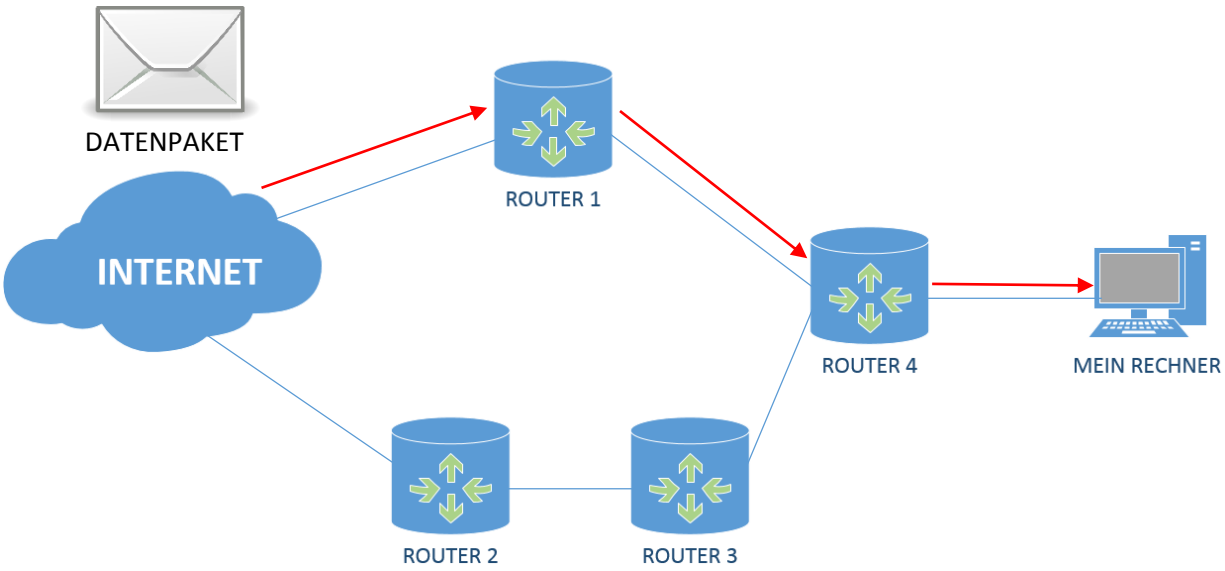
Internet Protokoll Version 6

1. Das IP-Protokoll

Bevor wir uns tiefgehender mit dem Internet Protokoll in der Version 6 beschäftigen, sollten wir vorab klären, was IP ist und wie es allgemein funktioniert.
Das IP-Protokoll ist für den Datentransport zuständig. Dieser läuft im Internet so ab, dass die zu versendenden Daten in gleich große Teilstücke zerhackt und fortlaufend nummeriert werden. Dabei werden die Teilstücke, oder auch Datenpakete, einem Empfänger und Absender zugeordnet und mit einer Prüfsumme, um die Gültigkeit dieses Pakets zu bewahren, versehen.
Im Internet vorhandene Router finden anhand der Empfängeradresse den Weg zum Empfänger, dabei greifen die Router auf sogenannte Routingtabellen zurück, die das Finden der optimalen Route ermöglichen. Dabei kennt aber jeder Router nur den Weg zum nächsten Router, das heißt dass die gesamte Route immer nur aus Teilrouten besteht. Dadurch können die Router automatisch auf Störungen oder Überlastungen reagieren und Alternativrouten suchen. Dies alles wird durch das IP-Protokoll unterstützt.



Auch um die Vergabe der Sender und Empfängeradressen kümmert sich das Protokoll, genauso um die gesicherte Datenübertragung. Also ob die Daten so ankommen, wie sie versendet wurden.

2. Warum IPv6?

Adressraum

Aktuell wird immer noch das Internet Protokoll in der Version 4 verwendet, welches bereits in den 1970 Jahren entwickelt wurde. Im Laufe der Jahre wurde nun das Internet beständig größer, also die Zahl der Nutzer und der zu vergebenden Adressen stieg weiter an. Im Adressbereich des IPv4-Protokolls sind 2^{32} (4.294.967.296) unterschiedliche Adressen möglich. Klingt nach einer hohen Zahl, aber bei der steigenden Zahl an Internetbenutzern und Geräten ist dieser vermeintlich hohe Adressraum ziemlich voll. Denn mittlerweile ist es nicht selten, dass jemand einen PC, ein Notebook und auch ein Smartphone besitzt. Außerdem muss man auch auf aktuelle Trends eingehen, wie das Internet of Everything, wobei wiederum eine Vielzahl an Geräten (wie vernetzte KFZ, Smart-Home-Devices) den Zugang zu Internet benötigen und auch diese Geräte müssen mit IP-Adressen versorgt werden. Erschwerend kommt hinzu, dass in den Anfängen die IP-Adressen sehr großzügig an Firmen und Einrichtungen vergeben wurden. Oft wurden komplette Klasse B Netze, das sind 65.000 Adressen, wobei aber niemals alle verwendet werden können. Trotzdem sind diese Blöcke bereits vergeben und es ist nicht mehr möglich diese wieder zu verwenden.
Der neue Adressraum von IPv6 beträgt 340 Sextillionen ($3{,}4 \times 10^{38}$), was nun eine neue Dimension der möglichen Adressen ermöglicht.

Vereinfachung

Im Gegensatz zu IPv4 sind im neuen IPv6-Standard einige neue Features hinzugekommen welche den Umgang mit Computersystemen und deren Vernetzung erleichtern. So sind Hostsysteme in der Lage

sich automatisch und selbstständig eine eindeutige IP-Adresse zu wählen, dies wird auch Autokonfiguration genannt (Genaueres folgt weiter unten im Skript).

Headerformat

Nun ist es auch so, dass der Header des IPv6-Protokolls um einiges einfacher geworden ist. Unter anderem ist der neue Header immer exakt 40 Bytes groß, wodurch eine schnellere Verarbeitung gewährt wird. Es sind Empfänger- und Senderadressen mit jeweils 16 Bytes reserviert und 8 Bytes für allgemeine Headerinformationen. Somit ist der IPv6-Header viel einfacher und schlanker als der IPv4-Header.

Erweiterungen

Im alten IPv4-Protokoll sind die erweiterten Optionen im Basisheader integriert. Jedoch im IPv6-Protokoll sind solche Optionen in einem sogenannten „Extension Header“ eingebaut, dieser wird auch nur genutzt wenn wirklich der Bedarf an solchen Parametern auftritt. Genau durch diesen Header wird eine schnellere Verarbeitung von diesen Paketen gewährleistet. Es gibt eine Auswahl von sechs „Extension Headers“, unter anderem Headers für Routing, Mobile-IPv6, Quality-of-Service und für die Sicherheit.

Gründe für den Umstieg

Sobald global auf IPv6 Adressierung umgestiegen wird, ist es auch notwendig für den einzelnen mitzuziehen. Sonst würde man sich selbst vom weltweiten Internetverkehr, Kommunikationsnetz und der generellen Erreichbarkeit ausschließen. Grundsätzlich gilt trotzdem die goldene Regel: „Never touch a running system“. Also solange die aktuelle Struktur den Anforderungen entspricht, besteht auch kein Bedarf etwas zu ändern. Aber sobald neue Geräte angeschafft werden sollen, sollte man darauf achten, auf IPv6-fähige Geräte zu setzen.

Hauptgründe um auf IPv6 umzusteigen:

- Anstehende Erweiterung des IPv4-Netzwerk
- Der bestehende Adressraum droht auszugehen
- Ende-zu-Ende-Sicherheit für eine große Benutzerzahl
- Alte Geräte werden ausgewechselt, dabei könnte man immer bereits darauf achten, IPv6-kompatible Geräte zu kaufen
- Man will auf bereits jetzt problemlos auf IPv6 umzusteigen und nicht erst dann, wenn es gefordert wird

3. Funktion von IPv6

IPv6 Header

Zur Wiederholung gebe ich hier nochmal kurz die Grafik des IPv4-Headers an, um damit das leichtere Verständnis für die Änderungen von Version 4 auf Version 6 zu ermöglichen.
Der IPv4-Header sieht in etwa so aus:

0–3	4–7	8–11	12–15	16–18	19–23	24–27	28–31
Version	IHL	Type of Service		Gesamtlänge			
Identifikation				Flags	Fragment Offset		
TTL		Protokoll		Header-Prüfsumme			
Quell-IP-Adresse							
Ziel-IP-Adresse							
evtl. Optionen …							

Abbildung - Quelle http://de.wikipedia.org/wiki/IPv4#Header-Format

Im Gegensatz dazu der IPv6-Header, welcher diesen Aufbau besitzt:

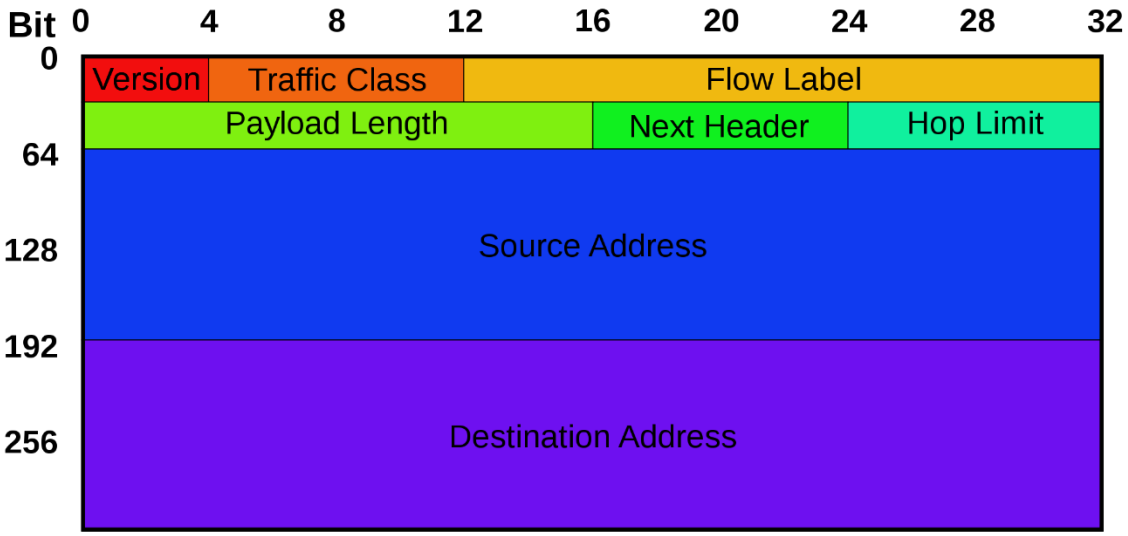


Abbildung - Quelle http://de.wikipedia.org/wiki/Datei:IPv6_header_rv1.svg

Was fällt uns dabei auf?
Es sind fünf Felder aus dem IPv4-Header entfallen und finden sich nicht mehr im IPv6-Header:

- IHL (IP Header Length), also die Information, wie lange der Header ist.
- Identifikationsfeld
- Flags
- Fragment-Offset
- Header-Prüfsumme

Die Headerlänge kann entfallen, da bei IPv6 der Header eine fixe Länge von 40 Bytes hat, somit wird auch die Gesamtlänge hinfällig.
Das Identifikationsfeld, Flags und der Fragment-Offset werden in IPv4 benötigt um mit der Fragmentierung umgehen zu können, welche auftritt, wenn große Pakete über ein Netz gesendet werden, indem nur kleine Pakete erlaubt sind. Dabei zerteilt der Router die Pakete und schreibt die benötigten Informationen in diese Felder. Der Empfänger baut dann anhand von diesen Informationen wieder die ursprünglichen Pakete zusammen, ist dabei eines fehlerhaft oder fehlend, so wird das gesamte verworfen und alles wieder neu gesendet. Dies ist aber äußerst ineffizient. Nun läuft dies in IPv6 so ab, dass der Sender lernt, die maximale Paketgröße für das jeweilige Netz anzupassen (MTU = Maximum Transmission Unit). Um die MTU bestimmen zu können, wird eine eigene Routine ausgeführt. Muss aber trotzdem Fragmentiert werden, dann verwendet IPv6 den „Extension Header“ und schreibt genau diese Felder hinein.

Die Header-Prüfsumme wurde entfernt um die Verarbeitungsgeschwindigkeit von IPv6 erhöhen zu können. Heutzutage ist die Gefahr von unentdeckten Fehler und falschgerouteten Paketen minimal, außerdem sind Prüfsummen im TCP/UDP-Header enthalten. Hinzu kommt auch noch, dass IP ein Protokoll für die korrekte Zustellung von Paketen ist und für die Nachrichtenintegrität bzw. –Richtigkeit sind Protokolle aus den höheren Sichten zuständig.

Felder im IPv6-Header

Nun stellen wir uns die Frage, was machen die einzelnen Headerfelder im Detail? Nochmal zur Erinnerung der Aufbau des IPv6-Headers:

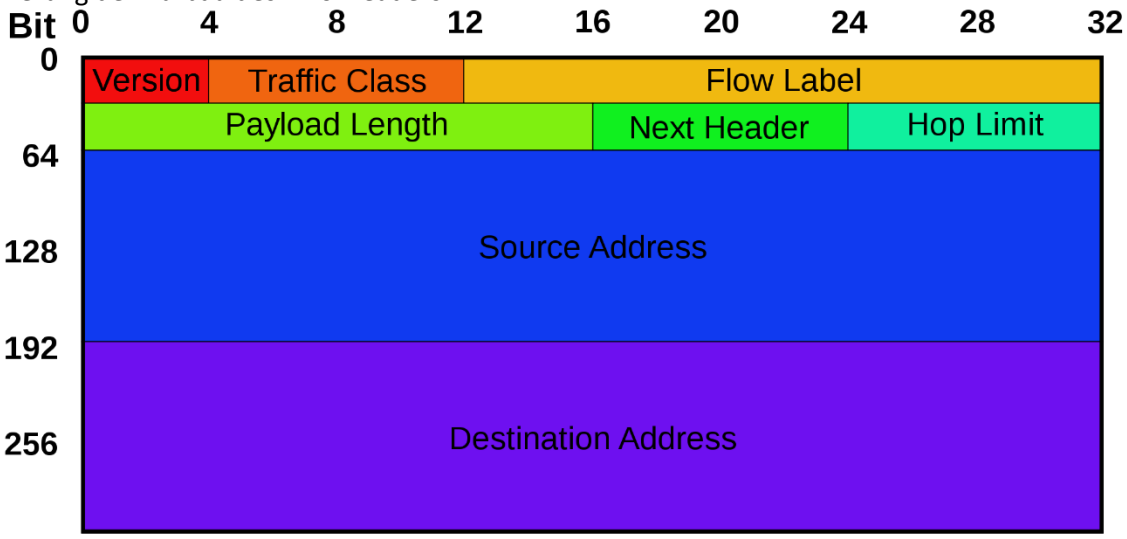


Abbildung - Quelle http://de.wikipedia.org/wiki/Datei:IPv6_header_rv1.svg

Version

Das Versionsfeld ist 4 Bit groß und enthält die Version des Protokolls. Also ist es also ein IPv6-Header steht bei Version eine 6.

Traffic Class

Das Feld Traffic Class ist 8 Bits (1 Byte) groß und ermöglicht die Unterscheidung von verschiedenen Paketklassen und Priorität der einzelnen Pakete. Auch werden Pakete, die gesonderte Verarbeitung benötigen dadurch gekennzeichnet.

Flow Label

Das Flow Label Feld ist 20 Bits groß und differenziert Pakete welche die gleiche Verarbeitung benötigen von anderen nicht spezifizierten, dies erleichtert das Prozessieren von Echtzeitdaten. Ein Sender kann nun gewisse Sequenzen von Paketen mit dieser Option setzen und signalisieren, dass diese zu einem Fluss zusammengehören. Dadurch können Router den Datenfluss verfolgen und somit Pakete, welche zum selben Fluss gehören, effizienter routen, weil nicht für jedes einzelne Paket der Header neu verarbeitet werden muss.

Payload Length

Die Payload Length ist ein Feld von der Größe von 16 Bits (2 Bytes). Dadurch wird die gesamte Payload näher spezifiziert, zum Beispiel die Länger der Daten welche nach dem Header im Paket folgen. Dabei unterscheidet sich der IPv6 Berechnungsansatz von dem des IPv4s in dem Sinne, dass IPv6 nur mehr die Daten, welche nach dem Header kommen, zur Berechnung der Länge verwendet. Dabei werden die Extension Headers mit einbezogen. Weil die Payload Length 2 Bytes groß ist, ist die maximale Paketgröße auf 64 KB limitiert.

Next Header

Dieses Feld ist ebenfalls 8 Bit (1 Byte) groß und kann als Protokolltypfeld gesehen werden, d.h. es wird angegeben welches Transportschicht-Protokoll dieses Paket verwendet, also zum Beispiel TCP oder UDP. Wird also TCP für dieses Paket verwendet, so steht im Feld Next Header „6“. Außerdem ist noch interessant, dass IPv6 und IPv4 hier die gleichen Protokollfelder verwenden. Werden aber für ein Paket Extension-Headers verwendet, dann steht in diesem Feld nicht der verwendete Protokolltyp, sondern dann steht hier der Typ des nächsten Extension-Headers drinnen. Dieser Extension-Header befindet sich zwischen dem IPv6-Header und des TCP/UDP-Headers.

Hop Limit

Ist genauso 8 Bit groß und ersetzt das von IPv4 bekannte Time-To-Live-Feld (TTL). Eine TTL gibt somit an wie viele Sekunden das Paket im Netzwerk bleiben darf bis das es zerstört wird. In IPv6 hingegen wurde das Feld in Hop Limit ungenannt. Nun wird hier die Anzahl der zulässigen Hops angegeben und nicht mehr die Anzahl der Sekunden. Jeder Knoten bzw. Router zählt diese Zahl der zulässigen Hops um eines herunter, empfängt nun ein Router ein Paket mit dem Hop Limit von 1 so wird er auf 0 herunter gezählt, das Paket verworfen und eine ICMPv6-Nachricht „Hop Limit ist während der Übertragung abgelaufen“.

Source Adresse

Enthält die IPv6-Adresse der Sendestation und ist 16 Bytes (128 Bits) groß.

Destination Adresse

Enthält die IPv6-Adresse der Empfängerstation bzw. Empfängerstationen und ist ebenfalls 16 Bytes (128 Bits) groß.

Extension Headers

Grundsätzlich gilt je einfacher der Header eines Paketes ist, desto schneller kann die Verarbeitung erfolgen. In IPv6 werden die erweiterten Paketoptionen in sogenannten zusätzlichen Extension Headers gespeichert. Nur wenn der Bedarf an Extension Headers besteht werden sie in ein Paket eingefügt, dadurch wird ein effizienter Umgang gewährleistet.
Es gibt sechs wichtige Extension-Headers:

- Hop-by-Hob-Options Header
  Für genau bzw. erweiterte Routinginformationen und Routingpfad.
- Routing Header
  Gibt eine Liste von Routern an, die das Paket auf seinem Weg passieren muss.
- Fragment Header
  Wird ein Paket über ein Netz gesendet, welches nur eine kleinere Paketgröße unterstützt, so werden die Pakete aufgeteilt und die benötigten Informationen hier drinnen gespeichert.
- Destination-Options Header
  Optionale Informationen für den Empfänger
- Authentication Header
- Encrypted-Security-Payload Header

Die Zahl der angehängten Header kann zwischen Null, einem und mehreren sein, dabei werden die Headers einfach nach dem IPv6-Header und vor dem Protokoll-Header der höheren Schicht. Mittels dem Feld Next Header (siehe Next Header) werden die jeweiligen Extension-Headers identifiziert. In jedem angefügten Extension Header ist wiederum das Next Extension – Feld, welches auf den nachfolgenden Header verweist (z.B. weiterer Extension Header oder Protokoll-Header der höheren Schicht), enthalten. Der Extension Header wird nur von den zuständigen Empfänger bzw. Empfängern bei Multicast-Adressen verarbeitet, was ja die gute Geschwindigkeit ermöglicht. Ein weiterer Vorteil ist, dass jederzeit neue Extension Headers entwickelt und angehängt werden können ohne dass der IPv6 Header neu definiert werden muss.

4. Adressierung

Schon alleine wenn man 20% der weltweiten Bevölkerung einen einigermaßen akzeptablen Internetzugang gewähren möchte, so reicht der IPv4 Adressraum nicht mehr aus. Deshalb wurde IPv6 auch eingeführt, denn hier beträgt der zulässige Adressraum 128 Bit, würde heißen, dass 2^{128} unterschiedliche, gültige Adressen möglich sind. 2^{128} unterschiedliche Adressen sind ungefähr so viel wie Sandkörner am gesamten Planten.

Adresstypen

In IPv4 wurden Unicast-, Multicast- und Broadcastadressen verwendet. Aber seit IPv6 sind Broadcastadressen hinfällig, anstelle werden Multicastadressen verwendet, weil Broadcasts in vielen Netzwerken zu Schwierigkeiten führen.
Nun gibt es in IPv6 wieder drei verschieden Adresstypen:

- Unicast
  Eine Unicastadresse identifiziert eindeutig einen IPv6 Host. Wird ein Paket via Unicast gesendet, wird es ausschließlich dem einen Empfänger zugestellt, welcher genau die angegeben Adresse enthält.
- Multicast
  Eine Multicastadresse identifiziert eindeutig eine Gruppe von IPv6 Schnittstellen/Hosts. Ein an eine Multicastadresse gesendetes Paket wird ausschließlich an alle Mitglieder dieser Multicast-Gruppe weitergeleitet.
- Anycast
  Eine Anycastadresse ist allen Schnittstellen/Hosts in einem Netz zugeordnet. Ein an eine Anycastadresse gesendetes Paket wird aber nur an eines dieser Schnittstellen zugestellt, meistens ist es jenes, welches am nächsten liegt.

Generelle Regeln

Bei IPv6 werden Adressen an Schnittstellen vergeben, also braucht jede Schnittstelle mindestens eine Unicastadresse. Auch kann eine einzige Schnittstelle mehreren IPv6-Adressen von beliebigem Typ (Uni-, Multi-, Anycast) zugeordnet werden. Somit kann natürlich eine Schnittstelle mit jeder ihrer Adressen eindeutig identifiziert werden, auch kann zum Beispiel eine Unicastadresse an mehrere Schnittstellen vergeben werden, um unteranderem die Auslastung der einzelnen zu verringern.
Auch wird in verschiedene Geltungsbereiche unterschieden, wie global oder lokal (auch als private Adressen bekannt).

Adressnotation

Eine IPv6-Adresse hat, wie bereits öfters erwähnt, 128 Bit, das sind 16 Byte. Dabei ist eine Adresse in acht, 16 Bit große, Hexadezimalblöcke aufgeteilt, welche durch einen Doppelpunkt getrennt sind.
Ein Beispiel:

2001:DB8:0000:0000:0202:B3FF:FE1E:8329

Die Adressen können vereinfacht werden, indem führende Nullen weggelassen werden können. In unserem Beispiel sieht dies so aus:

2001:DB8:0000:0000:0202:B3FF:FE1E:8329
2001:DB8:0:0:202:B3FF:FE1E:8329

Bleiben nach dieser Vereinfachung Bereiche, wo zwei Nullen aufeinander folgen, so können sie weggelassen werden. Dabei müssen wir beachten, dass zwei aufeinanderfolgende Doppelpunkte nur einmal in der gesamten Adresse vorkommen dürfen, weil der Computer immer auf 128 Bit aufrechnet. Findet nun der Computer so eine Sequenz füllt er die Adresse mit so vielen Nullen auf, wie benötigt. Dies würde aber bei mehreren nicht klappen, da nicht eindeutig hervorgeht, wo wieviel Nullen eingefügt werden sollen. Die Notation sieht so aus:

2001:DB8::202:B3FF:FE1E:8329

Integration von IPv4-Adressen in IPv6-Adressen

Wenn IPv4 und IPv6 Netze gemischt werden, gibt es einen recht einfachen Weg eine IPv4-Adresse in einer IPv6-Adresse zu integrieren. Dies funktioniert in dem man die IPv4-Adresse in die vier letzten Byte Sequenzen der IPv6-Adresse integriert.

- Nehmen wir zum Beispiel die IPv4-Adresse 192.168.0.2.
- Diese können wir auch als x:x:x:x:x:x:192.168.0.2 in IPv6-Notation anschrieben.
- x:x:x:x:x:x:192.168.0.2 könnten wir auch als 0:0:0:0:0:0:192.168.0.2 sehen, oder in vereinfachter Form als ::192.168.0.2 (wir erinnern uns, dass bei :: so viele Nullen angehängt werden um auf 128 Bit zu gelangen).
- Bevorzugt schreiben wir diese in der Hexadezimalschreibweise ::C0A8:2 an.

Globale Routing Präfixe

Zuerst sollten wir mal klären, wie so ein Präfix angegeben wird. Durch ein Präfix wird ein bestimmtes Subnetz oder eine gewisse Art von Adressen definiert. Dabei können wir die Präfix-Notation mit der Art, wie IPv4-Adressen in Classless Interdomain Routing (CIDR) geschrieben werden, vergleichen. Dabei wird einfach die Präfixlänge an die IPv6-Adresse angehängt, dies führt zu folgender Notation:

IPv6-Adresse/Präfixlänge

Wobei die Präfixlänge angibt wie viele Bits der jeweiligen Adresse, von links gelesen, die Präfix definieren. Wir können diese Art der Notation mit der Angabe der Subnetzmaske vergleichen, also welcher Bereich der Adresse zur Netzwerkidentifikation gehört. Wir merken uns, dass Präfixe ein Subnetz definieren zu dem die Schnittstellen gehören und es hilft den Routern die Pakete richtig weiterzuleiten.
Ein kleines Beispiel wird uns dies vor Augen führen:

2E78:DA53:1200:2D25::/40 (wir erinnern uns, :: steht für die restlichen Nullen)

Präfix sind also laut Angabe die ersten 40 Bits (siehe /40).

2E78	= 16 Bit
DA53	= 16 Bit
1200	= 8 Bit

Also können wir sagen, dass der Abschnitt *2E78:DA53:1200* den sogenannten Netzanteil identifiziert. Sämtliche Schnittstellen die mit *2E78:DA53:1200* beginnen gehören somit zum selben Subnetz, sprich sie befinden sich im selben Netz.
Der Rest der Adresse gibt dann an, welche Schnittstelle angesprochen wird, in diesem Fall ist *2D25::* die Identifikation im Subnetz.

Es gibt auch mittlerweile eine definierte Anzahl an Präfixen welche Adresstypen und -räume festlegen. Ich zähle hier nur eine kleine Auswahl der möglichen Präfixe auf:

- Global Unicast Adressen … 2000::/3
  - Sind weltweit eindeutige Adressen
- Link Local Adressen … FE80::/10
  - Gelten nur in lokalen, abgeschlossen Netzwerken, z.B. Heimnetzwerk
  - Findet Verwendung bei Autokonfiguration und Neighbour-Discovery
- Multicast Adressen … FF00::/8

Autokonfiguration

Ich erkläre kurz den generellen Ablauf der Autokonfiguration, um nicht den Rahmen dieser Arbeit zu sprengen.

Die Stateless Address Autoconfiguration (SLAAC, zustandslose Adresskonfiguration) ermöglicht einem Host bzw. Schnittstelle eine funktionsfähige Internet- oder Netzwerkverbindung aufbauen zu können. Dafür tritt die Schnittstelle in Kontakt mit dem zuständigen Router. Den grundsätzlichen Ablauf können wir uns so vorstellen:

- Der Host weist sich selbst Link Local Adresse zu (durch Hardware errechnet)
- Durch Neighbour-Discovery wird zuständiger Router gesucht
- Router antwortet mit allen verfügbaren Präfixen → Adressräume
- Daraus kann sich der Host selber eine Unicast-Adresse zuweisen

5. Übergangsstrategie von IPv4 auf IPv6

Migrationsplanung

Der Umstieg hat entscheidende Auswirkungen auf die Kommunikation von Unternehmen mit Kunden, Partnern und Zulieferern. Dabei ist es wichtig, dass man die Folgen, welche durch einen Wechsel entstehen können, vorab schon klarstellt. Einige dieser Aspekte sind:

- Die Aktivierung von IPv6 im lokalen Netz ist notwendig, um eine gesicherte Kommunikation zu gewährleisten. Webseiten und webbasierte Anwendungen müssen ebenfalls für IPv6 konfiguriert werden.
- Neue Netzwerke werden nur mehr über IPv6 erreichbar sein, da der Adressraum von IPv4 in nächster Zeit verbraucht sein wird.
- Es sollte klar sein, dass ein Umstieg Geld und Ressourcen kosten wird.

Laut einer Studie von SolarWinds (siehe http://www.solarwinds.com/) sagen 47% der Befragten, dass sie „überhaupt nicht sicher“ sind, ob ihr Unternehmen einen umsetzbaren Plan zu Einführung von IPv6 besitzt. Noch schockierender ist es, dass nur weniger als drei Prozent der Umfrageteilnehmer bereits auf IPv6 umgestiegen sind.

Strategien

Um einen Umstieg vollziehen zu können, muss man etliche Faktoren berücksichtigen. Unteranderem die aktuelle Netzwerkumgebung im Unternehmen, die voraussichtliche Anzahl an IPv6-Schnittstellen, den aufkommenden Netzwerkverkehr und ob die Verfügbarkeit von IPv6-Anwenungen auf Endsystemen und Applikationen gegeben sein muss. Logischerweise können wir ja nicht von heute auf morgen den Umstieg von IPv4 auf IPv6 vollziehen, sondern der Übergang wird wohl eher stufenweise vollzogen werden. Anfangs wird man dabei auf eine hybride (beide Technologien vereint) Netzwerkumgebung setzen, wodurch wir die IPv4/IPv6-Kompatibilität gewährleisten können.
Doch ist es auch verständlich, dass wir für den Umstieg keine neue Infrastruktur benötigen, sondern die bereits bestehende verwenden können. So sind üblicherweise keine neuen Leitungen, Netzwerkkarten und Geräte nötig, jedoch muss das Betriebssystem die neuere IPv6-Technologie unterstützen. Noch dazu gibt es kein Gerät welches ausschließlich IPv6 beherrscht, aber IPv4 nicht. Um jetzt aber ausschließlich auf IPv4 ausgelegte Geräte weiterhin benützen zu können, aber nur in einem IPv6-Netz betrieben werden, benötigen wir Übersetzungsverfahren.
Im Grunde können wir drei unterschiedliche Übergangsstrategien festlegen:

- Dual-Stack, auch Parallelbetrieb genannt
- Tunneling
- Translation

Wobei Dual-Stack und Tunneling seitens des Betriebssystems voraussetzt wird, dass dieses beide Protokolle beherrscht.

Dual-Stack-Umgebung (Parallelbetrieb)

In Dual-Stack-Umgebungen können Geräte IPv4 und IPv6 parallel nutzen. Dadurch ist große Flexibilität für Anwendungen beider Protokolle gewährleistet, weil ja die Schnittstelle gleichzeitig IPv6 und

IPv4 Inhalte abrufen können. Im Grunde können wir uns das so vorstellen, als ob einer IPv4-Schnittstelle mindestens eine IPv6-Adresse und die notwendigen Routinginformationen zugewiesen werden. Dies ist die am häufigsten eingesetzte Übergangsstrategie.
Vorteile sind:

- Beide Protokolle können zusammen aber auch unabhängig voneinander betrieben werden und somit einen schrittweisen Übergang von Geräten und Anwendungen gewährleisten.
- Die meisten Netzwerke sind so aufgebaut, dass sie intern private IPv4-Adressen verwenden, aber nach außen hin keine öffentlichen routingfähigen Adressen. Die setzen dann Dual-Stack-Umgebungen ein, um ihr nach außen gewandtes Netz auf IPv6 umzustellen, im internen aber weiterhin IPv4 einzusetzen.
- Der Parallelbetrieb wird von allen bedeutenden Betriebssystemen unterstützt.

Die Nachteile davon liegen ebenfalls auf der Hand:

- Es werden zwei Netzwerke gefordert, um den Parallelbetrieb zu ermöglichen.
- Es sind doppelt so viele IP-Verwaltungsaufgaben durchzuführen, wodurch auch die IT-Ausgaben steigen.

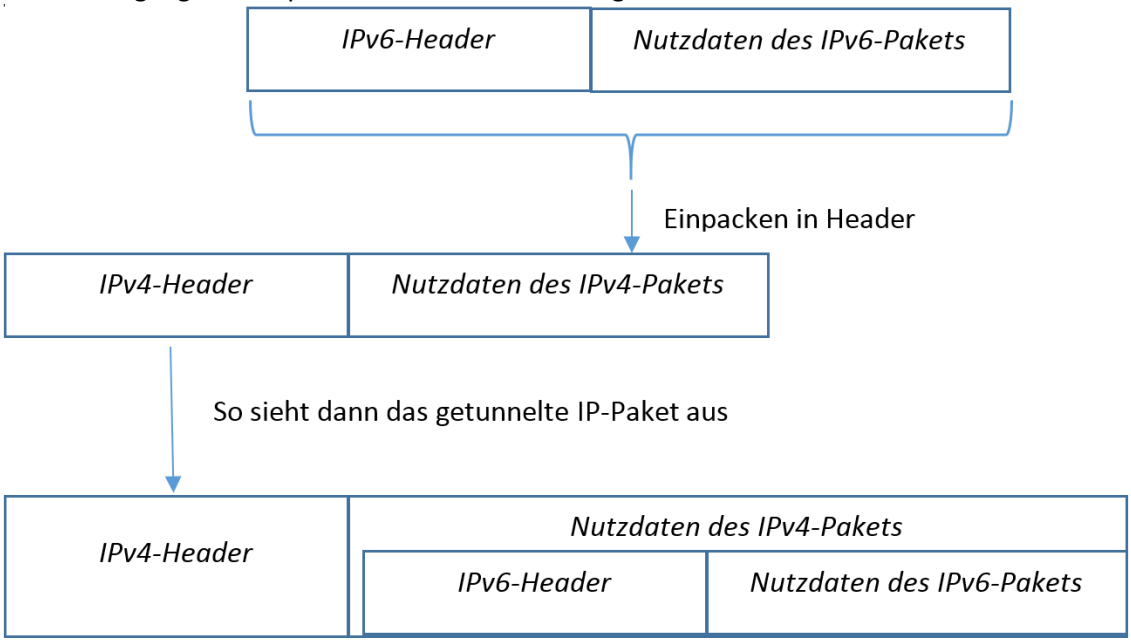
Im Allgemeinen sollten wir den Übergang so durchführen, dass die nach außen gewandten Netze zuerst in die Dual-Stack-Umgebung integriert werden. Danach identifizieren wir Geräte, welche wir nicht in diese Umgebung inkludieren wollen und welche schon, damit wir nicht unnötig hohen Aufwand betreiben müssen.

Tunneling 6in4

Eine weitere Art des Überganges stellt das Tunneling dar, in dem Router IPv6-Netzwerke überbrücken können. Unter Tunneling versteht man das Einpacken eines Protokolls in den Header eines anderen. Also in unserem Beispiel des IPv6-Tunneling wird das IPv6-Protokoll in den Header des IPv4-Protokolls verpackt. Generell unterscheidet man zwei Arten des Tunnelings von IPv6:

- Manuelles Tunneling von IPv6 über IPv4
    - IPv6-Pakete werden in IPv4-Pakete eingepackt
    - Danach über IPv4-Infrastruktur transportiert
    - Ein Punkt zu Punkt Tunneling
- Automatisches Tunneling von IPv6 über IPv4
    - IPv6-Knotenpunkte verwenden unterschiedliche Adressen um dynamisch IPv6 über IPv4 zu tunneln
    - Dabei enthält ein IPv6-Header auch IPv4-Adressen, die dann zum Transport über die jeweilige Infrastruktur verwendet werden.

Diesen Vorgang des Verpackens können wir uns folgendermaßen Vorstellen:



Tunneling Generell

Grundsätzlich wird ein IPv6-Paket in ein IPv4-Paket eingepackt und in weiterer Folge über ein IPv4-Netzwerk versendet, wenn es dann am Ende ankommt, wird es wieder ausgepackt und weiter über das Ziel-IPv6-Netz dem Empfänger zugestellt. Dies können wir uns dann wieder so vorstellen:

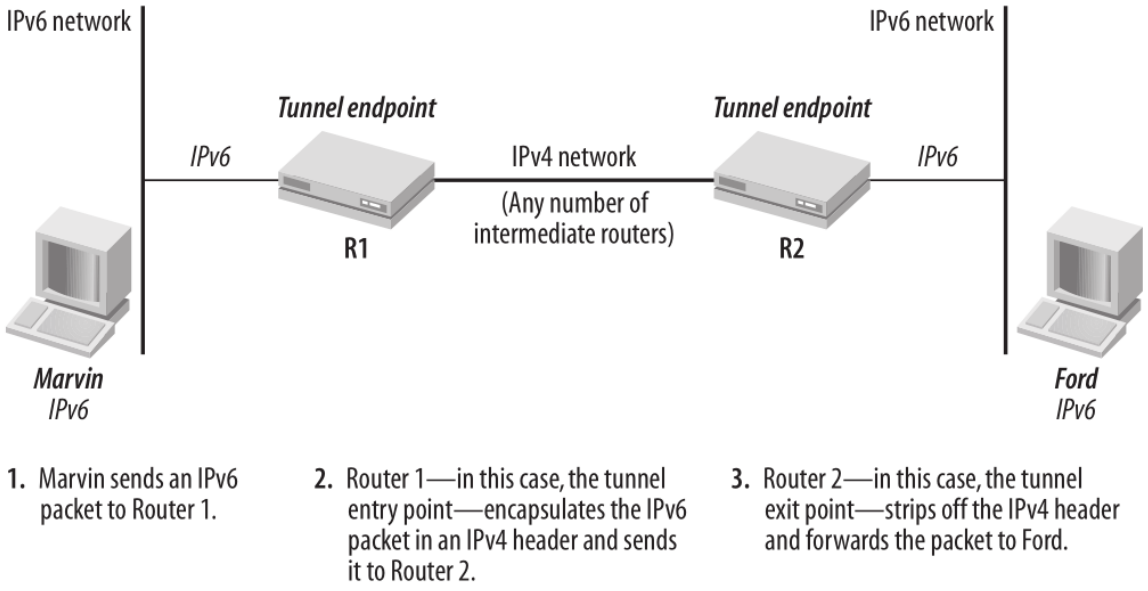


Abbildung - IPv6 Essentials (O´Reilly)

Diesen Vorgang möchte ich hier noch etwas genauer erläutern und beziehe mich dabei auf die Grafik aus dem Buch „IPv6 Essentials“. Also dieser PC namens Marvin sendet ein IPv6-Paket an Ford, welcher auch in einem IPv6-Netz hängt. Dazwischen ist ein IPv4-Netzwerk welches von den Ausgangs- und Zielnetzwerk jeweils durch einen Router abgegrenzt wird.

- Marvin sendet ein IPv6-Paket an Router R1

- R1 empfängt das Paket und sobald er erkennt, dass der Empfänger, Ford, in einem anderen Netz als dieses liegt, so verpackt er das IPv6-Paket in ein IPv4-Paket.
- R1 versendet das getunnelte Gerät über das IPv4-Netz
- R2 erhält das Paket von R1, auch Tunnelausgangspunkt genannt
- R2 entpackt das IPv4-Paket und leitet das gültige IPv6-Paket an den Empfänger Ford weiter

Funktionsweise von 6in4

6in4 ist ein Übergangsmechanismus, bei dem IPv6-Pakete in IPv4-Pakete getunnelt werden. Wie auch schon oben erklärt, wird der gesamte Datenverkehr in IPv4-Datenpakete gepackt, wobei das IPv6-Paket direkt nach dem IPv4-Header angefügt wird. Es müssen alle involvierten IPv6-Knoten, welche einen Übergang auf IPv4 markieren, statisch konfiguriert werden, damit sie die Pakete tunneln.

4in6

Bei 4in6 werden IPv4-Datenpakete über ein IPv6-Netzwerk übertragen. Dabei wird, genau umgekehrt von 6in4, der Datenverkehr in IPv6-Pakete verpackt und ermöglicht somit ein Tunneling durch ein IPv6-Netz. Die hierbei vorgenommene Verschachtelung der IPv4-Pakete in IPv6-Pakete wird es möglich, dass IPv4-Knotenpunkte, welche keine IPv6-Unterstützung besitzen, trotzdem über ein IPv6-Netzwerk kommunizieren können.

Translation NAT64

NAT64 ist ein bekannter IPv6-Übergangsmechanismus und hilft dabei IPv4-Adressen in IPv6-Adressen zu übersetzen. Es ermöglicht die Kommunikation von ausschließlich IPv6-beherrschenden Hosts und ausschließlich IPv4-beherrschenden Hosts, der Hauptzweck besteht auch darin, IPv4-Server von IPv6-Netzwerken aus ansprechen zu können.

Adressschreibweise

Eine IPv6-Adresse kann ohne weiteres eine 32-Bit lange IPv4-Adresse enthalten. Dazu ein kleines Beispiel:

- Die IPv4-Adresse *192.0.2.1*
- In Hexadezimal ist *192.0.2.1* gleichbedeutend mit *c0.00.02.01*
- Nun wird das IPv6-Präfix *64:ff9b::/96* verwendet, welches genau für diesen Mechanismus reserviert wurde
- Die gültige IPv6-Adresse wäre dann *64:ff9b::c000:0201*
- Oder in der einfacheren Mischschreibweise *64:ff9b::192.0.2.1*, wobei hier automatisch in die Hexadezimalschreibweise umgewandelt wird

Routing und DNS64

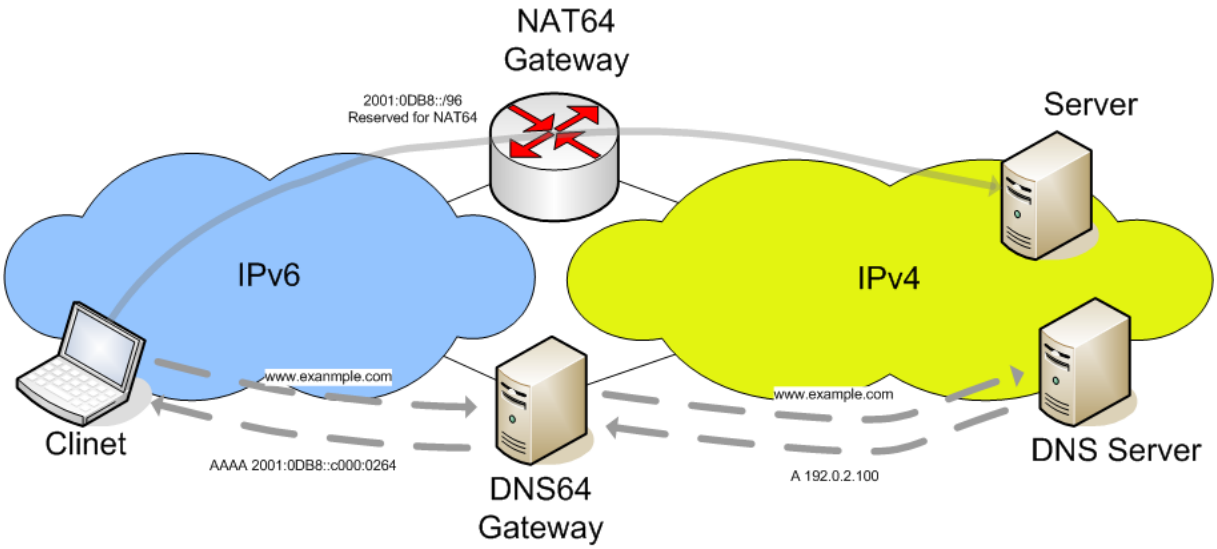


Abbildung - Quelle: http://www.swissipv6council.ch/sites/default/files/docs/nat64-dns64.png

Das Routing funktioniert so, dass ein IPv6-Host, der Kontakt mit einem IPv4-Server aufnehmen will, seine Pakete mit dem */96*-Präfix ins lokale Netz versendet. Dann nimmt der NAT64-Router die Pakete entgegen, ermittelt die IPv4-Adresse und sendet das Paket an einen IPv4-Host/Server weiter.

Bei DNS64 ermöglicht die Namensauflösung über beide Netzwerkarchitekturen. Dabei wird zuerst eine Anfrage nach einer IPv4-URL gestellt, wobei der IPv6-Host nicht weiß, ob es sich dabei um eine IPv4-Adresse handeln könnte. Dann versucht DNS64 die URL im IPv6-Netz aufzulösen, findet aber keinen passenden. Dafür erhält er zuständigen Nameserver im IPv4-Netz eine IPv4-Adresse. Diese Adresse wird dann wie oben dargestellt in eine IPv6-Adresse übersetzt und an den Host zurückgegeben. Für eben diesen IPv6-Host wirkt es so, als ob er eine gültige IPv6-Adresse von einem IPv6-Server erhalten hat. Dann wird das Paket versendet und durch den NAT64-Router zugestellt.

6. Sicherheit

Im IPv4-Standard wurde nicht groß auf Sicherheit geachtet, da in den Anfängen einfach den Teilnehmern von Grund auf vertraut wurde. Deshalb enthält das Internetprotokoll der Version 4 keinerlei Sicherheitsframeworks, falls doch mal Sicherheit gebraucht wurde, wurden rudimentäre Authentifikation- und Autorisierungsmechanismen in den jeweiligen Anwendungen implementiert. Später wurde IPsec eingeführt, jedoch wurde es nie zu einem weitverbreiteten Standard in IPv4. Erst seit IPv6 wurde IPsec fest in die Basis des Protokolls integriert, damit wollte man erreichen, dass eine gesicherte Verbindung gewährleistet wird und auch sichergestellt ist, dass IPsec tatsächlich zum Einsatz kommt.

Allgemein Sicherheit

Um sensible Daten zu schützen, sollten wir nicht nur von mögliche Gefahren aus einem fremden Netz ausgehen, sondern auch von viele weitere Schwachstellen, wie zum Beispiel:

- Unzulängliche Sicherheitsrichtlinien in Infrastruktur
- Zu wenig Kontrolle über die Vorgänge im Netz
- Das Entwenden von Benutzerrechten (Passwortdiebstahl)
- Fehlerhafte Administration des IT-Systems
- Missbrauch von Benutzerrechten (Weitergabe)
- Schwachstellen in Software

- Manipulation, Diebstahl oder Zerstörung von Geräten, Daten und Software
- Viren, Trojaner
- Fehlfunktion von Routing
- Attacken auf IT-System (z.B. DoS, Man-in-the-middle-Attacken)
- Lauschangriffe auf Netzwerkstruktur (wie zum Beispiel Sniffing von Datenpaketen)

Außerdem sollten wir uns darüber im Klaren sein, dass nur ein kleiner Bruchteil der Attacken von Außenstehenden erfolgt.
Es gibt einige Standardpraktiken, davon sind der CIA und der AAA Ansatz die bekanntesten.
CIA-Strategie bedeutet:

- Confidential (Vertraulichkeit)
  Gespeicherte und übertragene Daten können von unautorisierten Teilnehmern weder gelesen noch bearbeitet werden.
- Integrity (Integrität)
  Jede Änderung von übertragenen oder gespeicherten Daten kann erkannt werden.
- Availability (Verfügbarkeit)
  Alle Daten sind jederzeit für jeden berechtigten Benutzer zugänglich.

AAA-Strategie besagt:

- Authentifikation
  Stellt sicher, dass jeder Benutzer wirklich derjenige ist, als wessen er sich ausgibt. Es wird somit der Benutzer identifiziert. Üblicher Ansatz zur Authentifikation erfolgt durch Benutzernamen und Passwörter, bzw. durch Karten.
- Autorisation
  Stellt sicher, dass authentifizierte Benutzer auch wirklich genügen Rechte besitzen, um die angeforderten Daten wirklich einsehen zu dürfen. Üblicherweise wird dies durch eine Art Zugriffskontrollliste gelöst.
- Accounting
  Protokolliert sämtliche Vorgänge von Ressourcennutzung und Zugriff mit und speichert diese. Zum Beispiel die Speicherung des HTTP-Verkehrs im Netz ist eine Möglichkeit des Accountings.

IPsec

IPsec beschreibt eine Sicherheitsarchitektur für beide Versionen des IP-Protokolls. Seit IPv6 ist das IPsec fest in das Regelwerk für eine gültige Konfiguration verankert. Im Grunde gibt es sechs Elemente die für das IPsec-Rahmenwerk relevant sind:

- Eine allgemeine Beschreibung der Sicherheitsanforderungen und –Mechanismen für das jeweilige Netzwerk.
- Das Protokoll für die Verschlüsselung (ESP (Encapsulation Security Payload))
- Das Protokoll für die Authentifizierung (AH (Authentication Header))
- Bestimmung für die verwendeten Verschlüsselungsalgorithmen für die Verschlüsselung und die Authentifizierung
- Bestimmung der Sicherheitsregeln und Sicherheitsvereinbarungen zwischen den Kommunikationspartnern
- Schlüsselverwaltung

IPsec grenzt geschützte und ungeschützte Bereiche in einem Netz voneinander ab. Dabei wird von Administratoren bestimmt, was mit Paketen passiert, welche die Grenze passieren. Entweder sie werden abgelehnt oder durchgelassen.

Sicherheitsvereinbarungen (Security-Assoziation)

Ist ein Übereinkommen der kommunizierenden Partner. Dabei sind drei Elemente Teil der Vereinbarung: einen Schlüssel, einen Verschlüsselungs- oder Authentifizierungsmechanismus und zusätzliche Parameter für den Algorithmus. Dabei gelten sie nur für einen Teilnehmer

und jeder braucht seine eigene Sicherheitsvereinbarung. IPsec kennt zwei Arten der Übertragung:

- Transportmode
  Gilt nur zwischen zwei Endknoten, dabei werden die Vereinbarungen auf alle IP-Pakete angewandt. Wobei der Header nicht verschlüsselt wird.
- Tunnelmode
  Gilt zwischen zwei Gateways, wobei das ganze Paket, wie beim Tunneling, in ein neues Paket gepackt wird und Verschlüsselt oder Authentifiziert wird. Dies ist auch die Basis von VPN-Tunneling.

Databases

Wir können drei wichtige Datenbanken unterscheiden:

- Security-Policy-Database (SPD)
  Definiert alle Regeln bezüglich IP-Verkehr (eingehenden und ausgehenden) von einem Endknoten oder Gateway, also ob verworfen wird oder nicht.
- Security-Assoziation-Database (SAD)
  Enthält für jede Vereinbarung einen Eintrag und die dafür definierten Parameter.
- Peer-Autorisation-Database (PAD)

IPv6 Sicherheitsbausteine

IPsec beschreibt Sicherheitsmechanismen die mit beiden Protokollen funktionieren. Dies bedeutet, dass IPv6 nicht sicherer ist als IPv4. Der Unterschied besteht darin, dass IPsec bei IPv4 speziell Installiert werden muss und bei IPv6 ist es bereit verpflichtender Bestandteil. Grundsätzlich unterscheidet man in IPsec das Protokoll für Authentication Headers (AH) und für Encapsulation Security Payload (ESP). Diese beiden Protokolle werden in die Extension Headers gepackt. Auch ist der ESP mittlerweile verpflichtend und AH nur optional.

Der Authentication Header

Der AH bietet Schutz vor Manipulation oder Fälschung der übertragenen Daten, dies erfolgt dadurch, dass eine Prüfsumme über das gesamte Datenpaket berechnet wird. Der Authentication Header ist, wenn er zum Einsatz kommt, nach dem IPv6-Header und vor dem Header einer höheren OSI-Schichtanwendung (zum Beispiel TCP-, UDP-, ICMP-Header) angesiedelt. Der Aufbau des Authentication Header ist folgendermaßen:

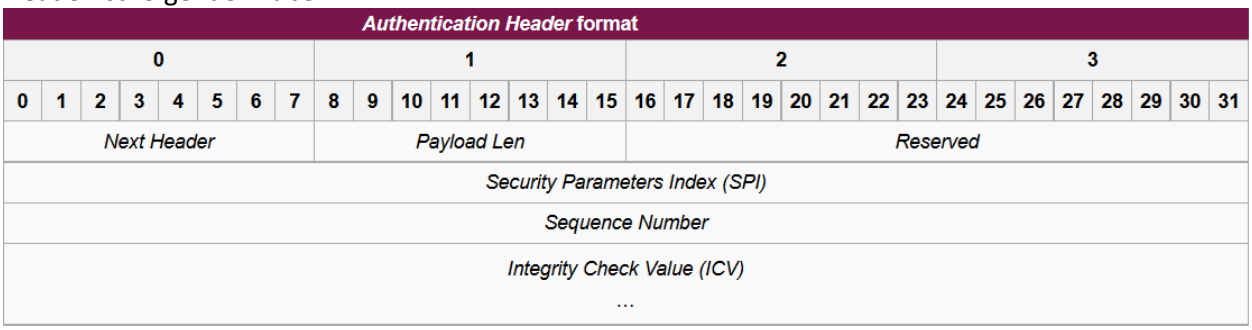


Abbildung - Quelle: https://en.wikipedia.org/wiki/IPsec#Authentication_Header

Next Header

Das Next Header Feld ist 1 Byte groß und gibt an welcher Header nach dem Authentication Header im Datenpaket folgt.

Payload Length

Das Payload Length Feld ist ebenfalls 1 Byte groß. Es gibt die Länge des Authentication Headers in vier Byte-Einheiten an, wobei die ersten acht Bytes nicht in der Berechnung nicht berücksichtigt werden.
Die Angabe der Länge ist notwendig, weil sich die Länge der Authentifikationsdaten im Authentication Header je nach verwendeten Berechnungsalgorithmus unterscheiden.

Reserved

Das Reserved Feld ist 2 Bytes groß, wird aber aktuell noch nicht verwendet. Deswegen ist der Wert dieses Feldes auf 0 zu setzen.

Security Parameters Index (SPI)

Das Security Parameters Index Feld ist ein willkürlich vergebener 32-Bit (bzw. 4 Byte) Wert. Es wird vom Empfänger verwendet um das eingehende Datenpaket einer Security Assoziation eindeutig zuordnen zu können. Die SPI-Werte von 1 bis 255 sind von seitens der Internet Assigned Numbers Authority für weitere Anwendungsfälle reserviert, der Wert 0 ist für die lokale Anwendung reserviert.

Sequence Nummer

Die Sequence Nummer Feld beinhaltet 32 Bit (bzw. 4 Byte) und ist eine sich ständig erhöhende Zählvariable und wird vom Sender gesetzt. Es stellt sicher, dass identische Daten nicht noch einmal gesendet werden. Dies schützt vor sogenannten Reply-Attacken (also ein Angreifer dupliziert genau dasselbe Paket und sendet nochmal) in einer Unicast Security Assoziation. Für mehrere-Sender Security Assoziation gibt es jedoch diesen Schutz nicht, weil der Authentication Header das Synchronisieren von Paketzählvariablen bei mehreren Sendern nicht unterstützt.

Bei der Einführung einer Security Assoziation wird die Sequence Nummer beim Sender und Empfänger auf 0 gesetzt. Danach hat das erste Paket den Wert 1, welcher dann bei jedem nachfolgenden Paket um eins erhöht wird. Wird der Wert 232 erreicht, wird die Sequence Nummer wieder auf 0 gesetzt.

Integrity Check Value (ICV)

Das Integrity Check Value ist von variabler Größe und enthält die Prüfsumme, welche über das gesamte Datenpaket hinweg berechnet wird. Dabei hängt die Länge von den verwendeten Algorithmus der Berechnung ab. Aber wir können bestimmen, dass es immer ein Vielfaches von 4 Bytes sein muss.

Die Prüfsumme

Die Prüfsumme des Authentication Headers, welche im ICV-Feld steht, wird über folgende Felder berechnet:

- Alle Felder des IPv6-Headers
- Alle Felder der Extension-Headers, welche vor dem Authentication Header im Paket angesiedelt sind und die sich während der Übertragung bzw. während des Sende- und Empfang-Vorganges nicht verändern.
- Alle Felder des Authentication Headers
- Weitere Extension-Headers und deren Nutzdaten, wenn sie statisch sind
- Die Bits welche für die Prüfsummenberechnung benötigt werden

Passende Algorithmen

Folgende Algorithmen gelten als angemessen für die IPsec:

- Keyed Massage Authentication Codes (MAC)
  - Gewährt Gewissheit über Ursprung
  - Sichert die Integrität (Schutz vor Veränderung)
  - Erfordert zwei Eingabeparameter
    - Die zu schützenden Daten
    - Einen geheimen Schlüssel
  - Aus den Eingabeparameter wird Prüfsumme berechnet
- Einweg Hashfunktionen (zum Beispiel SHA-1, SHA-256, MD5)
  - Prüfsumme wird über komplexe Berechnung gebildet
  - Eindeutiger Prüfwert für digitale Daten
  - Prüfwert wird verwendet, um die Integrität einer Nachricht zu sichern

AH in Tunnelmodus und Transportmodus

Der Authentication Header kann in beiden (Tunnel- und Transportmodus) zur Anwendung gebracht werden, dabei ist der Aufbau der Pakete aber unterschiedlich:

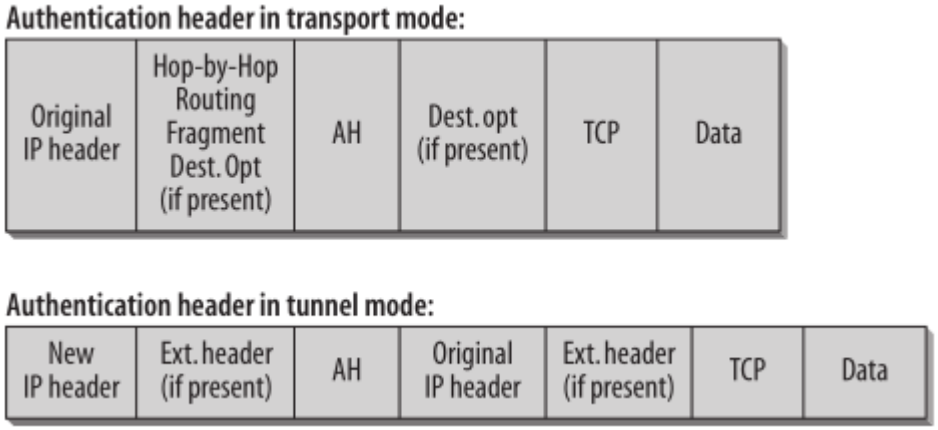


Abbildung - IPv6 Essentials (O´Reilly)

- Transportmodus
  - Die gesamten Nutzdaten und der IPv6-Header wird durch die Prüfsumme des Authentication Headers geschützt
- Tunnelmodus
  - Der äußere IPv6-Header enthält die IP-Adresse der Tunnel-Endpunkte
  - Hier wird der innere und äußere IPv6-Header samt Nutzdaten durch den Authentication Header bzw. durch deren Prüfsumme geschützt

Die Encapsulation Security Payload

Der Encapsulation Security Payload Header (auch ESP genannt) ist ein Protokoll, welches Integrität (unautorisierter Modifikation von Daten), Vertraulichkeit (also ob der Sender und Empfänger vertrauenswürdig ist), Verifikation der Datenherkunft (sicherstellen, dass der Sender nicht modifiziert wird) und Schutz vor Datenduplizierung gewährleistet. Um dies zu können werden neben einer Prüfsumme auch die gesamten Daten durch einen, in den Security Assoziationen festgelegten, Verschlüsselungsalgorithmus geschützt, somit sind die Herkunft und Integrität der Daten gewährleistet.

Die zwischen zwei Hosts zum Einsatz kommenden Verfahren des ESP werden durch die Security Assoziationen geregelt.

Der ESP Header folgt nach dem IPv6-Header und ist aber vor Transport- (TCP, UDP), Netzwerk- (ICMP) und Routing-Headers (OSPF) im IPv6-Datenpaket angesiedelt.

Der Encapsulation Security Payload Header ist so aufgebaut:

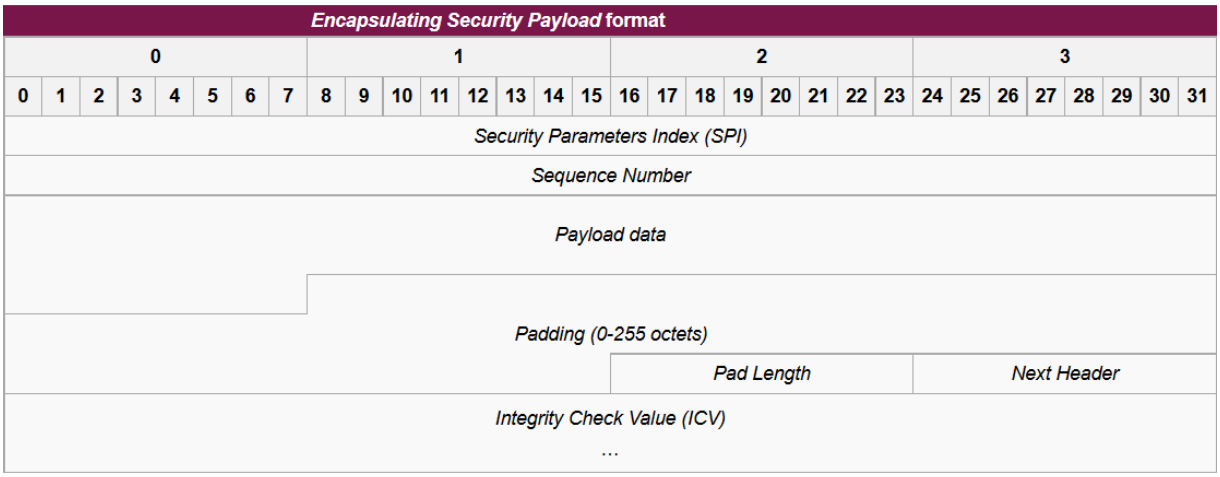


Abbildung - Quelle: https://en.wikipedia.org/wiki/IPsec#Encapsulating_Security_Payload

Security Parameter Index (SPI)

Das Security Parameters Index Feld ist ein willkürlich vergebener 32-Bit (bzw. 4 Byte) Wert. Es wird vom Empfänger verwendet um das eingehende Datenpaket einer Security Assoziation eindeutig zuordnen zu können. Die SPI-Werte von 1 bis 255 sind von seitens der Internet Assigned Numbers Authority für weitere Anwendungsfälle reserviert, der Wert 0 ist für die lokale Anwendung reserviert.

Sequence Nummer

Die Sequence Nummer Feld beinhaltet 32 Bit (bzw. 4 Byte) und ist eine sich ständig erhöhende Zählvariable und wird vom Sender gesetzt. Es stellt sicher, dass identische Daten nicht noch einmal gesendet werden. Dies schützt vor sogenannten Reply-Attacken (also ein Angreifer dupliziert genau dasselbe Paket und sendet nochmal) in einer Unicast Security Assoziation. Für mehrere-Sender Security Assoziation gibt es jedoch diesen Schutz nicht, weil der Authentication Header das Synchronisieren von Paketzählvariablen bei mehreren Sendern nicht unterstützt.

Bei der Einführung einer Security Assoziation wird die Sequence Nummer beim Sender und Empfänger auf 0 gesetzt. Danach hat das erste Paket den Wert 1, welcher dann bei jedem nachfolgenden Paket um eins erhöht wird. Wird der Wert 232 erreicht, wird die Sequence Nummer wieder auf 0 gesetzt.

Payload Data

Das Payload Data Feld ist ein Feld von variabler Länge. Es enthält die verschlüsselten Daten und, falls von dem verwendeten Algorithmus gefordert, den Initialisierungsvektor (= ein Block von Zufallsdaten).

Padding

Das Padding Feld ist zwischen 0 und 255 Bytes groß, es gleicht das Paket mit einem Vielfachen von 4 Bytes an, damit die minimale Paketlänge erreicht wird, falls dies nicht bereits gegeben ist und falls dies vom Algorithmus gefordert wird.

Pad Length

Pad Length Feld ist 1 Byte groß und gibt die Anzahl der Bytes des vorangestellten Padding Feldes an.

Next Header

Das Next Header Feld ist 1 Byte groß und gibt an ob und welcher Header nach dem Encapsulation Security Payload Header im Datenpaket folgt (Zum Beispiel TCP-, UDP-, ICMP-Headers).

Integrity Check Value

Das Integrity Check Value Feld (auch ICV genannt) ist von variabler Größe enthält die Prüfsumme, welche über den ESP-Header, den Nutzdaten und des ESP-Trailers gebildet werden. Jedoch ist dieses Feld, anders als beim Authentication Header, optional und wird nur verwendet, wenn vom ESP auch die Nachrichtenintegrität gewährleistet werden soll und wenn der Integritätsalgorithmus, bzw. der Algorithmus für die Verschlüsselung einen ICV verwendet. Die Länge dieses Feldes wird von den, in den Security Assoziationen festgelegten, Algorithmus festgelegt.

Das Padding-, Pad Length- und Next Header – Feld zählen zu dem ESP-Trailer, was also heißt, dass diese mitverschlüsselt werden. Der Verschlüsselungsalgorithmus wird entweder durch die vereinbarten Security Assoziationen oder manuell festgelegt.

ESP im Tunnelmodus und Transportmodus

Wie auch schon der Authentication Header kann auch die Encapsulation Security Payload im Tunnel- oder Transportmodus verwendet werden:

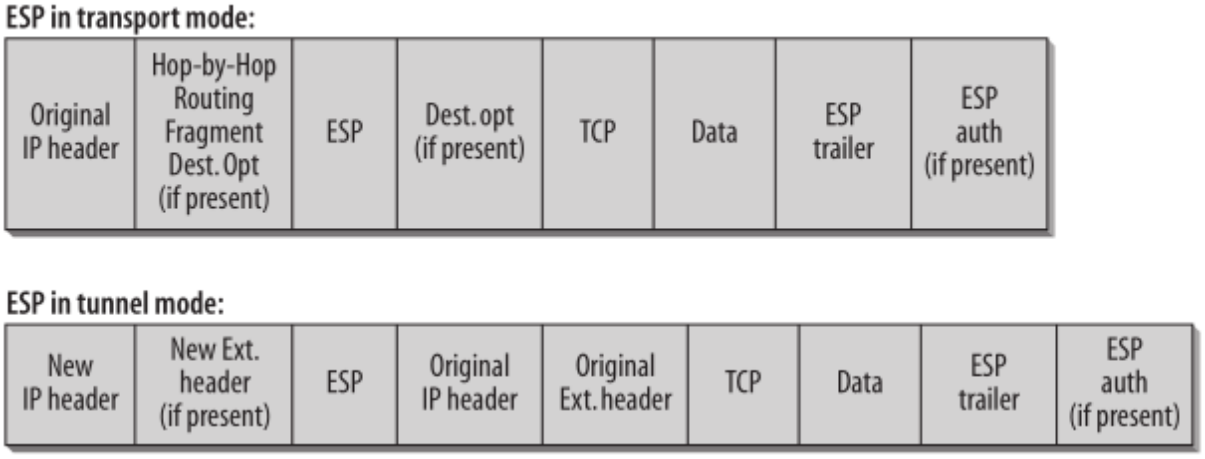


Abbildung - IPv6 Essentials (O´Reilly)

- Transportmodus
  - Der IP-Header und die nachfolgenden Extension-Headers werden beim ESP nicht verschlüsselt, da ansonsten die Pakete nicht mehr geroutet und weitergeleitet werden könnten
  - Es werden lediglich die Nutzdaten, welche im IP-Paket angehängt sind verschlüsselt
- Tunnelmodus
  - Hier wird das gesamte originale Paket verschlüsselt, also auch der IP- und Extension-Header
  - Das innere Paket enthält die originalen Empfänger- und Senderadressen
  - Das äußere Paket enthält die Adresse der Tunnelendpunkte
  - Es ist also so, dass das originale Paket ab den Tunnelendpunkten komplett verschlüsselt wird.

Wechselwirkung von IPsec und IPv6

Der verpflichtende Einsatz von IPsec im IPv6-Standard ist eine große Errungenschaft für die Sicherheitsstandards im Internet. Aber es gibt leider immer noch Bereiche in denen IPsec sich nicht so reibungslos mit anderen Diensten bzw. Anforderungen von IPv6 vereinen lässt. Einige von diesen Problemstellen sind:

- Tunneling
    - Schwierigkeiten mit bereits bestehenden Firewalls und Sicherheitsknotenpunkten, welche das interne Netz nach außen hin abgrenzen.
    - Wird nun Tunneling eingesetzt, wobei Tunneling ja ein Ende-zu-Ende Sicherheitsmodell bereitstellt, kann die Firewall die Pakete nicht mehr auf gefährliche oder unautorisierte Inhalte prüfen.
    - Um dieses Problem zu lösen sollten die Security Assoziationen zwischen den Knotenpunkten vereinbart werden und nicht von den Endbenutzern.
    - Es könnte auch sein, dass das getunnelte Paket wichtige Informationen über das interne Netz preisgibt, wie zum Beispiel Routinginformationen oder ICMP-Nachrichten.
- NAT
    - NAT führt Adressübersetzung und Portübersetzung durch, was heißt, dass die Adress-/Portkonfiguration eines Netzes für einen Host in einem anderen Netz zu Verfügung gestellt werden. Diese werden via IP-Header durchgeführt.
    - Diese Übersetzung bereitet aber Problem mit Autorisierung von Teilnehmern und bei Verschlüsselung von Datenpakete. Da diese ja teilweise durch das gesamte Internet erfolgen kann, dabei können die Pakete leicht gefälscht werden.
- Quality of Service
    - Erlaubt Routern gewisse Datenpakete zu verwerfen und nicht weiterzuleiten
    - Das Problem ist, dass in den IPsec-Bestimmungen den Verlust/das Verwerfen von Paketen nicht erlaubt sind
    - Dadurch können einige Services von IPsec nicht angewendet werden
- Erweiterte mobile Komptabilität
    - Die ständig ändernden IP-Adressen im mobilen Bereich kann zu Problemen mit der IPsec-Umgebung
    - Dynamische Adressen verursachen zum Beispiel Schwierigkeiten mit der Identitätsprüfung, welche bei IKE (Schlüssel- oder Assoziationsvereinbarungsprotokoll) zum Einsatz kommt.

Fehlerquellen IPv6

Die Sicherheit eines IPv6-Netzwerkes bietet nur begrenzt mehr Sicherheit als ein IPv4-Netzwerk, da nur bestimmte Schwachstellen behoben wurden, aber der grundlegende Sicherheitsaspekt in Netzwerken bleibt erhalten. Deshalb sind auch die meisten bekannten IPv4-Angriffe auch in einem IPv6-Netzwerk durchführbar, was also bedeutet, dass der Ansatz der Datensicherung ähnlich ist.
Weiters ist bereits IPv6 in den gängigsten Betriebssystemen implementiert und ist entsprechend einfach zu konfigurieren. Oft sind auch Tunnelmechanismen bzw. andere Übergangsmechanismen (siehe weiter oben: Übergangsstrategie von IPv4 auf IPv6) standardmäßig aktiviert. Somit kann in einem IPv4-Netzwerk bereits von Hackern ein Angriff über IPv6-Pakete erfolgen, ohne dass dies großes Aufsehen erregt.
Auch sollten wir uns im Klaren sein, dass IPsec nicht die Nonplusultra-Lösung ist, sondern nur eine gute Basis für ein guter Ansatz und Leitfaden zu einem sicheren Netzwerk ist.

Firewalls

Die Ende-zu-Ende-Verschlüsselung des IPsec gilt als einer der bekanntesten Vorteile, welche die IPv6-Implementierung bietet. Jedoch treten genau durch diese Verbesserung auch wieder neue Fehler mit bereits bestehenden Firewall-Architekturen, wenn zum Beispiel ESP (siehe

„Die Encapsulation Security Payload“ weiter oben) mit aktivierter Verschlüsselung eingesetzt wird. Denn wenn Pakete von Ende zu Ende verschlüsselt werden, wie soll dann die Firewall die gesendeten Daten überprüfen, wenn diese nicht in der Lage ist, sie wirklich zu entschlüsseln? Also bereiten bestehende Firewall Lösungen Schwierigkeiten mit verschlüsselten Paketen. Eine mögliche Lösung könnte die zentrale Speicherung der verwendeten Schlüssel sein. Diese bietet wiederum eine Schwachstelle für mögliche Eindringlinge, denn somit wären sämtliche Informationen für die Ent- und Verschlüsselung preisgegeben und dadurch nicht mehr sicher.

Neighbour Discovery

Eine weitere Fehlerquelle ist das Neighbour Discovery Protokoll (NDP). Dieses Protokoll ermöglicht einen „Denial of Service“-Angriff, weil ja mehrere IPv6-Adressen einem einzigen Interface, bzw. Host, zugeordnet werden können. Einer einzigen Workstation können bis zu mehreren Tausend Adressen zugeordnet werden. Somit werden allen Adressen welche im Netz zu vergeben sind als belegt gekennzeichnet und das gesamte Netzwerk lahmgelegt, weil keine neuen Adressen mehr vergeben werden können.
Deshalb wird beim Einsatz von NDP auch die Verwendung von IPsec empfohlen. Dies wird auch Secured Neighbour Discovery (SEND) genannt. Dabei wird der Inhaber von sämtlichen Adressen zugeordnet.

Portscanning

Seit IPv6 ist das bekannte Portscanning (also das Zuordnen von Anwendung und Port) sehr komplex geworden. Da ein IPv6-Interfaceidentifier bis zu 64 Bits beträgt, würde es einen extrem hohen Zeitaufwand bedeuten, um einen einzigen 64 Bit Raum zu scannen. Somit ist es auch schwer einen, mit Virus/Malware/etc. infizierten, Port zu finden. Dieses Problem kann jedoch einfach vermieden werden, indem nicht so einfach zu erratende Adressierungsschemen eingesetzt werden könnten.

Übergangs- und Tunneling-Mechanismen

Die Mechanismen, welche es IPv4-Netzwerken und Geräten ermöglicht mit IPv6-Netzwerken zu kommunizieren, bieten nicht nur Vorteile, sondern auch Schwachstellen. Unter anderem können die Übergangsmechanismen als Hintertür zu gewöhnlichen und ausschließlich für IPv4 ausgelegte Netzwerke missbraucht werden. Es wird also IPv6 als Hintertür zu IPv4-Netzen verwendet. Wir können uns dies so vorstellen: Einen Eindringling gelingt es, dass er die Daten in unserem IPv4-Netzwerk über die Tunneling-Mechanismen von IPv6 aus dem internen Netzwerk in ein anderes Netzwerk, bzw. zu seinem PC tunnelt. Somit könnte er sämtliche für IPv4 ausgelegte Kontroll- und Sicherheitsmechanismen umgehen.
Also sollten wir generell beim Tunneling darauf achten, dass die Pakete keine, für die Sicherheit des Netzwerkes relevanten, Paketfilter und Firewalls umgehen können. Somit könnte via Tunnel ein IPv4-Paket zu unserem Endpunkt im IPv6-Netzwerk gelangen, dort wird es dann ausgepackt und an den Empfänger weitergeleitet. Nun sieht es für den Empfänger in unserem Netzwerk so aus, als ob dieses IPv6-Paket direkt von dem internen Netz stammt, obwohl es ja von außerhalb, und in ein IPv4-Paket verpackt, stammt.
Eine Gegenmaßnahme könnte sein, dass wir einfach unseren Endpunkt so konfigurieren, dass er nur von bekannten und vorher definierten Tunnelanfangspunkten Pakete entgegen nimmt.

7. Quality of Service

Einleitung

Zu Beginn war das Internet nur für einfache Kommunikationsmechanismen entwickelt worden, wie zum Beispiel E-Mail oder Datenübertragung. Jedoch seit den letzten Jahrzehnten hat sich das Internet in ein komplexes, globales Kommunikationsnetzwerk weiterentwickelt, wo auch eine Vielzahl an

unterschiedlichen Anwendungen und Diensten zur Verfügung gestellt wird. Im Grunde ist IPv4 ein einfaches Protokoll, welches stur Datenpakete auf dem günstigsten Pfad an sein bestimmtes Ziel bringt. Dabei wird aber vorerst nicht berücksichtigt ob und wie die Daten angekommen sind. Dafür wurde dann TCP eingeführt, dieses Protokoll bietet nun eine garantierte, sichere und komplette Zustellung der Pakete, jedoch sind erweiterte Kontrollparameter (wie zum Beispiel Verzögerung der Zustellung oder das Durchführen der Bandbreitenallokation) nicht im TCP inkludiert.

Die Entwicklung von IPv6, in welche auch der Aspekt der immer größer werdenden Nachfrage an Echtzeit-Diensten berücksichtigt wurde, berücksichtigt nun auch die geforderten Quality of Service (kurz QoS genannt) Funktionen. Jedoch sind bis heute noch der eine oder andere Bestandteile des QoS in Entwicklung.

Allgemein Quality of Service

Das aktuelle IP-Modell behandelt alle Pakete gleich, denn alle werden so weitergeleitet, dass die Geschwindigkeit und Übertragungsrate am höchstmöglichsten und die Fehlerrate am geringsten ist. Dabei hängt der Pfad des Paketes von den verfügbaren Routern, der Routingtabellen (also, welcher Pfad am effizientesten sein mag) und von der allgemeinen Netzauslastung ab.
Quality of Service Protokolle haben nun also die Aufgabe verschieden Datenströme (bzw. Paketströme) mit unterschiedlichen Prioritäten zu versehen und garantieren Qualitätsanforderungen, wie zum Beispiel die zugesicherte Bandbreite oder Verzögerungszeiten von Paketen es sein können.
Wir können beim QoS grundsätzlich zwei verschiedene Architekturen unterscheiden:

- Integrated Services (auch kurz als IntServ)
- Differentiated Services (kurz geschrieben als DiffServ)

Beide Architekturen bieten und verwenden Richtlinien für den Paketverkehr bzw. Datenverkehr. Des Weiteren können die beiden Ansätze auch kombiniert werden, dadurch wird in lokalen aber auch in globalen Netzwerken (wir können auch von LAN und WAN sprechen) der Quality of Service ermöglicht.
Richtlinien für den Paketverkehr können dazu verwendet werden um die Datenübertragung von bestimmten unterschiedlichen Kriterien abhängig zu machen. Zum Beispiel könnte festgelegt werden wie viele Ressourcen für die Datenübertragung benötigt werden, oder die Richtlinien können auch den Datenverkehr überwachen und, falls notwendig, Anpassungen oder Beschränkungen an den geforderten Standard ausüben.
Unteranderem kann der QoS die Anforderungen an den Datenverkehr bei Echtzeitdaten (zum Beispiel Streaming) oder auch wegen kommerziellen Gründen (zum Beispiel Kostenberechnung für Dienstnutzung) bestimmen.

Integrated Services

Die Integrated Services Architektur baut darauf auf, dass alle relevanten Ressourcen und Bandbreiten anhand einer Ende-zu-Ende-Basis (also vom Sender bis zum Empfänger) bestimmt und reserviert werden. Dies bedeutet aber auch, dass sämtliche betroffene Router die Informationen für jeden Datenstrom (dies meint die Verbindung von einem Sender zu einem Empfänger) speichern und jedes Paket analysieren müssen. Anhand dieser Informationen können sie dann bestimmen zu welchem Datenstrom dieses Paket gehört und somit können die Router das Paket, den von QoS geforderten Kriterien entsprechend, weiterleiten.

Das RSVP (Resource Reservation Protocol) ist Teil der Integrated Services Architektur. Dieses Protokoll ist für Reservierung bzw. Bestimmung der geforderten Bandbreite und anderer QoS-Ressourcen bezüglich Netzwerke zuständig.

Differentiated Services

Während die Integrated Services Architektur (kurz DiffServ) jedem einzelnen Datenstrom die benötigte Bandbreite zuordnet, bietet die Differentiated Services Architektur eine weniger

feine Unterscheidung zwischen den einzelnen Verbindungen. Dadurch ist nicht nur eine bessere Skalierbarkeit, sondern auch eine bessere Verwendbarkeit in großen Netzwerken bzw. im Internet selbst möglich.

Für diese Architektur gibt es im IPv6-Header ein eigenes Differentiated Services Feld (kurz DS-Feld) und wird im „Traffic Class" Feld des IPv6-Headers implementiert. Dieses Feld wird von Routern verwendet um für jedes Paket die QoS-Weiterleitungsbestimmungen festlegen zu können.

Es legt auch das Per-Hop Behavior (PHB) fest, welches dazu dient die Richtlinien und Prioritäten für ein bestimmtes Paket festzulegen bzw. zuzuordnen, wenn es eine bestimmten Hop (wie zum Beispiel ein Router es ist) passiert. Eben diese Regeln, wie Pakete innerhalb des DiffServ-Netzwerkes gehandhabt werden, werden dadurch festgelegt.

Pakete können anhand von Informationen im Paketheader und anderen vorbestimmten Regeln klassifiziert werden. Es gibt zwei Typen von solchen Klassifikationen:

- Behavior Aggregate Classifier (auch BA genannt)
  Teil Pakete anhand ihrer DS-Felder ein.
- Multi Field Classifier (kurz MF)
  Teil Pakete anhand mehrere Felder, dies können unter anderem Sender- und Empfängerfeld, DS-Feld oder Protokollnummer sein, oder anderen Informationen, wie das Interface der eingehenden Pakete, ein.

Quality of Service und IPv6

IPv6 unterstützt so viele Quality of Service Mechanismen wie möglich. Dabei können zum einen die IPv6-Header sowie die Extension-Headers für die unterschiedlichsten QoS-Dienste verwendet werden.

IPv6 Header

Im gewöhnlichen IPv6-Header können zwei verschiedene Felder für QoS verwendet werden:

- das Traffic Class Feld
- und das Flow Label Feld

Traffic Class Feld

Das 1 Byte große Traffic Class Feld wird beim Einsatz von QoS auch als DiffServ-Feld geführt, wie wir schon sehen, kommt dies also bei der Differentiated Services Architektur vor. Das Ziel von diesem Feld ist, dass DiffServ-Router unterschiedliche Routinen ausführen können, welche durch den Wert des DS-Feldes bestimmt werden. Diese Routinen haben wir vorhin schon unter dem Namen PHB kennengelernt.
Das DS-Feld sieht im Grunde so aus:

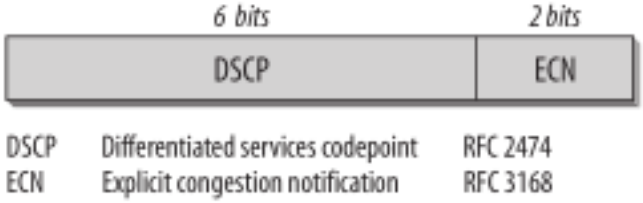


Abbildung - IPv6 Essentials (O´Reilly)

In diesem DS-Feld gibt es den 6-Bit großen Differentiated Services Codepoint, kurz und bündig DSCP genannt, und das 2-Bit große Explicit Congestion Notification (ECN)

Das DSCP-Feld

Das DSCP-Feld dient zur Klassifizierung der Prioritäten und kann theoretisch bis zu 64 Prioritäten differenzieren (da es ja 6-Bit groß ist), die als Codepoints bezeichnet werden. Die 64 Codepoints sind in drei Gruppen unterteilt.

- Gruppe 1
  Diese umfasst 32 Codepoints und dient den Anweisungen und dem Management von QoS-Anforderungen.
- Gruppe 2
  Hat 16 Codepoints und ist für experimentelle und lokale Benutzung vorgesehen.
- Gruppe 3
  Dient als Erweiterung, falls Gruppe 1 bereits voll ausgenutzt wird.

Dabei wird dieses Feld für die Bestimmung der Dienstleistungsqualität benutzt.

Das ECN-Feld

Das ECN-Feld ist 2-Bit groß und ist eine Erweiterung des TCP/IP und dient zur Überlastkontrolle. Die zwei Bit-Werte dienen zur Anzeige der Überlastung, damit können also Router anzeigen wie sehr oder wenig das Netzwerk ausgelastet ist. Die Werte können sein:

- 00: Paket verwendet kein ECN.
- 01/10: Sender und Empfänger unterstützen ECN.
- 11: Router signalisiert Überlastung.

PHB (Per-Hop-Behavior)

Die Qualitätsanforderung, welche durch die Markierung des DSCP kenntlich gemacht wird, heißt Per-Hop-Behavior (PHB). PHB beschreibt die Zuteilung bestimmter Bandbreiten und Speicherressourcen, ebenso die Anforderungen an die Leitung (von Sender zum Empfänger) wie zum Beispiel Verzögerungszeit oder Paketverluste. Dadurch ist, falls gewünscht, eine Einteilung in verschiedene Dienstklassen möglich.
Es gibt dabei verschieden PHB-Klassen:

- Best-Effort-Prinzip (BE)
  Es wird die schnellste mögliche Verbindung gefordert.
- Expedited-Forwarding
  Es wird zwischen höher und niedrigeren Prioritäten bei Paketen unterschieden.
- Assured-Forwarding
  Es wird gefordert, dass die Pakete bestimmt ankommen (niedrigste Fehlerrate).

Flow Label Feld

Das 20-Bit große Flow Label Feld im IPv6-Header kann dazu verwendet werden, wenn bestimmte Pakete eine besondere Behandlung von IPv6-Routern verlangen. Zum Beispiel könnte dies sein, wenn Echtzeit-Dienste zum Einsatz kommen. Dabei wird das Flow Label einem gewissen Datenfluss zugeordnet, was somit heißt, dass jedes Paket einem Datenfluss zugeordnet wird. Somit können Datenflüsse priorisiert werden und es muss nicht für jedes Paket einzeln die QoS-Anforderungen definiert werden. Auch können Router das Flow Label besonders effizient verarbeiten und es kommt auch bei aktivierten IPsec zum Einsatz.

IPv6 Extension-Headers

Wie bereits weiter oben erwähnt, werden die Extension-Headers auch für QoS-Anforderungen verwendet.

Routing Extension Header

Der Routing Extension Header kann verwendet werden, um eine bestimmte Route für den Datenverkehr anzufordern, indem eine Sequenz von Knoten (Router) angegeben wird. Jedoch muss hierbei die bevorzugte Route bekannt sein (wie zum Beispiel die Netzwerktopologie, QoS-Parameter (z.B. Datendurchsatz)).

Hop-By-Hop Options

Dies wird verwendet um ein Maximum von Router-Benachrichtigungen an alle Router, welche entlang des, durch QoS bestimmten, Pfades vorkommen, senden zu können. Damit kann allen Empfängern mitgeteilt werden, dass dieses Paket eine gesonderte Behandlung erfordert. Es erlaubt unteranderem das schnellere Verarbeiten von Paketen, da diese keine Protokolle von höheren Level verarbeiten müssen. Eine kleine Übersicht von Router-Benachrichtigungen:

WERT	AUFGABE
0	Das IP-Paket enthält eine Multicast-Listener-Discovery Nachricht
1	IP-Paket enthält eine RSVP Nachricht
2	IP-Paket enthält ein aktives Netzwerksegment, der Sender möchte zum Beispiel ein Programm in den Router laden, welches dann angepasste Funktionen ausführt.
3–35	IP-Paket enthält ein Aggregated Reservation Nesting Level
36–65,535	Von der IANA reserviert

8. Mobiles IPv6

Mobiles IPv6 (MIPv6) ist ein Abkömmling des IPv6-Protokolles, es unterstützt aber auch mobile Verbindungen. Also es ermöglicht das Bewegen von einem Netzwerk zu einem anderen ohne die Verbindung zu verlieren.
Im normalen IPv6 wird auch TCP verwendet, eine TCP-Verbindung ist jedoch durch IP- und Port-Adresse von Sender und Empfänger definiert. Sobald sich eine davon ändert, ist ein erneuter Verbindungsaufbau notwendig. Die Schwierigkeit liegt dabei darin, dass sich das mobile Gerät von einer Verbindung zur anderen bewegen können muss, ohne dabei eine neue IP-Adresse anzufordern, indem es sich bei beiden Verbindungen einwählen können soll.
Hier gibt es die Heimadresse, welche statisch und nicht veränderbar ist, somit identifiziert sie das Gerät mit einer TCP-Verbindung. Dann gibt es noch eine zweite IP-Adresse (auch care-of-Adresse genannt), diese ändert und passt sich dem Netzwerk an, in welchem sich das Gerät gerade befindet.
Es bleibt also eine „virtuelle“ Verbindung zwischen Quell- und Zielgerät bestehen, welche durch die Heimadresse gewährleistet wird, und für die einzelnen Netzwerke wird mit der Care-of-Adresse gearbeitet.

9. Quellen

Web

- http://www.ipv6-portal.de/informationen/einfuehrung/warum-ipv6.html
- http://www.rdfnuernberg.de/iav1a/theorie/ipv4.html
- http://www.ip-insider.de/themenbereiche/standards-und-protokolle/tcp-ip/articles/402191/
- http://www.holzke.de/documents/IPv6-Ausarbeitung.pdf
- http://pixabay.com/ (Quelloffene Grafiken, Wordart)

- https://de.wikipedia.org (mehrere Artikel, Grafiken)
- https://ipv6.net (mehrerer Artikel)
- http://www.itwissen.info (mehrere Artikel)
- http://www.cisco.com/c/en/us/td/docs/ios-xml/ios/qos_classn/configuration/xe-3s/qos-classn-xe-3s-book/ip6-qos-xe.html#GUID-5C8EBEB3-3763-4FA2-AFC9-23EA68D7371C
- https://docs.oracle.com/cd/E19683-01/817-0573/chapter1-25/index.html

Bücher und Fremddokumente

- IPv6 Essentails O´Reilly-Verlag von Silvia Hagen (ISBN-13: 978-0-596-10058-2)
- PRRV 5te Klasse von Mayrbäurl, Hinterholzer, Seyer, Ortner und Aschauer

Benutzte Software

- Microsoft Office Word 2013 (https://products.office.com/)
- Microsoft Visio Professional 2013 (https://products.office.com/)
- PAINT.NET (http://www.getpaint.net/)
- SumatraPDF (http://www.sumatrapdfreader.org/)
- Firefox (https://www.mozilla.org/)