IPv4 Handout Tanenbaum Wetherall

Donnerstag, 8. Oktober 2020 19:33

05H IPv4 Handout ...

Kostenarten

Vermittlungsschicht im Internet

Nun ist es an der Zeit, die Vermittlungsschicht im Internet genauer zu untersuchen. Doch bevor wir ins Detail gehen, lohnt sich ein Blick auf die Prinzipien, die in der Vergangenheit für den Entwurf verantwortlich waren und heutzutage seinen Erfolg ausmachen. Es scheint zu oft, dass diese heute in Vergessenheit geraten sind. Diese Prinzipien werden in RFC 1958, der sehr lesenswert ist (und zur Pflichtlektüre aller Protokollentwickler gehören sollte — mit entsprechender Abschlussprüfung) aufgeführt und erörtert. Dieser RFC stützt sich stark auf Konzepte, die in Clark (1988) und Saltzer et al. (1984) vorgestellt werden. Das Folgende ist eine Übersicht der nach unserem Dafürhalten wichtigsten zehn Prinzipien (vom wichtigsten zum am wenigsten wichtigen):

1. Stellen Sie sicher, dass es funktioniert. Schließen Sie das Design oder den Standard nicht ab, bevor nicht mehrere Prototypen erfolgreich miteinander kommuniziert haben. Zu oft erstellen die Entwickler zuerst einen 100 Seiten langen Standard, lassen diesen genehmigen und entdecken dann, dass so viele Fehler enthalten sind, dass er nicht funktioniert. Dann erstellen Sie die Version 1.1 des Standards. Dies ist nicht der richtige Weg.
2. Halten Sie den Entwurf einfach. Wählen Sie in Zweifelsfällen die einfachste Lösung. William von Occam formulierte dieses Prinzip im 14. Jahrhundert (das Ökonomieprinzip von Occam). In modernen Begriffen formuliert: Vermeiden Sie zu viele Funktionen. Ist eine Funktion nicht absolut notwendig, lassen Sie sie weg, Insbesondere dann, wenn die gleiche Wirkung durch die Kombination anderer Funktionen erzielt werden kann.
3. Treffen Sie eine klare Auswahl. Wenn verschiedene Möglichkeiten bestehen, das Gleiche zu erledigen, entscheiden Sie sich für eine. Wenn Sie die gleiche Sache auf zwei oder mehr Arten ausführen, sieht dies nach Problemen aus. Standards verfügen oftmals über mehrere Optionen, Modi oder Parameter, da verschiedene einflussreiche Parteien darauf bestehen, dass ihr Weg der beste ist. Entwickler sollten sich dieser Tendenz mit ganzer Kraft widersetzen. Sagen Sie einfach nein.
4. Nutzen Sie die Modularität. Dieses Prinzip führt direkt zum Konzept von Protokollstapeln, bei denen jede Schicht unabhängig von allen anderen ist. Wenn es erforderlich wird, dass ein Modul oder eine Schicht geändert wird, so sind auf diese Weise keine anderen betroffen.
5. Erwarten Sie Heterogenität. In jedem großen Netz treten verschiedene Typen von Hardware, Übertragungseinrichtungen und Anwendungen auf. Um diese handhaben zu können, muss der Netzentwurf einfach, allgemein und flexibel sein.
6. Vermeiden Sie statische Optionen und Parameter. Können Parameter nicht vermieden werden (wie z.B. die maximale Paketgröße), so ist es am besten, wenn der Sender und der Empfänger einen Wert aushandeln, anstatt feste Optionen vorzugeben.
7. Streben Sie nach einem guten Entwurf, er muss nicht perfekt sein. Oftmals haben Entwickler einen guten Entwurf, können aber einen merkwürdigen Sonderfall nicht behandeln. Entwickler sollten dann, anstatt sich den Entwurf zu verderben, den guten Entwurf nehmen und den Entwicklungsaufwand für eine Lösung des Sonderfalls den Leuten mit den ausgefallenen Anforderungen überlassen.
8. Handhaben Sie das Senden streng, das Empfangen aber tolerant. Anders ausgedrückt, senden Sie nur Pakete, die den Standards hundertprozentig entsprechen. Akzeptieren Sie auch ankommende Pakete, die die Standards nicht vollständig erfüllen, und versuchen Sie, diese zu verarbeiten.
9. Berücksichtigen Sie die Skalierbarkeit. Wenn das System Millionen von Hosts und Milliarden von Benutzern leistungsstark verwalten soll, sind zentrale Datenbanken absolut nicht möglich, und die Last muss gleichmäßig über die verfügbaren Ressourcen verteilt werden.
10. Berücksichtigen Sie Leistung und Kosten. Wenn ein Netz eine schlechte Leistung oder enorme Kosten aufweist, wird es niemand verwenden.

Wenden wir uns nun von den allgemeinen Prinzipien hin zu den Einzelheiten der Vermittlungsschicht im Internet. Auf der Vermittlungsschicht kann das Internet als Sammlung von Teilnetzen oder autonomen Systemen (AS) betrachtet werden, die miteinander verbunden sind. Es gibt keine echte Struktur, sondern mehrere größere Backbones. Diese Backbones werden aus Leitungen mit hoher Bandbreite und schnellen Routern gebildet. Die größten dieser Backbones, mit denen sich alle anderen verbinden, um den Rest des Internets zu erreichen, werden Tier-1-Netze genannt. An die Backbones sind ISPs angeschlossen, die einen Internetzugang für private und geschäftliche Anwendungen, für Datenzentren und Housing-Einrichtungen mit vielen Servern sowie für regionale Netze (mittlere Ebene) zur Verfügung stellen. Die Datenzentren verarbeiten einen Großteil des Inhalts, der über das Internet gesendet wird. Mit den regionalen Netzen sind weitere ISPs, die LANs vieler Universitäten und Unternehmen sowie andere Randnetze verbunden. Diese quasi hierarchische Organisation ist in »Abbildung 5.45 dargestellt.

Der Klebstoff, der das alles zusammenhält, ist das Protokoll der Vermittlungsschicht — IP (Internet Protocol). Im Gegensatz zu den meisten älteren Protokollen der Vermittlungsschicht wurde IP von Anfang für das Internetworking ausgelegt. Stellen Sie sich die Vermittlungsschicht auf die folgende Weise vor: die Aufgabe der Vermittlungsschicht ist die Bereitstellung einer Möglichkeit, nach bestem Bestreben (d.h. nicht garantiert) Pakete von der

Quelle zum Ziel zu befördern, unabhängig davon, ob sich diese Rechner im gleichen Netz befinden oder ob andere Netze dazwischenliegen.

Die Kommunikation im Internet funktioniert so, dass die Transportschicht Datenströme entgegennimmt und so aufgeteilt, dass sie als IP-Pakete gesendet werden können. Theoretisch können Pakete jeweils bis zu 64 KB groß sein, doch in der Praxis liegt die Größe bei nicht mehr als 1 500 Byte (so passen Sie in einen Ethernet-Rahmen). IP-Router leiten jedes Paket durch das Internet entlang eines Pfads von einem Router zum nächsten weiter, bis das Ziel erreicht ist. Am Ziel reicht die Vermittlungsschicht die Daten an die Transportschicht weiter, welche diese dem empfangenen Prozess übergibt. Wenn alle Teile schließlich das Ziel erreicht haben, werden sie von der Vermittlungsschicht wieder zum ursprünglichen Datagramm zusammengesetzt. Dieses Datagramm wird dann an die Transportschicht übergeben.

Im Beispiel von Abbildung 5.45 muss ein Paket, das seinen Ursprung bei einem Host im Heimnetz hat, vier Netze und eine große Anzahl von IP-Routern passieren, bevor es überhaupt das Firmennetz erreicht, in dem sich der Ziel-Host befindet. Dies ist in der Praxis nicht unüblich und es gibt viele noch längere Pfade. Außerdem gibt es viel redundante Konnektivität im Internet, da sich Backbones und ISPs an mehreren Stellen miteinander verbinden. Dies bedeutet, dass es viele mögliche Pfade zwischen zwei Hosts gibt. Es ist die Aufgabe des IP-Routing-Protokolls zu entscheiden, welche Pfade benutzt werden sollen.

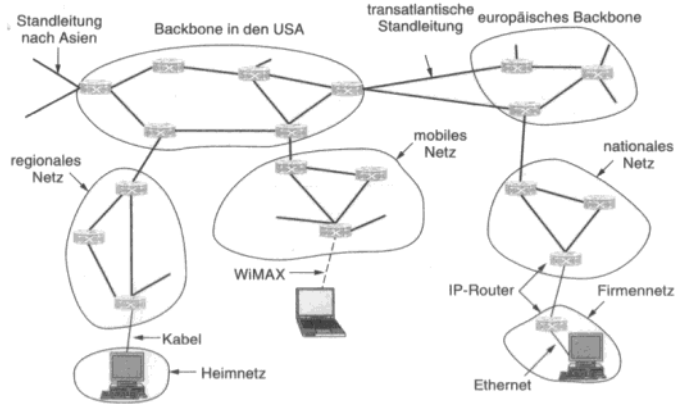


Abbildung 5.45: Das Internet besteht aus vielen miteinander verbundenen Netzen.

IPv4

Das Format der IP-Datagramme ist ein guter Ausgangspunkt für unsere Erforschung der Vermittlungsschicht des Internets. Ein IPv4-Datagramm besteht aus einem Header- und einem Body- oder Nutzdatenteil. Der Header besteht aus einem festen, 20 Byte großen Abschnitt und einem optionalen Abschnitt mit variabler Länge. Das Header-Format ist in »Abbildung 5.46 dargestellt. Die Bits werden von links nach rechts und von oben nach unten übertragen, beginnend mit dem höchstwertigen Bit des Version-Feldes. (Dies ist eine Bytereihenfolge aus „Big-Endian"-Netzen. Auf Little-Endian-Rechnern, wie den Intel-x86-Computern, ist eine Softwarekonvertierung bei der Übertragung und beim Empfang erforderlich.) Rückblickend wäre Little-Endian die bessere Wahl gewesen, aber zu der Zeit, als IP entworfen wurde, wusste niemand, dass es einmal die Computerwelt beherrschen würde.

Das *Version*-Feld verfolgt die Version des Protokolls, zu dem das Datagramm gehört. Version 4 dominiert heute das Internet, und damit haben wir unsere Betrachtung begonnen. Durch Einbindung der Version am Anfang jedes Datagramms ist es möglich geworden, dass sich ein Übergang zwischen Versionen über eine lange Zeitspanne hinzieht. In der Tat wurde IPv6, die nächste IP-Version, vor mehr als einem Jahrzehnt definiert und dennoch beginnt die Installation von IPv6 gerade erst. Wir werden später in diesem Abschnitt darauf zurückkommen. Die Verwendung von IPv6 wird letztendlich erzwungen werden, wenn jeder der fast 231 Einwohner Chinas einen Desktop-PC, einen Laptop und ein IP-Telefon hat. Nebenbei zur fortlaufenden Nummerierung: IPv5 war ein experimentelles Echtzeitprotokoll, das nie große Verbreitung fand.

Da die Header-Länge nicht konstant ist, wird das Header-Feld *IHL* bereitgestellt. In diesem Feld wird die Header-Länge in 32-Bit-Wörtern angegeben. Der Mindestwert ist 5, was zutrifft, wenn keine Optionen vorhanden sind. Der Höchstwert dieses 4-Bit-Feldes ist 15, was den Header auf 60 Byte und damit das *Options*-Feld auf 40 Byte begrenzt. Für manche Optionen, z.B. eine, die den Weg eines Pakets aufzeichnet, ist das viel zu klein, sodass die Option unbrauchbar ist.

Das Feld *Differentiated Services* ist eines der wenigen Felder, die ihre Bedeutung im Laufe der Jahre (leicht) verändert haben. Ursprünglich hieß das Feld *Type of Service*. Es wurde und wird zur Unterscheidung diverser Dienstklassen verwendet. Hier sind verschiedene Kombinationen aus Zuverlässigkeit und Geschwindigkeit möglich. Für digitalisierte Sprache ist Schnelligkeit wichtiger als Genauigkeit. Für Dateiübertragung ist die fehlerfreie Übertragung wichtiger als die Schnelligkeit. Das *Type of Service*-Feld stellt 3 Bits für die Anzeige der Priorität und 3 Bits, um mitzuteilen, ob einem Host eher Verzögerung, Durchsatz oder Zuverlässigkeit wichtig ist. Da jedoch niemand wusste, was Router mit diesen Bits anfangen sollten, blieben sie viele Jahre ungenutzt. Als differenzierte Dienste eingeführt wurden, warf die IETF das Handtuch und belegte dieses Feld neu. Nun werden die ersten 6 Bits benutzt,

um das Paket mit seiner Dienstklasse zu markieren; wir haben die Dienstklassen Expedited Forwarding und Assured Forwarding bereits früher in diesem Kapitel beschrieben. Die letzten 2 Bits werden zur Übertragung von ECN-Informationen verwendet, zum Beispiel, ob das Paket Überlastung erfahren hat; wir haben ECN als Teil der Überlastungsüberwachung weiter vorne in diesem Kapitel beschrieben.

Das Feld *Total Length* beinhaltet das komplette Datagramm — Header und Daten. Die Höchstlänge ist 65 535 Byte. Derzeit ist diese Obergrenze akzeptabel, aber in künftigen Netzen können größere Datagramme nötig sein.

Das Feld *Identification* ist erforderlich, damit der Ziel-Host feststellen kann, zu welchem Paket ein neu angekommenes Fragment gehört. Alle Fragmente eines Pakets enthalten den gleichen Wert in *Identification*.

Danach folgt ein nicht verwendetes Bit, was erstaunlich ist, da verfügbare Fläche im IP-Header extrem rar gesät ist. Bellovin (2003) hat — als Aprilscherz — vorgeschlagen, dieses Bit zu benutzen, um bösartigen Verkehr aufzudecken. Dies würde die Sicherheit sehr vereinfachen, da man wüsste, dass Pakete, bei denen das „Böse"-Bit gesetzt ist, von Angreifern gesendet wurden und somit einfach verworfen werden könnten. Leider ist Netzwerksicherheit doch nicht ganz so einfach.

Als Nächstes kommen zwei 1-Bit-Felder, die sich auf die Fragmentierung beziehen. *DF* steht für „Don't Fragment" (nicht fragmentieren). Das ist ein Befehl an die Router, dass ein Paket nicht fragmentiert werden darf. Ursprünglich war es dazu gedacht, Hosts zu unterstützen, die nicht in der Lage sind, die einzelnen Stücke wieder zusammenzusetzen. Jetzt wird es innerhalb des Prozesses verwendet, mit dem die Path MTU herausgefunden wird — die Größe, die angibt, wie groß ein Paket sein darf, das ohne Fragmentierung einen Pfad entlang reisen kann. Durch Kennzeichnung des Datagramms mit dem *DF*-Bit weiß der Sender, dass das Paket entweder in einem Stück ankommt oder dass eine Fehlermeldung an den Sender zurückgeschickt wird.

MF bedeutet „More Fragments" (mehr Fragmente). Dieses Bit ist bei allen Fragmenten außer dem letzten gesetzt. Es wird benötigt, um feststellen zu können, wann alle Fragmente eines Datagramms angekommen sind.

Fragment Offset gibt an, an welche Stelle im aktuellen Paket ein Fragment gehört. Alle Fragmente außer dem letzten müssen in einem Datagramm ein Vielfaches von 8 Byte sein. Das ist die elementare Fragmenteinheit. Da 13 Bit verfügbar sind, sind maximal 8 192 Fragmente pro Datagramm möglich, wodurch eine maximale Paketlänge bis zur Grenze des Felds *Total Length* unterstützt wird. Die Felder *Identification*, *MF* und *Fragment Offset* werden zusammen verwendet, um Fragmentierung wie in Abschnitt 5.5.5 beschrieben zu implementieren.

Das Feld *TTL* (*Time to Live*) ist ein Zähler, mit dem die Lebensdauer von Paketen begrenzt werden kann. Ursprünglich sollte das Feld die Zeit in Sekunden erfassen. Zulässig ist eine maximale Lebensdauer von 255 Sekunden. Der Zähler muss bei jeder Teilstrecke um 1 reduziert werden und bei einer längeren Zwischenspeicherung in einem Router mehrmals. In der Praxis werden nur die Teilstrecken gezählt. Erreicht der Zähler null, wird das Paket verworfen und ein Warnpaket an den Quell-Host gesendet. Durch dieses Merkmal werden Pakete daran gehindert, ewig herumzuschwirren, was manchmal passiert, wenn die Routing-Tabellen beschädigt wurden.

Hat die Vermittlungsschicht ein komplettes Paket zusammengestellt, muss sie wissen, was sie damit anfangen soll. Durch das Feld *Protocol* kann sie erkennen, an welchen Transportprozess das Paket weiterzugeben ist. TCP ist eine Möglichkeit, UDP und andere sind weitere. Die Nummerierung der Protokolle ist im ganzen Internet einheitlich. Protokolle und andere zugewiesene Nummern wurden früher im RFC 1700 aufgeführt, sind aber heutzutage in einer Onlinedatenbank unter *www.iana.org* zu finden.

Da der Header wesentliche Informationen wie Adressen beinhaltet, legt er seine eigene Prüfsumme zum Schutz an, das Feld *Header Checksum*. Der Algorithmus addiert alle 16-Bit-Halbwörter des Headers bei ihrer Ankunft in der Einerkomplement-Arithmetik und nimmt dann das Einerkomplement des Ergebnisses. Zum Zweck dieses Algorithmus wird angenommen, dass *Header Checksum* initial null ist. Eine solche Prüfsumme ist nützlich zum Erkennen von Fehlern, während das Paket durch das Netz reist. Beachten Sie, dass es bei jeder Teilstrecke neu berechnet werden muss, weil sich mindestens ein Feld immer ändert (das TTL-Feld). Es sind aber verschiedene Tricks möglich, um die Berechnung zu beschleunigen.

Die Felder *Source Address* und *Destination Address* bezeichnen die IP-Adresse der Netzschnittstellen von Quelle und Ziel. Internetadressen werden im nächsten Abschnitt behandelt.

Das Feld *Options* wurde vorgesehen, um es späteren Versionen des Protokolls zu ermöglichen, Informationen aufzunehmen, die im ursprünglichen Entwurf nicht vorhanden sind. Damit können Experimentierfreudige neue Ideen ausprobieren und es wird gleichzeitig vermieden, dass Header-Bits für Informationen genutzt werden, die selten benötigt werden. Das Feld hat eine variable Länge. Jede Option beginnt mit einem 1-Byte-Code, der die Option identifiziert. Bei einigen Optionen folgt darauf ein 1-Byte-Feld mit der Optionslänge und dann eines oder mehrere Datenbytes. Das Feld Options wird auf ein Vielfaches von 4 Byte aufgefüllt. Ursprünglich waren fünf Optionen definiert, [...].

Die Option *Security* (Sicherheit) bestimmt, wie geheim die Informationen sind. Theoretisch kann ein Router in einer militärischen Anwendung diese Option benutzen, um festzulegen, dass die Übertragung der Pakete nicht durch bestimmte Länder fließen soll. In der Praxis

wird es von allen Routern ignoriert; deshalb liegt seine einzige praktische Verwendung darin, dass Spione schneller bestimmte Leckerbissen finden können.

Die Option *Strict Source Routing* (exaktes Source-Routing) gibt den vollständigen Pfad von der Quelle zum Ziel als Folge von IP-Adressen an. Das Datagramm muss diese Route genau einhalten. Sie ist besonders für Systemmanager nützlich, die Notpakete aussenden müssen, wenn Routing-Tabellen beschädigt wurden, oder für bestimmte Zeitmessungen.

Die Option *Loose Source Routing* (lockeres Source-Routing) verlangt, dass das Paket die spezifizierten Router in der angegebenen Reihenfolge durchläuft, es kann aber auf dem Weg auch andere Router durchqueren. Normalerweise stellt diese Option nur ein paar Router zur Verfügung, um einen bestimmten Pfad zu erzwingen. Wenn beispielsweise durchgesetzt werden soll, dass ein Paket von London nach Sydney in westlicher und nicht in östlicher Richtung übertragen werden soll, können mit dieser Option Router in New York, Los Angeles und Honolulu angegeben werden. Diese Option ist am nützlichsten, wenn Datenströme aus politischen oder wirtschaftlichen Gründen bestimmte Länder durchlaufen oder aber meiden sollen.

Die *Option Record Route* (Wegaufzeichnung) weist jeden Router auf dem durchquerten Weg an, seine IP-Adresse an das Feld *Options* anzuhängen. Dadurch können Systemmanager Fehler in den Routing-Algorithmen verfolgen („Warum besuchen Pakete von Houston nach Dallas zuerst Tokio?"). Als das ARPANET erstmals eingerichtet wurde, überquerte nie ein Paket mehr als neun Router, sodass die 40-Byte-Option reichlich bemessen war. Heute ist das zu wenig.

Die *Option Time Stamp* (Zeitstempel) schließlich ist mit der Option *Record Route* vergleichbar, außer dass hier jeder Router zusätzlich zur 32-Bit-IP-Adresse auch einen 32-Bit-Zeitstempel erfasst. Auch diese Option wird vorwiegend für Messungen verwendet.

Heutzutage sind IP-Optionen in Ungnade gefallen. Viele Router ignorieren sie oder verarbeiten sie nicht effizient, schieben sie als unüblichen Fall beiseite. Das heißt, Optionen werden nur teilweise unterstützt und selten benutzt.

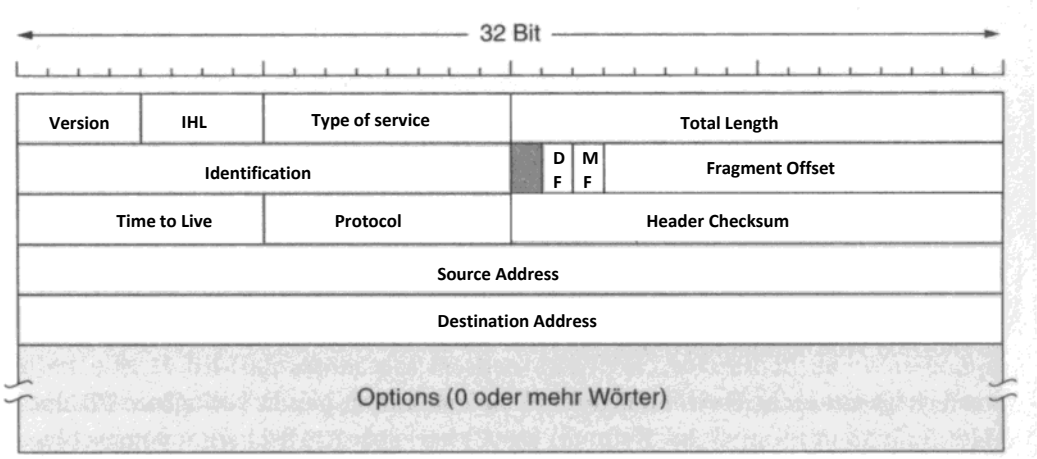


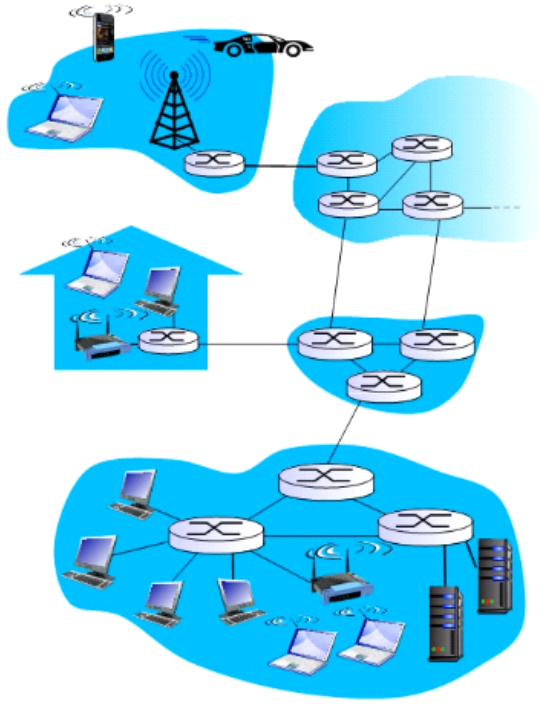
Abbildung 5.46: Der IPv4-Header.

Vermittlungsschicht

Montag, 25. Januar 2021 12:39

Vermittlungsschicht

- Segment der Transport-schicht von Host zu Host
- Sender kapselt Segment in IP-Paket
- Empfänger übergibt Segment an Transportschicht
- Protokolle der Vermittlungsschicht laufen in jedem Host & Router
- Router lesen Steuerdaten im Header der IP-Pakete



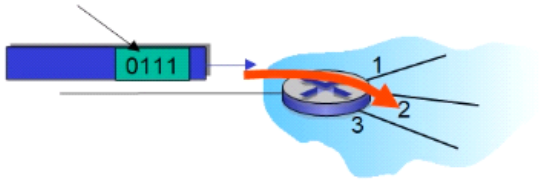
Network Layer: Data Plane 4-2

Vermittlungsschicht

2 Funktionen:

- *Weiterleitung* eines Packets vom Router-Eingang zum richtigen Router-Ausgang

  ➔Weiterleitungstabelle



- *Routing:* ermitteln der Route von Quell-Host zu Ziel-Host

  ➔*Routing Algorithmen*

Dienstmodelle der Vermittlungsschicht

Network Architecture	Service Model	Guarantees ?				Congestion feedback
		Bandwidth	Loss	Order	Timing	
Internet (IP)	best effort	none	no	no	no	no (inferred via loss)

schlechte Nachricht:

Das Internet (IP) tut sein Bestes ("best-effort"), aber es gibt keinerlei Garantien für eine schnelle und/oder vollständige Zustellung.

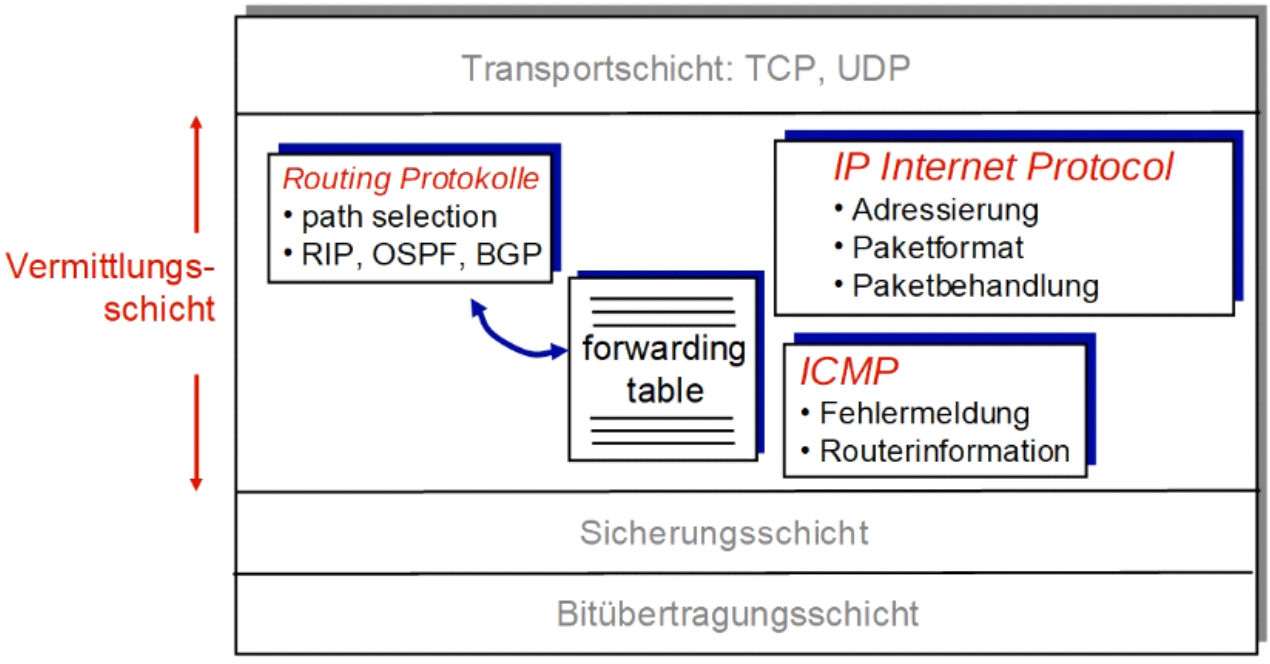
es gibt auch andere:

Asynchronous Transfer Mode (ATM)

Asynchronous Transfer Mode (ATM)

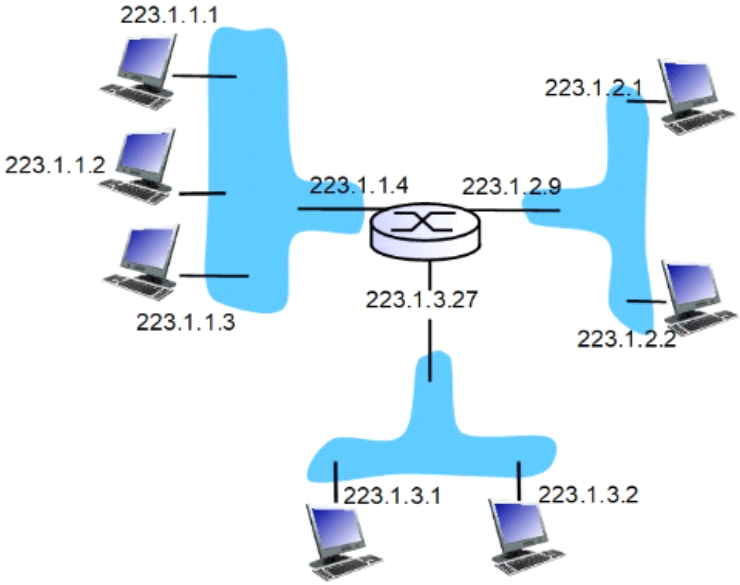
Lesetipp: https://www.stefan-marr.de/pages/atm-traffic-management/

Vermittlungsschicht im Schichtenmodell



IP Adressen

- 32-Bit (IPv4) oder 128-Bit (IPv6) Kennnummer für jede Host- und Router-Schnittstellen
- *Schnittstelle:* Verbindung zwischen Host/Router und dem Übertragungsmedium (physical link)
  - Router haben mehrere Schnittstellen
  - Host haben meistens eine oder zwei Schnittstellen (z.B., Ethernet 802.3, WLAN 802.11)



223.1.1.1 = 11011111 00000001 00000001 00000001

223 1 1 1

IP Adressen: Das letzte Wort...

Q: Wer vergibt die IP-Adressen an die ISPs?

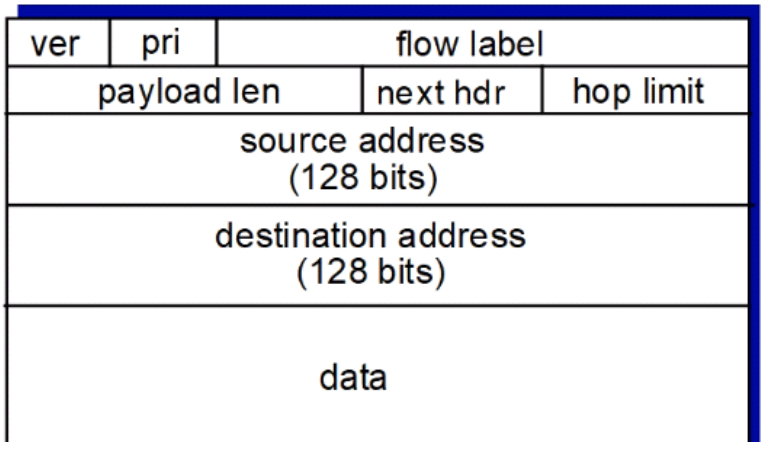
A: ICANN: Internet Corporation for Assigned Names and Numbers http://www.icann.org/

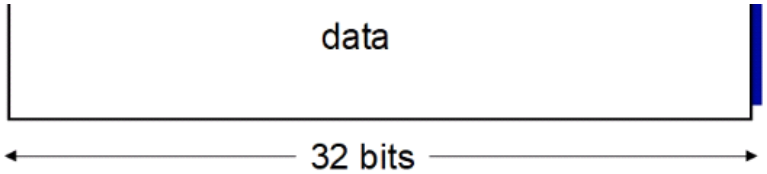
- Zuordnung von Adressen
- Verwaltung des DNS
- Vergabe von Domainamen
- Beilegung von Streitigkeiten

IPv6: Motivation

- *Initialmotivation (90er?)*
  32-Bit Adressen bald verbraucht
- zusätzliche Motivation
  Headerformat von IPv4 kompliziert und unpraktisch
- ... ??? ...

IPv6 Paketformat





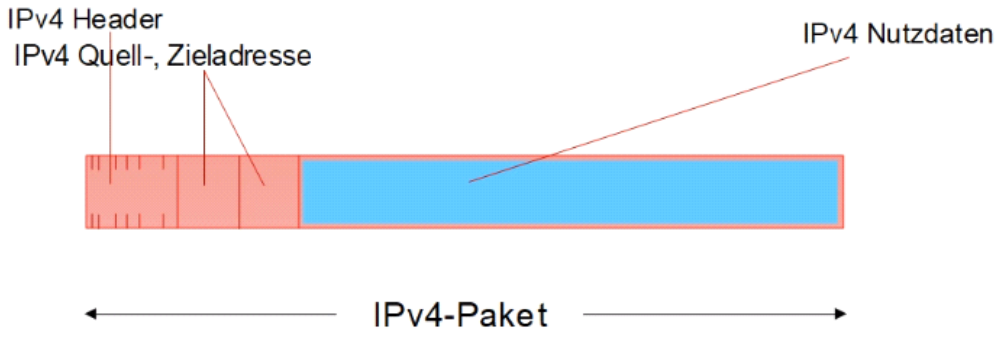
IPv6: Unterschiede zu IPv4

- Adressen: 16 Byte
- fixe Länge des Headers: 40 Byte
- aber: Erweiterungsheader mit Optionen (→ next header)
- keine Fragmentierung, *jumbograms*
- keine Prüfsumme
- automatisierte Konfiguration und Adressvergabe
- …

Migration von IPv4 zu IPv6

- *dual-stack:* ???
  Arbeitsauftrag
- *tunneling:* IPv6-Paket als Nutzdaten in einem IPv4-Paket – IPv4 Router bekommen nichts mit (geht auch umgekehrt)

(geht auch umgekehrt)



Network Layer: Data Plane 4-11

IPv6 Adressen Abkürzen

Abkürzen von IPv6-Adressen

Die meisten IPv6-Adressen belegen nicht alle verfügbaren 128 Bit. Dies führt zu Feldern, die entweder mit Nullen aufgefüllt werden oder nur Nullen enthalten.
Die IPv6-Adressierungsarchitektur ermöglicht Ihnen eine Notation mit zwei Doppelpunkten (: :), um zusammenhängende 16-Bit-Felder mit Nullen darzustellen. So können Sie die IPv6-Adresse aus Abbildung 3–2 beispielsweise schreiben, indem Sie die zwei zusammenhängenden Felder mit Nullen in der Schnittstellen-ID durch zwei Doppelpunkte ersetzen. Die resultierende Adresse lautet dann **2001:0db8:3c4d:0015::1a2f:1a2b**. Andere aus Null bestehende Felder können als einzelne 0 dargestellt werden. Sie können führende Nullen in einem Feld weglassen, d. h. **0db8** kann beispielsweise als **db8** geschrieben werden.
Die Adresse **2001:0db8:3c4d:0015:0000:0000:1a2f:1a2b** kann also zu **2001:db8:3c4d:15::1a2f:1a2b** verkürzt werden.
Sie können die Notation mit zwei Doppelpunkten verwenden, um alle zusammenhängenden Felder mit Nullen in der IPv6-Adresse zu ersetzen. So kann die IPv6-Adresse **2001:0db8:3c4d:0015:0000:d234::3eee:0000** zu **2001:db8:3c4d:15:0:d234:3eee**:: verkürzt werden.

Aus <https://docs.oracle.com/cd/E19957-01/820-2980/ipv6-overview-24/index.html>