

# TCP – UDP

## Allgemeines

- TCP und UDP setzen auf IP auf
- Transportprotokolle, d.h. sie geben die Daten an die einzelnen Programme weiter
- Auf Schicht 4 (Transportschicht) des OSI-Referenzmodelles angesiedelt
- Es werden Ports (0 – 65.535) verwendet, um die Daten den Programmen zuzuordnen
- Daten werden in einzelne Pakete aufgeteilt, um die Daten für die unteren Schichten vorzubereiten, die über Größenbeschränkungen verfügen

## TCP

- TCP = **T**ransmission **C**ontrol **P**rotocol
- IP-Protokollnummer: 06
- Wird für komplexe Protokolle verwendet (z.B. FTP, HTTP, SMTP)
- Verbindungsorientiertes Protokoll  
→ Daten kommen trotz Paketfragmentierung so beim Empfänger an, wie sie versendet wurden
- Datenpakete werden mit Sequenznummern sortiert, um sie beim Empfänger wieder zusammensetzen zu können
- Zuverlässiges Protokoll, Empfänger muss jedes Datenpaket bestätigen, in dem er das nächste Paket anfordert (Sender antwortet mit einem Paket ohne Daten, in dessen Header das ACK-Bit gesetzt wurde und eine *Acknowledgment number* enthält)
- Pakete können gleichzeitig in beide Richtungen versendet werden (Vollduplex)
- Verwendet Drei-Wege-Handshake, um zu garantieren, dass die Verbindung korrekt aufgebaut wurde und dass der Datenweg nicht gestört ist:
  - Client sendet SYN-Paket
  - Server beantwortet mit gesetztem SYN- und ACK-Flag
  - Client beantwortet Paket des Servers
- Verwendet Flusssteuerung, damit ein langsamer Empfänger nicht mehr Daten bekommt, als er verarbeiten kann (*Window*-Feld des Headers enthält eine Zahl von Paketen, die er noch bereit ist, anzunehmen)
- Aufgrund der Paketprüfung langsam

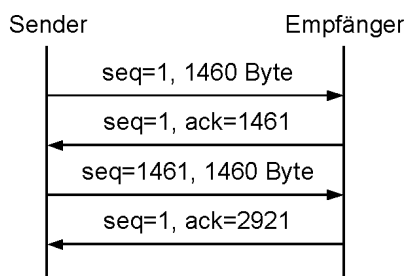
## UDP

- UDP = **U**ser **D**atagram **P**rotocol
- IP-Protokollnummer: 17
- Wird zumeist für einfache Frage-Antwort-Protokolle verwendet (z.B. DNS, DHCP, NTP)
- Verbindungsloses Protokoll  
→ Datenpakete werden wie bei einem Brief einfach hinausgeschickt
- Pakete werden nicht geprüft, nicht bestätigt und nicht sortiert
- Schnell, da Daten nicht überprüft und direkt an das Programm weitergegeben werden  
→ Eignet sich für die Übertragung von Sprache

## Vergleich von TCP und UDP

	TCP	UDP
Verbindungsorientiert	ja	nein
Zuverlässig	ja	nein
Duplex	ja	nein
Sortierung der Pakete	ja	nein
Flusssteuerung	ja	nein
Fehlererkennung	ja	optional
Fehlerbehebung	ja	nein
Drei-Wege-Handshake	ja	nein
Headergröße	20-60 Byte	8 Byte
Geschwindigkeit	langsam	schnell
Systembelastung	normal	gering

## Beispiel einer TCP-Datenübertragung



### Erklärung:

Senden der Bytes 1-1460

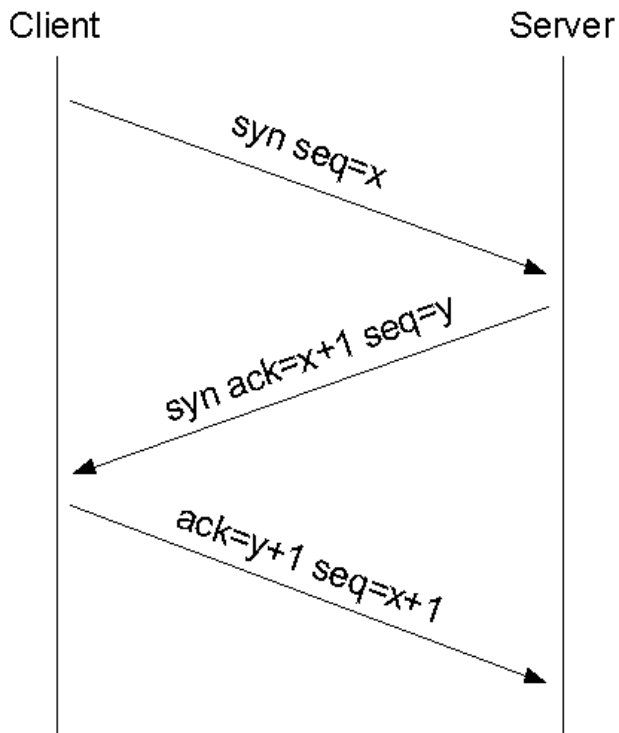
Hat Bytes 1-1460 erhalten, fordert Daten ab Byte 1461 an

Senden der Bytes 1461-2920

Hat Bytes 1461-2920 erhalten, fordert Daten ab Byte 2921 an

**Wichtig:** Die Sequenznummern errechnen sich anhand der Bytes, die übertragen werden! Im obigen Beispiel überträgt der Sender zunächst das Paket, das mit dem ersten Byte der zu übertragenden Daten beginnt und insgesamt 1460 Byte Daten enthält. Anschließend fordert der Empfänger das Paket an, das mit 1461. Byte der zu übertragenden Daten beginnt. Der Sender verschickt anschließend ein Paket mit der Sequenznummer 1461, da dieses Paket ja mit dem 1461. Byte der zu übertragenden Daten beginnt.

## *Drei-Wege-Handshake*

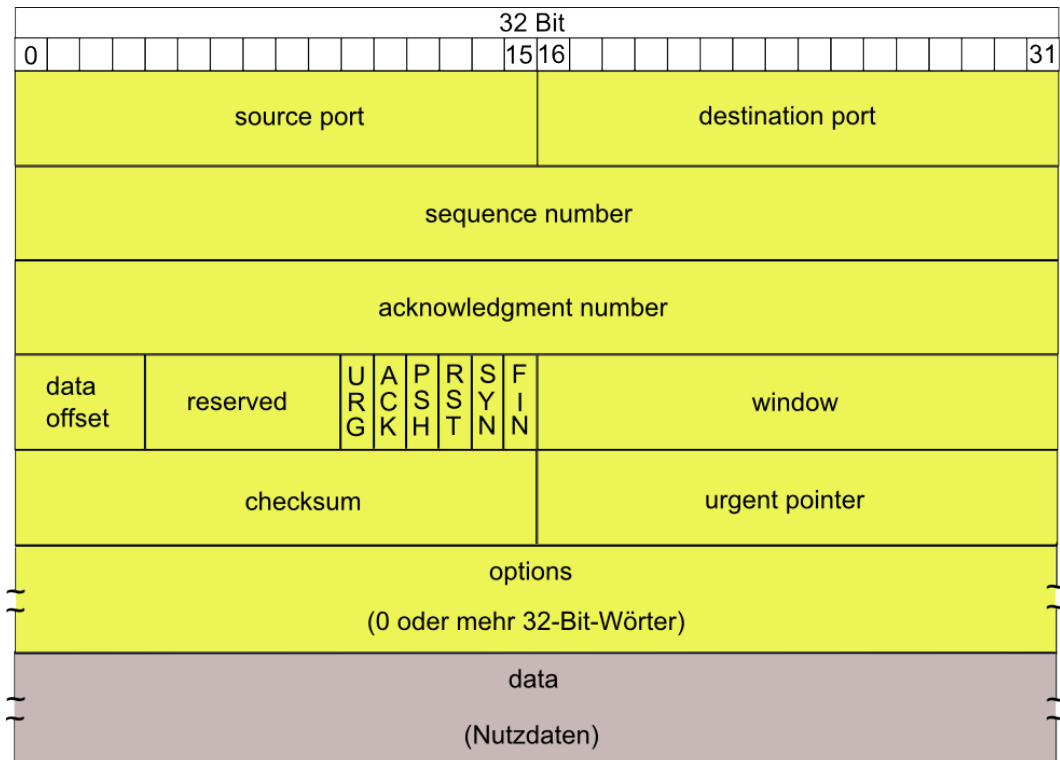


### **Ablauf:**

1. Client sendet ein Paket ohne Daten, mit gesetztem SYN-Bit und einer Sequenznummer  $x$
2. Server antwortet ebenfalls mit einem Paket ohne Daten, in dem das SYN- und das ACK-Bit gesetzt sind, die *Acknowledgment number* enthält den Wert  $x+1$ , zusätzlich wird noch die Sequenznummer  $y$  versendet
3. Client antwortet mit einem Paket ohne Daten, in dem das ACK-Bit gesetzt ist, es wird die *Acknowledgment number*  $y+1$  versendet, zusätzlich noch die Sequenznummer  $x+1$

Nach diesen drei Schritten ist die Verbindung aufgebaut.

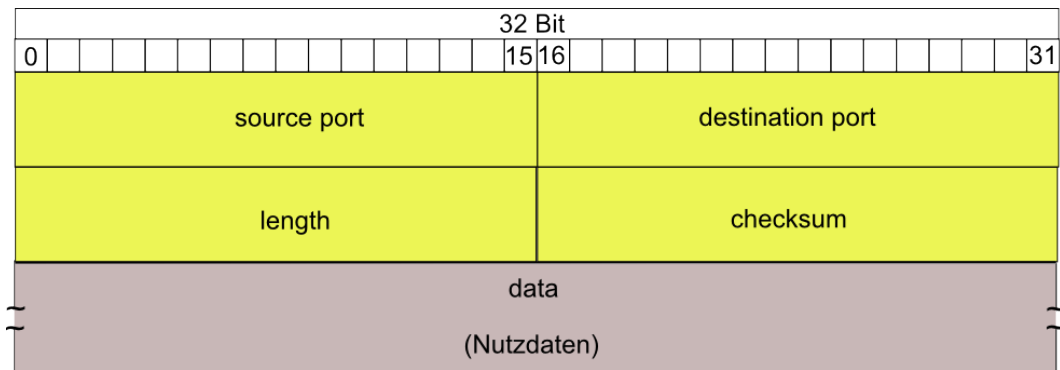
## TCP-Paket



TCP-Header

Quelle: Wikipedia

## UDP-Paket



UDP-Header

Quelle: Wikipedia