

Systeme II

5. Die Transportschicht

Christian Schindelbauer

Technische Fakultät

Rechnernetze und Telematik

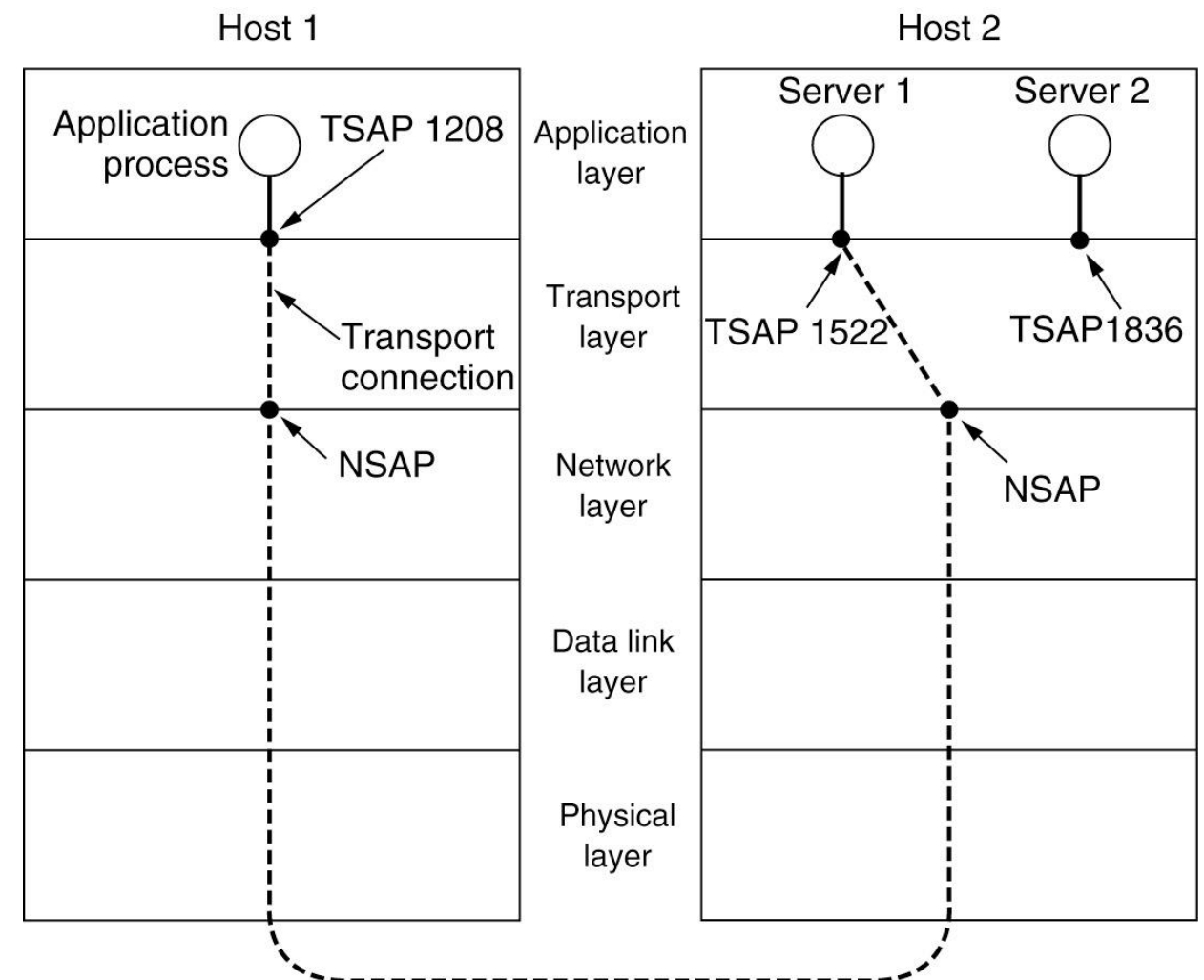
Albert-Ludwigs-Universität Freiburg

Version 26.06.2017

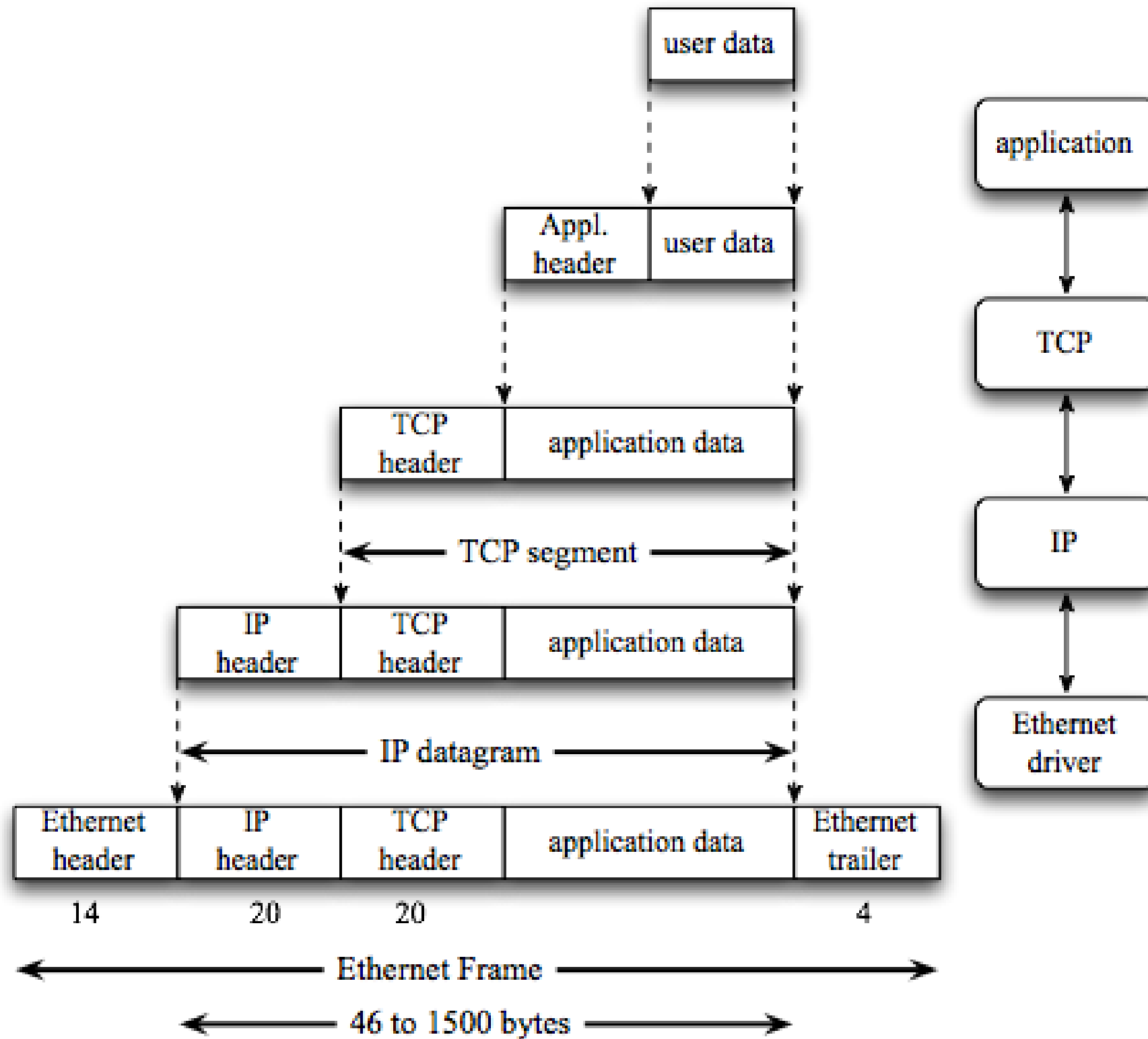
- Verbindungslos oder Verbindungsorientiert
 - Beachte: Sitzungsschicht im ISO/OSI-Protokoll
- Zuverlässig oder unzuverlässig
 - Best effort oder Quality of Service
 - Fehlerkontrolle
- Mit oder ohne Congestion Control
- Möglichkeit verschiedener Punkt-zu-Punktverbindungen
 - Stichwort: Demultiplexen
- Interaktionsmodelle
 - Byte-Strom, Nachrichten, „Remote Procedure Call“

Multiplex in der Transportschicht

- Die Netzwerkschicht leitet Daten an die Transportschicht unkontrolliert weiter
- Die Transportschicht muss sie den verschiedenen Anwendungen zuordnen:
 - z.B. Web, Mail, FTP, ssh, ...
 - In TCP/UDP durch Port-Nummern
 - z.B. Port 80 für Web-Server

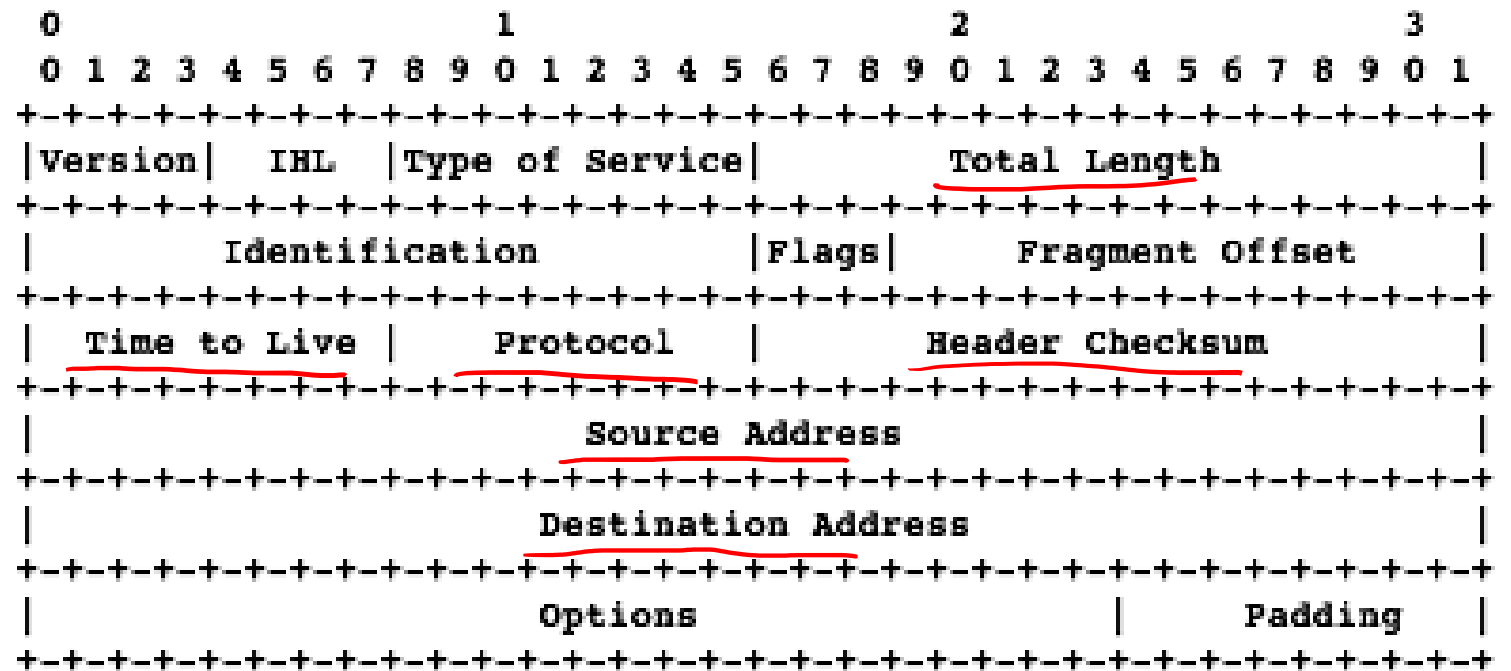


Datenkapselung

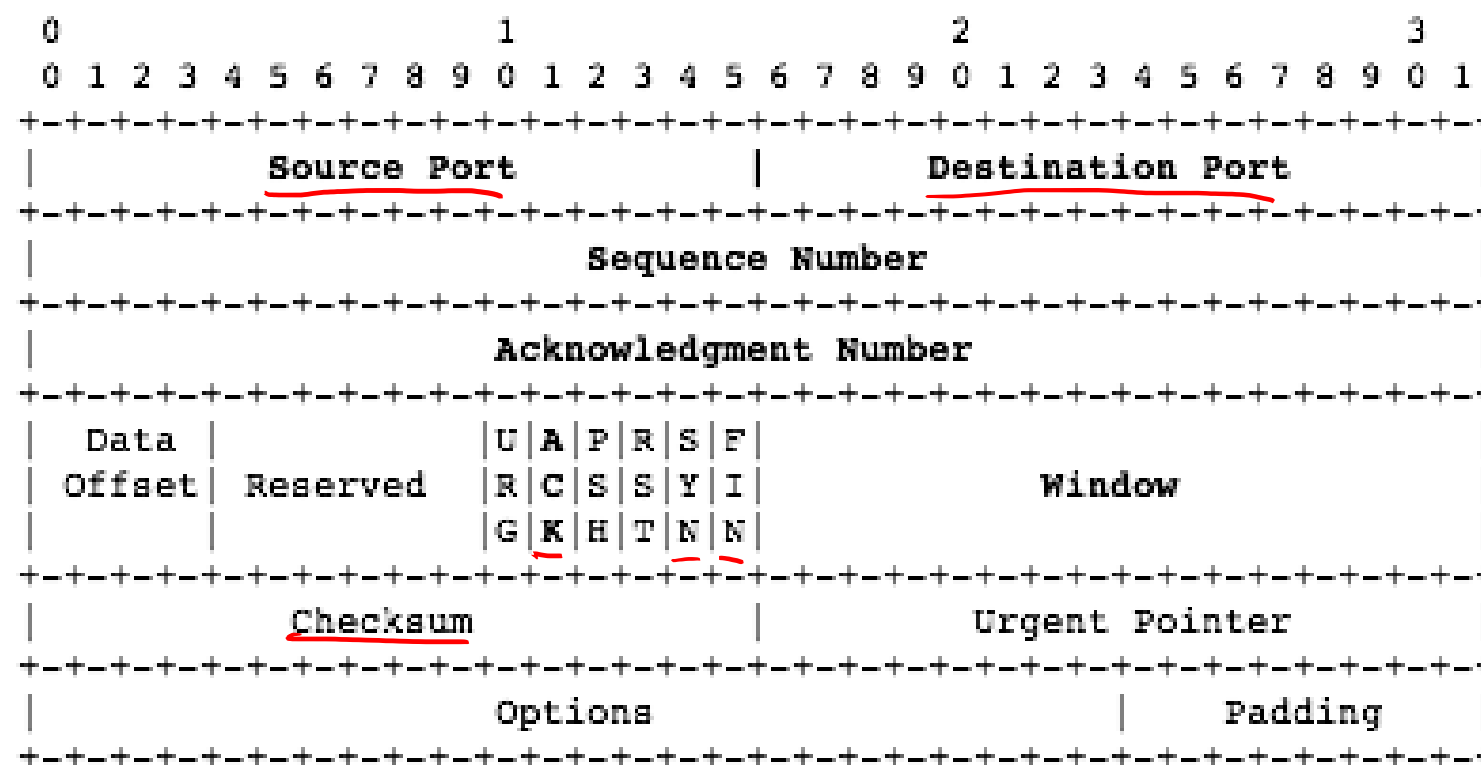


IP-Header (RFC 791)

- Version: 4 = IPv4
- IHL: Headerlänge
 - in 32 Bit-Wörter (>5)
- Type of Service
 - Optimierte delay, throughput, reliability, monetary cost
- Checksum (nur für IP-Header)
- Source and destination IP-address
- Protocol, identifiziert passendes Protokoll
 - Z.B. TCP, UDP, ICMP, IGMP
- Time to Live:
 - maximale Anzahl Hops



- Sequenznummer
 - Nummer des ersten Bytes im Segment
 - Jedes Datenbyte ist nummeriert modulo 2^{32}
- Bestätigungsnummer
 - Aktiviert durch ACK-Flag
 - Nummer des nächsten noch nicht bearbeiteten Datenbytes
 - = letzte Sequenznummer + letzte Datenmenge:
- Port-Adressen
 - Für parallele TCP-Verbindungen
 - Ziel-Port-Nr.
 - Absender-Port
- Headerlänge
 - data offset
- Prüfsumme
 - Für Header und Daten



- TCP (transmission control protocol)
 - Erzeugt zuverlässigen Datenfluß zwischen zwei Rechnern
 - Unterteilt Datenströme aus Anwendungsschicht in Pakete
 - Gegenseite schickt Empfangsbestätigungen (Acknowledgments)
- UDP (user datagram protocol)
 - Einfacher unzuverlässiger Dienst zum Versand von einzelnen Päckchen
 - Wandelt Eingabe in ein Datagramm um
 - Anwendungsschicht bestimmt Paketgröße
- Versand durch Netzwerkschicht
- Kein Routing: End-to-End-Protokolle

- TCP ist ein verbindungsorientierter, zuverlässiger Dienst für bidirektionale Byteströme
- TCP ist verbindungsorientiert
 - Zwei Parteien identifiziert durch Socket: IP-Adresse und Port
(TCP-Verbindung eindeutig identifiziert durch Socketpaar)
 - Kein Broadcast oder Multicast
 - Verbindungsaufbau und Ende notwendig
 - Solange Verbindung nicht (ordentlich) beendet, ist Verbindung noch aktiv

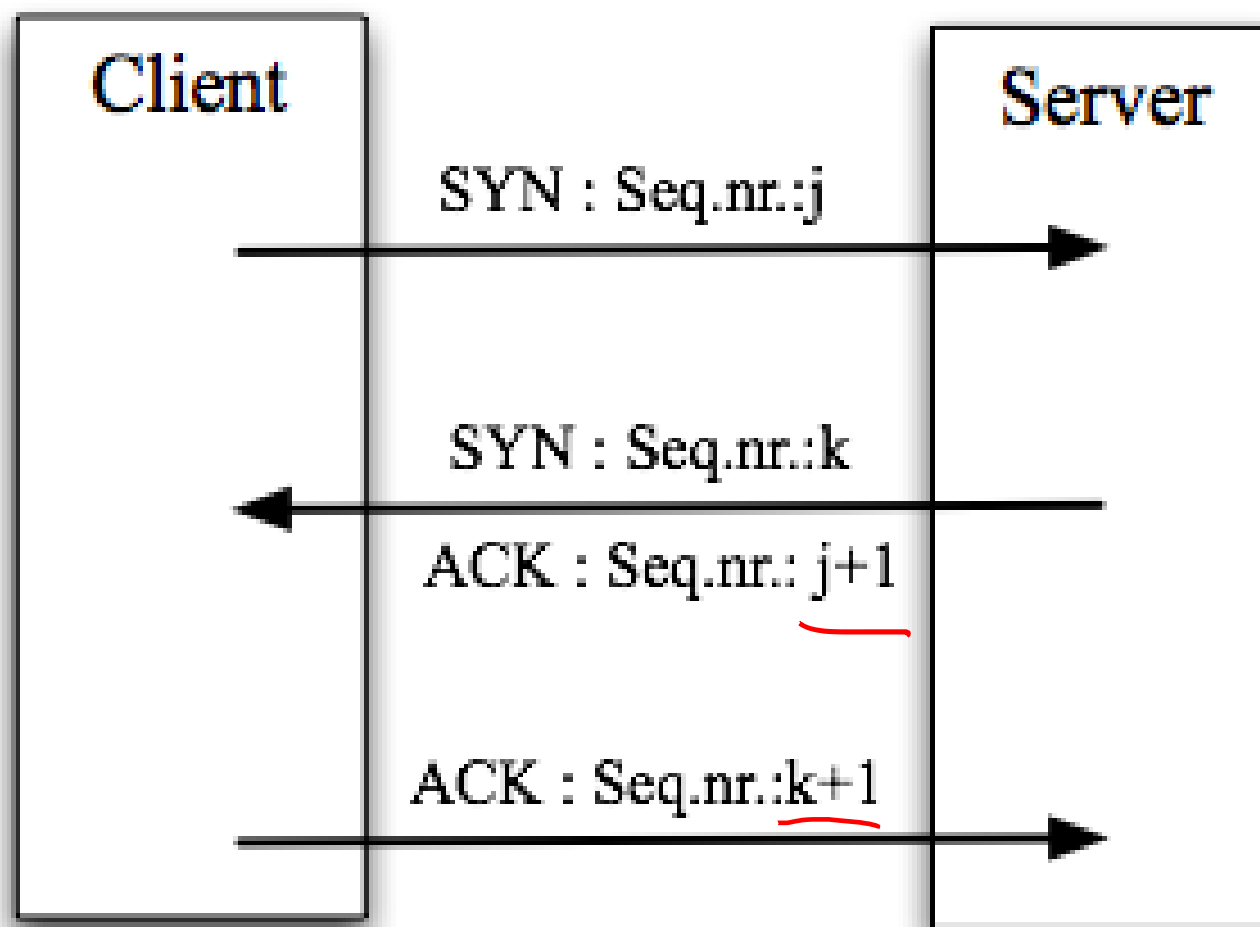
- TCP ist ein verbindungsorientierter, zuverlässiger Dienst für bidirektionale Byteströme

- TCP ist zuverlässig
 - Jedes Datenpaket wird bestätigt (acknowledgment)
 - Erneutes Senden von unbestätigten Datenpakete
 - Checksum für TCP-Header und Daten
 - TCP nummeriert Pakete und sortiert beim Empfänger
 - Löscht duplizierte Pakete

- TCP ist ein verbindungsorientierter, zuverlässiger Dienst für bidirektionale Byteströme
- TCP ist ein Dienst für bidirektionale Byteströme
 - Daten sind zwei gegenläufige Folgen aus einzelnen Bytes (=8 Bits)
 - Inhalt wird nicht interpretiert
 - Zeitverhalten der Datenfolgen kann verändert werden
 - Versucht zeitnahe Auslieferung jedes einzelnen Datenbytes
 - Versucht Übertragungsmedium effizient zu nutzen
 - = wenig Pakete

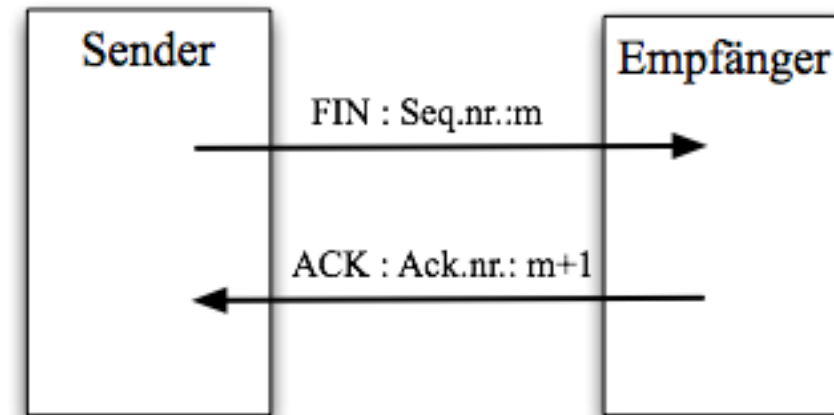
TCP-Verbindungsaufbau

- In der Regel Client-Server-Verbindungen
 - Dann Aufbau mit drei TCP-Pakete (=Segmente)
 - Mit ersten SYN-Segment auch Übermittlung der MSS (maximum segment size)

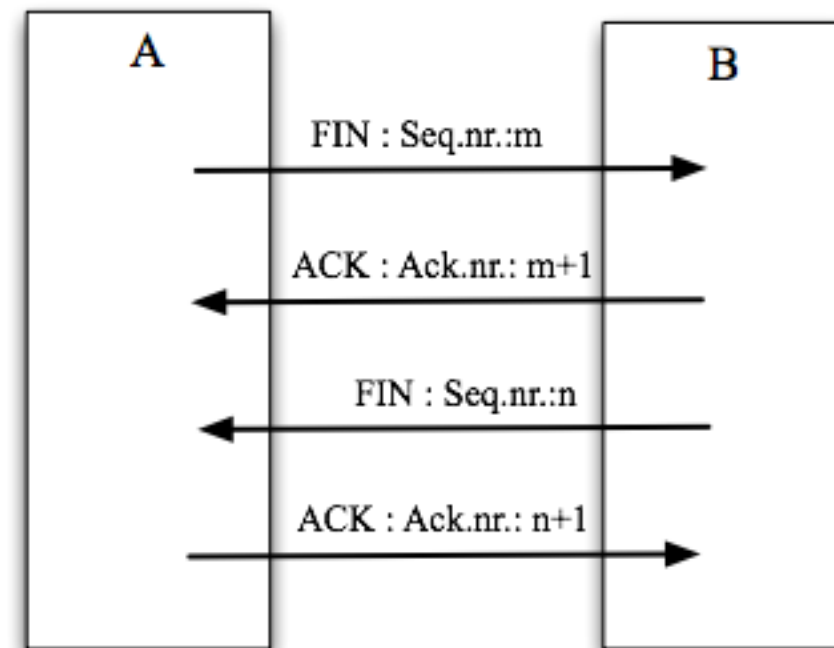


■ Half-Close

- Sender kündigt Ende mit FIN-Segment an und wartet auf Bestätigung
- In Gegenrichtung kann weitergesendet werden



■ 2 Half-Close beenden TCP-Verbindung



- Huckepack-Technik
 - Bestätigungen „reiten“ auf den Datenpaket der Gegenrichtung
- Eine Bestätigungssegment kann viele Segmente bestätigen
 - Liegen keine Daten an, werden Acks verzögert

