Kapitel 8: Layer 4 / TCP und UDP

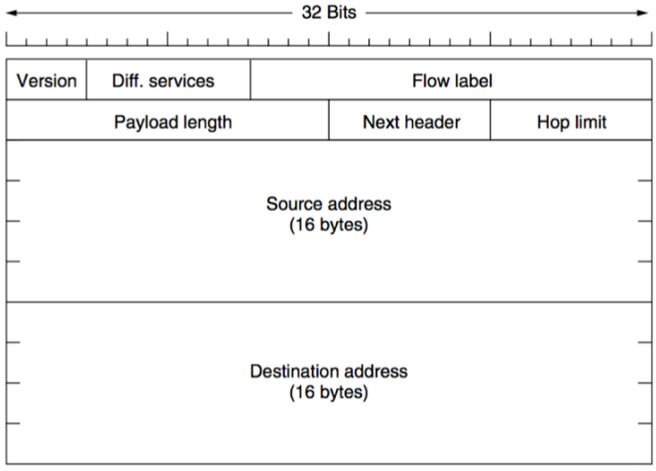
Internet Protocol v6 (IPv6)

Wozu IPv6? Die Hauptgründe sind:

- Die Zahl der Geräte (im Internet), übersteigt die Zahl der vorhandenen IPv4-Adressen.
- *Private IPv4-Adressen und Network Address Translation erlauben zwar die Nutzung von anderen Rechnern/Diensten im Internet, machen aber die direkte Kommunikation zwischen zwei Rechnern mit privater IP-Adresse schwierig. Sicherheitsaspekte wurden bei der Entwicklung von IPv4 nicht in dem Maße bedacht, wie es heute gefordert wäre.*

Ziele von IPv6

- Vergrößerung des Adressraumes auf 128 Bit *($2^{128} \approx 3,4 \cdot 10^{38}$)*
- Verbesserung der Header-Struktur(Feste grösse).
  - Erweiterungs-Header sind allerdings möglich.
- Neue Adresstypen wie z. B. link local-Adressen.
- Unterstützung von Autokonfiguration.
- Verbesserung der Sicherheit.



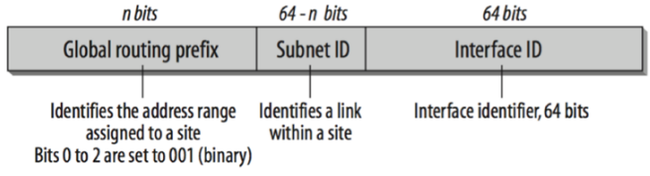
Die Felder im IPv6 Header

- **Version**: Hier steht die Nummer, also eine 6.
- **Diff. Services**: Wirf für Quality of Service benötigt.
- **Flow Label**: Markierung von „zusammengehörigen" Paketen, die gleich behandelt werden sollten.
- **Payload Length**: Größe der Nutzlast, sie kann maximal 64 KB betragen.
- **Next Header**: Beschreibt den Inhalt hinter dem Kopf. (Das kann ein TCP-Paket sein (kommen wir später zu) oder ein weiterer IPv6 Header, genannt Extension Header.)
- **Hop Limit**: Hiess früher bei IPv4 TTL. Verhindert Endlosschleifen beim Routing.
- **Source / Destination Address**: Absender und Empfänger IPv6-Adressen.

Schreibweisen von IPv6-Adressen

Bilde 8 Blöcke mit jeweils 16 Bit. Trenne Blöcke durch **:**
Schreibe jeden Block als Folge von 4 Hexadezimalzahlen. *Beispiel: ADCF:0005:0000:0000:0000:0000:0600:FEDC*
Erste Vereinfachung: Lasse führende 0 im Block weg. *Beispiel: ADCF:5:0:0:0:0:600:FEDC*
Zweite Vereinfachung: Ersetze Folge von 0-Blöcken durch **::** *Beispiel: ADCF:5::600:FEDC*

Netzanteil und Host-Anteil in IPv6



- Eine (globale) IPv6-Adresse besteht aus einem 64 Bit Netzanteil (Präfix) und 64 Bit Host-Anteil (interface ID).
- Notation: IPv6Adresse „/" Präfix-Länge Beispiel: 2000::/16
- **Der Netzanteil kann variabel aufgeteilt werden.**
- **Der Host-Anteil beträgt immer 64 Bit.**
- Die HsH bekommt eine /48 Adresse. Sie kann also $2^{64-48} = 2^{16} \approx 65$ Tausend eigene Subnetze bilden. (Die Abteilung Informatik hat eine /56 Adresse bekommen; d. h. 256 Subnetze.)

Pv6 Adresstypen

- **Unicast**: Identifiziert ein Netzinterface eines Knoten eindeutig. Pakete an diese Adressen werden nur diesem einen Knoten zugestellt.
- **Multicast**: Identifiziert eine Gruppe von IPv6-Netzinterfaces. Pakete an diese Adressen werden von allen Mitgliedern der Gruppe empfangen und bearbeitet.
- **Anycast**: Verschiedenen IPv6-Netzinterfaces können dieselbe Anycast-Adresse besitzen (typischerweise auch auf verschiedenen Rechnern). Pakete an diese Adresse werden einem der Interfaces zugestellt.

Allgemeine Regeln:

- IPv6-Adresse haben verschiedenen scope (Gültigkeitsbereich)
  - Global: Weltweit eindeutige Adresse.
  - Link Local: Innerhalb eines (lokalen) Netzes eindeutige Adresse.

Alle Kombinationen von Adresstyp und Gültigkeitsbereich sind möglich.

Wichtige Präfixe

- Link Local Adressen sind nur innerhalb eines LANs gültig!
- Link Local Adressen werden nicht „geroutet“ und daher auch nicht in der Routing-Tabelle enthalten.

Präfix Hex	Präfix Binär	Bedeutung
2000::/3	001	Global Unicast Address
FE80::/10	1111 1110 10	Link Local Unicast
FF00::/8	1111 1111	Multicast
FF02::/16	1111 1111 0000 0010	Link Local Multicast

Einige besondere IPv6-Adressen:

Adresse	Bedeutung
0:0. . . :0	Nicht spezifizierte Adresse
::1	loopback Address

Unterschied zwischen Multicast und Broadcast

Broadcast:

- Spezielle Adresse für alle Empfänger.
- Jede Station bekommt Broadcast zugestellt.
- Adresse besteht typischerweise aus lauter Einsen.
- *Beispiel Ethernet: FF-FF-FF-FF-FF-FF*
- *Beispiel IPv4: DHCP-Anfragen gehen an 255.255.255.255*

Multicast:

- Spezielle Adresse für eine Empfängergruppe.
- Stationen müssen sich in die Gruppe eintragen.
- Es kann mehrere Multicast-Gruppen geben.
- *Beispiel Ethernet: 33-33-xx-xx-xx-xx*
- *Beispiel IPv6: Solicited Node Multicast; All Routers; . . .*

Transmission Control Protocol (TCP)

Motivation der Transportschicht

Funktionen der Vermittlungsschicht:

- Vermittlungsschicht (IP) realisiert einzig und allein Internetworking, d. h. sie transportiert Pakete von einem Rechner zu einem beliebigen anderen Rechner

Noch offene Probleme:

- Problem 1: Pakete können verloren gehen, verfälscht werden, sich gegenseitig überholen, usw.
- Problem 2: An jedem Router unterwegs kann Überlast auftreten. Abender sollte dann langsamer senden.
- Problem 3: Vermittlungsschicht lässt es offen, welcher Prozess beim Empfänger ein Datenpaket verarbeiten soll.

Fokus der Transportschicht:

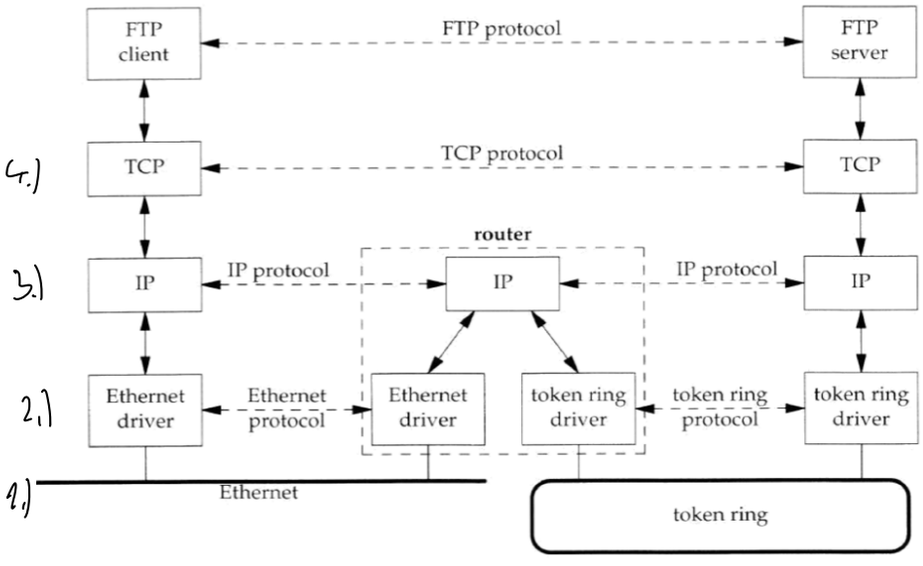
- Datentransport von einem Absender-Prozess auf Quellrechner zu einem Empfänger-Prozess auf Zielrechner.
- Bereitstellung einer Programmierschnittstelle für Anwendungen

Kernaufgaben:

- Bereitstellung einer Kommunikationsschnittstelle für einzelne Anwendungen durch Erweiterung des Adressierungsschemas um Ports.
- Gegebenenfalls Sicherstellung der Zuverlässigkeit und Reihenfolge der Daten, d. h. einen zuverlässigen Transportdienst über ein unzuverlässiges Netz anbieten.
- Anbieten von Flusssteuerung, Überlastungsüberwachung, . . .

Layer 4: Erste Ende-zu-Ende Schicht

- TCP muss nur auf den Endpunkten implementiert sein.

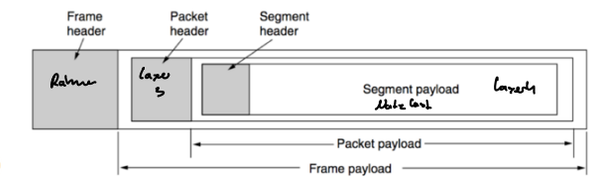


Transmission Control Protocol (TCP)

- TCP arbeitet verbindungsorientiert, d. h. am Anfang wird eine Verbindung aufgebaut, dann werden Daten übertragen, am Ende wird die Verbindung abgebaut.
- TCP bietet zuverlässige Kommunikation, d. h.
  - (1) es erkennt verloren gegangene Pakete anhand von Bestätigungsnummern (engl. acknowledgement number) und überträgt diese erneut und
  - (2) es erkennt sich überholende Pakete mit Hilfe von Sequenznummern (engl. sequence number) und bringt diese in die richtige Reihenfolge.
- TCP identifiziert Prozesse durch Portnummern.
  - Portnummern sind 16 Bit Binärzahlen (0 bis 65 535)
  - Nummern bis 1024 sind well known ports, die für spezielle Dienste vorgesehen sind.

Segmente und verschachtelte Übertragung

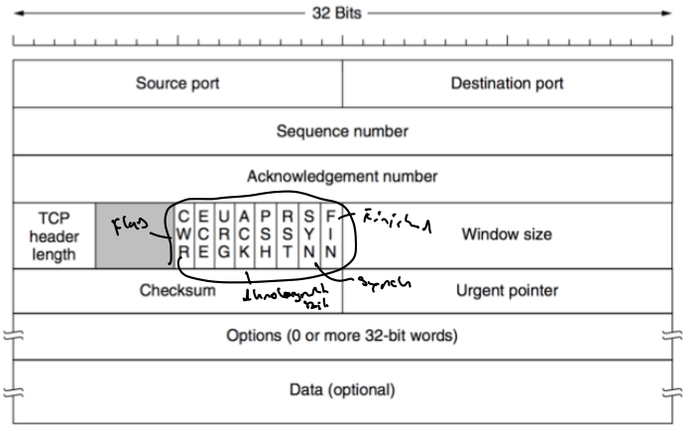
- Auf der Transportschicht sprechen wir von Segmenten, (engl. segment), die übertragen werden.
- Hinweis: Oft wird auch auf dieser Schicht von Paketen gesprochen.
- TCP-Segmente (TCP-Pakete) haben eine Nutzlast (engl. payload) und werden als Nutzlast in IP-Pakete gepackt, welche ihrerseits die Nutzlast in einem Datenrahgmen sind



b

Der TCP-Kopf (engl. TCP header)

- **Source port**: Quell-Portnummer, identifiziert den Absender-Prozess.
- **Destination port:** Ziel-Portnummer, identifiziert den Empfänger-Prozess.
- **Sequence number**: Folgenummer, hilft bei der Identifizierung der Position der Daten im Bytestrom.
- **Acknowledgement number**: Bestätigungsnummer, hilft bei der Prüfung, welche Daten schon beim Empfänger angekommen sind.

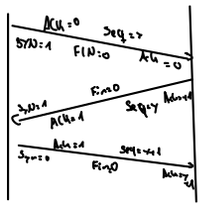


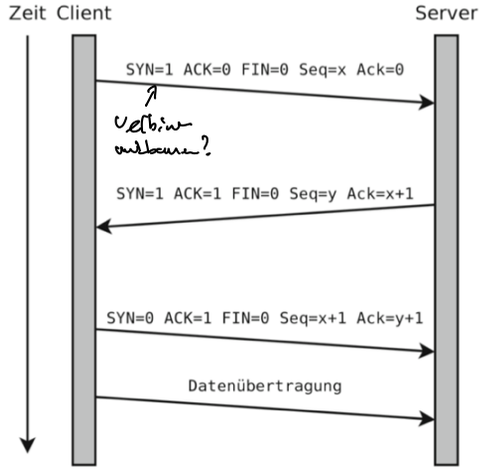
Adressierung bei TCP

- TCP verwendet Ports zur Adressierung der Prozesse
  - Socket: IP-Adresse und Port-Nummer
  - Ein Socket identifiziert einen Prozess.
- Eine TCP-Verbindung wird identifiziert durch ein Socket-Paar, d. h. Socket des Senders und Socket des Empfängers
- ~~Quell-Portnummern: „Frische (kurzlebige)“ Port-Nummern, üblicherweise zwischen 1024 und 5000~~
- Ziel-Portnummern:
  - Benötigen langlebige Port-Nummern!
  - Für Standard-Applikationen ~~sogenannte Well-Known-Ports~~ zwischen 1 und 1023 Beispiele: 25 = SMTP; 80 = HTTP; ~~110~~ = POP3, usw.
  - ~~Sonst „frische“ Portnummern vom Betriebssystem größer 5000~~

TCP-Handshake / Drei-Wege-Handshake / Three-Way-Handshake

1. Client sendet Segment mit gesetztem SYN-Bit und einer zufällig gewählten initialen Sequenz-Nr. (ISN) Seq = x.
2. Server antwortet mit gesetzten SYN- und ACK-Bit, wählt eigene ISN Seq = y und bestätigt im ACK-Feld, was bisher empfangen wurde (x + 1).
3. Client antwortet mit Segment mit gesetztem ACK-Bit und der neuen Seq = x + 1 (es wurde ein Byte übertragen). Client bestätigt erfolgreichen Empfang durch Ack = y + 1





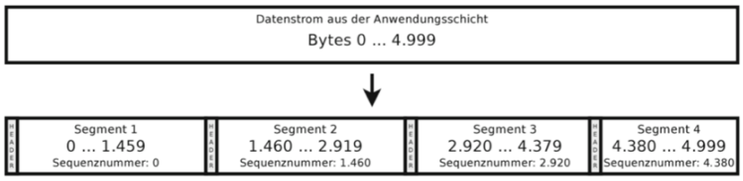
Datenübertragung:

- Absender zählt übertragene Bytes und erhöht die Sequenz-Nr. entsprechend. Überholen sich Segmente, so erkennt der Empfänger das anhand der Sequenz-Nr.
- Anhand der ACK-Nummer in (Antwort-)Segmenten des Empfängers erkennt Absender verloren gegangene Segmente und kann diese erneut abschicken.

Verbindungsabbau:

- Hat Client keine Daten mehr zu übertragen, dann sendet der Client ein Segment mit gesetztem FIN-Bit. Verbindung ist nun „halb geschlossen“.
- Server kann weiter senden. Hat der Server auch nichts mehr zu übertragen, dann sendet dieser auch ein Segment mit gesetztem FIN-Bit.
- Die Segmente mit gesetztem FIN-Bit werden durch ein Segment mit gesetztem ACK quittiert. Der Server kann auch als erster ein FIN senden.

Segmente und Sequenznummern



- Die Transportschicht teilt größere Datenströme in Segmente ein.
- Die Sequenznummern eines Segments ist die Nummer des ersten Bytes.
- In der Praxis wird statt der Bytenummer n immer n + ISN (initial sequence number) benutzt.
- Beim Verbindungsaufbau werden meist leere, bzw. nur 1 Byte große Segmente versendet.
- Segmentgrößen können im Laufe der Übertragung auch variieren.

Eigenschaften von TCP

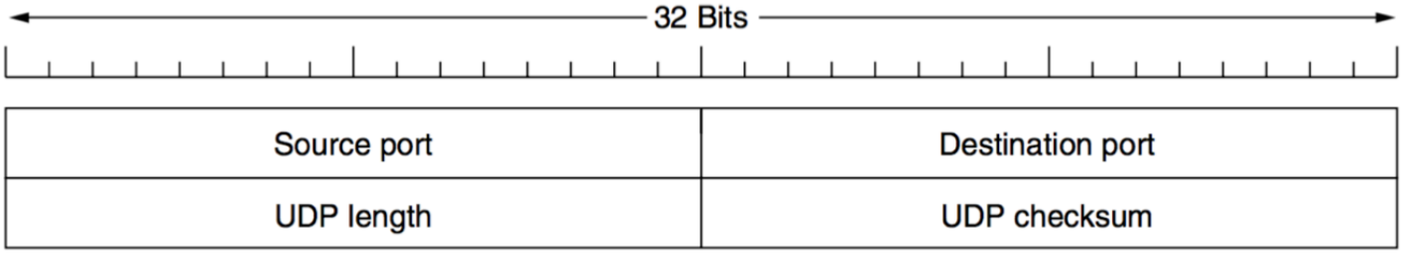
Vorteile:

- Bietet Anwendungen einen zuverlässigen Datenstrom über ein unzuverlässiges Netz.
- Mit den Portnummern können Prozesse auf dem Empfänger-Host angesprochen werden.
- Erhöht Übertragungsge- schwindigkeit bis Pakete verloren gehen.

Nachteile:

- Braucht einen Verbindungsaufbau bevor Daten übertragen werden können.
- Zuordnung von Portnummer zu Prozess obliegt dem Administrator des Empfänger-Hosts.
- Beginnt mit langsamer Datenübertragung (engl. slow start) und steigert sich dann je nach Netzkapazität.

User Datagram Protocol (UDP)



- **Source port**: Quell-Portnummer, identifiziert den Absender-Prozess.
- **Destination port**: Ziel-Portnummer, identifiziert den Empfänger-Prozess.
- **UDP length**: Länge des Segments (Kopf und Nutzlast) in Bytes.
- **UDP checksum**: Optionale Prüfsumme.

Eigenschaften von UDP

Vorteile:

- Sehr einfaches Protokoll
- Ermöglicht Kommunikation zwischen Prozessen
- Kein Verbindungsaufbau erforderlich
- Geeignet für Multimedia-Daten (VoIP, Videostreaming, usw.)

Nachteile:

- Unzuverlässig; verloren gegangene Pakete sind weg.
- Keine Flusskontrolle; zu schnelles senden kann Netze überlasten.

Zusammenfassung

- Da IPv4-Adressen nahezu alle vergeben sind, aber bspw. für Internet of Things (IoT) immer mehr Adressen benötigt werden, wurde IPv6 entwickelt. IPv6-Adressen sind 128 Bit lang und bestehen aus Netzanteil und Interface Identifier.
- IPv6-Adressen haben verschiedene Gültigkeitsbereiche (engl. scope) (global, link local) und verschiedenen Typ (unicast, multicast).
- Mit TCP ist ein zuverlässiges Transportprotokoll verfügbar, das auf unzuverlässigen Netzen arbeitet.
- Mit Portnummern lassen sich spezielle Prozesse auf dem Empfängerrechner adressieren.
- Ein Socket enthält die beiden Endpunkte einer TCP-Verbindung. TCP-Verbindungen werden mit dem Dreiwege-Handshake aufgebaut.
- Mit sequence und acknowledge numbers werden TCP-Segmente in die richtige Reihenfolge gebracht und verloren gegangene Segemente erneut übertragen.