abgebildet werden, und ein Nameserver eingerichtet, der die DNS-Anfragen via IPv4 abwickelt und die Resultate in diesem Adressraum abbildet.

Greift dann ein IPv6-Client-Programm auf die so aufgelöste Adresse zu, wird vom Router eine Protokollumsetzung von IPv6 auf IPv4 vorgenommen, und die so erzeugten reinen IPv4-Pakete regulär geroutet.

Diese Methode ist bereits mehrfach von der Internet Engineering Taskforce (IETF) angewandt worden, die hierbei eingesetzten Programme waren der `totd` (Trick or Treat Daemon) als Nameserver und der `faithd` als Protokoll-Umsetzer.

6.3. NAT-PT nach rfc2766

Network Address Translation und Protocol Translation ist eine weitere Methode, um reinen IPv6-Rechnern den Zugriff auf reine IPv4-Hosts zu ermöglichen. Die Vorgehensweise ähnelt der von NAT bei IPv4.

Es wird ein Adresspool von IPv4-Adressen definiert, die dann dynamisch mit IPv6-Adressen verknüpft werden. Hierbei wird also für jeden Host, der IPv4-Verbindungen nutzt, eine eigene IPv4-Adresse benötigt.

Eine andere Form, die Network Address Port Translation und Protocol Translation (NAPT-PT) erweitert dieses Konzept, indem zusätzlich die Portnummern angepasst werden. Dadurch können mehrere IPv6-Rechner auf eine einzige IPv4-Adresse abgebildet werden, der Einsatz ist allerdings auf Protokolle beschränkt die mit Portnummern arbeiten.

Diese Konzepte haben allerdings die gleichen Nachteile die bereits NAT bei IPv4 mit sich bringt. Daher sollte diese Art der Adressumsetzung möglichst vermieden werden, und findet beispielsweise ihre Grenzen bei allen Protokollen, die mit eingebetteten Adressen arbeiten. Dann müssen auch diese Informationen vom Gateway umgeschrieben werden.

6.4. Kommunikation zwischen IPv6-Rechnern über IPv4 (IPv4-compatible IPv6)

IPv4-compatible IPv6 kommt immer dann zum Einsatz, wenn zwar beide an einer Kommunikation beteiligten Hosts IPv6-fähig sind, ihre Verbindung allerdings eine reine IPv4-Verbindung ist.

Die ausgetauschten Pakete sind dann IPv6-Pakete, die in IPv4-Paketen verpackt sind. Damit dies funktioniert müssen natürlich beide an der Kommunikation beteiligten Hosts diese Technik beherrschen.

Unter Windows ist dies automatisch konfiguriert, das heisst man kann ohne weitere Konfiguration einen anderen Rechner über IPv4-compatible IPv6-Adressen ansprechen. Um dies auch unter Linux zu können muss das entsprechende Tunnel-Interface aktiviert werden, indem der Befehl "ifconfig sit0 up" aufgerufen wird.

Abbildung 6-8. Aktivierung des Tunnel-Devices sit0

```
nws-04:~# ifconfig sit0 up
nws-04:~# ifconfig sit0
sit0      Link encap:IPv6-in-IPv4
          inet6 addr:  ::172.17.17.4/96 Scope:Compat
          inet6 addr:  ::127.0.0.1/96 Scope:Unknown
          UP RUNNING NOARP  MTU:1480  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 b)  TX bytes:0 (0.0 b)

nws-04:~# ping6 -c 4  ::172.17.17.6
PING  ::172.17.17.6( ::172.17.17.6) from  ::172.17.17.4 : 56 data bytes
64 bytes from  ::172.17.17.6: icmp_seq=1 ttl=128 time=0.249 ms
64 bytes from  ::172.17.17.6: icmp_seq=2 ttl=128 time=0.255 ms
64 bytes from  ::172.17.17.6: icmp_seq=3 ttl=128 time=0.215 ms
64 bytes from  ::172.17.17.6: icmp_seq=4 ttl=128 time=0.200 ms

---  ::172.17.17.6 ping statistics ---
4 packets transmitted, 4 received, 0% loss, time 2999ms
rtt min/avg/max/mdev = 0.200/0.229/0.255/0.029 ms
```

Da die IPv6-Adresse dieses Devices direkt aus den konfigurierten IPv4-Adressen hervorgeht, werden auch hier die entsprechenden Adressen automatisch konfiguriert.

6.5. Verbindung zwischen IPv4-Rechnern und IPv6-Rechnern (IPv4-mapped IPv6)

Kontaktiert ein reiner IPv4-Rechner einen reinen IPv6-Dienst auf einem Rechner mit einem dualen Stack, oder umgekehrt, so wird die IPv4-Adresse auf eine IPv4-Mapped IPv6-Adresse abgebildet, indem ihr 0::FFFF/96 vorangestellt wird. Aus der IPv4-Adresse 172.17.17.165 wird also ::FFFF:AC11:11A5. Diese Umsetzung geschieht vollständig innerhalb des dualen IP-Stacks, zwischen den Rechnern läuft also eine reine IPv4-Kommunikation.

Dadurch muss ein Dienst keinen Unterschied zwischen Verbindungen von IPv4- und IPv6-Clients machen, sondern kann alle eingehenden Anforderungen gleich behandeln, was den Übergang zu IPv6 erleichtert.

Kapitel 7. Fazit

Abschliessend ist festzustellen, dass sich IPv6 eindeutig auf dem Weg zur Benutzbarkeit befindet. Vor einem praktischen Einsatz müssen allerdings noch ein paar Elemente, die zu einer vollständigen IPv6-Infrastruktur noch fehlen, bis zur produktiven Einsatzreife entwickelt werden.

Als Hauptschwachstelle ist bisher die Unterstützung des Domain Name System zu bezeichnen. So gibt es beispielsweise bisher keine Einträge um IPv6-Adressen, die nach dem 6to4 Schema vergeben werden, wieder zu Namen aufzulösen. Ausserdem ist noch mehr Entwicklungsarbeit in Hinsicht auf die statische Autoconfiguration per DHCPv6 notwendig.

Auch die Unterstützung durch Serverdienste ist noch verbesserungswürdig. Unter Windows sind nur sehr wenige Dienste IPv6-fähig, und um unter Linux Dienste über IPv6 anzubieten muss meist ein Patch eingespielt und das Programm neu kompiliert oder eine instabile Testversion verwendet werden.

Nichtsdestotrotz sollte meiner Meinung nach jeder, der für den Betrieb eines Netzwerkes verantwortlich ist, vor diesem Thema nicht die Augen verschliessen und die Entwicklung aufmerksam beobachten.

Anhang A. Quellen

RFC-Editor (<http://www.rfc-editor.org>)

Tabelle A-1. RFCs zu IPv6

Nummer	Titel
rfc791	Internet Protocol
rfc1886	DNS Extensions to support IP version 6
rfc1924	A Compact Representation of IPv6 Addresses
rfc2050	Internet Registry IP Allocation Guidelines
rfc2373	IP Version 6 Addressing Architecture
rfc2374	An IPv6 Aggregatable Global Unicast Address Format
rfc2460	Internet Protocol, Version 6 (IPv6)
rfc2461	Neighbor Discovery for IP Version 6 (IPv6)
rfc2462	IPv6 Stateless Address Autoconfiguration
rfc2463	Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6)
rfc2732	Format for Literal IPv6 Addresses in URLs
rfc2766	Network Address Translation - Protocol Translation (NAT-PT)
rfc2874	DNS Extensions to Support IPv6 Address Aggregation and Renumbering
rfc2893	Transition Mechanisms for IPv6 Hosts and Routers
rfc3056	Connection of IPv6 Domains via IPv4 Clouds
rfc3142	An IPv6-to-IPv4 Transport Relay Translator
rfc3152	Delegation of IP6.ARPA

6bone (<http://www.6bone.net>)

freenet6 (<http://www.freenet6.net>)

6bone (<http://www.6bone.net>)

KAME-Projekt (<http://www.kame.net>)