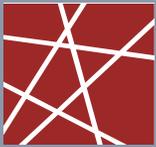




Motivierende Fragen

- ❑ Welche Transportdienste gibt es ?
- ❑ Welche Probleme können beim Transport entstehen ?
- ❑ Welche Aufgaben werden von der Transportschicht erledigt ?
- ❑ Wie funktioniert TCP ?
- ❑ Wie funktioniert UDP ?



Grundlagen: Rechnernetze und Verteilte Systeme

Kapitel 6: Transport-Protokolle

TCP, UDP

Prof. Dr.-Ing. Georg Carle
Lehrstuhl für Netzarchitekturen und Netzdienste
Technische Universität München
carle@net.in.tum.de
<http://www.net.in.tum.de>





Übersicht

1. Einführung und Motivation
 - Bedeutung, Beispiele
2. Begriffswelt und Standards
 - Dienst, Protokoll, Standardisierung
3. Direktverbindungsnetze
 - Fehlererkennung, Protokolle
 - Ethernet
4. Vermittlung
 - Vermittlungsprinzipien
 - Wegwahlverfahren
5. Internet-Protokolle
 - IP, ARP, DHCP, ICMP
 - Routing-Protokolle
6. **Transportprotokolle**
 - **UDP, TCP**
7. Verkehrssteuerung
 - Kriterien, Mechanismen
 - Verkehrssteuerung im Internet
8. Anwendungsorientierte Protokolle und Mechanismen
 - Netzmanagement
 - DNS, SMTP, HTTP
9. Verteilte Systeme
 - Middleware
 - RPC, RMI
 - Web Services
10. **Netzsicherheit**
 - Kryptographische Mechanismen und Dienste
 - Protokolle mit sicheren Diensten: IPSec etc.
 - Firewalls, Intrusion Detection
11. Nachrichtentechnik
 - Daten, Signal, Medien, Physik
12. Bitübertragungsschicht
 - Codierung
 - Modems



- In diesem Kapitel wollen wir vermitteln
 - Arten von Transportdiensten
 - Verbindungsaufbau und -abbau
 - Aufgaben der Transportschicht
 - Funktionalität TCP
 - Funktionalität UDP



6.1. Der Transportdienst (nach ISO/OSI-Begriffswelt)

6.1.1. Phasen des verbindungsorientierten Dienstes

6.1.2. Fehler beim Verbindungsaufbau

6.1.3. Verbindungsabbau

6.2. Aufgaben der Transportschicht

6.2.1. Ende-zu-Ende Kommunikation in Internet

6.2.2. TCP

6.2.2.1. TCP-Paketformat

6.2.2.2. TCP: Mechanismen

6.2.3. UDP



6.1. Der Transportdienst (nach ISO/OSI-Begriffswelt)

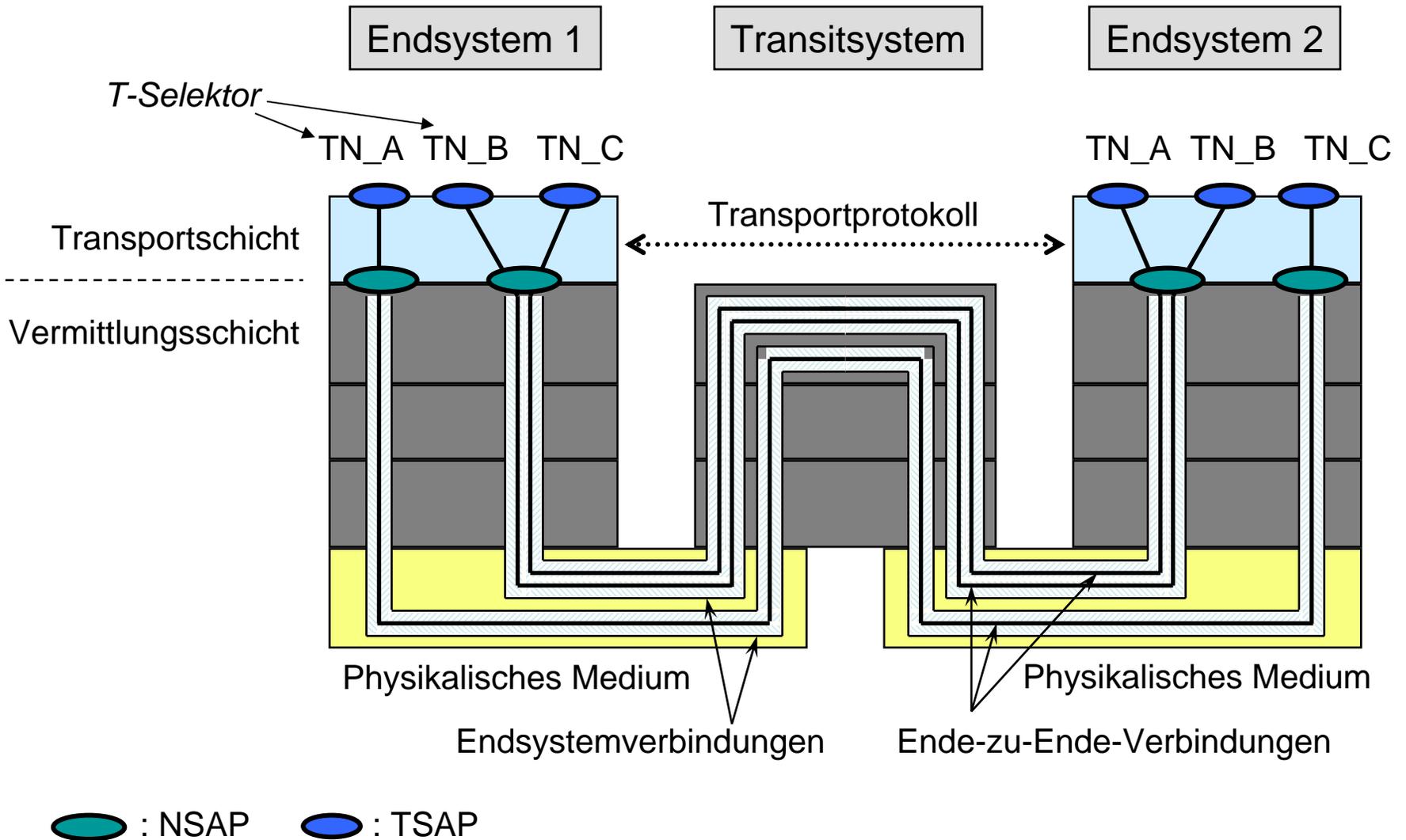
- Man unterscheidet die folgenden Transportdienste:
 - **verbindungsorientiert**
 - **verbindungslos**

- Beim **verbindungsorientierten** Dienst unterscheidet man drei Phasen:
 - Verbindungsaufbauphase (Dienstelement: T-Connect)
 - Datentransferphase (Dienstelement: T-Data)
 - Verbindungsabbauphase (Dienstelement: T-Disconnect)

- Adressierung eines Transportdienstbenutzers durch *TSAP-Adresse* (Transport Service Access Point), beinhaltet:
 - *NSAP-Adresse* (Network Service Access Point) zur Adressierung des Endsystems
 - *T-Selektor* zur Identifizierung des TSAP auf einem Endsystem



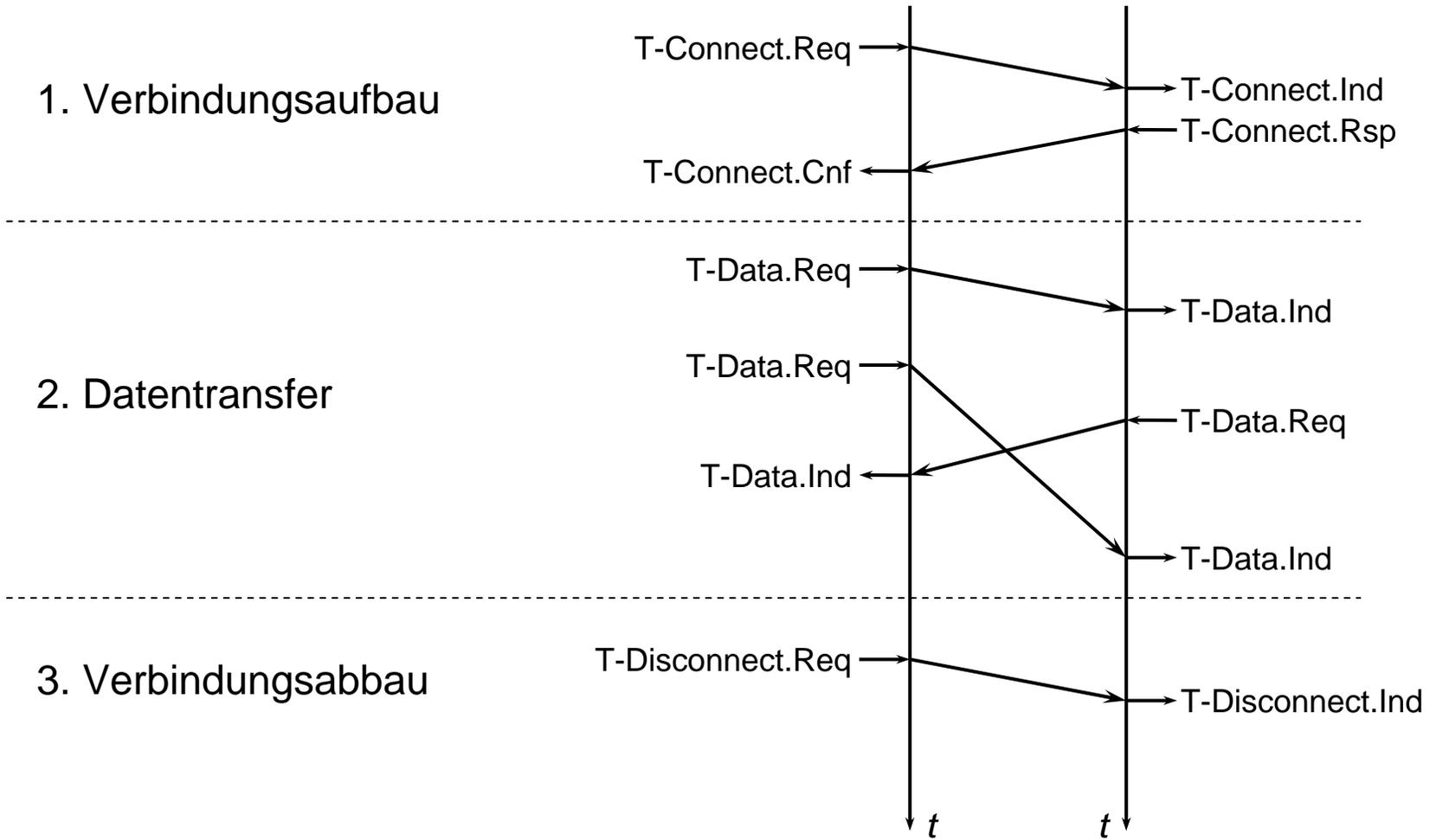
Abstraktionseigenschaft der Transportschicht





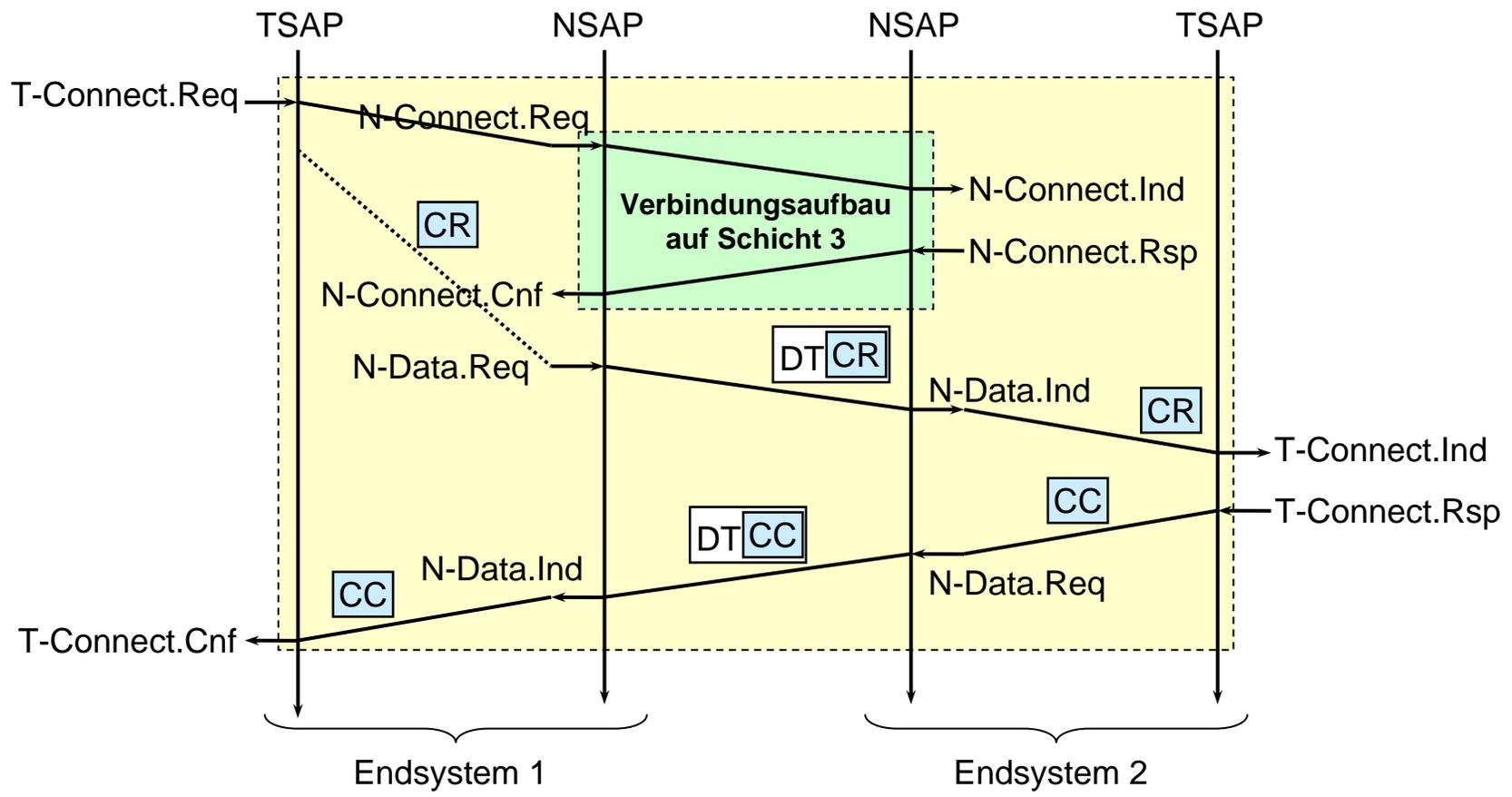
6.1.1. Phasen des verbindungsorientierten Dienstes

Beispiellauf:





Verbindungsaufbau auf Schicht 3 für verbindungsorientierten Transportdienst

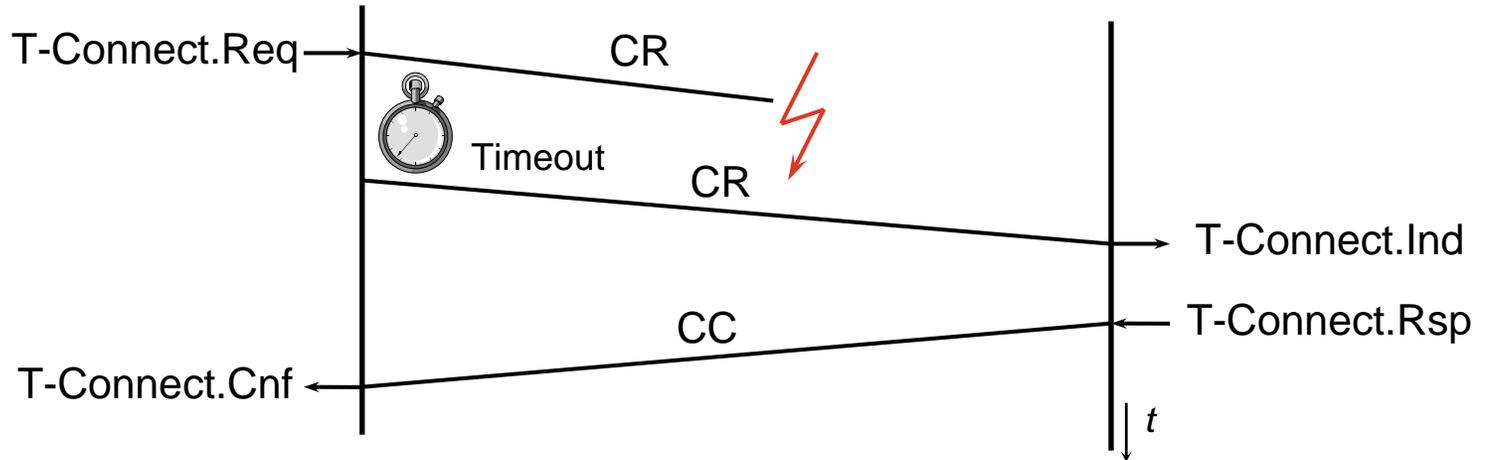


- **Hinweis:** Setzt die Transportschicht auf einem verbindungslosen Dienst der Vermittlungsschicht auf (z.B. IP) oder existiert bereits eine Schicht-3-Verbindung, so ist kein Verbindungsaufbau auf Vermittlungsebene notwendig!

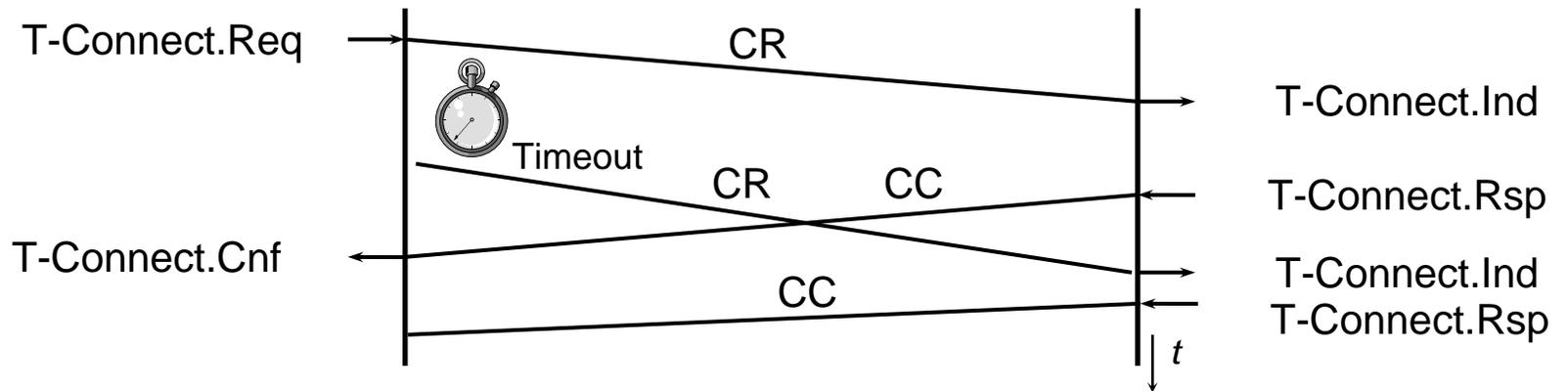


6.1.2. Fehler beim Verbindungsaufbau

- ❑ Verlust der CR oder CC TPDU:



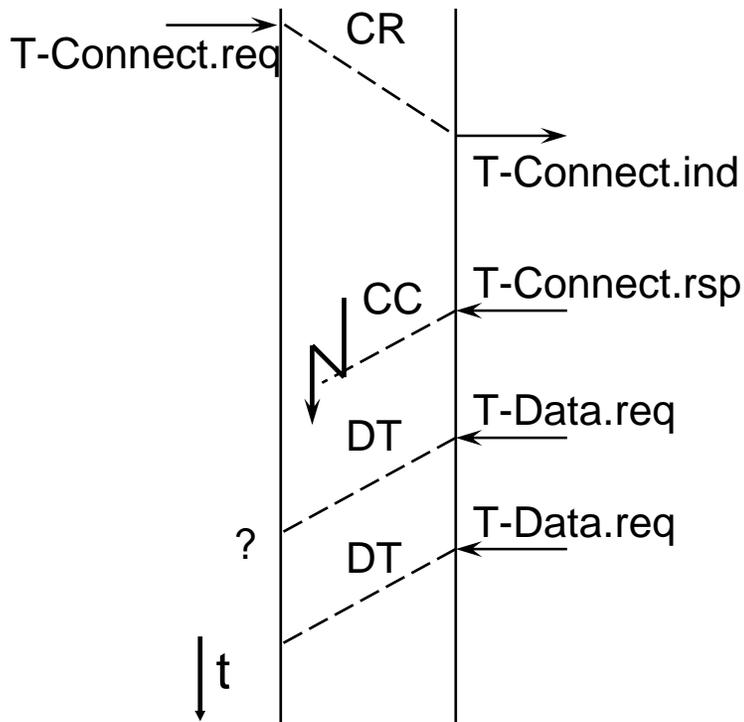
- ❑ Duplizierung von TPDU:



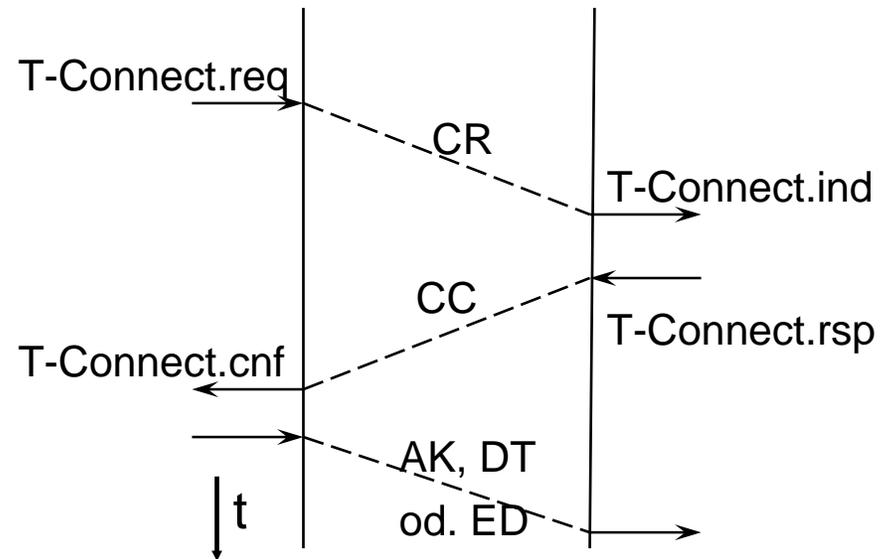


Three-Way Handshake

- ❑ **Problem:** Verlust der CC TPDU



- ❑ **Lösung:** Three-Way Handshake
Verbindung wird erst als aufgebaut anerkannt, wenn beide Verbindungsaufbau TPDU's (CR und CC) quittiert sind.

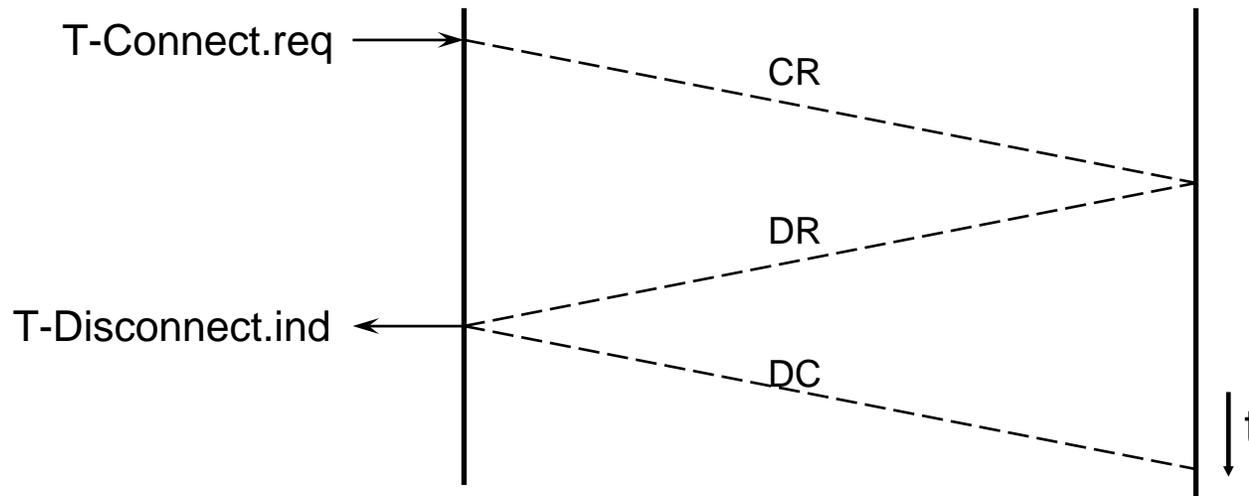




Verbindungsrückweisung

❑ Connection Refusal:

- Zurückweisung eines Verbindungsaufbauwunsches erfolgt durch Disconnect-Request (DR) oder Error-TPDU. Gründe für Zurückweisung werden angegeben.
- Gründe:
 - Zurückweisung durch den Transportdienstbenutzer.
 - Anforderungen an den Dienst können nicht erfüllt werden



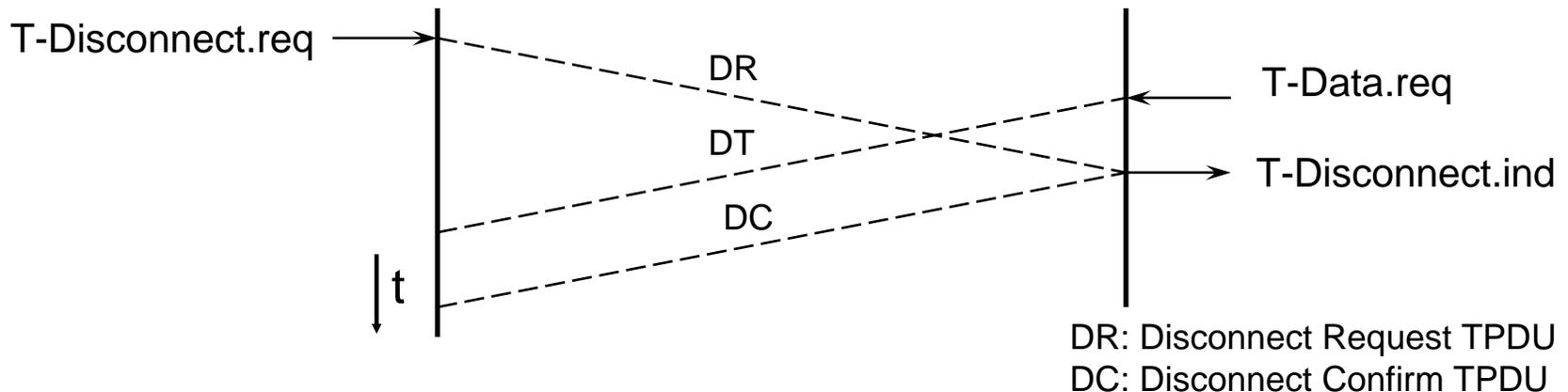
DR: Disconnect Request TPDU
DC: Disconnect Confirm TPDU



6.1.3. Verbindungsabbau

□ Normal Release:

- Beim Verbindungsabbau wird eine bestehende Transportverbindung aufgelöst. Dabei kann es zum Verlust von Daten kommen.
- Varianten:
 - implizit: Abbau der Vermittlungsschichtverbindung.
 - explizit: Abbauprozedur über Disconnect-TPDUs.

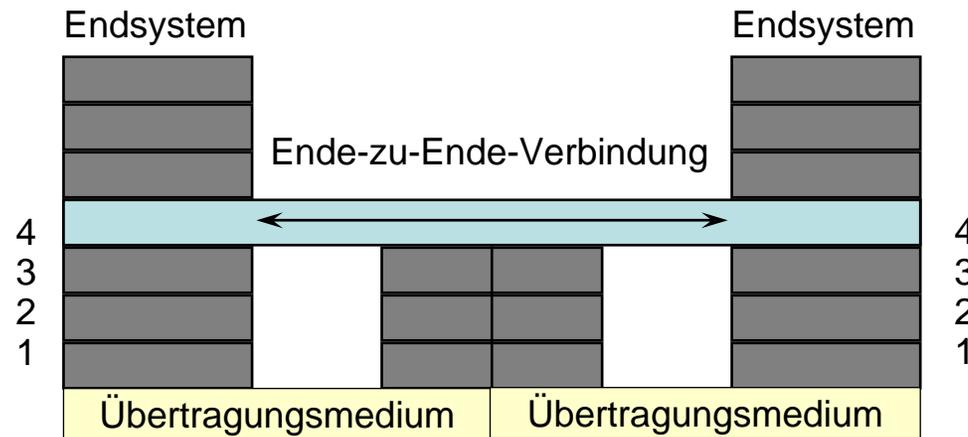


- Verbindungsabbau nach Fehler (Error Release): Kann nach einem Fehler (N-Disconnect oder N-Reset) keine geeignete Fehlerbehandlung erfolgen, wird eine Transportverbindung vom Transportdienstleister abgebaut.



6.2. Aufgaben der Transportschicht

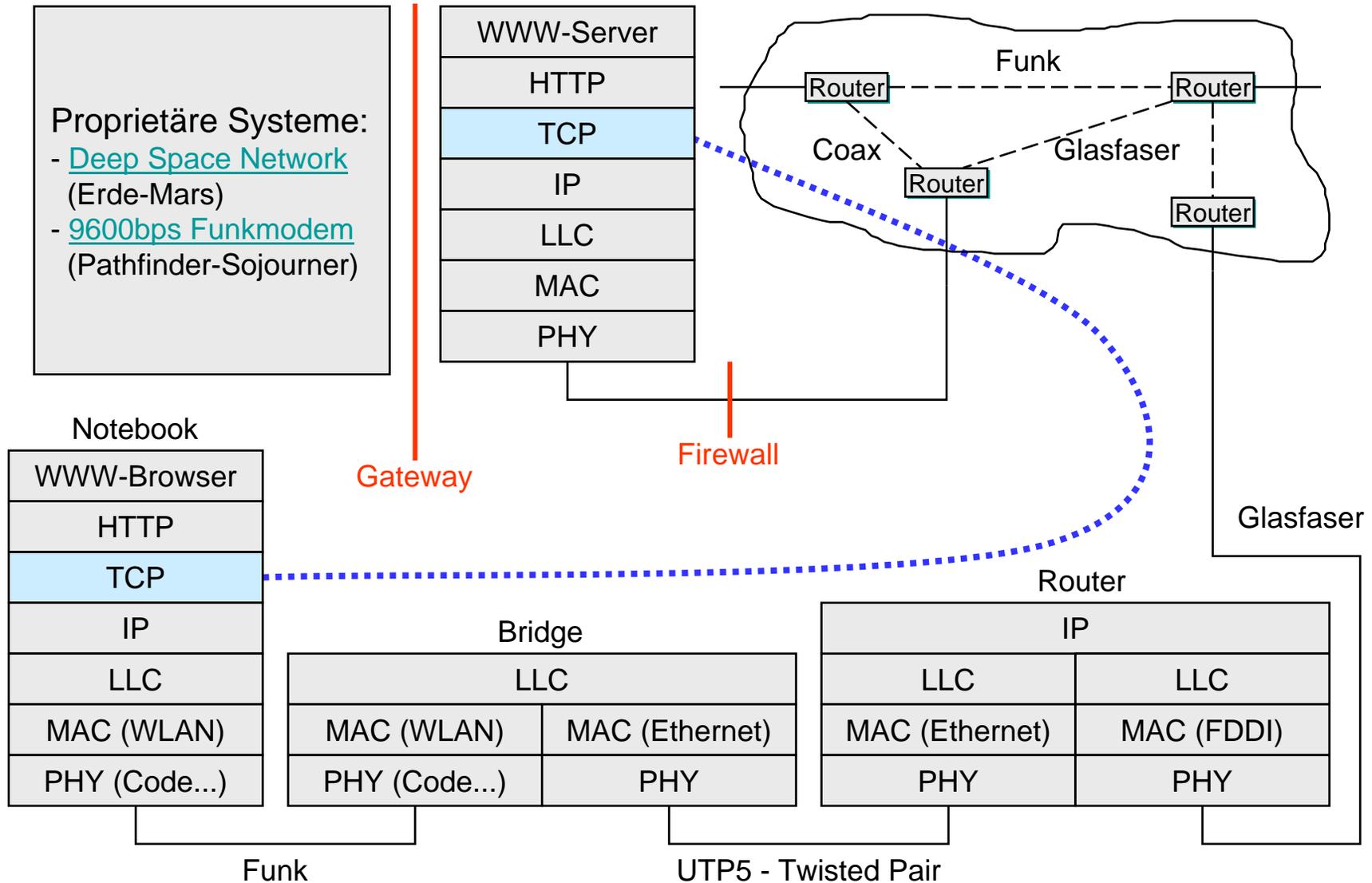
- **Ende-zu-Ende-Verbindung**
(Teilnehmer-zu-Teilnehmer statt Rechnerknoten-zu-Rechnerknoten)



- Isolierung der höheren Schichten von der Technologie, der Struktur und den Unvollständigkeiten der verwendeten Subnetze.
- Transparente Übertragung der Nutzdaten.
- Wahlmöglichkeiten für die Dienstgüte.
- Unabhängige Teilnehmeradressierung: globaler Adressraum für Teilnehmer, unabhängig von Adressen der unteren Schichten.
- **Ziel:** Effizienter und zuverlässiger Dienst soll angeboten werden



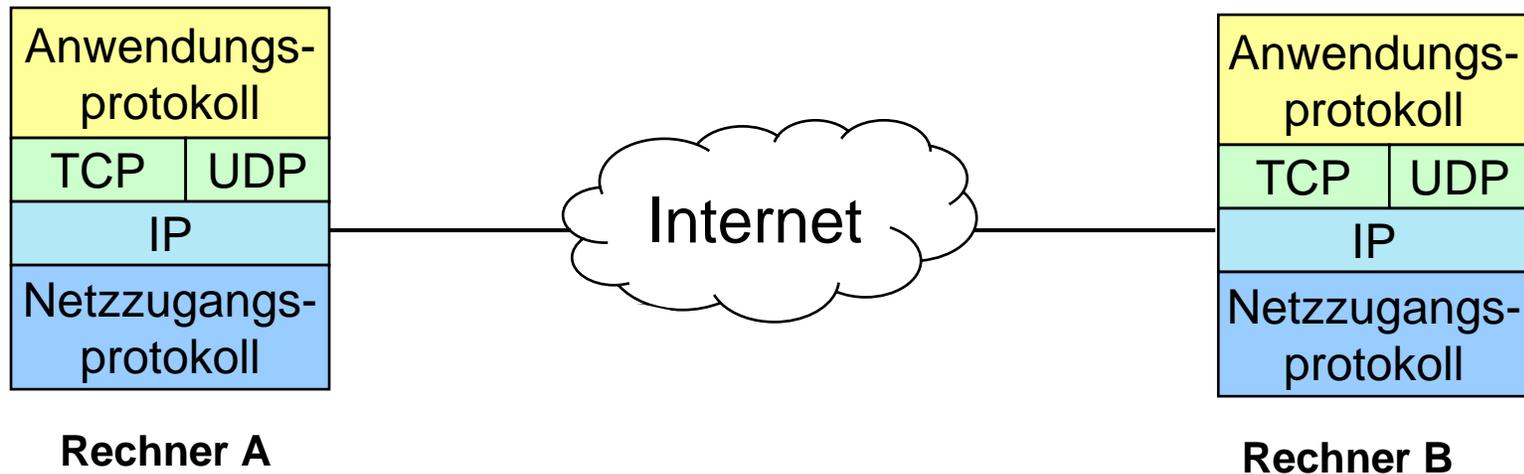
6.2.1. Ende-zu-Ende Kommunikation im Internet





Die Transportschicht im Internet

- Im Internet kommen auf Transportebene derzeit hauptsächlich zwei Protokolle zum Einsatz:
 - **TCP** (Transmission Control Protocol): Zuverlässiges, verbindungsorientiertes Transportprotokoll über unzuverlässigem IP
 - **UDP** (User Datagram Protocol): Verbindungsloses Transportprotokoll. Bietet eine Anwendungsschnittstelle zu IP, d.h. es verbessert den Dienst von IP nicht wesentlich.





6.2.2. TCP: Eigenschaften und Dienste (II)

□ Datenübertragung:

- Vollduplex
- Fehlerkontrolle durch Folgenummern (Sequenznummern), Prüfsumme, Quittierung, Übertragungswiederholung im Fehlerfall
- Reihenfolge, Flusskontrolle (durch Fenstermechanismus) und Staukontrolle
- Unterstützung von Sicherheitsstufen und Prioritäten
- Zeitbehaftete Daten: Falls die Auslieferung in einer bestimmten Zeit nicht möglich ist, wird der Dienstbenutzer informiert.

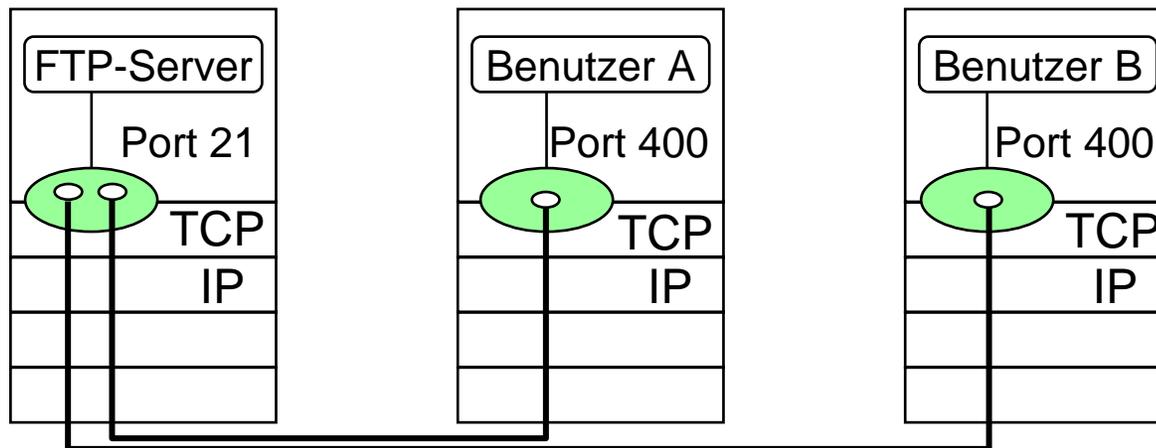
□ Fehleranzeige:

- Treten während der Verbindung Störungen auf, wird der Benutzer darüber in Kenntnis gesetzt.



TCP: Adressierung

- Identifikation von TCP-Diensten geschieht über Ports (TSAPs in der OSI-Terminologie)
- Portnummern bis 255 sind standardisiert ("well known ports") und für häufig benutzte Dienste reserviert (z.B. 21 für FTP, 23 für TELNET, 80 für HTTP)
- Ein FTP-Server ist z.B. über (IP-Adresse:Portnummer)129.13.35.7:21 erreichbar
- Socket: Kommunikationsendpunkt einer Kommunikationsbeziehung der Transportschicht, welche durch Fünftupel (Protokoll, lokale Adresse, lokale Portnummer, entfernte Adresse, entfernter Port) spezifiziert ist





TCP: fest vereinbarte port-Nummern (well-known ports)

Viele Anwendungen wählen TCP als Protokoll, allerdings muss der richtige *port* gewählt werden, um auf der Gegenseite mit der richtigen Anwendung zu kommunizieren.

- 13: Tageszeit
- 20: FTP Daten
- 25: SMTP
(Simple Mail Transfer Protocol)
- 53: DNS
(Domain Name Server)
- 80: HTTP
(Hyper Text Transfer Protocol)
- 119: NNTP
(Network News Transfer Protocol)

```
> telnet walapai 13
Trying 129.13.3.121...
Connected to leonis.
Escape character is '^]'.
Mon Aug 4 16:57:19 2002
Connection closed by foreign host
```

```
> telnet mailhost 25
Trying 129.13.3.161...
Connected to mailhost .
Escape character is '^]'.
220 mailhost ESMTP Sendmail 8.8.5/8.8.5; Mon,
4 Aug 2002 17:02:51 +0200
HELP
214-This is Sendmail version 8.8.5
214-Topics:
214-   HELO      EHLO      MAIL      RCPT      DATA
214-   RSET      NOOP      QUIT      HELP      VRFY
214-   EXPN      VERB      ETRN      DSN
214-For more info use "HELP <topic>".
...
214 End of HELP info
```



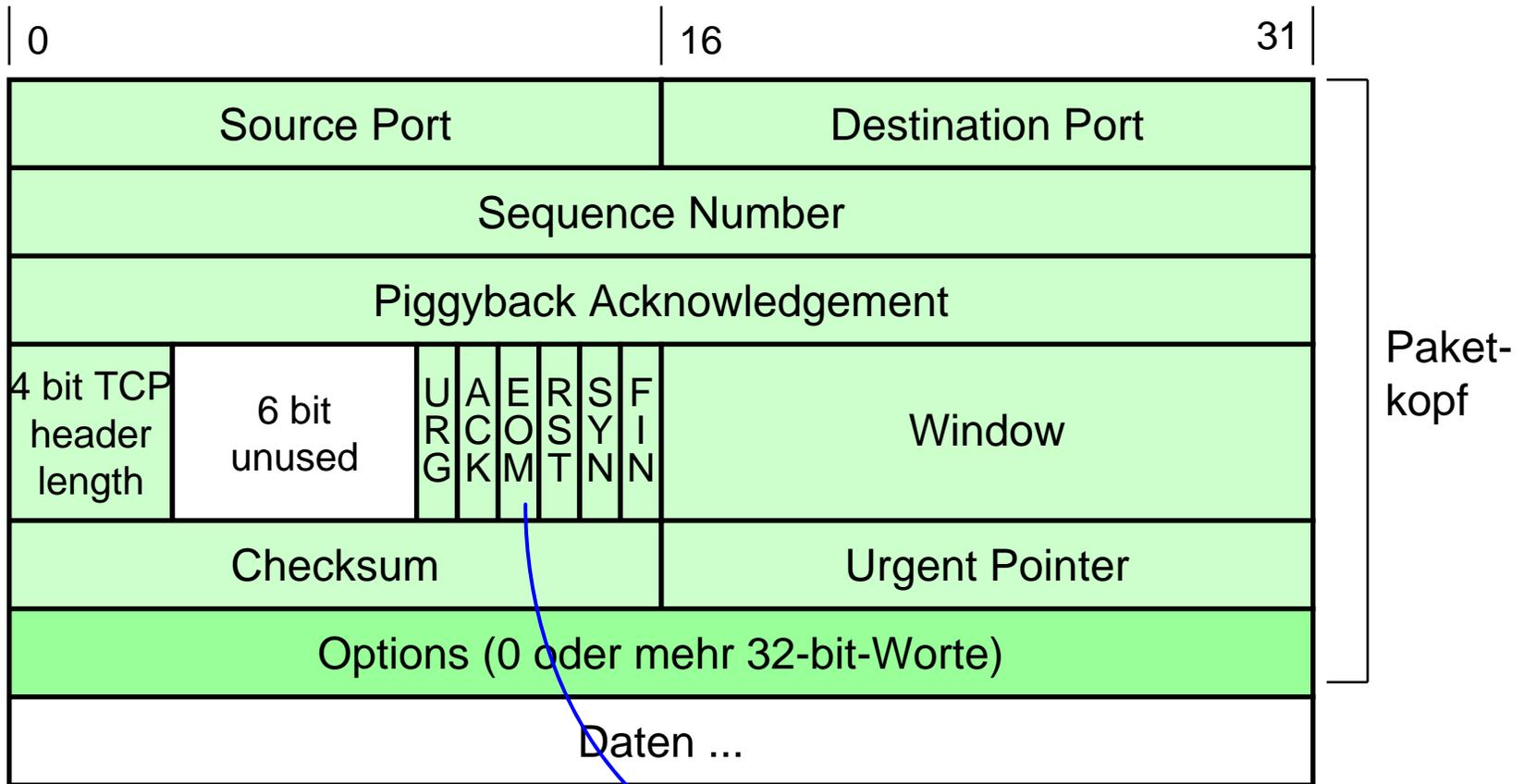
TCP: Verbindungsaufbau

- Verbindungen können nach der Erstellung eines Sockets **aktiv** (connect) oder **passiv** (listen/accept) aufgebaut werden.
 - Aktiver Modus: Anforderung einer TCP-Verbindung mit dem spezifizierten Socket.
 - Passiver Modus: Ein Dienstnutzer informiert TCP, dass er auf eine eingehende Verbindung wartet.
 - Spezifikation eines speziellen Sockets, von dem er eine eingehende Verbindung erwartet wird (fully specified passive open) oder
 - Alle Verbindungen annehmen (unspecified passive open).
 - Geht ein Verbindungsaufbauwunsch ein, wird ein neuer Socket erzeugt, der dann als Verbindungsendpunkt dient.

- Anmerkung:
 - Die Verbindung wird von den TCP-Instanzen ohne weiteres Eingreifen der Dienstbenutzer aufgebaut (es existiert z.B. kein Primitiv, das T-CONNECT.Rsp entspricht).

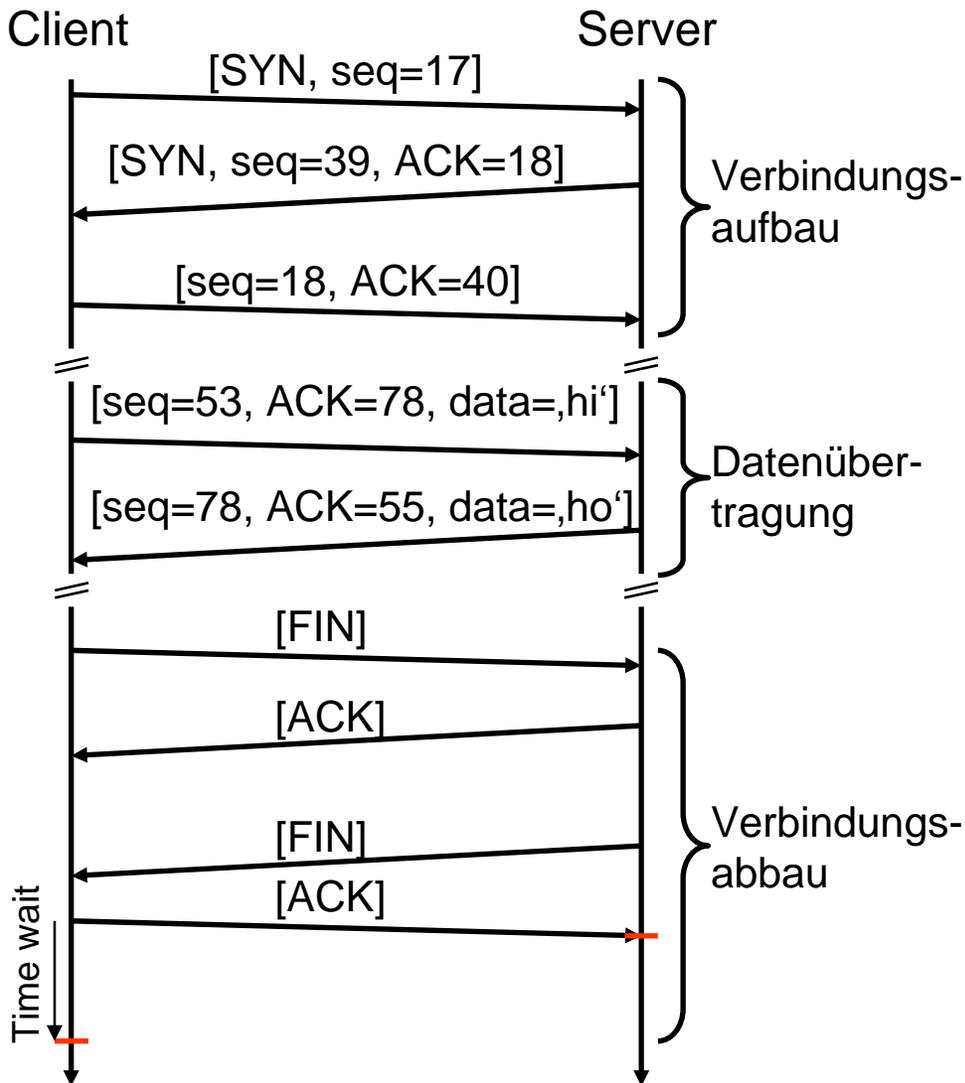


6.2.2.1. TCP-Paketformat: Aufbau





TCP-Verbindungsaufbau/Datenübertragung/ Verbindungsabbau

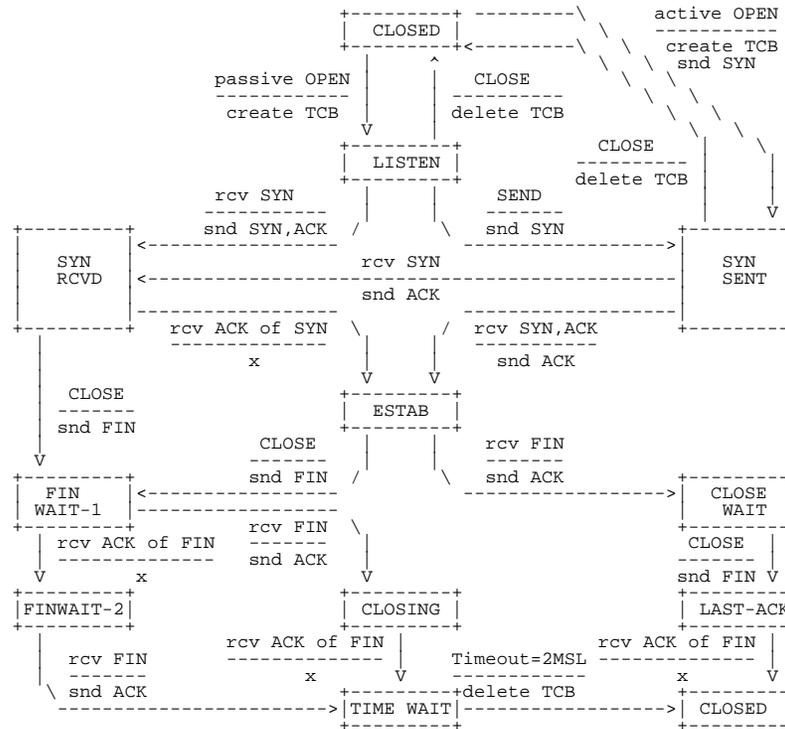


- Verbindungsaufbau
 - 3-Wege-Handshake
 - Aushandlung von Fenstergröße, Sequenznummern
- Datenübertragung
 - Bestätigung im Huckepack
- Verbindungsabbau
 - Bestätigt
 - Ressourcenfreigabe auf Client-Seite erst nach time-wait (frozen reference), typisch sind 30 s, 1 min, 2 min – beeinflusst stark die Leistungsfähigkeit!



TCP-Zustandsübergangsdiagramm

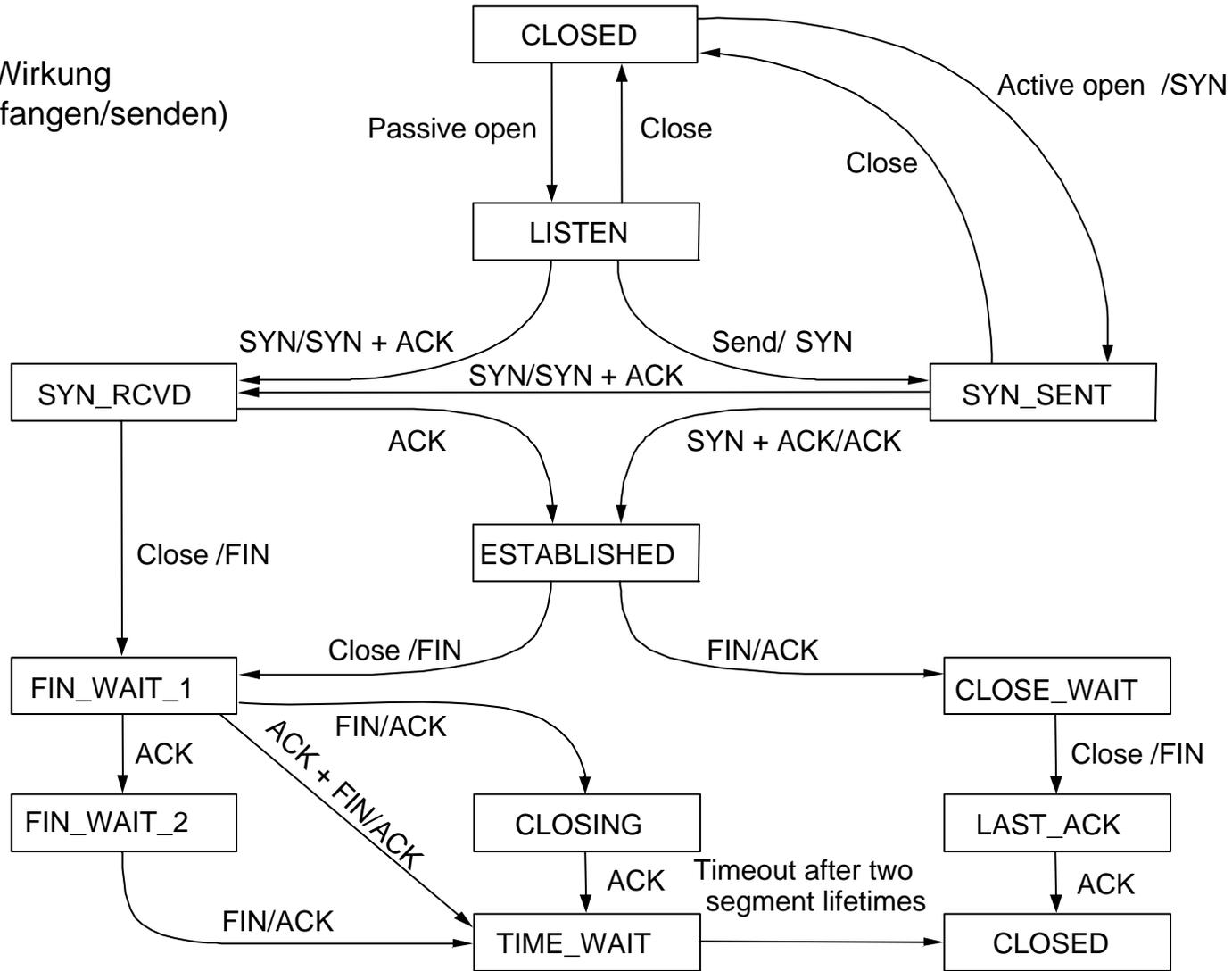
□ RFC 793





TCP-Zustandsübergangsdiagramm

Ursache/Wirkung
(z.B. empfangen/senden)





6.2.2.2. TCP: Mechanismen (I)

- Einheit der Datenübertragung: **Segment** (TCP-Header + Nutzdaten)
 - Größenbeschränkung durch max. IP-Nutzdatengöße von 65536 Byte
 - in Praxis: Größe von mehreren tausend Byte, um Fragmentierung auf IP-Ebene zu vermeiden

- **Mechanismen:**
 - Verwendung von **Timern:**
 - z.B. **Retransmission-Timer:** Wird beim Senden eines Segments gestartet
 - Übertragungswiederholung, falls keine Bestätigung vor Ablauf des Timers
 - komplexe Berechnung des Timer-Wertes
 - Verwendung des **Sliding-Windows-Verfahrens:**
 - Fenstergröße variabel, wird dem Sender im Feld *Window* mitgeteilt (auch Receiver Window genannt)
 - maximal 16 Bit ($2^{16}=65536$ Byte)
 - unzureichend für Hochleistungsnetze \Rightarrow Fensterskalierung bis 2^{32} Byte

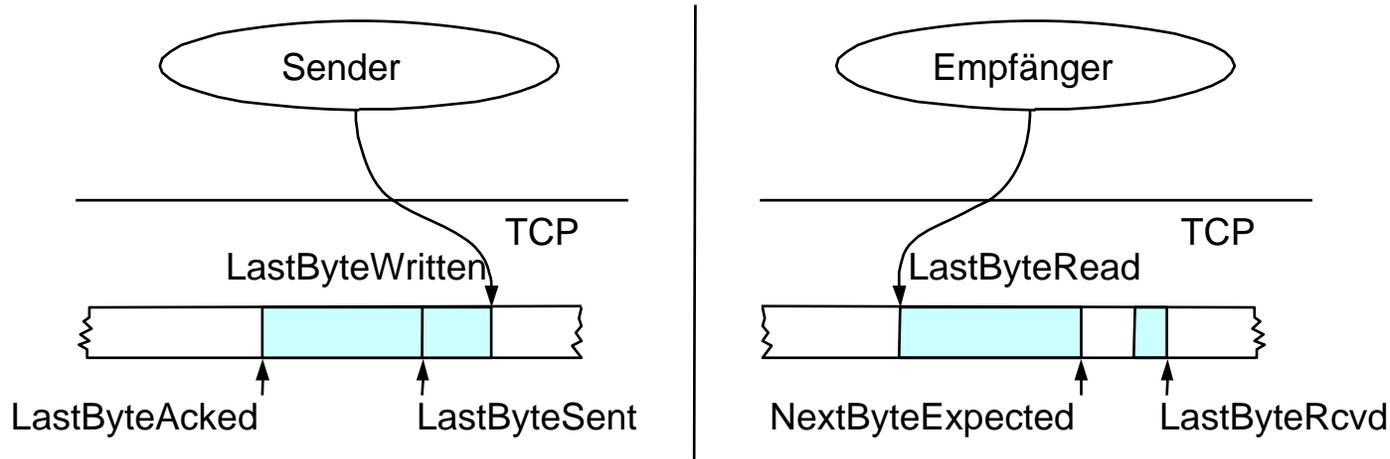


TCP: Mechanismen (II)

- Piggybacking (jeweils 16 Bit Sequenznummer)
- Prüfsummenbildung
 - Sicherung von TCP-Header, Nutzdaten und TCP-Pseudoheader
 - TCP-Pseudoheader = IP-Quell-/Zieladresse, IP-Protocol-Feld (6), TCP-Segmentgröße
- Ähnlich wie Go-Back-N
 - Bestätigungsnr. n bestätigt Empfang aller Bytes bis Seq.-Nr. $n-1$
 - Übertragungswiederholung aufgrund Ablauf von Retransmission Timer
 - Pakete, die in der falschen Reihenfolge ankommen, werden am Empfänger zwischengespeichert (Unterschied zu Go-Back-N).
- Aushandlung bei Verbindungsaufbau:
 - Selective-Repeat (RFC 1106)
 - durch NAK kann fehlerhaftes Segment explizit angefordert werden
 - Selective Acknowledge (1996, RFC 2018)
 - durch SACK werden einzelne, korrekt empfangene Segmente bestätigt
- Flusssteuerung und Staukontrolle
 - Siehe Kapitel 9



Sliding Window – Prinzip in TCP



Sender

- ❑ $\text{LastByteAcked} \leq \text{LastByteSent}$
- ❑ $\text{LastByteSent} \leq \text{LastByteWritten}$
- ❑ Puffern aller Daten zwischen LastByteAcked und LastByteWritten

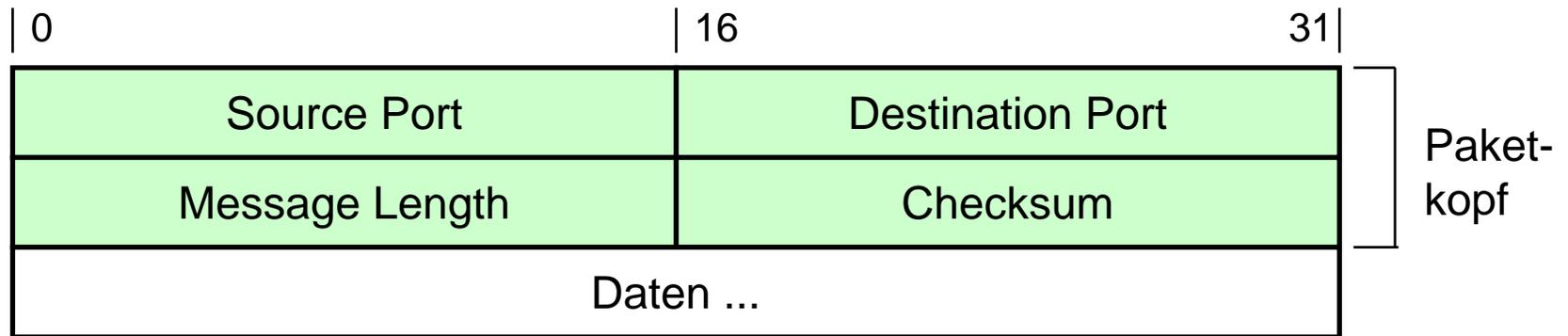
Empfänger

- $\text{LastByteRead} < \text{NextByteExpected}$
- $\text{NextByteExpected} \leq \text{LastByteRcvd} + 1$
- Puffern aller Daten zwischen NextByteRead und LastByteRcvd



6.2.3. UDP (User Datagram Protocol)

- ❑ Unzuverlässig, verbindungslos, einfacher und schneller als TCP
- ❑ Demultiplexing der empfangenen Pakete basiert auf der Port-Nummer
- ❑ Optionale Prüfsumme



- ❑ festgelegte, sog. „well-known“ ports:
 - 13: daytime
 - 53: domain name server
 - 123: network time protocol
- ❑ sehr viele Multimedia-Anwendungen nehmen UDP statt TCP wegen Leistungsvorteilen