



Wissenschaftliche Arbeit zum Thema

IPv4- und IPv6 Header – Analyse und Vergleich

von

Jan Arends

im Rahmen des Einstiegerprojekts
„Das neue Internetprotokoll“
WS 2013/14

Themenstellung: Martina Kannen
Verfasser: Jan Arends
E-Mail: jan.arends@smail.inf.h-brs.de
Eingereicht am: 13. Januar 2014

Abstract

IPv4 ist das gegenwärtig genutzte Internetprotokoll. Durch die wachsenden Anforderungen an das Internet soll IPv4 in den kommenden Jahren von seinem Nachfolger IPv6 abgelöst werden. Die Unterschiede zu dem Vorgänger IPv4 sind groß. So ist IPv6 keine Weiterentwicklung im eigentlichen Sinne, sondern ein völlig neues Protokoll. Diese Arbeit beschäftigt sich mit den IP-Headern des jeweiligen Protokolls. Anhand eines übersichtlichen Beispiels werden die einzelnen Felder der jeweiligen Header analysiert und im Anschluss verglichen.

Inhaltsverzeichnis

Abbildungsverzeichnis	ii
1 Einleitung	1
1.1 Motivation	1
1.2 Ziel der Arbeit	1
1.3 Aufbau der Arbeit	1
2 Analyse	2
2.1 IPv4 Header	2
2.2 IPv6 Header	5
3 Vergleich	8
3.1 Felder	8
3.2 Optionen	9
4 Zusammenfassung	10
5 Literaturverzeichnis	11

Abbildungsverzeichnis

Abbildung 1: Der IPv4 Header im Trace File	2
Abbildung 2: Die Aufteilung des DSCP-Pools [vgl. Nic98].....	3
Abbildung 3: Beispiel einer Fragmentierung.....	4
Abbildung 4: Der IPv6 Header im Trace File	5
Abbildung 5: Funktionsweise des Next Header Felds [Cis06]	7
Abbildung 6: IPv4- und IPv6 Header im Vergleich [Cis06].....	9

1 Einleitung

1.1 Motivation

„Die IP Version, welche wir heute alle in unseren Netzwerken und im Internet einsetzen, ist die IP Version 4, kurz IPv4 genannt. IPv4 wurde in den früheren 70er-Jahren von einer Gruppe Pionieren entwickelt, deren Ziel es war, ein paar staatliche und universitäre Netzwerke in den USA miteinander zu verbinden. Zu jener Zeit war ein Internet, wie wir es heute kennen, jenseits jeder Vorstellung. Es ging damals darum, ein Protokoll zu entwickeln, welches Tausende von Hosts verbinden würde. Entsprechend war es auch kein Design-Ziel des Protokolls, ein globales Netzwerk mit Milliarden von Hosts zu unterstützen. Umso faszinierender ist es, dass es diesen Pionieren gelang, ein Protokoll zu entwickeln, welches so skalierbar und stabil ist, dass es heute – 30 Jahre später – als Grundlage fürs Internet dienen kann. Heute ist es jedoch eindeutig in die Jahre gekommen, wo es Zeit für eine neue Generation ist.

IPv6 ist eine Evolution von IPv4 und wurde aufgrund der reichen Erfahrung mit IPv4 entwickelt. Bewährtes wurde beibehalten, bekannte Einschränkungen wurden behoben, Skalierbarkeit und Flexibilität wurden erweitert. IPv6 ist das Protokoll, das in der Lage sein wird, der Wachstumsrate des Internet und den Anforderungen zukünftiger Dienste gewachsen zu sein“ [Hag09, S. 1].

1.2 Ziel der Arbeit

Es werden wohl noch einige Jahre vergehen, bis IPv6 das gegenwärtig genutzte IPv4 vollständig ablöst. Es ist jedoch von Bedeutung sich heute schon konkret mit der Einführung von IPv6 zu beschäftigen, da sie mittelfristig unumgänglich ist. „Bezieht man IPv6 frühzeitig in die strategische Planung mit ein, befasst man sich frühzeitig mit möglichen Einführungsszenarien und berücksichtigt man die bevorstehende Einführung beim Tätigen von Investitionen in die Infrastruktur, so lassen sich teilweise beträchtliche Kosten vermeiden“ [Hag09, S. 2]

Ziel der Arbeit ist es, den Lesern die technischen Details der Header von IPv4 und IPv6 zu erläutern.

1.3 Aufbau der Arbeit

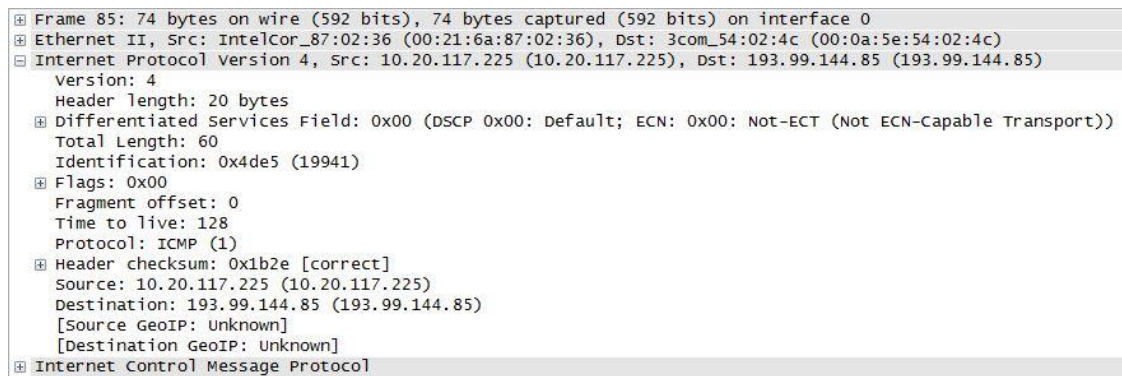
Nach der Einleitung beschäftigt sich Kapitel 2 mit der Analyse der beiden Header anhand eines Beispiels. In Kapitel 3 werden die beiden Header verglichen und die Gemeinsamkeiten und Unterschiede verdeutlicht. Der Umgang mit Optionen wird während dessen explizit erläutert. Abschließend erfolgt eine Zusammenfassung, die wesentliche Vor- und Nachteile beinhaltet.

2 Analyse

In diesem Kapitel werden die Header von IPv4 und IPv6 analysiert. Um Ihnen den Inhalt der einzelnen Felder zu veranschaulichen und Sie auf künftige Analysen eines Headers auf Bit-Ebene gefasst zu machen, stützt sich die Analyse auf einem konkreten Beispiel. Dazu betrachten wir jeweils einen Ping-Request zu der Homepage von Heise. Durch einen Ping lässt sich die Erreichbarkeit eines bestimmten Host in einem IP-Netzwerk prüfen. Zur Analyse der IP-Header wurde das Programm Wireshark benutzt.

2.1 IPv4 Header

Abbildung 1 zeigt den kompletten IPv4 Header des Ping-Requests in einem Trace File, welches im folgenden Abschnitt als Beispiel dient.



```

85 Frame 85: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface 0
85 Ethernet II, Src: IntelCor_87:02:36 (00:21:6a:87:02:36), Dst: 3com_54:02:4c (00:0a:5e:54:02:4c)
85 Internet Protocol Version 4, Src: 10.20.117.225 (10.20.117.225), Dst: 193.99.144.85 (193.99.144.85)
    Version: 4
    Header length: 20 bytes
    Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00: Not-ECT (Not ECN-Capable Transport))
    Total Length: 60
    Identification: 0x4de5 (19941)
    Flags: 0x00
    Fragment offset: 0
    Time to live: 128
    Protocol: ICMP (1)
    Header checksum: 0x1b2e [correct]
    Source: 10.20.117.225 (10.20.117.225)
    Destination: 193.99.144.85 (193.99.144.85)
    [Source GeoIP: Unknown]
    [Destination GeoIP: Unknown]
85 Internet Control Message Protocol
```

Abbildung 1: Der IPv4 Header im Trace File

- **Version**

Die ersten 4 Bits des IPv4 Headers beschreiben die eingesetzte Version des Internetprotokolls. Dementsprechend wurden diese Bits in unserem Beispiel auf 1000_2 gesetzt, was dem Wert 4 entspricht.

- **Internet Header Length (IHL)**

Dieses Datenfeld ist ebenfalls 4 Bit lang und gibt mittels 32 Bit Einheiten die Gesamtlänge des IP-Headers an. Durch den Wert dieses Feldes weiß der Empfänger, an welcher Stelle die Nutzdaten beginnen. Warum der IPv4 Header eine flexible Größe hat, wird später genauer erklärt. In unserem Beispiel ist das IHL-Feld mit 0101_2 belegt. Es entspricht also dem Wert 5. Durch die 32 Bit Einheiten lässt sich mit folgender Rechnung die Größe des Headers zeigen: $5 \cdot 32 \text{ Bit} = 160 \text{ Bit} \Rightarrow 20 \text{ Byte}$. Diese Größe entspricht der minimalen Größe eines IPv4 Headers. Die maximale Größe des Headers beträgt, durch die 4 Bits dieses Feldes, 60 Byte.

- **Differentiated Services (DS)**

Ursprünglich befand sich an dieser Stelle das Type of Service (ToS) Feld. Da dieses Feld nicht einheitlich definiert war und die Gefahr bestand, dass sein Gebrauch letztendlich eher zu Verzögerungen aufgrund von erhöhtem Verarbeitungsaufwand auf Routern führen würde, wurde das Feld jedoch nie wirklich benutzt. Das Redesign von IP für IPv6 stellte zusammen mit den absehbaren steigenden Anforderungen an Quality of Service (QoS) eine gute Gelegenheit dar, dieses Thema in Angriff zu nehmen. Um QoS Optimierung zu ermöglichen, gibt es z.Z. zwei Architekturen: Integrated Service (IntServ) und Differentiated Services (DiffServ). Ohne näher auf die Architekturen eingehen zu wollen, kann man sagen, dass IntServ aufwendig in der Implementation und aufgrund der limitierten Skalierbarkeit keine generelle QoS-Lösung für das globale Internet darstellt. DiffServ verbessert die Skalierbarkeit in großen Netzwerken und im Internet. Aus diesem Grund wurde DiffServ in den IPv4 Header implementiert. [vgl. Hag09, S471f.] „Das DS-Feld enthält die QoS- Anforderungen eines Paketes und wird von jedem

DiffServ Router benutzt, um die Art und Weise zu bestimmen, wie er dieses Paket weiterleitet. Damit können kommunizierende Knoten ihren Datenaustausch in verschiedene Kategorien einteilen, welche durch ein sogenanntes Per-Hop-Behaviour (PHB) identifiziert sind. Aufgrund des PHB erhalten die Pakete auf DiffServ Routern eine entsprechende Behandlung“ [Hag09, S. 474].

Die ersten 6 Bits des DS-Feldes beschreiben den sog. Codepoint (DSCP), mit dem ein Pool von Per-Hop-Behaviors (PHB) angegeben wird. „Ziel dieser Spezifikation ist es, dass die Router, welche DiffServ unterstützen, einen bekannten Satz von DS Routinen kenne, die aufgrund der Werte im DS-Feld angezeigt werden [Hag09, S476]. Mit dem DSCP-Feld lassen sich 64 verschiedene Codepoints darstellen. Um die Zuweisung von PHBs zu kontrollieren, wurde dieser Pool in drei Bereiche eingeteilt. Abbildung 2 zeigt diese Bereiche.

Pool	Codepoint	Zuweisung
1	xxxxx0	Standardbenutzung
2	xxxx11	Experimentelle/ lokale Benutzung
3	xxxx01	Experimentelle/ lokale Benutzung

Abbildung 2: Die Aufteilung des DSCP-Pools [vgl. Nic98]

Pool 1 stellt 32 empfohlene Standard-Codepoints zu Verfügung. Pool 2 dient zu experimentellen Zwecken und findet interne Verwendung. Pool 3 hat dieselbe Zuweisung wie Pool 2, jedoch mit dem Unterschied, dass dieser für Standardbenutzung freigegeben werden kann, wenn Pool 1 ausgeschöpft ist. Die PHBs definieren, wie ein Paket weitergeleitet wird. [vgl. Hag09, S. 471f.]. Der Default Wert des DSCP entspricht 0, welcher sich auch in unserem Beispiel wiederfinden lässt. „Dieser Default PHB entspricht dem normalen, best-effort Routingverhalten, wie es jeder Router unterstützt, d.h. es werden so viele Pakete wie möglich so schnell wie möglich weitergeleitet, je nach Kapazität und Auslastung“ [Hag09, S. 477].

- **Explicit Congestion Notification (ECN)**

ECN ist ein Teil des DS-Feldes und beinhaltet die letzten 2 Bits dieses Feldes. Es dient zur Überlastkontrolle und wird von Routern verwendet, welche durch eine einfache Markierung eines Bits, eine drohende Überlast den nachfolgenden Routern mitteilen können [vgl. Hag09, S. 478]. In unserem Beispiel wurde keins der beiden Bits gesetzt. Dies bedeutet, dass dieses Paket kein ECN benutzt. Ist der Wert des ECN Felds 01/10 bedeutet dies, dass Absender und Empfänger ECN aktiviert haben, es aber keine Überlast besteht. Sind beide Bits gesetzt, so signalisiert der Router eine bevorstehende Überlast [vgl. Ram01].

- **Total Length**

In diesem Feld wird die Gesamtlänge des IP-Pakets (Header + Nutzdaten) in Byte angegeben. Dieses Feld nutzt 16 Bit, woraus sich die maximale Größe eines kompletten IP-Pakets ergibt: $2^{16} \Rightarrow 65.536 \text{ Byte} \Rightarrow 64 \text{ KiB}$ [vgl. Inf81]. Das Paket aus unserem Beispiel hat eine Größe von 60 Bytes. Auf Bit-Ebene betrachtet steht hier also 00000000 00111100₂.

- **Identification**

Die nächsten 3 Felder „Identification“, „Flags“ und „Fragment Offset“ spielen bei der Fragmentierung von IP-Paketen eine wichtige Rolle. Eine Fragmentierung bedeutet das Aufteilen eines Datenpaketes in mehrere Datenblöcke, falls die Gesamtlänge des Datenpakets die maximal zulässige Paketlänge der Vermittlungsschicht (Maximum

Transmission Unit, kurz MTU) überschreitet. Das Feld „Identification“ dient zur Identifikation von fragmentierten Paketen durch eine eindeutige Kennung. D.h. alle Fragmente eines einzelnen Datagramms sind mit demselben Wert markiert. Dieser dient dem Empfänger dazu, die Zusammengehörigkeit von Fragmenten erkennen zu können, um diese nachher wieder zusammen zu setzen. Bei dem Ping aus unserem Beispiel war keine Fragmentierung von Nöten. Um das Prinzip der Fragmentierung und dem Sinn des Identifikationsfeldes zu verdeutlichen, nehmen wir jedoch an, dass eine Fragmentierung vorgenommen wurde. In unserem Beispiel hat das Feld den Wert 19941 ($01001101\ 11100101_2$) bekommen. Bei einer Fragmentierung des Pakets, würde jedes Fragment ebenfalls diese Nummer zugewiesen bekommen. Abbildung 3 veranschaulicht dieses Prinzip.

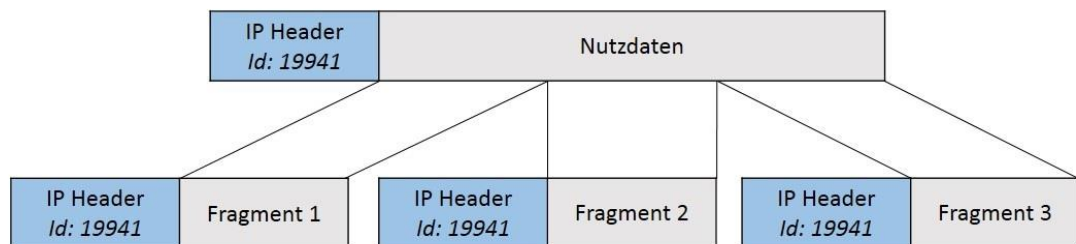


Abbildung 3: Beispiel einer Fragmentierung

- **Flags**

Dieses Feld ist 3 Bit groß und gibt Auskunft über die Fragmentierung des einzelnen Fragments. Das erste Bit ist reserviert und wird immer mit 0 belegt. Falls das zweite Bit gesetzt ist, also den Wert 1 hat, liegt Fragmentierung vor und das Fragment ist nur ein Teil eines Datagramms. Mit einem gesetztem dritten Bit wird angegeben, dass noch weitere Fragmente folgen [vgl. Inf81].

In unserem Beispiel hat das Flags-Feld den Wert 000_2 und zeigt somit an, dass es kein fragmentiertes Paket ist und dass auch dementsprechend keine weiteren fragmentierten Pakete folgen. Zumindest nicht von den gleichen Datagramm.

- **Fragment Offset**

Enthält ein IP-Paket fragmentierte Nutzdaten, steht in diesem Feld die Position der Daten im ursprünglichen IP-Paket. Anhand dieser Position kann das Paket beim Empfänger wieder in die ursprüngliche Reihenfolge defragmentiert werden [vgl. Inf81]. Da in unserem Beispiel keine Fragmentierung von Nöten war, wurde auch kein Bit in diesem Feld gesetzt.

- **Time To Live (TTL)**

Der IP-Header wurde weiterhin mit den 8 Bits großen Time To Live Feld ausgestattet, um zu verhindern, dass ein Paket unkontrolliert im Internet verweilt. Diese kann z.B. dann geschehen, wenn der Router die Zieladresse nicht findet. Um den unnötigen Traffic zu vermeiden, legt der Sender eine Lebensdauer für das Paket fest. Jede Station, die ein Paket weiterleiten muss, verringert den Wert um 1. Solch eine Station wird auch Hop genannt. Hat das TTL-Feld den Wert 0 erreicht, wird das IP-Paket verworfen und eine entsprechende Meldung an die Quelladresse gesendet [vgl. Inf81]. In unserem Beispiel wurde der Start-/ Initialwert auf 128 ($1000\ 0000_2$) festgelegt.

- **Protocol**

Die nächsten 8 Bits werden als Protocol-Feld bezeichnet und geben Auskunft über das übergeordnete Protokoll (engl.: Upper Layer Protocol), zu dem die transportierten Nutzdaten gehören [vgl. Inf81]. Das Feld repräsentiert eine Protokoll-Nummer, die von der Abteilung „Internet Assigned Numbers Authority“ (kurz IANA) der Organisation

„Internet Corporation for Assigned Names and Numerbs“ definiert wurde. Die Protokoll-Nummern lassen sich in einer Online-Datenbank nachschlagen. Der Ping aus unserem Beispiel benutzt das Protokoll ICMP. Dafür legte die IANA den Wert 1 fest, der auch in unserem Beispiel zu finden ist [IAN13].

- **Header Checksum**

Anhand der Checksumme des Headers wird die Korrektheit des IP-Headers sichergestellt. Für die Richtigkeit der Nutzdaten ist ein übergeordnetes Protokoll mit eigener Checksumme zuständig. Dadurch, dass sich der Header ständig ändert (z.B. durch das TTL-Feld), muss jeder Hop die Checksumme neu berechnen. Bekommt ein Router ein IP-Paket mit inkorrekt Header Checksumme, wird das Paket verworfen. Der Header unseres Beispiels wurde mit der Checksumme 42952 (10100111 11001000₂) versehen.

- **Source IP Address**

In diesem Feld wird die IP-Adresse eingetragen, die das Paket abgeschickt hat (Quelladresse). Da eine IPv4 Adresse 32 Bit lang ist, ist dieses Feld auch genauso groß.

- **Destination IP Address**

Analog zu der Source-IP Adresse wird in diesen Feld die IP-Adresse geschrieben, für die das Paket bestimmt ist (Zieladresse). Dafür wurden ebenfalls 32 Bit im IP-Header reserviert

- **Options und Padding**

Dieses Feld kann bei Bedarf genutzt werden. Es dient zum Anfügen von Zusatzoptionen wie z.B. Security-Optionen, Source Routing oder Timestamp-Informationen. Die Zusatzinformationen müssen ein vielfaches von 32 Bit lang sein. Die Wahrscheinlichkeit, dass die Informationen die Länge eines Vielfachen von 32 Bit hat, ist gering. Um dies jedoch trotzdem erreichen zu können wurde das Padding hinzugefügt. Es füllt die Optionen mit 0-Bits, solange ein Vielfaches von 32 erreicht ist [vgl. Inf81].

2.2 IPv6 Header

Nachdem der IPv4 Header vollständig analysiert wurde kommen wir jetzt zum IPv6-Header. Als Beispiel betrachten wir hierzu wieder einen Ping-Request zu der Homepage www.heise.de (mit der IPv6 Adresse 2A02:2E0:3FE:100::7). Abbildung 4 zeigt den IPv6 Header im Trace File. Dieser wird anschließend Stück für Stück erläutert.

```

⊞ Frame 2: 134 bytes on wire (1072 bits), 134 bytes captured (1072 bits) on interface 0
⊞ Ethernet II, Src: 3com_54:02:4c (00:0a:5e:54:02:4c), Dst: IPv6mcast_00:00:00:01 (33:33:00:00:00:01)
⊞ Internet Protocol Version 6, Src: fe80::20a:5eff:fe54:24c (fe80::20a:5eff:fe54:24c), Dst: ff02::1 (ff02::1)
  ⊞ 0110 .... = Version: 6
  ⊞ .... 0000 0000 .... = Traffic class: 0x00000000
    .... 0000 0000 0000 0000 0000 0000 = Flowlabel: 0x00000000
    Payload length: 80
    Next header: ICMPv6 (58)
    Hop limit: 255
    Source: fe80::20a:5eff:fe54:24c (fe80::20a:5eff:fe54:24c)
    [Source SA MAC: 3com_54:02:4c (00:0a:5e:54:02:4c)]
    Destination: ff02::1 (ff02::1)
    [Source GeoIP: Unknown]
    [Destination GeoIP: Unknown]
⊞ Internet Control Message Protocol v6

```

Abbildung 4: Der IPv6 Header im Trace File

- **Version**

Genau wie bei IPv4 stehen bei IPv6 die ersten 4 Bit für die IP-Versionsnummer. Anhand der Belegung 0110₂ in unserem Beispiel kann man erkennen, dass die Version 6 eingesetzt wurde.

- **Traffic Class**

Das 1 Byte große Feld dient dazu, verschiedene Klassen oder Prioritäten von Paketen zu unterscheiden und dementsprechend zu behandeln. Das Feld befasst sich mit Real-Time Daten und anderen Daten, die eine besondere Behandlung benötigen [vgl. Hag09, S. 22]. Das Feld benutzt wie der IPv4 Header das DS-Feld mit einem 6 Bit langem DSCP und eine 2 Bit lange Kennung für ECN (siehe Unterpunkte Differentiated Services und Explicit Congestion Notification in Kapitel 2.1). Wie im vorigen Beispiel bei IPv4 wurde hier ebenfalls kein Codepoint gesetzt und ECN nicht benutzt (0000 0000₂).

- **Flow Label**

Das Flow Label Feld dient ebenfalls zur Unterstützung von QoS. Es ist zu der erleichterten Verarbeitung von Real-Time Paketen behilflich, indem es Pakete, welche gleichartig behandelt werden müssen, mit einem sogenannten Label kennzeichnet. Das Flow Label wird vom Absender der Pakete zugewiesen und ist eine Zufallszahl aus dem Bereich von 00001₁₆ bis FFFFF₁₆, die in Kombination mit der Absenderadresse eindeutig identifizierbar ist. Ein Paket ohne QoS-Anforderungen hat alle Bits des Feldes auf 0 gesetzt. Pakete die demselben Flow angehören, müssen sowohl identische Absender- und Empfängeradressen sowie identische Absender- und Empfänger Ports haben. Ein Router kann daraufhin den Flow speichern und alle Pakete mit demselben Label auf die gleiche Weise verarbeiten, ohne weitere Informationen über das Paket zu kennen. Das Paket aus unserem Beispiel wurde mit keinem Label versehen, da keine erhöhte Priorität vorgesehen war.

- **Payload Length**

Dieses 2 Byte Große Feld gibt die Länge des sogenannten Payloads in Byte an. Dabei handelt es sich sowohl um Nutzdaten, als auch um sogenannte Extension Headers, welche zum Ende dieses Abschnittes beschrieben werden. Die Payload Länge in unserem Beispiel beträgt 40 Bytes. Die maximale Größe des Payloads ist durch die verfügbaren 2 Byte auf 64 KiB beschränkt.

- **Next Header**

In diesem Feld wird der nachfolgende Header beschrieben. Dieser kann ein Protokoll des nächst höheren Layers oder ein Extension Header sein. Wurde kein Extension Header angefügt und der Next Header ist dementsprechend ein Protokoll wie z.B. TCP oder UDP ist, enthält dieses Feld dieselben Werte wie das Protocol Feld im IPv4 Header. Wenn der nächste Header ein Extension Header ist, beschreibt dieses Feld den Typ des nächsten Extension Headers. Extension Header befinden sich immer zwischen dem IPv6 Header und den nachfolgenden Protokoll-Header [vgl. Hag09, S.23]. Abbildung 5 verdeutlicht, mit welchem Wert das Next Header Feld versehen wird.

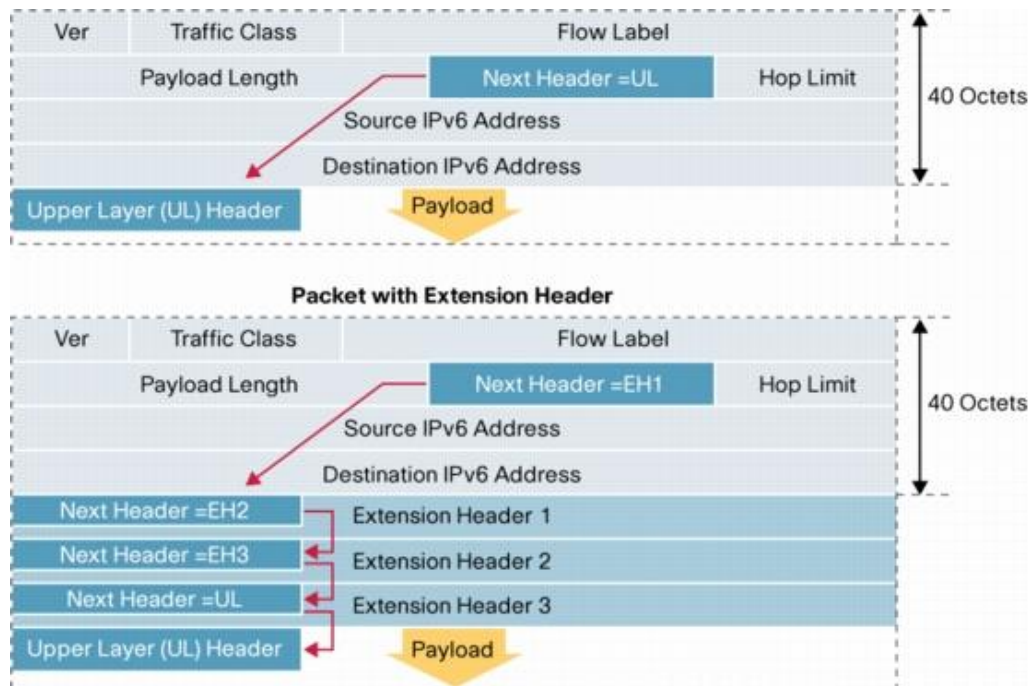


Abbildung 5: Funktionsweise des Next Header Felds [Cis06]

Das Programm Ping aus unserem Beispiel, verwendet das Protokoll ICMPv6. In der bereits genannten Datenbank für Protokoll-Nummern wurde für ICMPv6 die Protokoll Nummer 58 festgelegt. Dem entsprechend wurde das Next Header Feld in unserem Beispiel auf 0011 1010₂ gesetzt.

- **Hop Limit**

Dieses Feld entspricht dem TTL Feld im IPv4 Header und gibt die Anzahl der Hops an, die das IP Paket noch passieren darf. Jeder Hop zieht beim Weiterleiten den Wert 1 ab. Ist der Wert in diesem Feld auf 0 reduziert worden, verwirft der letzte Hop das Paket und schickt dem Absender des Pakets eine entsprechende ICMPv6-Meldung. In unserem Beispiel wurde der Hop Limit auf 128 festgelegt [vgl. Hag09, S. 25].

- **Source Address**

In diesem Feld wird die IPv6-Adresse des Absenders eingetragen. Folglich werden dafür 128 Bit in Anspruch genommen.

- **Destination Address**

Hier ist die IPv6-Adresse des Empfängers vermerkt. Anders als bei IPv4 muss hier nicht zwingend die endgültige Ziel-Adresse stehen, sondern ist z.B. auch die Adresse des nächsten Hops möglich. Diese wird gesetzt, wenn ein Routing Header vorhanden ist.

- **Extension Header**

IPv6 transportiert die Optionen in zusätzlichen Headers, den Extension Headers. Diese Extension Header enthalten Optionen und Informationen, die für die Netzwerkschicht (IP Layer) von Bedeutung sind. Extension Header werden nur eingefügt, wenn Optionen vorhanden sind [vgl. Hag09, S. 27]. Die aktuelle IPv6 Spezifikation definiert 11 Extension Header, wovon zwei für experimentelle Zwecke benutzt werden [IAN13].

3 Vergleich

In folgenden Abschnitt möchte ich auf die Unterschiede und die Gemeinsamkeiten der beiden Header eingehen. Der IPv6 Header wurde durch die Erneuerung stark vereinfacht. Von den 14 Feldern bei IPv4 blieben nur noch 7 Felder in IPv6 übrig. Diese Vereinfachung ermöglicht Routern, Pakete schneller verarbeiten zu können. Darüber hinaus werden nun Erweiterungen und Optionen durch die Extension Header besser unterstützt.

3.1 Felder

Das Version Feld hat sich bei der Erneuerung nicht verändert. In beiden Versionen steht es mit 4 Bits am Anfang des Headers.

Das Feld Header Length ist, durch das Auslagern der Optionen außerhalb des Headers, überflüssig geworden. Im Gegensatz zum IPv4 Header hat der IPv6 Header eine feste Länge von 40 Bytes. Aus diesem Grund kann auf die Berechnung der IP-Header-Größe verzichtet werden.

Das in IPv4 genannte ToS-Feld wurde durch das Feld Traffic Class bei IPv6 ersetzt. Trotz der unterschiedlichen Bezeichnungen erfüllen jedoch beide Felder die gleichen Aufgaben, und zwar bestimmen sie die Beförderungspriorität des Pakets, mittels des DSCP.

Das Feld „Flow Label“ wurde in IPv6 neu eingeführt. Dieses Feld dient zur Fluss-Kennzeichnung von Paketen. Anhand des eingetragenen Werts in diesem Feld, kann der Router erkennen, ob eine spezielle Behandlung gewünscht ist.

Das Feld Total Length ist vergleichbar mit dem Payload Length Feld in IPv6. Beide Felder geben Auskunft über die Größe des gesamten IP-Pakets an. Die Berechnung der Größe ist zwischen den Versionen jedoch unterschiedlich. Das Total Length Feld bezieht Header- und Nutzdaten in die Berechnung mit ein. IPv6 hingegen verzichtet auf die Berechnung des Headerbereichs, da dieser sowieso immer die gleiche Größe besitzt und eine ständige Berechnung dementsprechend überflüssig wäre. Dafür müssen die Extension Header bei IPv6 berücksichtigt werden. Die Länge der Nutzdaten werden in jeder Version in die Berechnung mit einbezogen.

Des Weiteren wurden die drei Felder Identification, Flags und Fragment Offset bei der Erneuerung aus dem eigentlichen Header gestrichen. Die genannten Felder werden bei IPv4 zur Realisierung und Verarbeitung von fragmentierten Paketen genutzt. Insgesamt werden dazu 32 Bit im IPv4 Header reserviert, die auch übertragen werden, wenn keine Fragmentierung vorgenommen wurde. Dies bedeutet unnötigen Header Overhead. Bei IPv6 wird die Fragmentierung mittels Extension Header realisiert, welcher nur dann eingefügt wird, wenn Fragmentierung notwendig ist.

Das Feld Time To Live aus dem IPv4 Header wurde bei IPv6 in „Hop Limit“ umbenannt. Sowohl die Funktion, als auch die Länge der Felder sind jedoch gleich geblieben. In beiden Fällen wird hier ein Wert angegeben, der Aussage darüber macht, wie viele Stationen ein Paket noch passieren darf. Jede Vermittlungsstation senkt diesen Wert um 1. Erreicht das Paket innerhalb dieses Limits nicht sein Ziel, wird es verworfen und der Empfänger wird dementsprechend benachrichtigt.

Das Gleiche gilt auch bei den Feldern Protocol und Next Header. Hier sind ebenfalls die Funktion und die Länge des Feldes identisch geblieben. Lediglich ein kleiner Unterschied ist vorhanden: Wird bei IPv6 ein Extension Header eingesetzt, so gibt das Feld nicht das Folgeprotokoll an sondern einen Wert zur Identifizierung des nachfolgenden Extension Headers. In unseren Beispielen wurde der Wert bei IPv4 auf 1 (0000 0001₂) für ICMP und bei IPv6 auf 58 (00111010₂) für ICMPv6 gesetzt.

Um die Verarbeitungsgeschwindigkeit von IP weiter zu erhöhen, wurde das Header Checksum Feld im IPv6 Header entfernt. Durch die Nutzung von Checksummen auf der Sicherungssicht, ist die Gefahr von nicht erkannten Fehler und fehlgeleiteten Paketen minimal. Das Bilden und Überprüfen von Checksummen auf dieser Schicht war jedoch zurzeit als IPv4 entwickelt wurde nicht üblich. So entschloss man sich, eine Checksumme in den IP Header zu implementieren [vgl. Hag09, S.20].

Die Quell- und Zieladressenfeldern wurden auf die 128 Bit IPv6 Adressen angepasst und erhalten somit eine Länge von 128 Bit für die jeweilige Adresse.

Abbildung 6 fasst die Gemeinsamkeiten und Unterschiede zusammen.

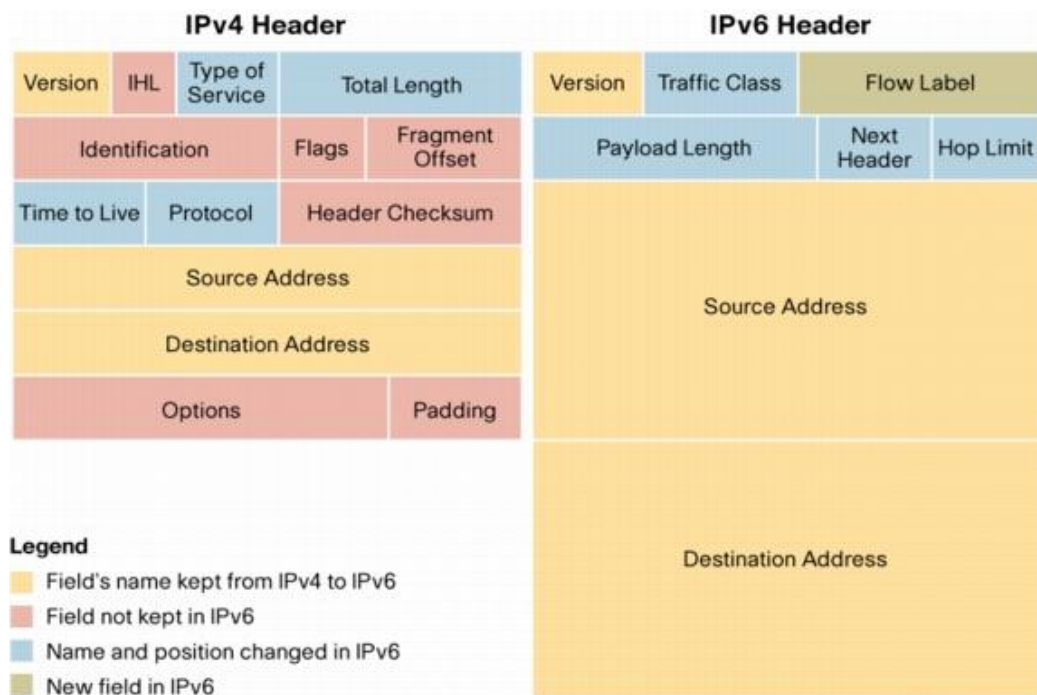


Abbildung 6: IPv4- und IPv6 Header im Vergleich [Cis06]

3.2 Optionen

Das Options Feld in IPv4 wurde aufgrund starker Performance Beeinträchtigung eher weniger genutzt. Diese Einbußen lassen sich z.B. anhand von Hardware-Forwarding-Geräten zeigen. Diese müssen alle Pakete mit Optionen an den Haupt-Prozessor übergeben und auf Software-Ebene verarbeiten, was eine langsamere Verarbeitung zur Folge hat. So kann man sagen, dass die Struktur des IPv4 Headers für heutige Ansprüche eher Ineffizient geworden ist. Denn desto einfacher ein Header strukturiert ist, desto schneller kann er verarbeitet werden. Aus diesem Grund werden bei IPv6 die Zusatzinformationen nicht mehr innerhalb des eigentlichen IP-Headers verarbeitet, sondern in zusätzlichen Extension Headers ausgelagert. Die Extension Headers werden, genau wie die Optionen im IPv4 Header, nur eingefügt, wenn Optionen vorhanden sind. [vgl. Hag09, S. 27].

4 Zusammenfassung

Die Umgestaltung des IP-Headers ist eine wesentliche Neuerung bei IPv6. Die Schlankheit des Headers und eine bessere Unterstützung für zusätzlichen Optionen erlauben eine schnelle Verarbeitung von IP Paketen. Des Weiteren wurde unnötiger Header Overhead vermieden, indem Felder, die zuvor Optional genutzt werden konnten, durch Extension Header ersetzt wurden. Neue Extension Header lassen sich jederzeit definieren und einführen, ohne die Grundstruktur des IPv6-Headers verändern zu müssen. So kann künftig schnell auf neue Anforderungen reagiert werden.

5 Literaturverzeichnis

- [Cis06] Cisco: "IPv6 Extension Headers Review and Considerations", Oktober 2006, http://www.cisco.com/en/US/technologies/tk648/tk872/technologies_white_paper0900aecd8054d37d.html, Stand 27 Dezember 2013
- [Dee98] S. Deering, Cisco, R. Hinden, Nokia, "Internet Protocol, Version 6 (IPv6) Specification" RFC 2460, Dezember 1998, <http://www.ietf.org/rfc/rfc2460.txt>, Stand 13. Dezember 2013
- [Hag09] Hagen, Silvia: „IPv6: Grundlagen - Funktionalität – Integration“, Sunny Edition, 2. Auflage, Dezember 2009
- [IAN13] Internet Assigned Numbers Authority: "Assigned Internet Protocol Numbers", Dezember 2013, <http://www.iana.org/assignments/protocol-numbers/protocol-numbers.xhtml#protocol-numbers-1>, Stand 09 Januar 2014
- [Inf81] Information Sciences Institute, University of Southern California "Internet Protocol" RFC 791, September 1981, <http://tools.ietf.org/html/rfc791>, Stand 14. Dezember 2013
- [Nic98] K. Nichols, Cisco Systems, S. Blaker, Torrent Networking Technologies, F. Baker, D. Black, EMC Corporation, "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers" RFC 2474, Dezember 1998, <http://www.ietf.org/rfc/rfc2474.txt>, Stand 14. Dezember 2013
- [Ram01] K. Ramakrishnan, TeraOptic Networks, S. Floyd, ACIRI, D. Black, EMC, "The Addition of Explicit Congestion Notification (ECN) to IP" RFC 3168, September 2001, <http://www.ietf.org/rfc/rfc3168.txt>, Stand 14. Dezember 2013