

# Rechnernetze - Tutorium

## zu Kapitel 6

Link zu den Folien 

[GitHub Pages](#)

1. Erläutere das Prinzip der IP Fragmentierung. Wer fragmentiert warum und wie?  
Wer setzt die Fragmente wieder zusammen?
2. Welche Vorteile hat der Einsatz von IPv6? Welche Hindernisse bei der Migration siehst du? Durch welche Ansätze versucht IPv6 die Verarbeitungsgeschwindigkeit in Routern zu erhöhen?
3. Grenze MTU und MSS voneinander ab.
4. Definiere die Begriffe Root Name Server, DNS Prefetching und DNS Round Robin.
5. Erläutere den Aufbau des TCP Rahmens und die Funktion der einzelnen Felder.
6. Erläutere die Begriffe Port und Socket, Three Way Handshake, ISN, Sliding Windows und Piggybacking.

7. Beschreibe den Einsatz der TCP Flags.
8. Wie erkennen und reagieren TCP und UDP auf ...
  - in falscher Reihenfolge zugestellte Pakete
  - Paketverlust und Paketverdopplung
  - fehlerhaft adressierte Ports
9. Was versteht man unter der TCP Flusssteuerung?
10. Erläutere das in der Vorlesung vorgestellte TCP Zustandsdiagramm.
11. Markiere die Felder des IP und TCP Header, die nach erfolgreichem Verbindungsaufbau während der Phase der Datenübertragung über das Internet nicht konstant bleiben.

**1.**

**Erläutere das Prinzip der IP Fragmentierung.**

**Wer fragmentiert warum und wie?**

**Wer setzt die Fragmente wieder zusammen?**

# Erläutere das Prinzip der IP Fragmentierung

Das Paket ist für den nachfolgenden Übertragungsweg zu lang, daher wird fragmentiert.

Dabei übernimmt jedes Fragment den IP Header.

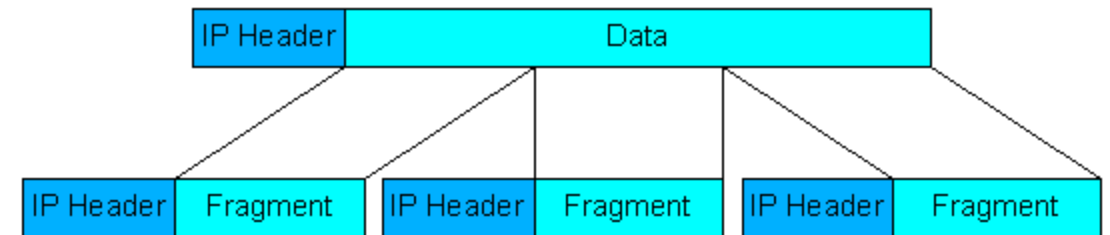


Bild Quelle: [http://www.cfd.tu-berlin.de/Lehre/EDV2/tcpip/kap\\_2\\_3.html](http://www.cfd.tu-berlin.de/Lehre/EDV2/tcpip/kap_2_3.html)

# Wer fragmentiert warum und wie?

- Der Router bei IPv4, sofern das Paket für den nachfolgenden Übertragungsweg zu lang ist.
- Dabei muss die MTU (Maximum Transmission Unit) berücksichtigt werden. Das Paket kann nicht länger als die MTU sein.
- Die Felder Identification, Fragment Offset und die Flags DF/MF werden angepasst
  - DF = 1 ➡ Fragmentierung verboten
  - MF = 1 ➡ weitere Fragmente folgen
  - MF = 0 ➡ letztes Fragment

# Wer setzt die Fragmente wieder zusammen?

- Das Gerät an der Ziel IP (bei IPv4 und IPv6).
- Ist ein Fragment fehlerhaft, werden alle Fragmente des Frames verworfen und das gesamte Paket wird erneut angefordert.

**2.**

**Welche Vorteile hat der Einsatz von IPv6?**

**Welche Hindernisse bei der Migration siehst du?**

**Durch welche Ansätze versucht IPv6 die Verarbeitungsgeschwindigkeit in Routern zu erhöhen?**



# Welche Vorteile hat der Einsatz von IPv6?

- großer Adressraum
  - IPv6  $2^{128} = 340.282.366.900.000.000.000.000.000.000.000.000.000$   
(Sextillionen)
  - IPv4  $2^{32} = 4.294.967.296$
- NAT von IPv4 hat ausgedient
  - es gibt jedoch auch NAT6
- Privacy Extensions
- keine IP Klassen
- einfacher Protokollaufbau (Performance)
- verbesserte Sicherheit (Authentifizierung und Verschlüsselung)
- bereit für zukünftige Erweiterungen durch das Prinzip des Next Header

# Welche Hindernisse bei der Migration siehst du?

## Beispiel von mir ..

- Bist du bei Unitymedia?
- Hast du einen DS Lite (Dual Stack Lite) Anschluss?
- Deine FritzBox hat nur eine IPv6?
- Du möchtest von unterwegs mit IPv4 einfach auf dein Heimnetz zugreifen?

Schonmal ausprobiert? Hat nicht geklappt? Lösungen heißen [tunnel6](#) oder [socat](#), brauchen jedoch bsp. einen vRoot außerhalb deines Netzwerks mit einer IPv4.

Alternativ den Power Upload bestellen, mit der Option von DS-Lite auf DS umgestellt zu werden.

Und hier sind wir schon bei den Problemen ..

# Welche Hindernisse bei der Migration siehst du?

- Koexistenz von IPv4 und IPv6
  - Ein komplettes Umstellen von IPv4 auf IPv6 ist noch nicht möglich, da nicht alle Geräte kompatibel sind.
- Aufgrund intelligenter Ressourcennutzung bei IPv4, herrscht weniger Druck zur vollständigen Umstellung
- Hardware Umstellungen
  - Carrier
  - Provider
  - User (alter Router)

Ein kompletter Umstieg bedeutet hohe Kosten.

# Durch welche Ansätze versucht IPv6 die Verarbeitungsgeschwindigkeit in Routern zu erhöhen?

- Header optimiert
  - ➡ wenig genutzte Pflichtfelder bei IPv4 werden optional
- Absender des IPv6 Pakets fragmentiert
- keine Checksumme
- Flow Label
  - ➡ Das Flow Label kennzeichnet Pakete für ein viel schnelleres Routing.

### **3. Grenze MTU und MSS voneinander ab.**

## MSS - Maximum Segment Size

- Die MSS gibt nur den Platz für den Payload im TCP/IP Paket an.

## MTU - Maximum Transmission Unit

- Die MTU gibt die Größe des kompletten TCP/IP Pakets an.

 **MTU ist die obere Grenze an maximaler Paketgröße der unteren Schichten.**

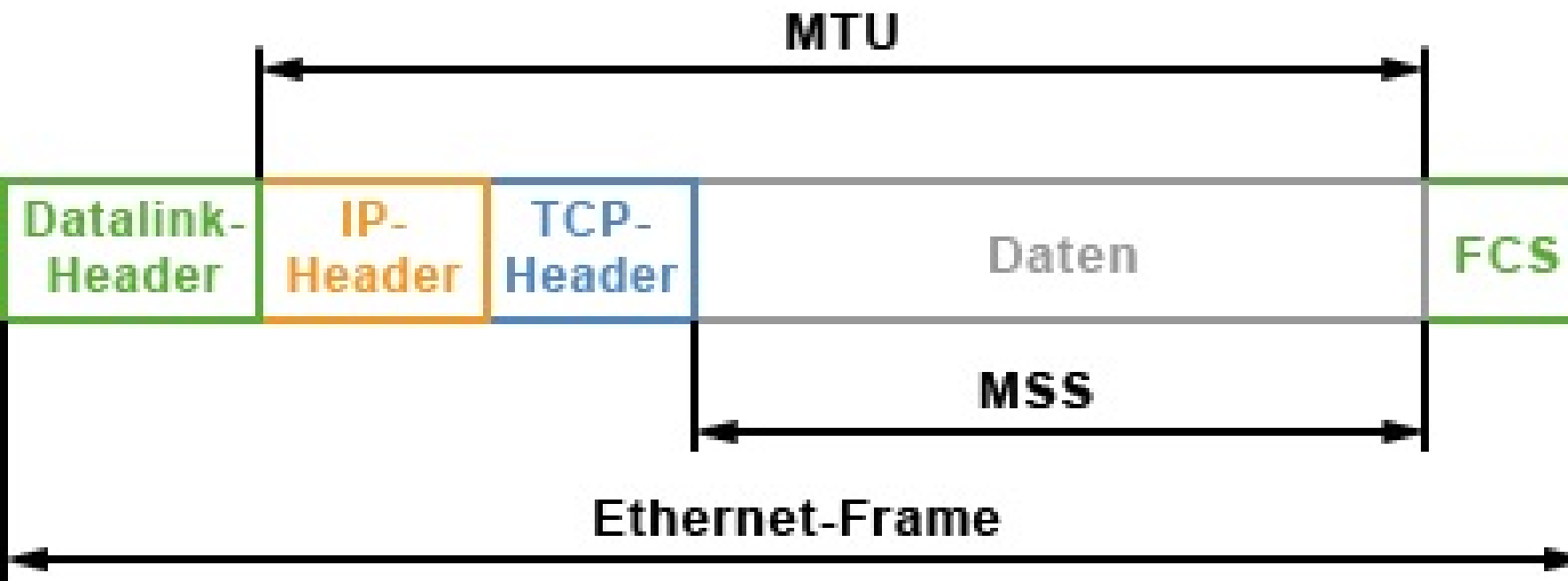


Bild Quelle:  
<https://www.elektronik-kompendium.de/sites/net/0812211.htm>

## **4. Definiere die Begriffe**

**Root Name Server**

**DNS Prefetching**

**DNS Round Robin**



# Root Name Server

- Die Domain Name System (DNS) Root Name Server ..
- .. sind für die Top Level Domain (TLD) Namensauflösung ( `.de` , `.com` , `.io` , `.org` , ...) zuständig.
  - 💡 DNS übersetzt den Namen einer Webseite in eine IP Adresse  
( `de.wikipedia.org` )

Die Root Name Server kennen `.org` und wissen wo die DNS Server für die *Second Level Domain* `wikipedia` sind und schauen dann bei der nächsten Hierarchiestufe nach.

**Heise Artikel** - [Hyperlocal: Wenn die DNS-Root-Zone zu Hause steht](#)

# DNS Prefetching

## Links auf Webseiten werden schon mit DNS aufgelöst.

Eine IP einer verlinkten Seite ist bereits geladen, wenn ein Benutzer darauf tippt.

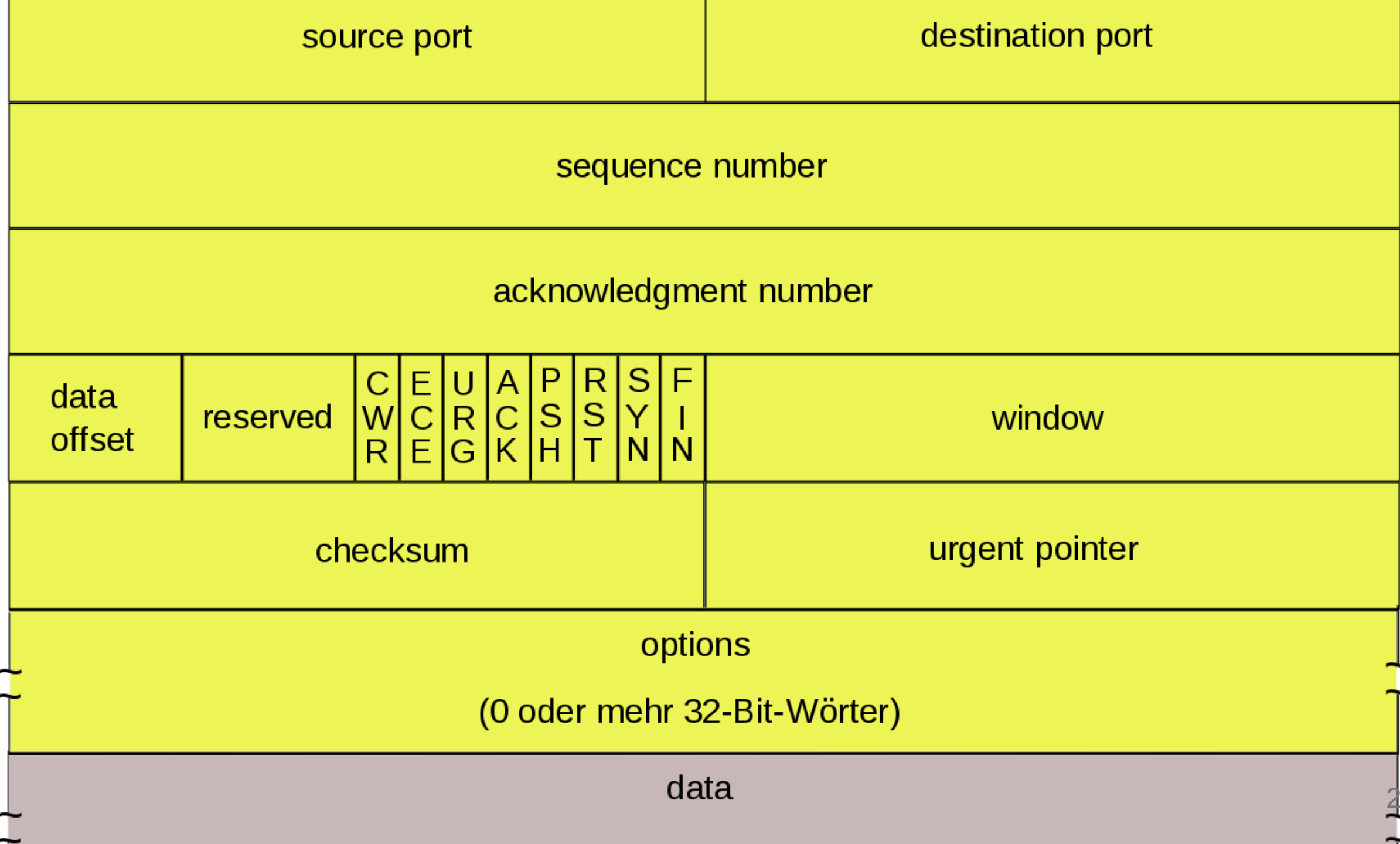
```
<html>
  <head>
    <\!-- Amazon S3 -->
    <link rel="dns-prefetch" href="//s3.amazonaws.com">
  </head>
</html>
```

# DNS Round Robin

## Lastverteilung für Netzwerkdienste per DNS.

- einer Domain werden mehrere IP Adressen zugewiesen
- wird der Name vom Resolver abgefragt, werden mehrere bekannte IP Adressen in wechselnder Reihenfolge zurück geliefert

## **5. Erläutere den Aufbau des TCP Rahmens und die Funktion der einzelnen Felder.**



# Funktionen der einzelnen Felder

## Source Port

- Port auf der Senderseite

## Destination Port

- Port an den Informationen weitergeleitet werden sollen

## Sequence Number

- Reihenfolge der Datenpakete

## Acknowledgment Number

- Bestätigung fehlerfrei empfangener TCP Datenpakete

## Data Offset

- Zeige auf den Beginn des Payload
- Länge des Header

## Reserved

- Reserviert für zukünftige Verwendungen

## Control Flags

- SYN, ACK, FIN, RST, URG, PSH

## Window Size

- gibt an wie viele Bytes der Sender unbestätigt senden darf

## Checksum

- Prüfsummenbildung zur Fehlererkennung

## Urgent Pointer

- Zeiger zur Lokalisierung wichtiger Daten (Priorisierung)

## Options

- enthält Zusatzinformationen

## Data

- der eigentliche Payload



## **6. Erläutere die Begriffe**

**Port und Socket**

**Three Way Handshake**

**ISN**

**Sliding Window**

**Piggybacking**

# Port

Wird für die Kennzeichnung von Diensten und Protokollen verwendet.

**Well Known Ports** (0 bis 1023) sind für Dienste reserviert

- 21 FTP, 22 SSH, 80 HTTP, 443 HTTP über TLS

**Registered Ports** (1024 bis 49151) sind für Anwendungen standardisiert

- 1194 OpenVPN, 5060 SIP, 8087 Kaspersky

**Dynamic Ports** (49152 bis 65535) für ausgehende Verbindungen

# Socket

Kombination aus IP Adresse und Port.



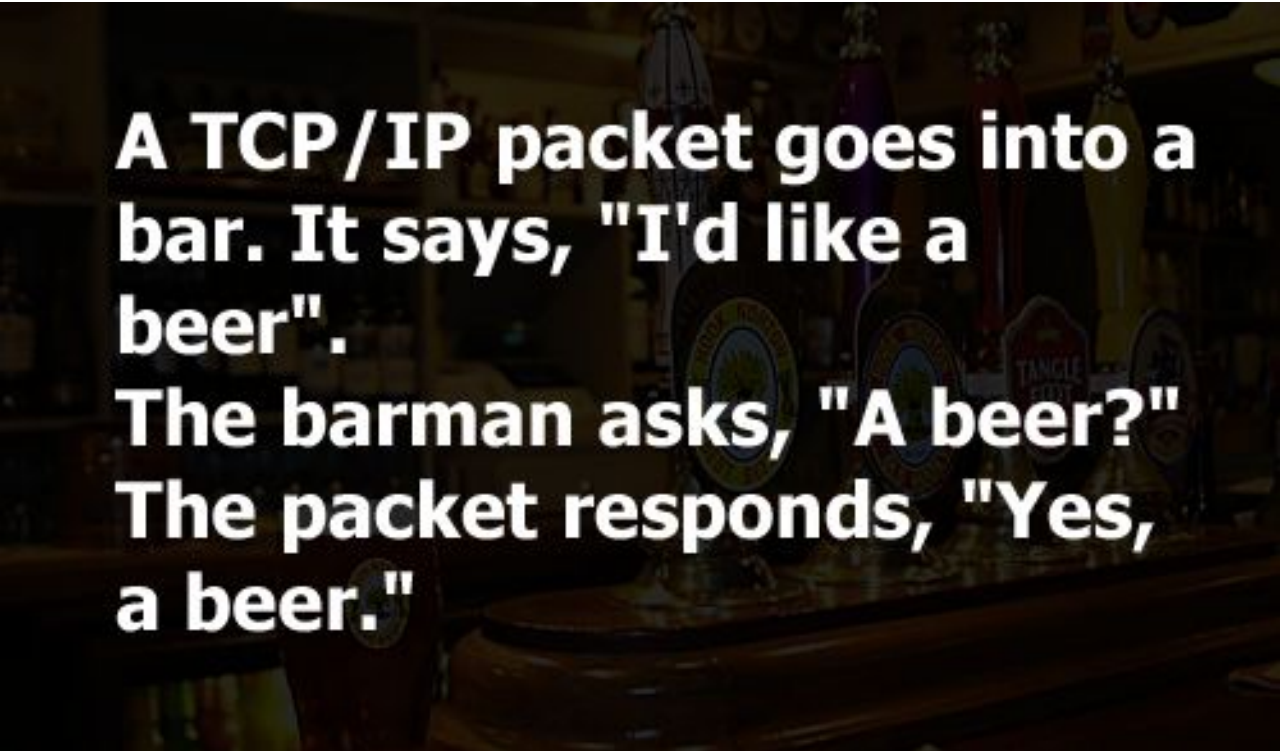
# Three Way Handshake

## Verbindungsaufbau

1. Client sendet SYN
2. Server bestätigt mit ACK und sendet ebenfalls SYN
3. Client bestätigt mit ACK

Bild Quelle:

<https://www.pinterest.de/pin/9710955419313046/>



**A TCP/IP packet goes into a bar. It says, "I'd like a beer".  
The barman asks, "A beer?"  
The packet responds, "Yes, a beer."**

# ISN - Initial Sequence Number

- zufällig gewählte Zahl, die die erste SEQ Nummer ist
- ist auf beiden Seiten der Verbindung bekannt
- wenn das SYN Flag gesetzt ist, wird die ISN benötigt

## Warum verwendet man eine Zufallszahl?

- Damit die Verbindung nicht einfach so übernommen werden kann
- Verbesserung der Sicherheit

# Sliding Window

**Dient der Datenflusssteuerung um einen Datenstau beim Empfänger zu verhindern.**

- Kleine Fenster eher bei Echtzeit Anwendungen
- Große Fenster ermöglichen einen höheren Datendurchsatz bei der Übertragung

# Piggybacking

**Anstatt das ACK Frame einzeln zu senden, wird es vom Payload eines Datenpakets gleich mitgenommen.**

## 7. Beschreibe den Einsatz der TCP Flags.

Lass uns mal eine Tabelle füllen.



<b>TCP Flag</b>		<b>Beschreibung</b>
SYN	Synchronize	Verbindungsanforderung wird gesendet
ACK	Acknowledge	Empfänger bestätigt den Erhalt der Daten
FIN	Final	Verbindungsabbau wird eingeleitet
RST	Reset	Abbruch einer Sitzung / Ablehnung einer Verbindungsanforderung
URG	Urgent	Pointer, welcher auf dringende Daten zeigt, dadurch Bevorzugung
PSH	Push	Daten sofort an höhere Protokolle (Layer) weiterleiten

## **8. Wie erkennen und reagieren TCP und UDP auf ...**

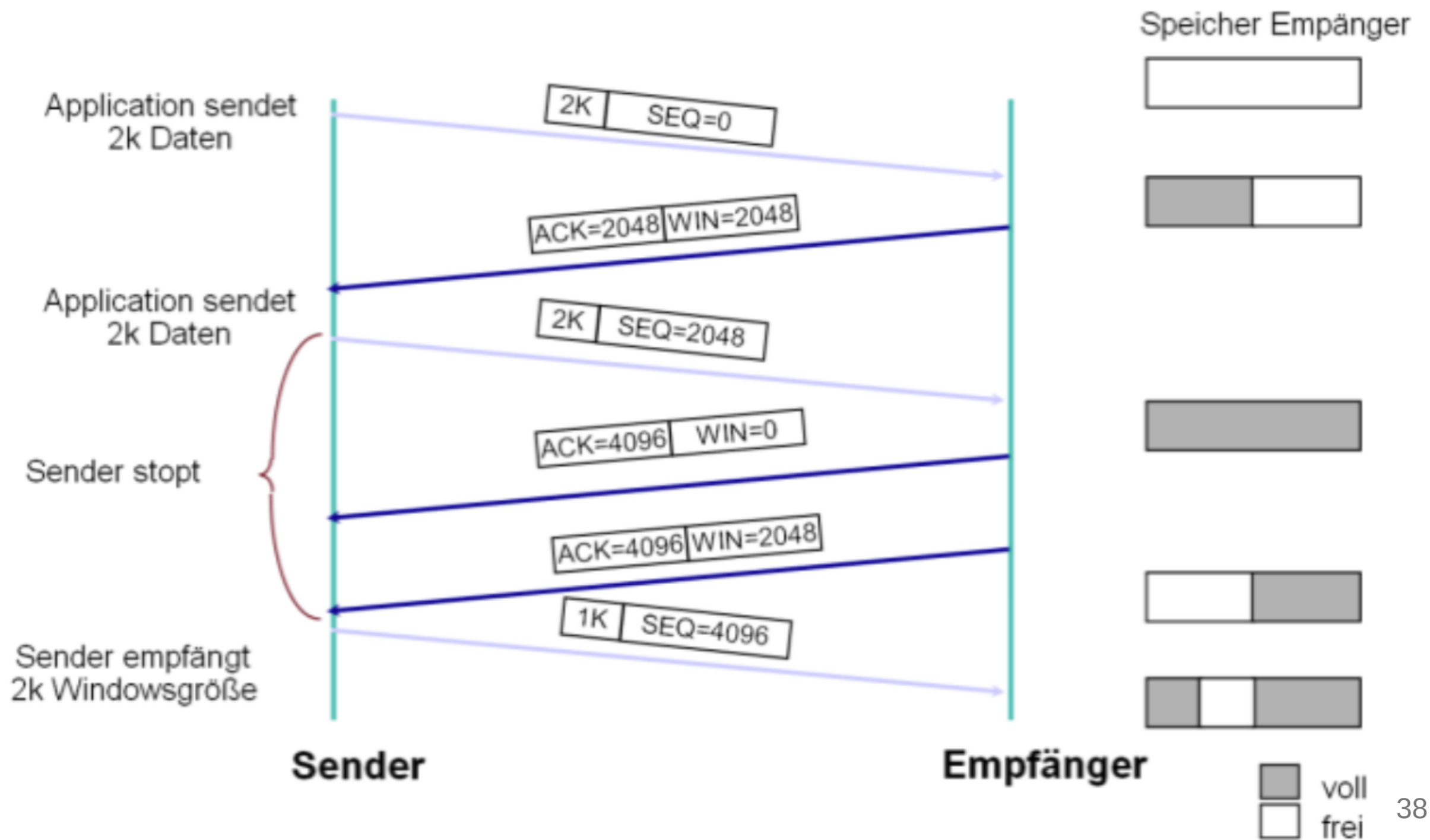
- in falscher Reihenfolge zugestellte Pakete**
- Paketverlust und Paketverdopplung**
- fehlerhaft adressierte Ports**

Szenario	TCP	UDP
falsche Reihenfolge	wird mit SEQ sortiert	nicht relevant
Paketverlust	SEQ Erkennung fehlender Pakete ➡ Neuanforderung	nicht relevant ➡ keine Neuanforderung
Doppelung	SEQ Erkennung doppelter Pakete ➡ Nur 1 Paket wird verarbeitet, rest verworfen	nicht relevant ➡ wird nicht verworfen
Fehlerhaft adressierte Ports	Paket mit gesetztem RST Flag wird zurück gesendet	Paket wird verworfen

## **9. Was versteht man unter der TCP Flusssteuerung?**

# TCP Flusssteuerung

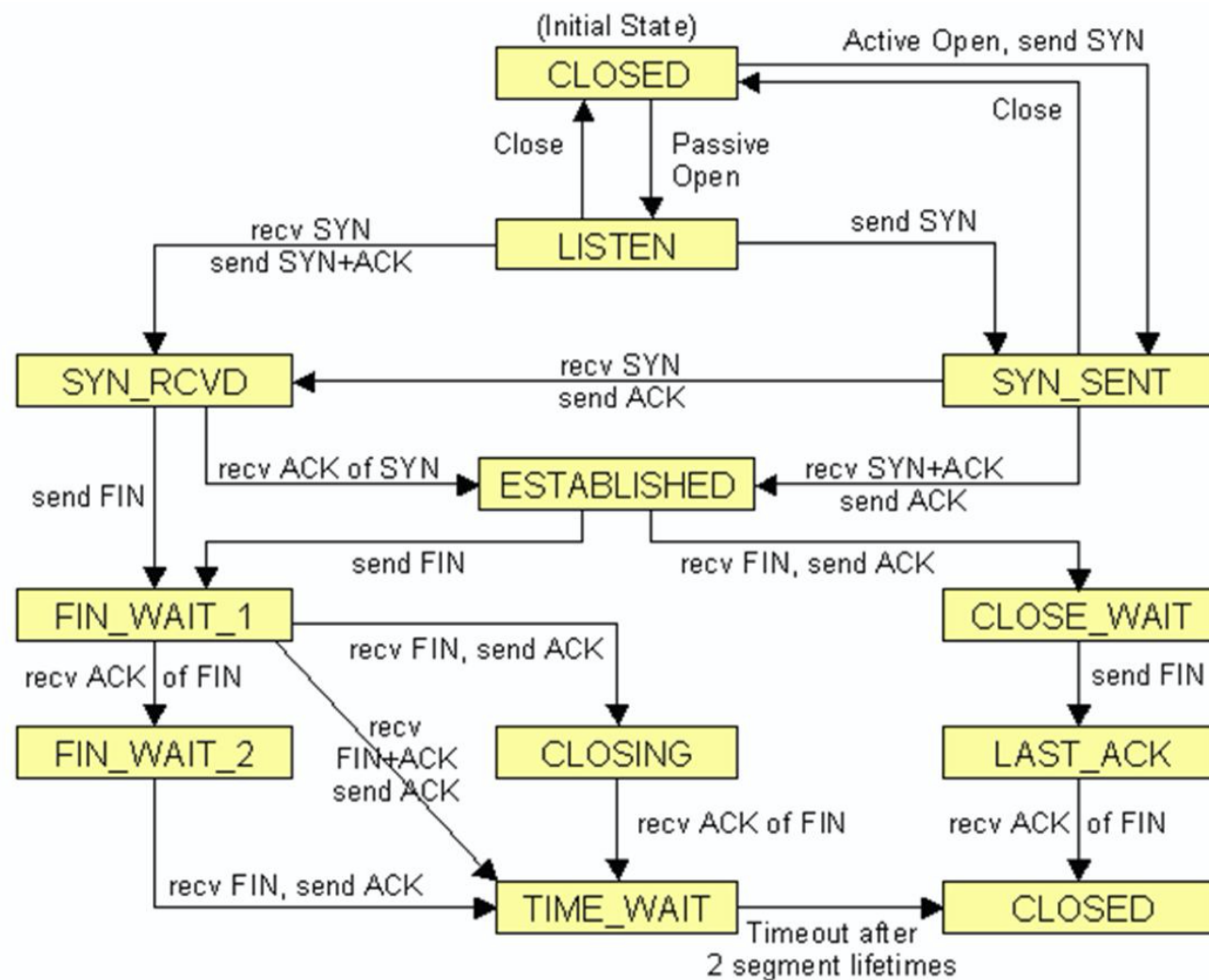
- Daten werden in einem Cache, beim Sender und auch beim Empfänger zwischen gespeichert
  - | Cache, Buffer, Puffer, ... hat man schon mal gehört
- Die aktuelle Cachegröße des Empfängers gibt an, wie viele Bytes der Sender unbestätigt senden darf
  - | Die Flusskontrolle sagt hier aus, das ist die "Aufnahmebereitschaft des Empfängers"
- Wenn der Cache voll ist, wird die Window Size auf 0 gesetzt.



**10. Erläutere das in der Vorlesung vorgestellte TCP Zustandsdiagramm.**



# TCP-Zustandsdiagramm



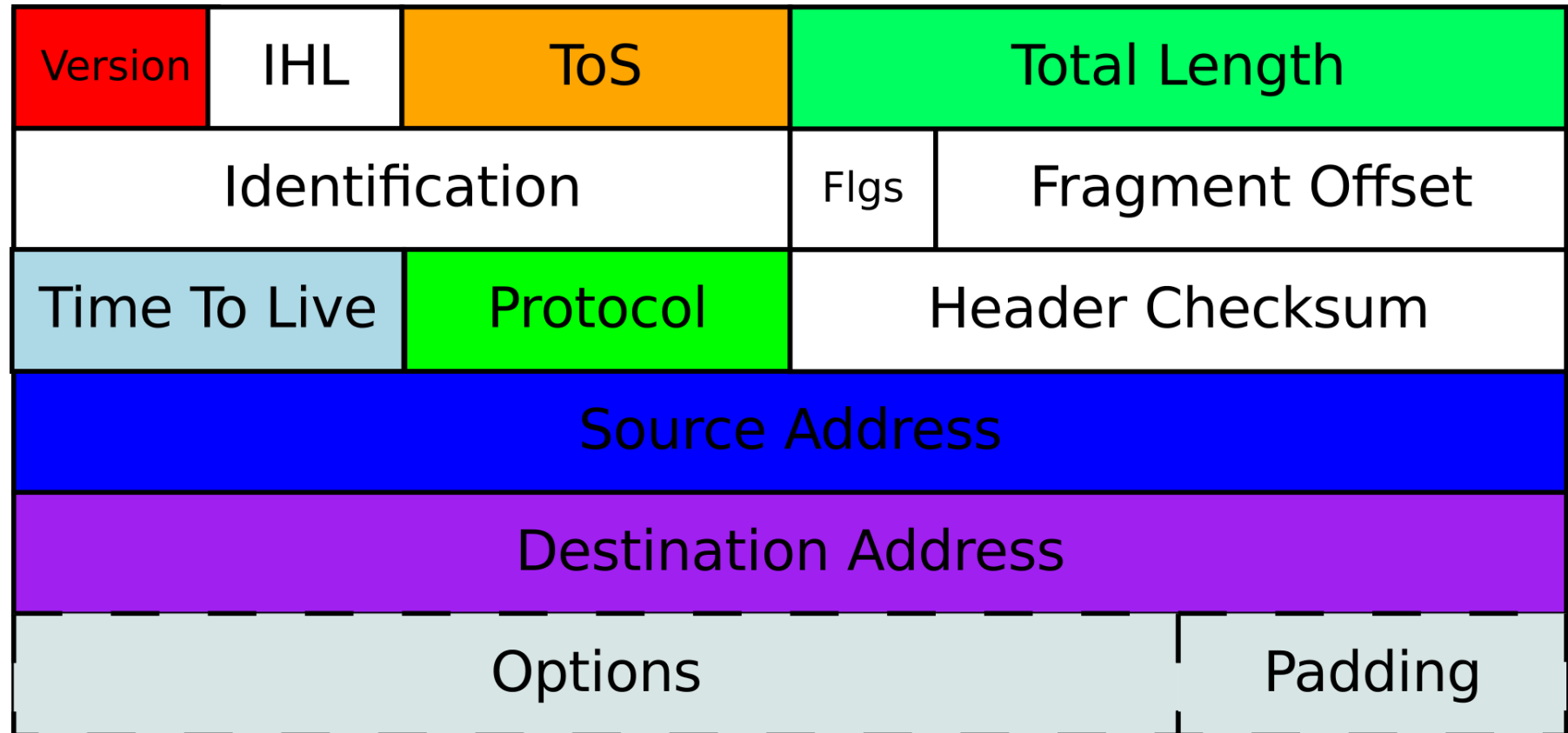


- **CLOSED**: Startzustand
- **LISTEN**: Server wartet auf Verbindungswünsche
- **SYN-RCVD**: Server hat Verbindungswunsch empfangen
- **SYN-SENT**: Client hat Verbindungsaufbau gestartet
- **ESTABLISHED**: Verbindung ist etabliert und betriebsbereit
- **FIN-WAIT-1**: Server/Client hat den Verbindungsabbau eingeleitet
- **FIN-WAIT-2**: Client/Server hat den Verbindungsabbau bestätigt
- **TIMED-WAIT**: Warten bis alle Segmente verschwunden sind
- **CLOSING**: Endpunkte beenden die Verbindung gleichzeitig
- **CLOSE-WAIT**: Entferntes System hat einen Verbindungsabbau gestartet
- **LAST-ACK**: Warten bis alle Segmente verschwunden sind

**11. Nenne die Felder des IP und TCP Header, die nach erfolgreichem Verbindungsaufbau während der Phase der Datenübertragung über das Internet nicht konstant bleiben.**

Bild Quelle

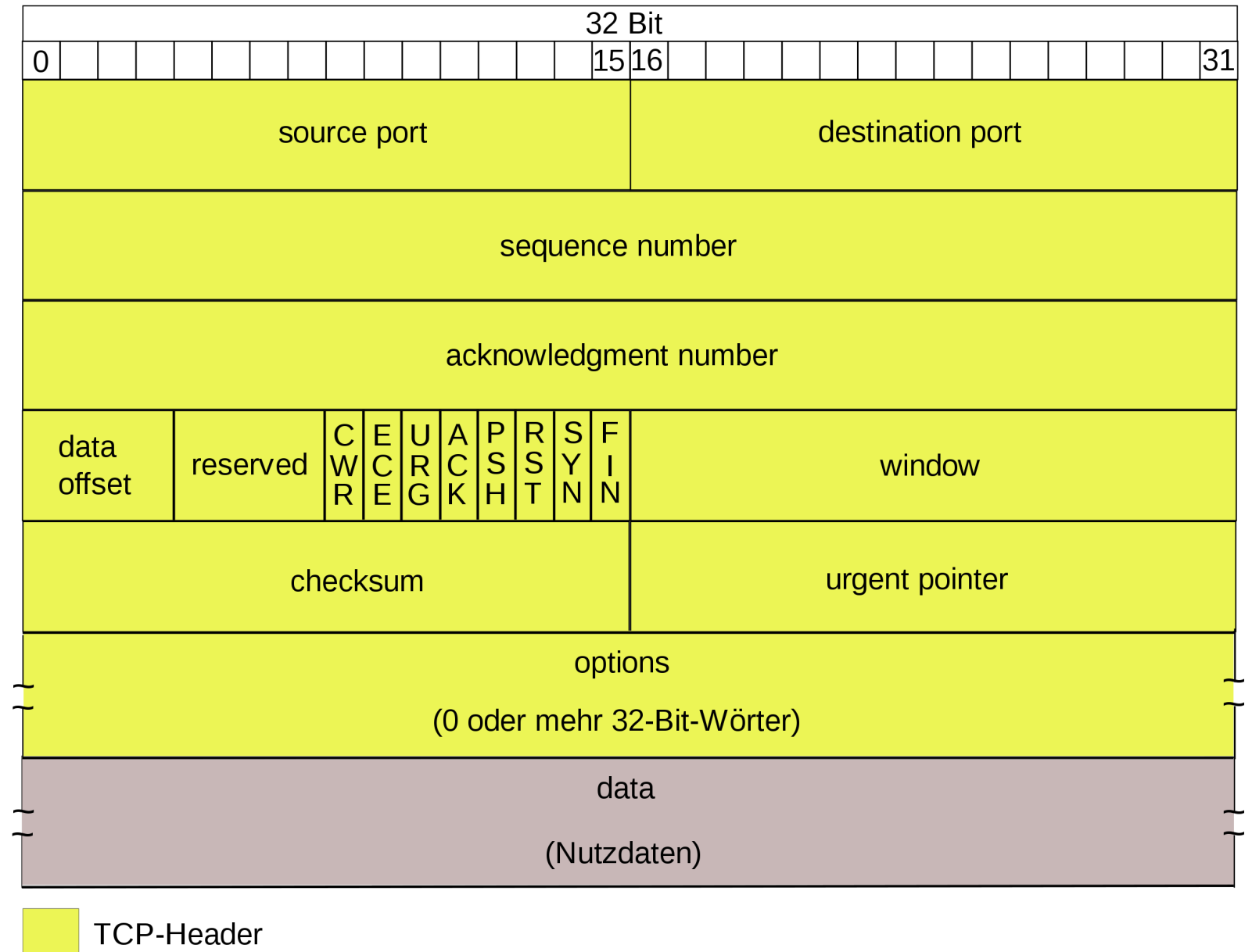
[https://commons.  
wikimedia.org/wi  
ki/File:Ipv4\\_head  
er.svg](https://commons.wikimedia.org/wiki/File:Ipv4_header.svg)



# IP

- Paketlänge
- Ident
- MF
- Fragment Offset (wenn DF/MF aktiviert)
- Lebenszeit
- Prüfsumme

Von Appaloosa  
 23:04, 6. Jul.  
 2007 (CEST) -  
 drawn by me with  
 inkscape, CC BY-  
 SA 3.0,  
[https://commons.  
 wikimedia.org/w/i  
 ndex.php?  
 curid=70129536](https://commons.wikimedia.org/w/index.php?curid=70129536)



# TCP

- Sequence Number
- Acknowledgment Number
- Flags
- Window Size
- TCP Checksum
- Urgent Pointer
- Payload

## Weitere Fragen?

Bitte per E-Mail an [kk172@hdm-stuttgart.de](mailto:kk172@hdm-stuttgart.de), [lg088@hdm-stuttgart.de](mailto:lg088@hdm-stuttgart.de). Wir beantworten die dann im kommenden Meeting ausführlich.

**Bis nächste Woche oder jetzt schon viel Erfolg für die Prüfung? 😊**

| `git pull` nicht vergessen