Karlsruher Institut für Technologie

Vorlesung Einführung in Rechnernetze
8. Die Transportschicht

Prof. Dr. Martina Zitterbart

Dipl.-Inform. Martin Florian, Markus Jung (M.Sc.), Matthias Flittner (M.Sc.)

[zitterbart | florian | m.jung | flittner]@kit.edu

Institut für Telematik, Prof. Zitterbart

Kapitelübersicht

1. Einführung
2. Netzwerkarchitekturen
3. Physikalische Grundlagen
4. Protokollmechanismen
5. Die Sicherungsschicht: HDLC
6. Die Sicherungsschicht: Lokale Netze
7. Netzkopplung und Vermittlung
8. Die Transportschicht
9. Sicherheit
10. Anwendungssysteme

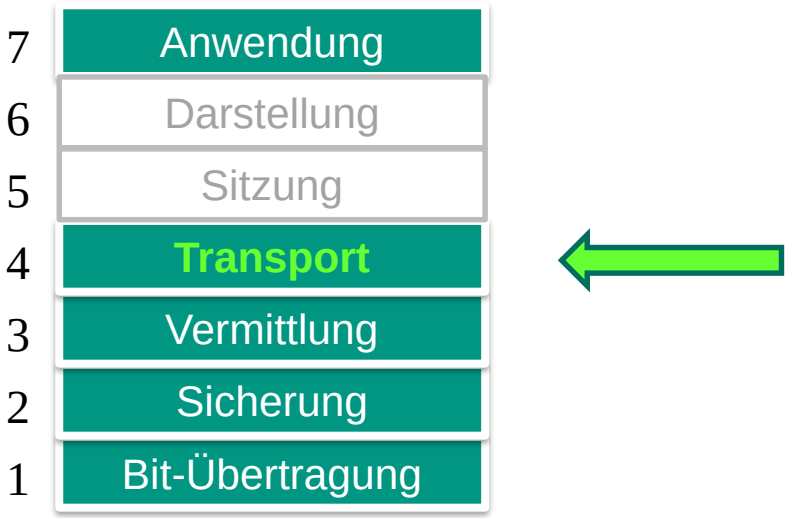
1. Einführung
2. Transportprotokolle im Internet
3. User Datagram Protocol (UDP)
4. Transmission Control Protocol (TCP)
5. Zusammenspiel der Protokolle

1. Einführung
2. Netzwerkarchitekturen
3. Physikalische Grundlagen
4. Protokollmechanismen
5. Die Sicherungsschicht: HDLC
6. Die Sicherungsschicht: Lokale Netze
7. Netzkopplung und Vermittlung
8. Die Transportschicht
9. Sicherheit
10. Anwendungssysteme

1. Einführung
2. Transportprotokolle im Internet
3. User Datagram Protocol (UDP)
4. Transmission Control Protocol (TCP)
5. Zusammenspiel der Protokolle

Einordnung

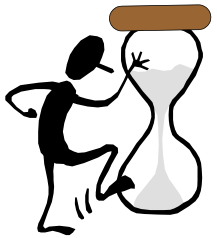
- Wir befinden uns auf Schicht 4 des OSI-Referenzmodells



8.1 Allgemeines

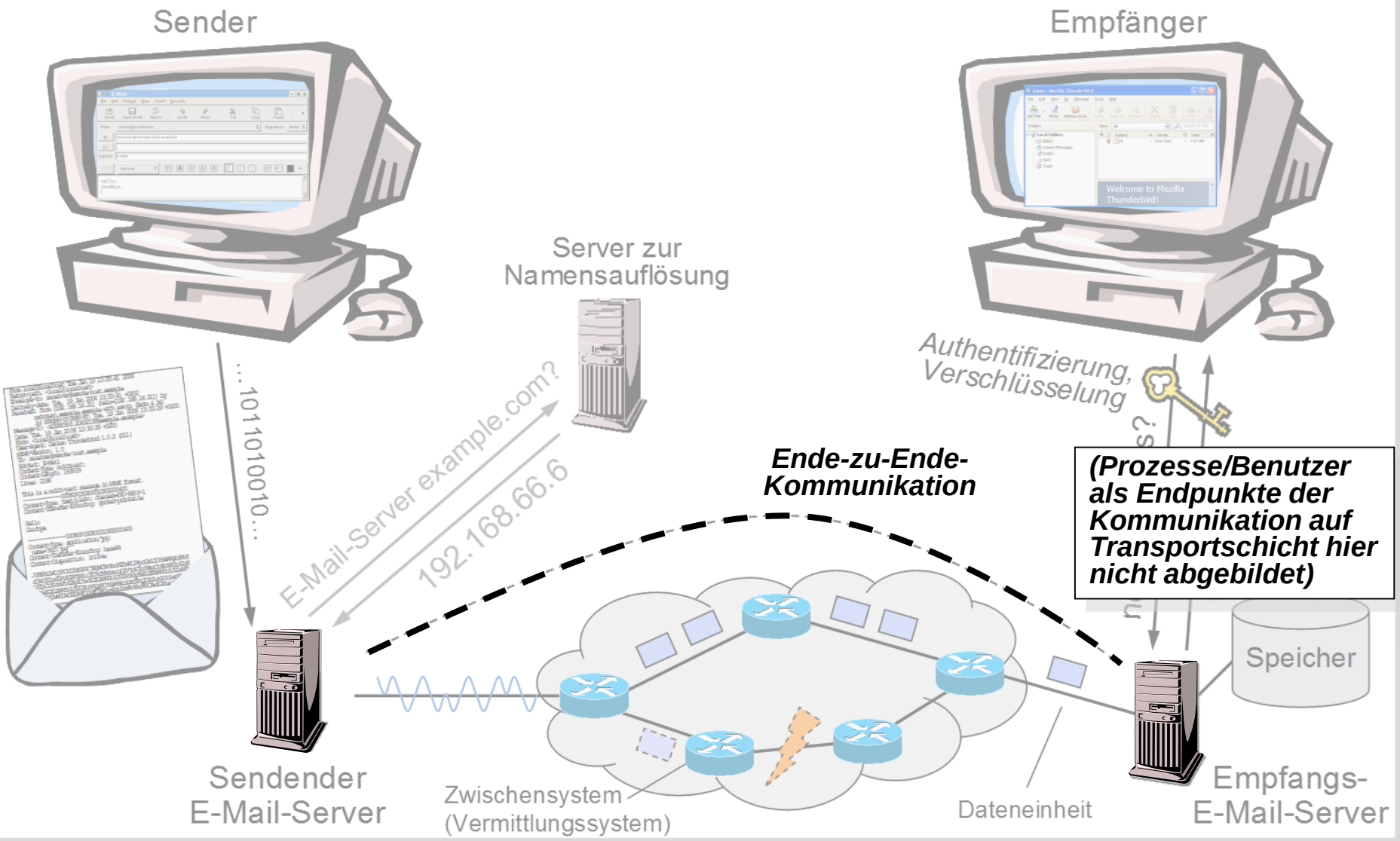
- Grundlegende Aufgabe
  - Verbergen von Details der transportorientierten Kommunikation vor den höheren Schichten (de-facto der Anwendung), z.B.
    - Technologien
    - Fehlercharakteristika

  … nicht alles lässt sich verbergen: Verzögerung bleibt „sichtbar“!



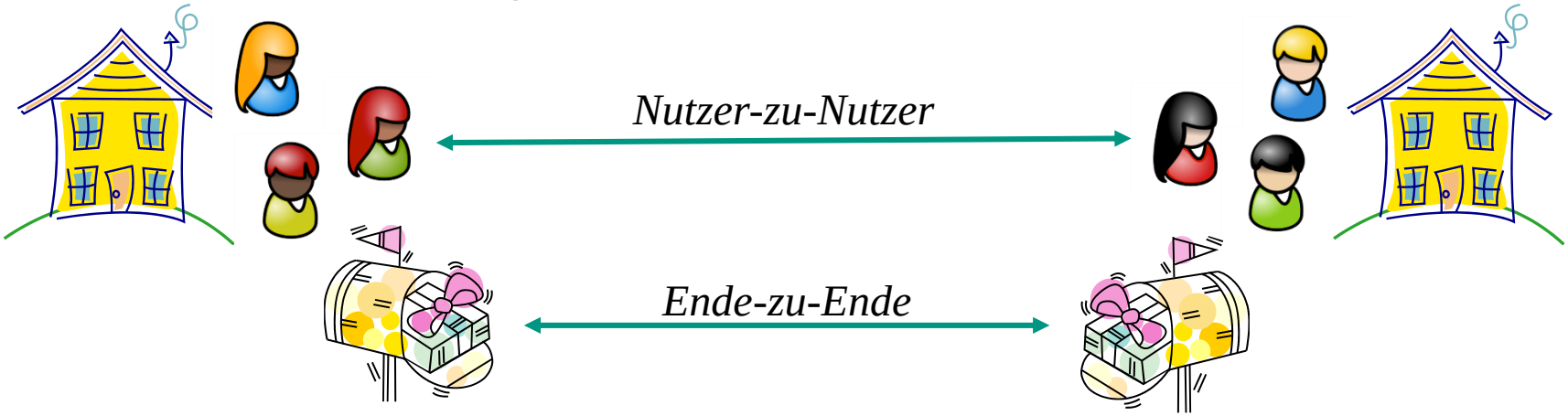
- Kommunikation
  - Logische Kommunikationsbeziehung zwischen Nutzern, i.d.R. Anwendungsprozessen
    - Nutzer-zu-Nutzer Kommunikation
  - Auf Schicht 3: Ende-zu-Ende Kommunikation
    - Kommunikation zwischen Endsystemen

Transport im Beispiel



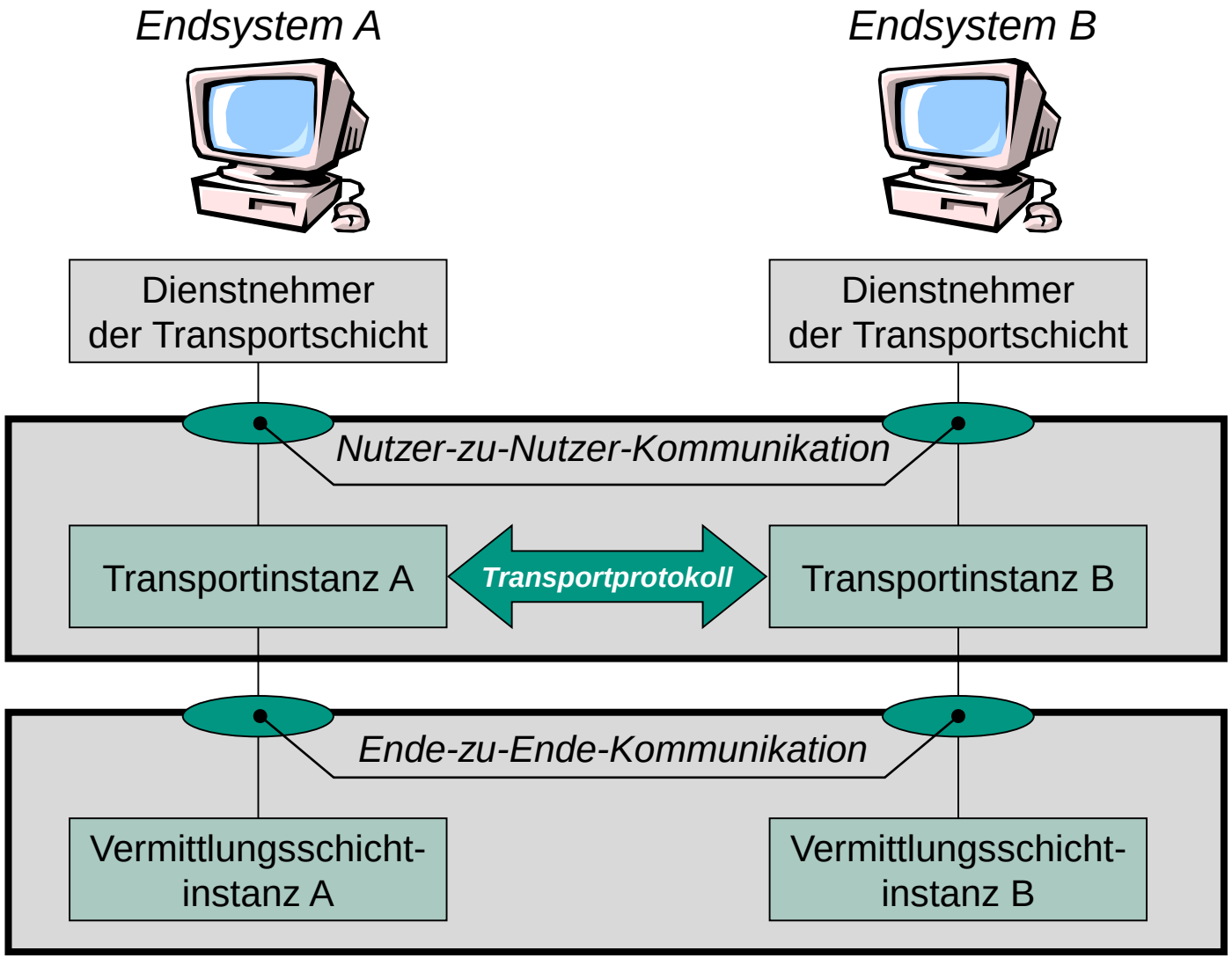
Nutzer-zu-Nutzer versus Ende-zu-Ende

- Anschauliches Beispiel
  - Briefkästen stehen weit vom Haus entfernt, so dass jeweils ein Mitglied der Großfamilie die Briefe sammelt, sie zum Briefkasten bringt bzw. wieder von diesem abholt.



  - Kommunikation zwischen Briefkästen
    - Ende-zu-Ende
  - Kommunikation zwischen Bewohnern der Häuser
    - Nutzer-zu-Nutzer

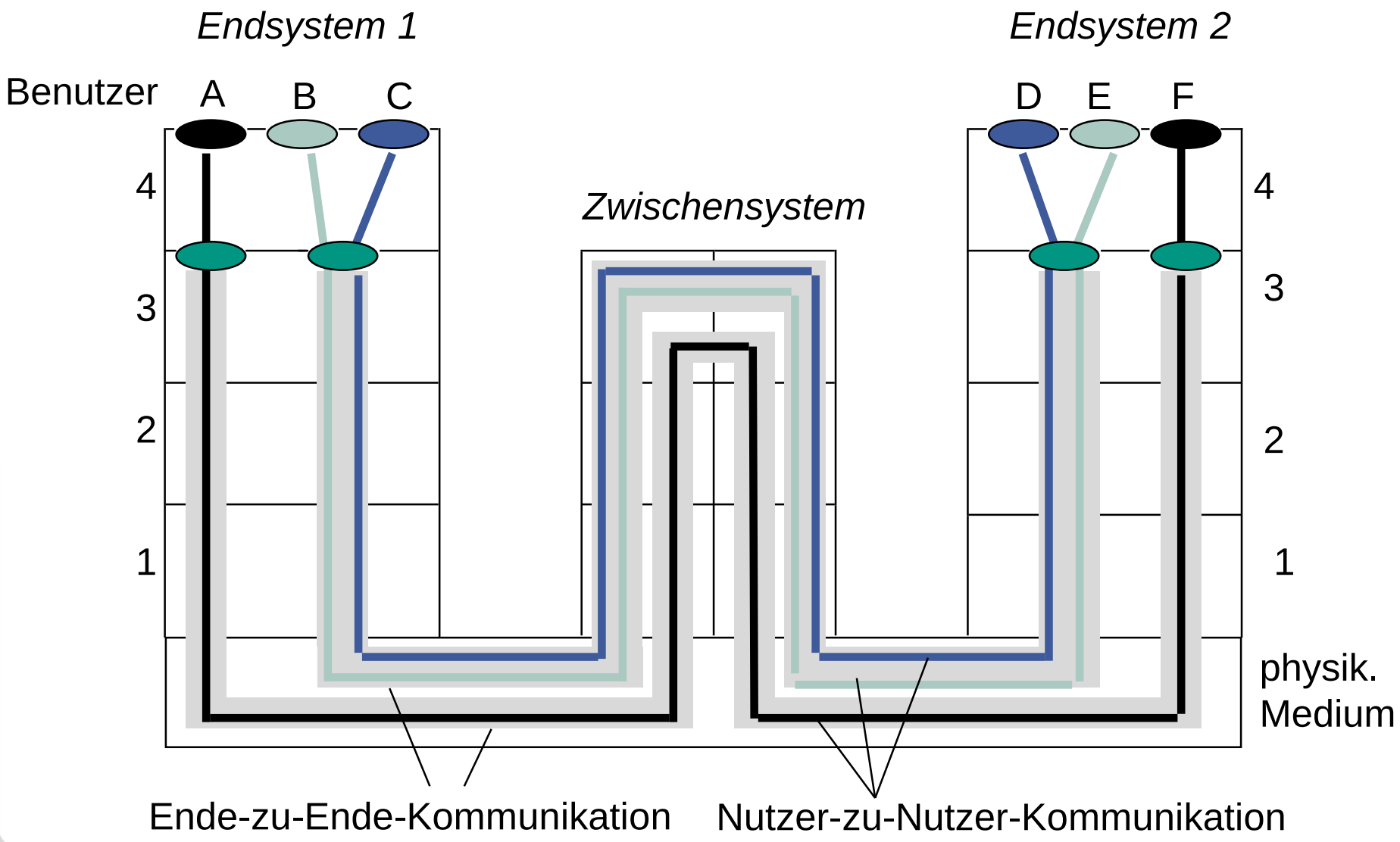
Nutzer-zu-Nutzer versus Ende-zu-Ende



Schicht 4: Transport-schicht

Schicht 3: Vermittlungs-schicht

Nutzer-zu-Nutzer versus Ende-zu-Ende



Pingo

- Pingo-Link für diese Vorlesung:

→ http://pingo.upb.de/6466

Pingo

http://pingo.upb.de/

1. Einführung
2. Netzwerkarchitekturen
3. Physikalische Grundlagen
4. Protokollmechanismen
5. Die Sicherungsschicht: HDLC
6. Die Sicherungsschicht: Lokale Netze
7. Netzkopplung und Vermittlung
8. Die Transportschicht
9. Sicherheit
10. Anwendungssysteme

1. Einführung
2. Transportprotokolle im Internet
3. User Datagram Protocol (UDP)
4. Transmission Control Protocol (TCP)
5. Zusammenspiel der Protokolle

8.2 Transportprotokolle im Internet

- Zwei grundlegende Transportprotokolle
  - UDP (User Datagramm Protocol)
    - Bietet verbindungslosen, unzuverlässigen Transportdienst
  - TCP (Transmission Control Protocol)
    - Bietet verbindungsorientierten, zuverlässigen Transportdienst

- Weitere Transportprotokolle
  - vor allem im Forschungsbereich z.B. zur
    - Unterstützung von Anforderungen multimedialer Anwendungen
    - Unterstützung von Gruppenkommunikation
    - Unterstützung von Signalisierung
    - Unterstützung von mehreren Pfaden
    - …

… in der Vorlesung „Next Generation Internet“ näher betrachtet

Adressierung auf Transportschicht

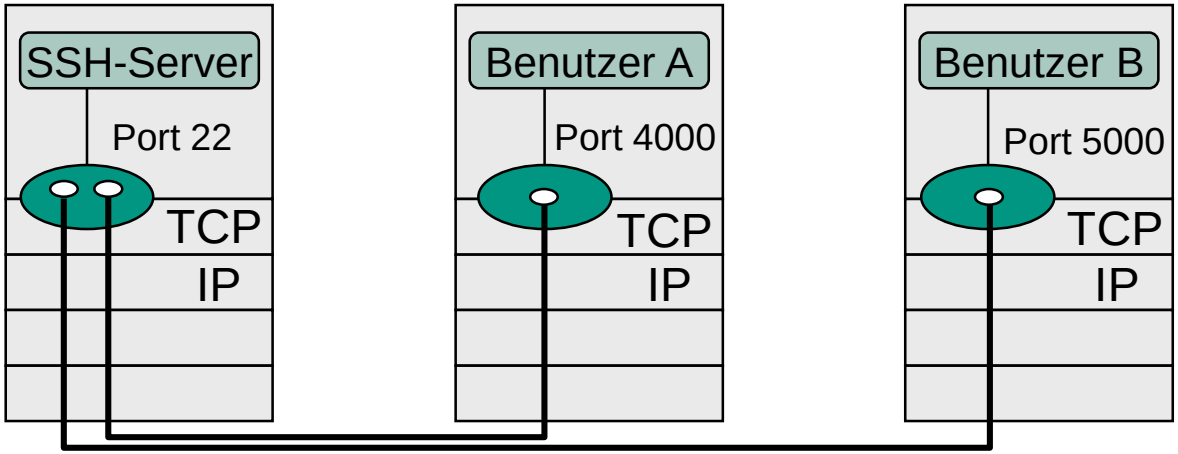
- Ziel
  - Eindeutige Kennzeichnung der Nutzer (i.d.R. Anwendungsprozesse)
- Ports
  - Adressen der Transportschicht
  - Unstrukturierte Nummer der Länge 16 Bit
    - Werte: 0 … 65535
  - Viele Portnummern kleiner 1024 für häufig benutzte Anwendungen reserviert
    - z.B. 21 für FTP, 22 für SSH, 23 für Telnet, 80 für HTTP

… Internet-weit eindeutige Adressierung eines Anwendungsprozesses

- IP-Adresse + Port

Adressierung auf Transportschicht

- Beispiel
  - SSH-Server am Telematik-Institut
  - Erreichbar über 141.3.70.4:22
    - <IP-Adresse>:<Port>



Port-Nummern-Konventionen (well-known ports)

- Von einer Anwendung muss der richtige *Port* gewählt werden, um auf der Gegenseite mit der gewünschten 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 osiris 13
Trying 129.13.3.121...
Connected to osiris.
Escape character is '^]'.
Mon Aug  4 16:57:19 1997
Connection closed by foreign host
```

„well-known“ und traditionell

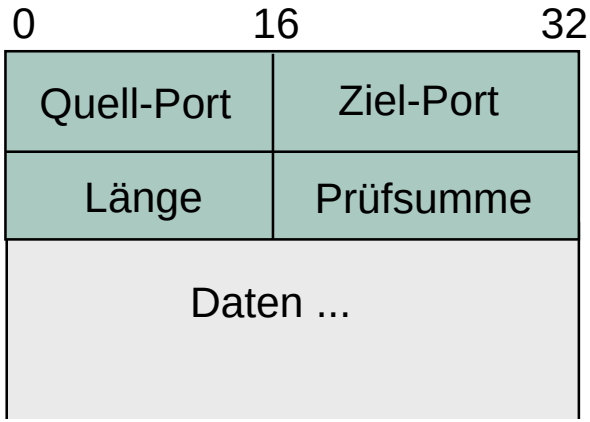
```
> telnet sokrates 25
Trying 129.13.3.161...
Connected to sokrates .
Escape character is '^]'.
220 sokrates ESMTP Sendmail 8.8.5/8.8.5;
Mon, 4 Aug 1997 17:02:51 +0200
HELP
214-This is Sendmail version 8.8.5
214-Topics:
214-    HELO    EHLO    MAIL    RCPT    DATA
214-    RSET    NOOP    QUIT    HELP    VRFY
214-    EXPN    VERB    ETRN    DSN
214-For more info use "HELP <topic>".
...
214 End of HELP info
```

1. Einführung
2. Netzwerkarchitekturen
3. Physikalische Grundlagen
4. Protokollmechanismen
5. Die Sicherungsschicht: HDLC
6. Die Sicherungsschicht: Lokale Netze
7. Netzkopplung und Vermittlung
8. Die Transportschicht
9. Sicherheit
10. Anwendungssysteme

1. Einführung
2. Transportprotokolle im Internet
3. User Datagram Protocol (UDP)
4. Transmission Control Protocol (TCP)
5. Zusammenspiel der Protokolle

8.3 UDP (User Datagram Protocol)

- Sehr einfaches Transportprotokoll
  - RFC 768, August 1980
- Eigenschaften
  - Multiplexen / Demultiplexen von Dateneinheiten für Anwendungsprozesse
  - Überprüfung von Bitfehlern: Prüfsumme
  - Keine Verbindung
  - Keine Zuverlässigkeit
- UDP-Dateneinheit



[RFC768]

Zu den Eigenschaften von UDP

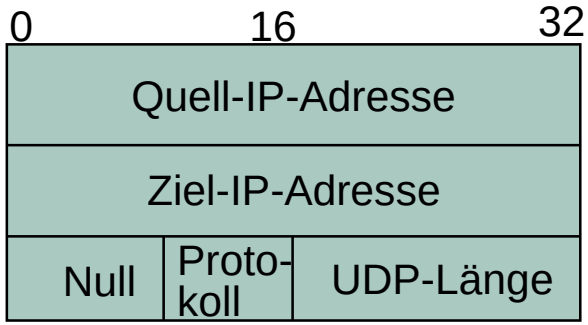
- Keine Verbindungsaufbauphase
  - Daten können sofort gesendet werden
    - Verzögerung zum Aufbau einer Verbindung entfällt
- Kein Verbindungszustand
  - Keine verbindungsrelevanten Informationen im Endsystem
    - z.B. Flusskontrollfenster, Sequenznummern, Timer
  - Skaliert z.B. für Server besser
    - Können mittels UDP typischerweise mehr Kommunikations-beziehungen unterstützen als mit TCP
- Geringer Overhead in der Dateneinheit
  - 8 Byte: Ports, Längenfeld und Prüfsumme
- Unreguliertes Senden
  - UDP kann Daten so schnell senden wie sie von der Anwendung geliefert werden und wie sie vom Netz abgenommen werden
    - Verluste durch Überlastungen im Netz möglich
- Vorteil
  - Sehr einfaches Protokoll mit äußerst geringem Overhead

UDP-Prüfsumme

- UDP-Prüfsumme wird berechnet über
  - UDP-Kopf (Prüfsummenfeld ist initial mit 0 zu belegen)
  - UDP-Daten
  - Pseudoheader
- Gleicher Algorithmus wie bei IP
  - Einer-Komplement der Summe der Einer-Komplemente von 16-Bit-Worten
- Mit IPv4
  - Berechnung kann deaktiviert werden
- Mit IPv6
  - Berechnung verpflichtend
    - … keine Prüfsumme mehr im IPv6-Kopf

Pseudoheader

- Teil des IP-Kopfs
  - Quell-/Ziel-IP-Adresse
  - Protokoll
  - Länge der UDP-Dateneinheit (ohne Pseudoheader)
- Funktion
  - Schutz gegen fehlgeleitete UDP-Dateneinheiten
    - Überprüfung, ob Daten zwischen den beiden korrekten Endpunkten ausgetauscht wurden
- Optionale Prüfsumme
  - Prüfsummenfeld = 0 : keine Prüfsummenberechnung gewünscht
  - Bei berechneter Prüfsumme 0 wird `0xFFFF` übertragen

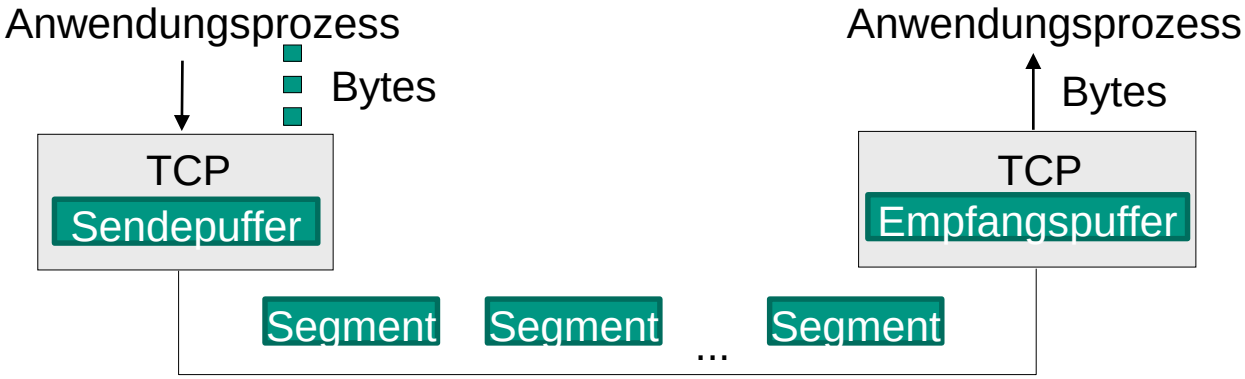


1. Einführung
2. Netzwerkarchitekturen
3. Physikalische Grundlagen
4. Protokollmechanismen
5. Die Sicherungsschicht: HDLC
6. Die Sicherungsschicht: Lokale Netze
7. Netzkopplung und Vermittlung
8. Die Transportschicht
9. Sicherheit
10. Anwendungssysteme

1. Allgemeines
2. Transportprotokolle im Internet
3. User Datagram Protocol (UDP)
4. Transmission Control Protocol (TCP)
5. Zusammenspiel der Protokolle

8.4 Transmission Control Protocol (TCP)

- TCP-Grundlagen
  - Stellt zuverlässigen Transportdienst zum Datentransfer zwischen Anwendungsprozessen zur Verfügung
    - RFC 793, September 1981
  - TCP erhält von der Anwendung einen Bytestrom und übergibt an IP TCP-Dateneinheiten, so genannte TCP-Segmente
  - In Anlehnung an Unix Pipes



[RFC793]

Bytestrom → TCP-Dateneinheit?

- Problem
  - Wann wird aus Bytestrom eine TCP-Dateneinheit an IP weitergegeben?
  - RFC 793
    - *„TCP should „send that data in segments at its own convenience“*

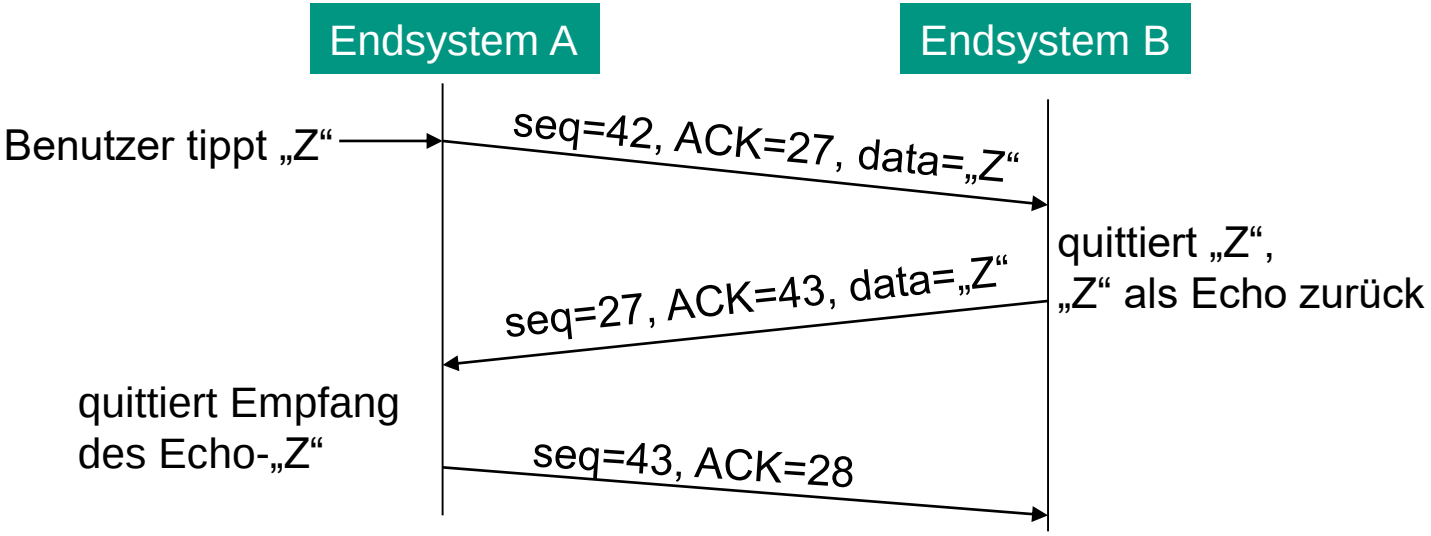
- Alternativen
  - MSS: Maximum Segment Size
    - Länge der Anwendungsdaten, nicht Länge der TCP-Dateneinheit
    - Typische Größen (vermeiden das Segmentieren durch IP)
      - 1460 Byte, 536 Byte, 512 Byte
  - Push (PSH-Flag im Kopf der TCP-Dateneinheit)
    - Sender verlangt sofortiges Versenden der Daten (Vorrangdaten)
    - Z.B. bei Telnet verwendet
  - Zeitgeber
    - Nach Zeitintervall der Inaktivität werden vorhandene Daten gesendet

Zu den Eigenschaften von TCP

- Verbindungsorientierter, zuverlässiger Dienst
  - Phasen: Verbindungsaufbau, Datentransfer, Verbindungsabbau
- Fehlerkontrolle
  - Sequenznummern, Prüfsumme, Quittierung, Sendewiederholungen im Fehlerfall
- Reihenfolgetreue Auslieferung der Daten
- Flusskontrolle
  - Ziel: Empfänger wird nicht überlastet
- Staukontrolle
  - Ziel: Netz wird nicht überlastet

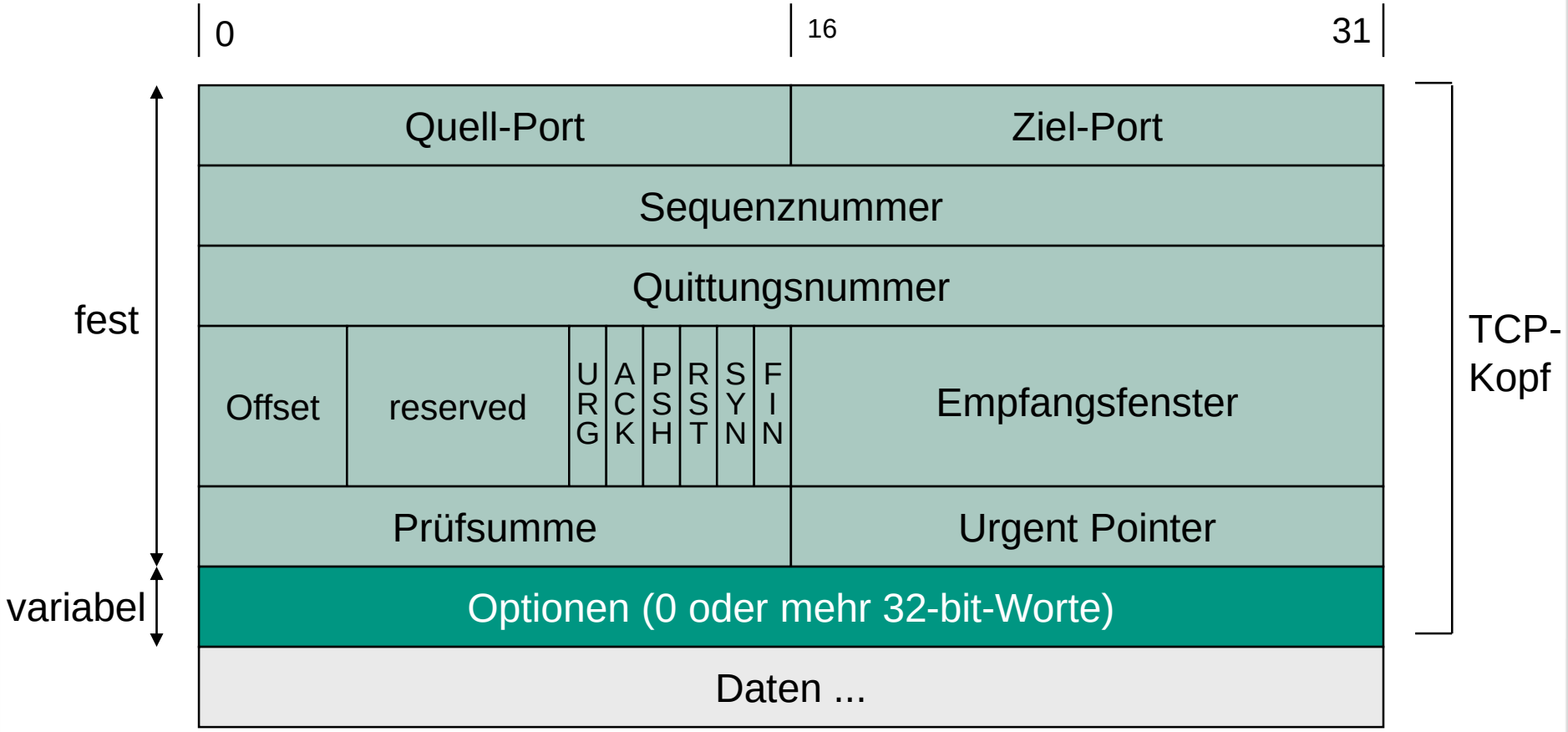
TCP-Sequenznummern und -Quittungen

- Sequenznummern
  - pro Byte, nicht pro Dateneinheit (erstes Byte in Dateneinheit)
  - Initiale Sequenznummer von Endsystemen zufällig gewählt
- Quittungen
  - positive kumulative Quittungen
  - Sequenznummer des nächsten Bytes, das vom Kommunikationspartner erwartet wird
- Beispiel (Telnet-Szenario)



Format einer TCP-Dateneinheit

- Als TCP-Segment bezeichnet



Felder einer TCP-Dateneinheit

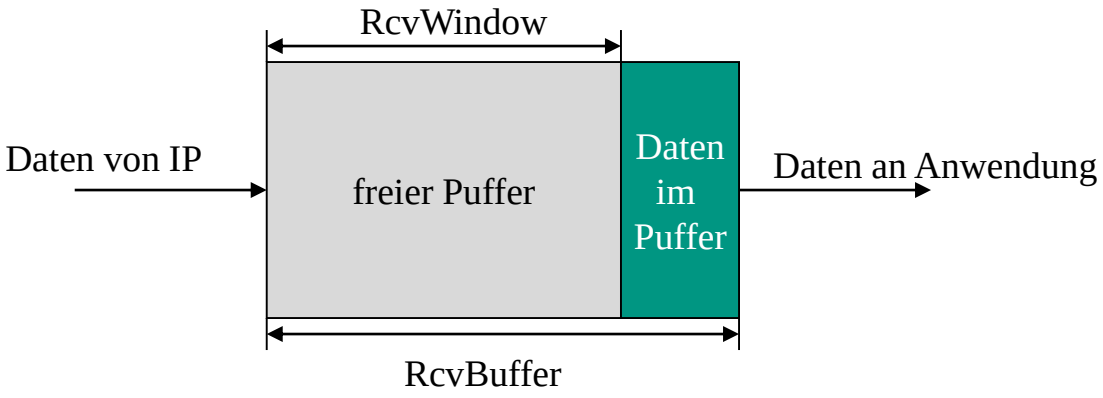
- Quell-Port und Ziel-Port
  - Identifizieren Endpunkte der Verbindung
- Sequenznummer
  - Gemessen in Byte (nicht pro Dateneinheit)
- Quittung
  - Die nächste vom Empfänger erwartete Sequenznummer
- Offset
  - Anzahl der 32-Bit-Wörter im TCP-Kopf
- URG
  - Wird auf 1 gesetzt, falls der Urgent Pointer verwendet wird
    - *… in der Regel nicht benutzt*
- SYN
  - Wird beim Verbindungsaufbau verwendet, um *Connection Request* (TConReq) bzw. *Connection Confirmation* (TConCnf) anzuzeigen
- ACK
  - Unterscheidet bei gesetztem SYN-Bit eine TConReq-PDU von einer TConCnf-PDU
  - Signalisiert die Gültigkeit des Quittungs-Feldes

Felder einer TCP-Dateneinheit

- FIN
  - Gibt an, dass der Sender keine Daten mehr senden möchte
- RST
  - Wird benutzt, um eine Verbindung zurückzusetzen
- PSH
  - Signalisiert, dass übergebene Daten sofort weitergeleitet werden sollen
  - Gilt sowohl für den Sender als auch für den Empfänger
  - Wird in der Regel nicht benutzt
- Empfangsfenster
  - Dient zur Flusskontrolle
- Prüfsumme
  - Enthält Prüfsumme über TCP-Kopf, Daten und Pseudoheader (wie bei UDP)
- Urgent-Zeiger
  - Relativer Zeiger auf wichtige Daten
- Das Optionen-Feld
  - kann Optionen variabler Länge aufnehmen $(n * 32\ \text{Bit})$

Flusskontrolle

- Ziel
  - Überlastung beim Empfänger vermeiden
- Vorgehensweise
  - Empfänger reserviert Pufferplatz pro Verbindung
    - Explizite Kreditvergabe
  - Variablen in der Empfängerinstanz
    - RcvBuffer: gesamter Pufferplatz für zu empfangende Daten
    - RcvWindow: momentan noch freier Pufferplatz
      - Empfangsfenster
      - Feld Empfangsfenster im Kopf der TCP-Dateneinheit



Flusskontrolle

- Szenario
  - Endsystem A schickt große Datei über TCP an Endsystem B
- Variablen beim *Empfänger*
  - LastByteRead
    - Letzte Sequenznummer, die von der Anwendung aus dem Empfangspuffer gelesen wurde
  - LastByteRcvd
    - Letzte Sequenznummer, die über das Netz empfangen und in den Empfangspuffer geschrieben wurde
  - Es muss immer gelten

    LastByteRcvd – LastByteRead <= RcvBuffer
  - Empfangsfenster

    RcvWindow = RcvBuffer – (LastByteRcvd – LastByteRead)

Flusskontrolle

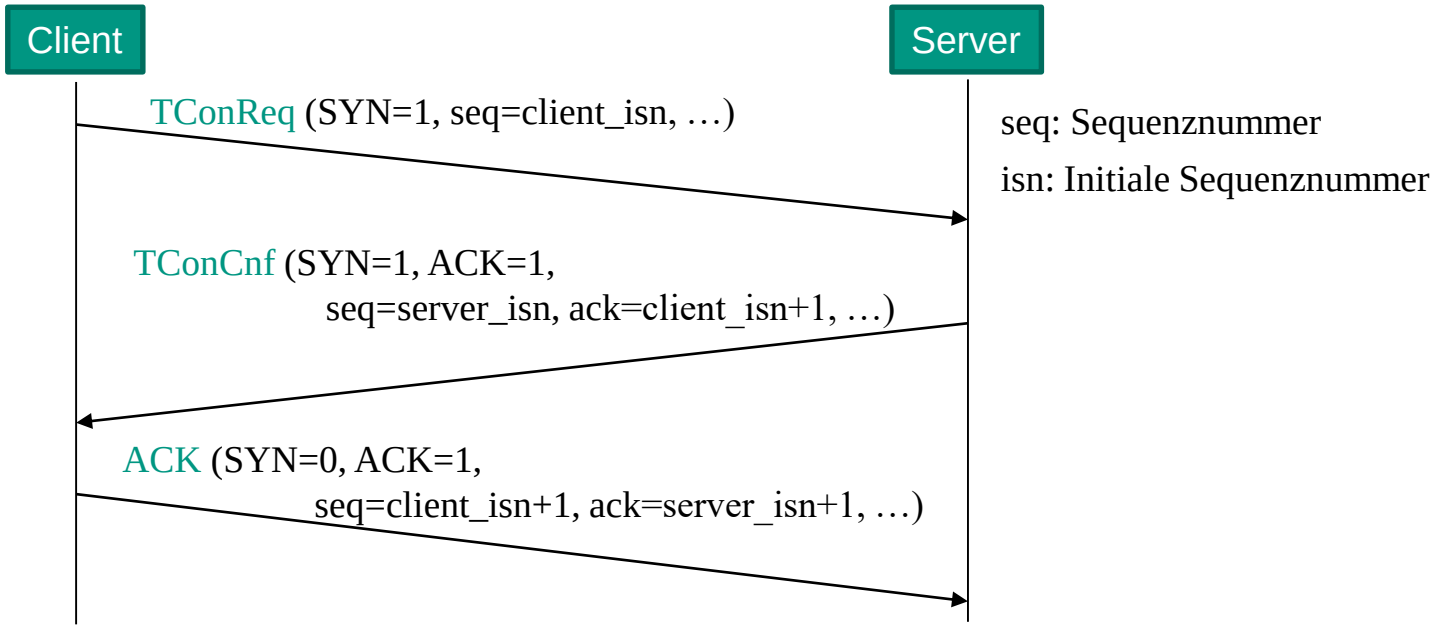
- Variablen beim *Sender*
  - LastByteSent
    - letzte Sequenznummer, die gesendet wurde
  - LastByteAcked
    - letzte Sequenznummer, die quittiert wurde

  - Es muss immer gelten

LastByteSent – LastByteAcked <= RcvWindow

TCP-Verbindungsverwaltung

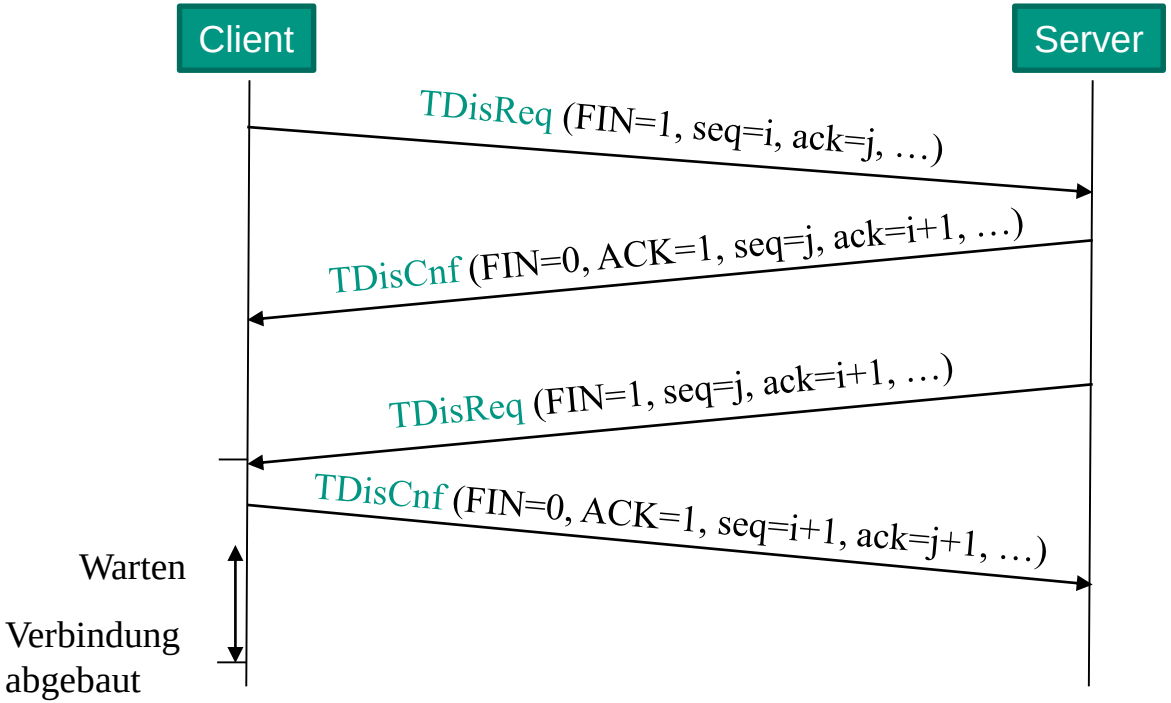
- Verbindungsaufbau
  - Client initiiert Verbindung und Server wartet auf Verbindungswünsche



  - Connection request/confirm führen keine Nutzdaten mit sich
  - 3-Wege-Handshake. Weshalb genügt 2-Wege-Handshake nicht?
  - Festlegung der initialen Sequenznummern (..._isn). Weshalb erforderlich?
  - Bekanntgabe der Größe des Flusskontrollfensters
  - Allokation der Puffer …
  - Ab wann dürfen Client und Server Daten senden?

TCP-Verbindungsverwaltung

- Verbindungsabbau
  - Kann sowohl vom Client als auch vom Server initiiert werden



  - Nach letztem ACK wird noch gewartet, bevor lokaler Kontext gelöscht wird
    - Zur Vermeidung der Zustandshaltung: senden von RST-Segment
    - Problem damit?
  - Weitere Varianten im Ablauf möglich

Verbindungsabbau

- Ordnungsgemäßer Abbau (vgl. vorangegangene Folie)
  - 4-Wege-Handshake
  - „Richtungen“ werden unabhängig voneinander geschlossen
    - FIN-, ACK- und FIN/ACK-Dateneinheiten
  - Lediglich eine Richtung geschlossen (simplex Datentransfer möglich)
    - Unidirektionaler Datenfluss
      - Neue Nutzdaten in „offener Richtung“ möglich
    - In „Gegenrichtung“
      - Kontrolldaten (z.B. ACK), keine neuen Nutzdaten

- Abbruch einer Verbindung
  - Reset (RST-Segment)
  - Verbindung wird unmittelbar geschlossen
    - Z.B. keine weiteren Sendewiederholungen
    - Kontext wird unmittelbar gelöscht

Pingo

http://pingo.upb.de/

Pingo

http://pingo.upb.de/

Stau im Internet!

… Netze werden schneller und Nutzung steigt

- Beobachtung
  - Drastischer Einbruch des Durchsatzes von TCP im Oktober 1986
    - Dann Serie so genannter „Staukollapse“
  - Stau im Netz … Puffer im Router füllen sich
    - Pufferüberlauf → Verwerfen von Dateneinheiten
      - … keine entsprechende Quittung vom Sender empfangen
      - … Sendewiederholungen durch den TCP-Sender
        → Verstärkung der Stausituation!

Staukontrolle

- Ziel
  - Vermeidung von Überlastsituationen im Netz
- Verfahren
  - Einführung eines Staukontrollfensters (CWnd, Congestion Window)
    - Beim Senden muss auch Empfangsfenster beachtet werden
      - … Minimum beider Fenstergrößen bestimmt maximal zu sendende Datenmenge

$$\text{LastByteSent} - \text{LastByteAcked} \leq \min\{\text{CWnd}, \text{RcvWindow}\}$$

  - … und eines Schwellenwertes (SSThresh)
  - Wie Stau erkennen?
    - Nutzung von Zeitgebern als Hilfsmittel für Stausignal
    - Ausbleibende Quittung … Vermutung einer Stausituation
  - Reaktion nach Erkennung?
    - Reduktion von CWnd
    - Langsames Erhöhen von CWnd … „herantasten“ an Netzkapazität

Ablauf TCP-Staukontrolle

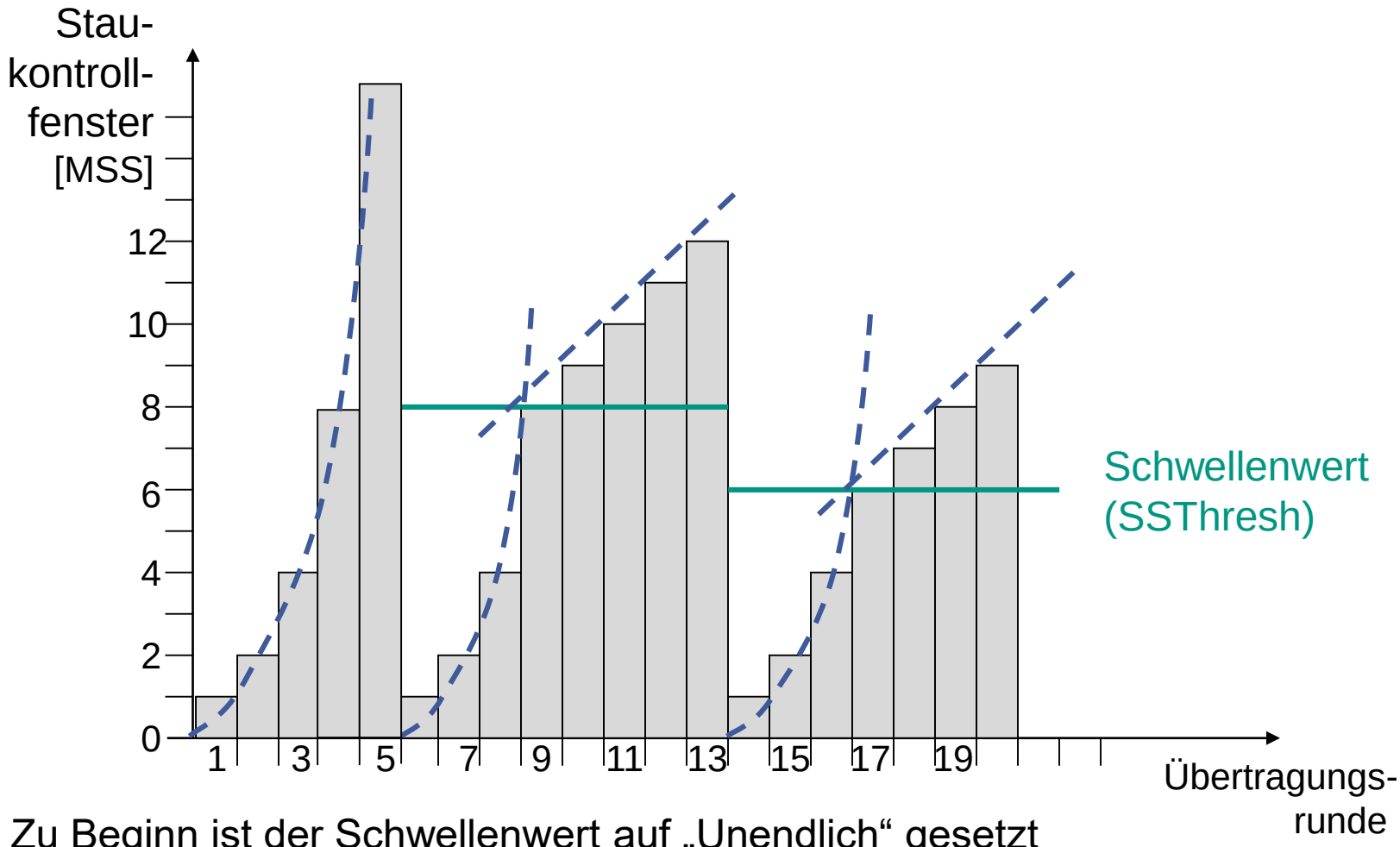
- Start: $\mathrm{CWnd} = 1\ MSS$ [Maximum Segment Size]
- $\mathrm{CWnd} \leq \mathrm{SSThresh}$ & Quittungen rechtzeitig empfangen: Slow-Start
  - Exponentielles Erhöhen des Staukontrollfensters
    - Bei jeder empfangenen Quittung: $\mathrm{CWnd} \mathrel{+}= 1$

- $\mathrm{CWnd} > \mathrm{SSThresh}$ & Quittungen rechtzeitig empfangen: Congestion Avoidance
  - Lineares Erhöhen des Staukontrollfensters
    - Erhöhen des Staukontrollfensters um 1 pro vollständig gesendetem und bestätigtem Sendefenster
    - Bei jeder empfangenen Quittung: $\mathrm{CWnd} \mathrel{+}= 1/\mathrm{CWnd}$

- Quittung nicht rechtzeitig empfangen (Timer abgelaufen)
  - Stau vermutet
  - $SSThresh = \max(FlightSize/2,\ 2 * MSS)$
    - FlightSize: Daten, die gesendet, aber noch nicht quittiert wurden
  - SCWnd zurücksetzen (neue Slow-Start-Phase): $\mathrm{CWnd} = 1\ \mathrm{MSS}$

Entwicklung des Staukontrollfensters

- Beispiel



- Zu Beginn ist der Schwellenwert auf „Unendlich“ gesetzt

Pingo

http://pingo.upb.de/

„TCP-Tools“

- Ziel
  - Beobachtung und Messung des Verhaltens bzw. der Leistungsfähigkeit von TCP
- Dazu erforderlich
  - Datensammlung
    - Erfassung und Sammlung von Daten
  - Datenanalyse
    - Weiterverarbeitung von Daten, beispielsweise um Eigenschaften, wie Minima, Maxima oder Durchschnittswerte zu berechnen
  - Visualisierung
    - Erzeugung von Hilfsmitteln wie Diagrammen oder Tabellen
  - Interpretation
    - Sinnvolle Schlussfolgerungen als Ziel der Beobachtung und Messung
- Kurze Vorstellung zweier nützlicher Tools im Kontext von TCP
  - Teilweise für verschiedene Betriebssysteme frei verfügbar


[HaJa03]

tcpdump

- Paketsniffer
  - Liest an einem Netzwerk-Interface, z.B. an einer Netzwerkkarte, alle empfangenen Dateneinheiten aus
  - Beispiel

```
$ tcpdump
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 96 bytes
11:37:41.729131 IP HostA.42980 > Server.www: S 62713440:62713440(0) win 5840
<mss 1460,sackOK,timestamp 8835394 0,nop,wscale 2>
11:37:41.732888 IP Server.www > HostA.42980: S 1853613617:1853613617(0) ack 62713441 win 5792
<mss 1460,sackOK,timestamp 1199883883 8835394,nop,wscale 0>
11:37:41.732907 IP HostA.42980 > Server.www: . ack 1 win 1460 <nop,nop,timestamp 8835395 1199883883>
11:37:41.729297 IP HostA.42980 > Server.www: P 1:112(111) ack 1 win 1460
<nop,nop,timestamp 8835395 1199883883>
11:37:41.729724 IP Server.www > HostA.42980: . ack 112 win 5792 <nop,nop,timestamp 1199883883 8835395>
```

  - Offenbar TCP-Verbindungsaufbau zu Port 80 eines Empfängers
    - 3-Wege Handshake (SYN, SYN/ACK, ACK)
  - Aller Wahrscheinlichkeit nach ein HTTP-Request eines Clients an einen WWW-Server
    - Tcpdump zählt die Sequenznummer nach erfolgreichem Verbindungsaufbau von 1 beginnend (relativ)

tcpdump

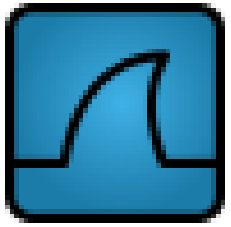
- Fortsetzung

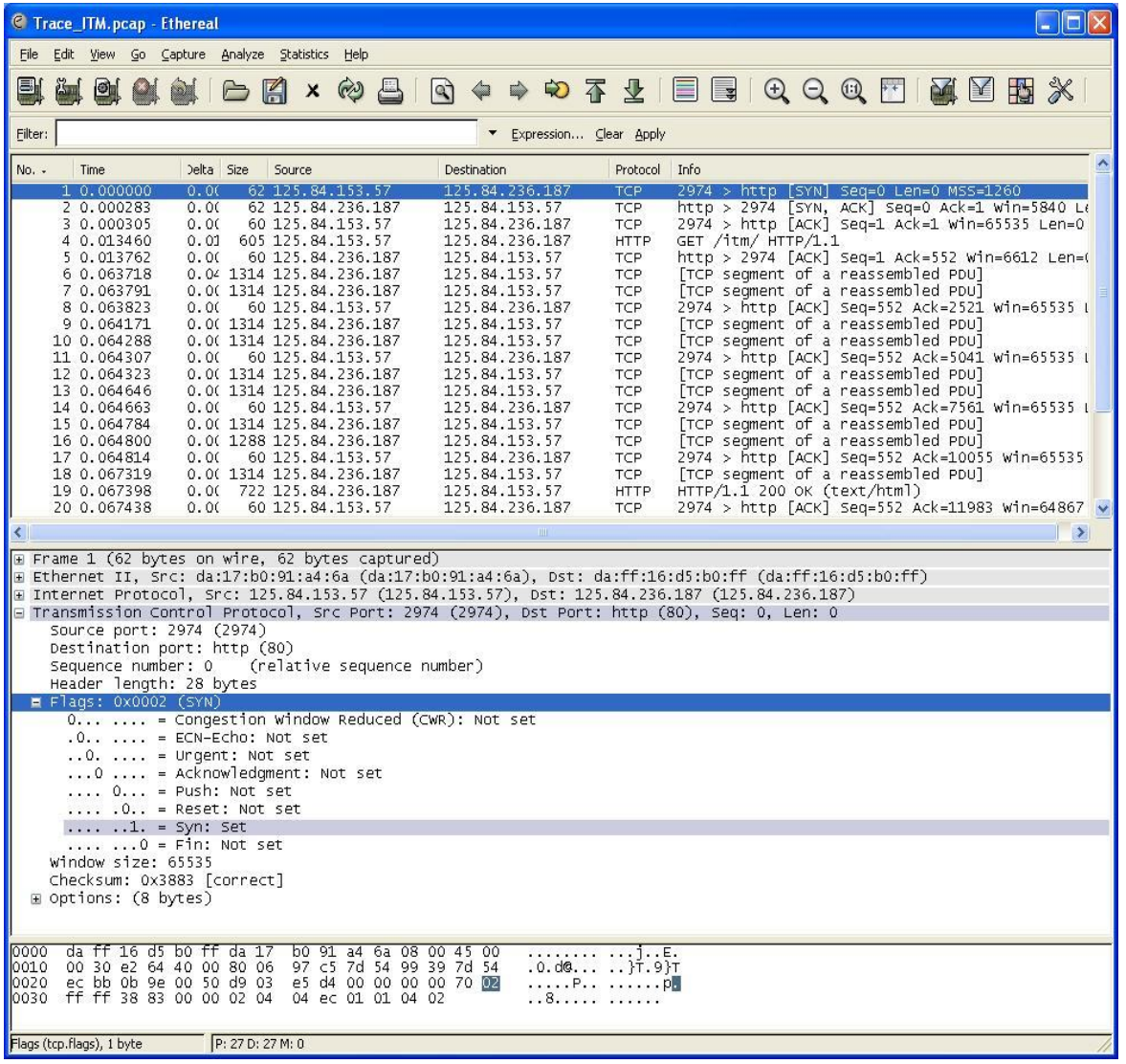
```
11:37:41.730116 IP HostA.42980 > Server.www: . ack 365 win 1728 <nop,nop,timestamp 8835396 1199883884>
11:37:41.730103 IP Server.www > HostA.42980: P 365:1131(766) ack 112 win 5792
<nop,nop,timestamp 1199883884 8835395>
11:37:41.730128 IP HostA.42980 > Server.www: . ack 1131 win 2111 <nop,nop,timestamp 8835396 1199883884>
11:37:41.730577 IP HostA.42980 > Server.www: F 112:112(0) ack 1131 win 2111
<nop,nop,timestamp 8835396 1199883884>
11:37:41.731096 IP Server.www > HostA.42980: F 1131:1131(0) ack 113 win 5792
<nop,nop,timestamp 1199883885 8835396>
11:37:41.731157 IP HostA.42980 > Server.www: . ack 1132 win 2111 <nop,nop,timestamp 8835397 1199883885>
```

- TCP-Verbindungsabbau
  - FIN, FIN/ACK, ACK

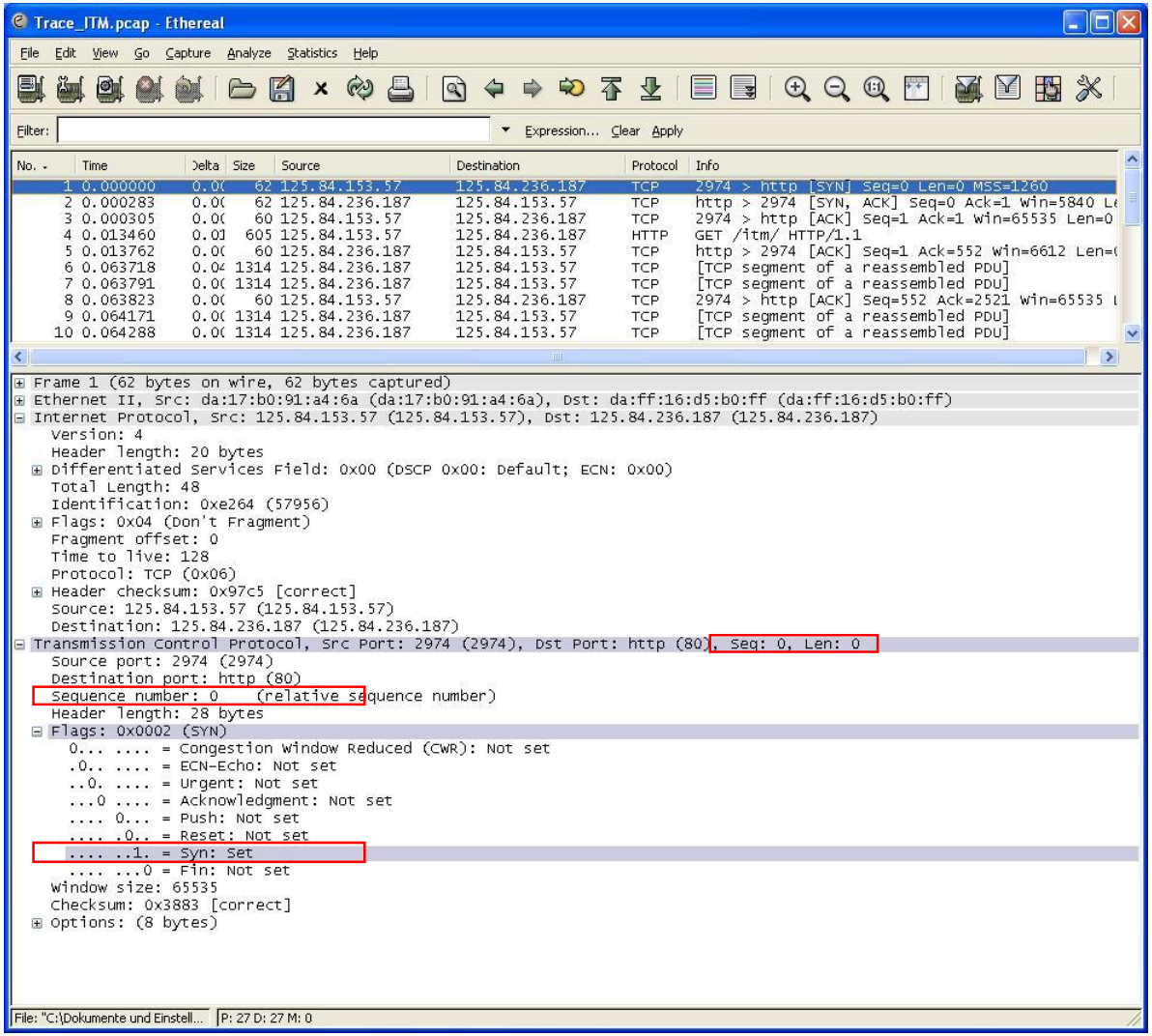
Wireshark

- Grafischer Paketsniffer
  - Dekodierung zahlreicher Protokolle
  - Diagramm-erstellung möglich

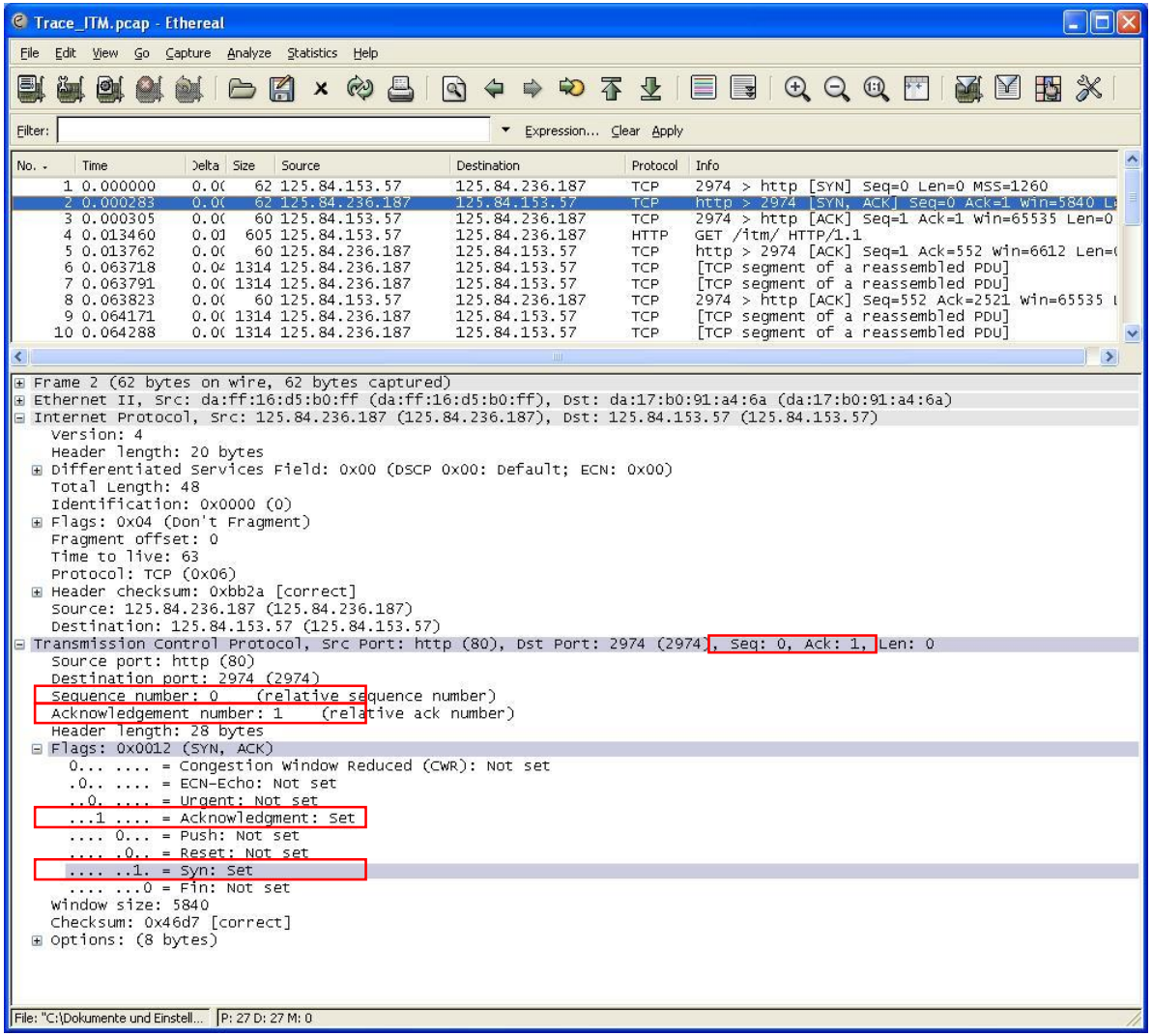




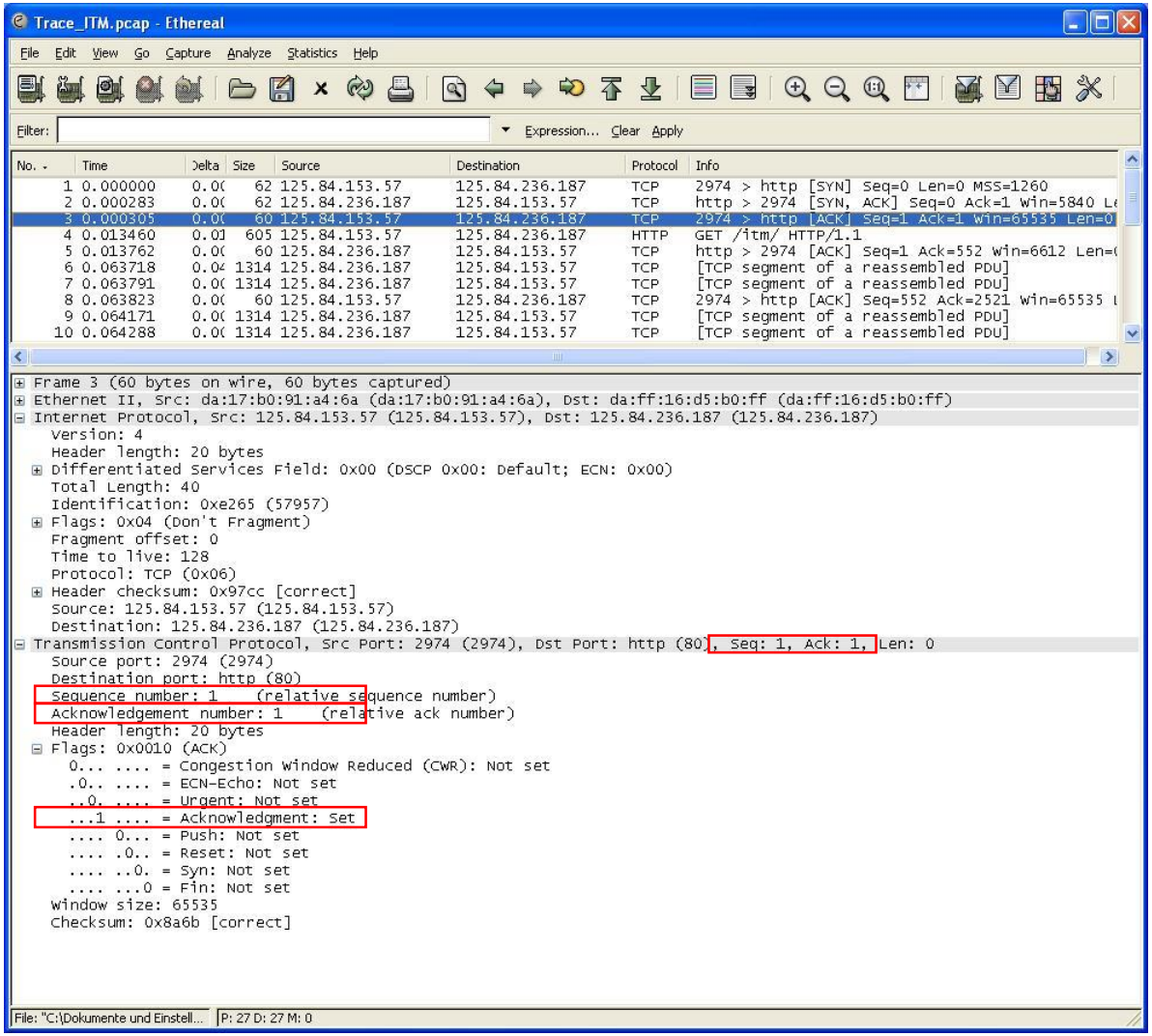
Beispiel TCP Verbindungsaufbau (Connection Request)



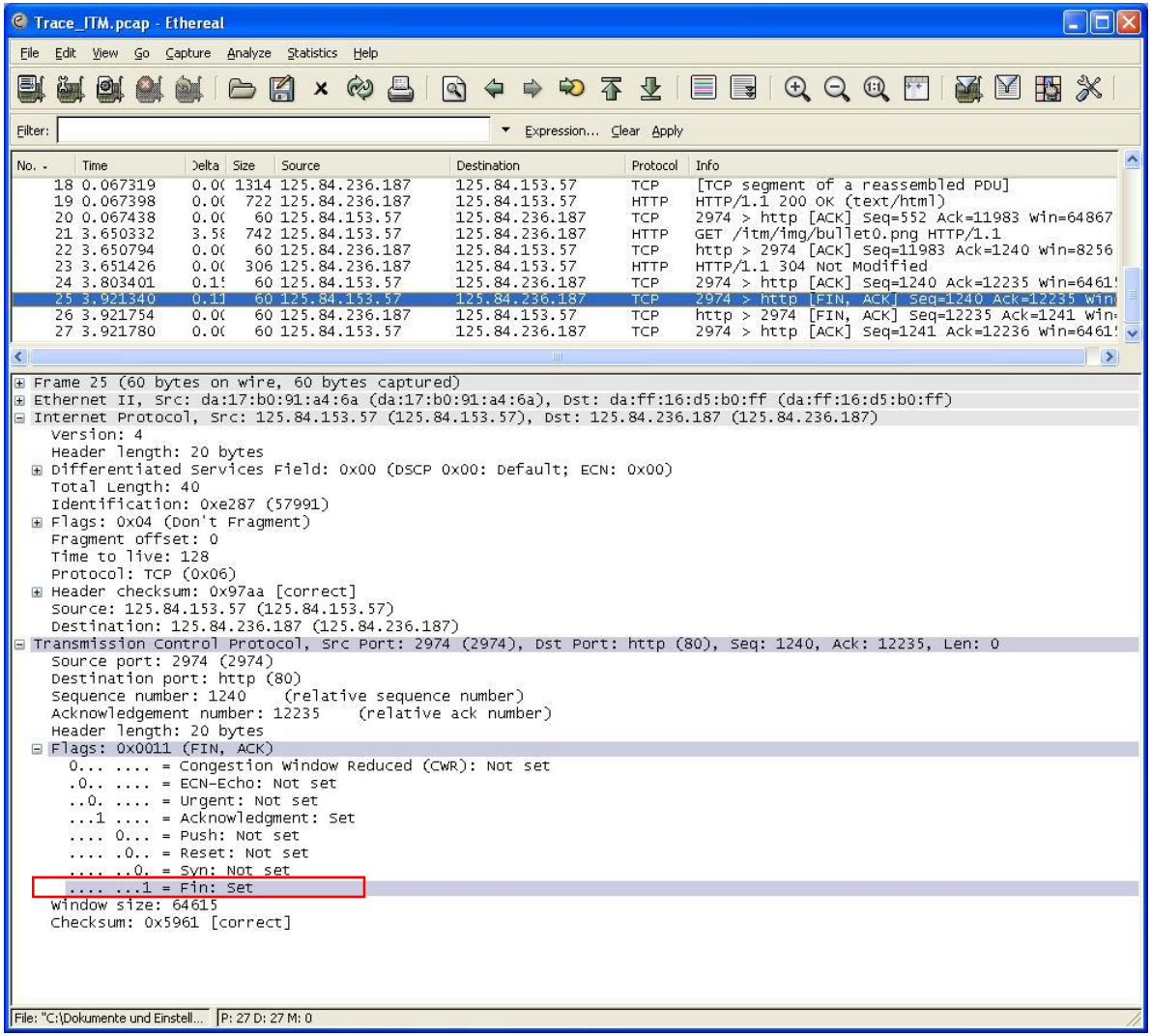
Beispiel TCP Verbindungsaufbau (Connection Confirm)



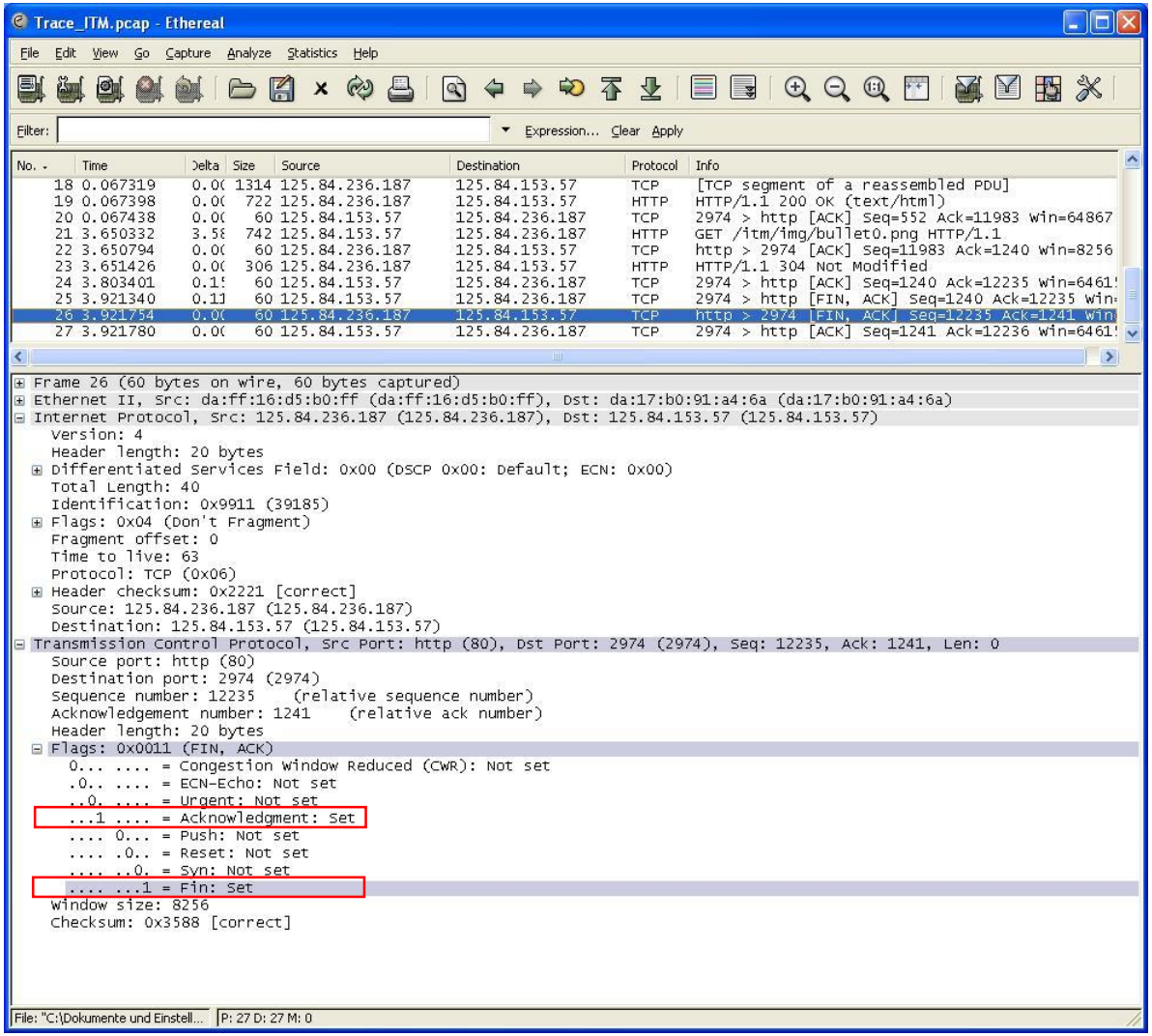
Beispiel TCP Verbindungsaufbau (ACK)



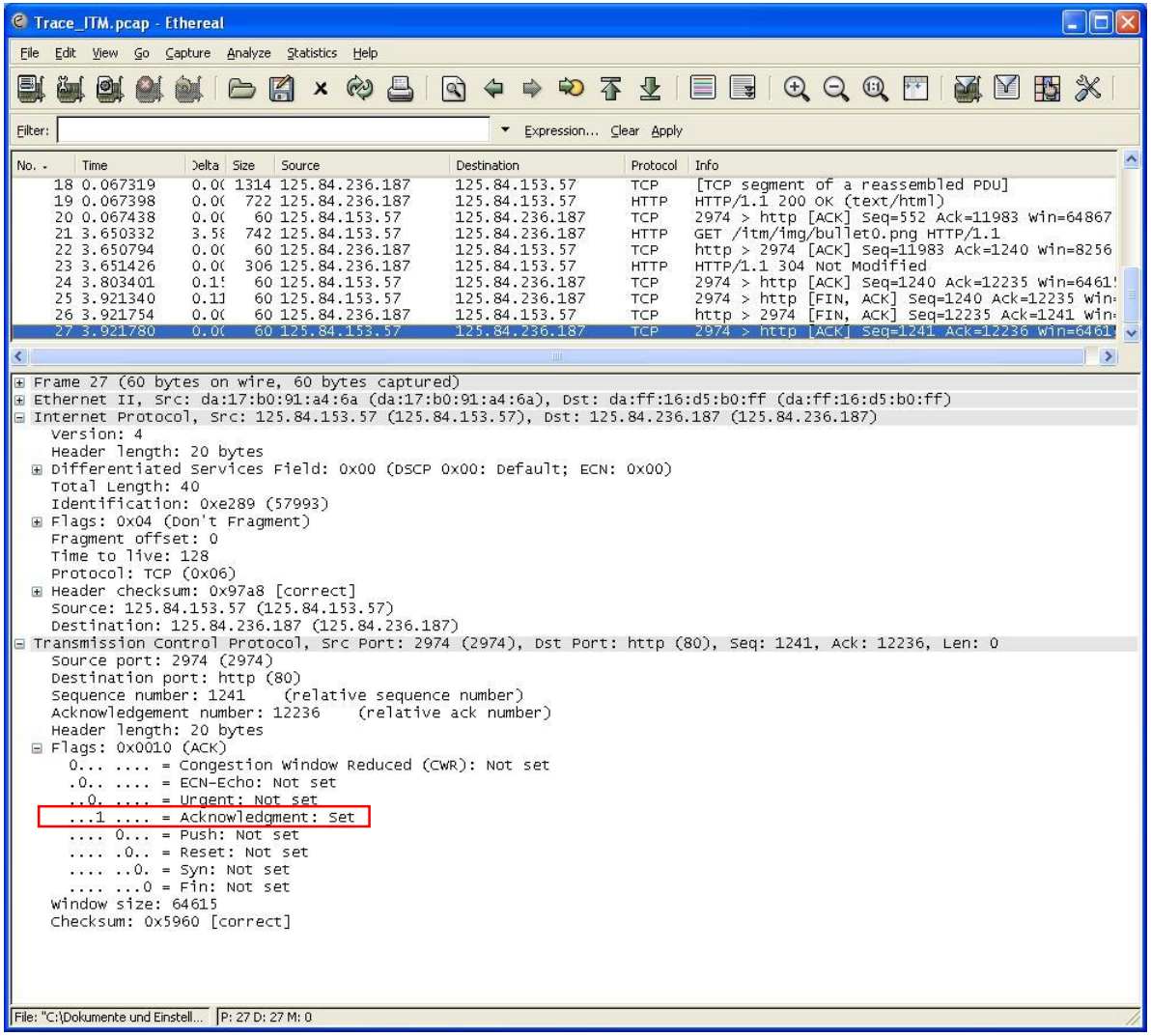
Verbindungsabbau (Client FIN)



Verbindungsabbau (Server ACK + FIN)



Verbindungsabbau (Client ACK)

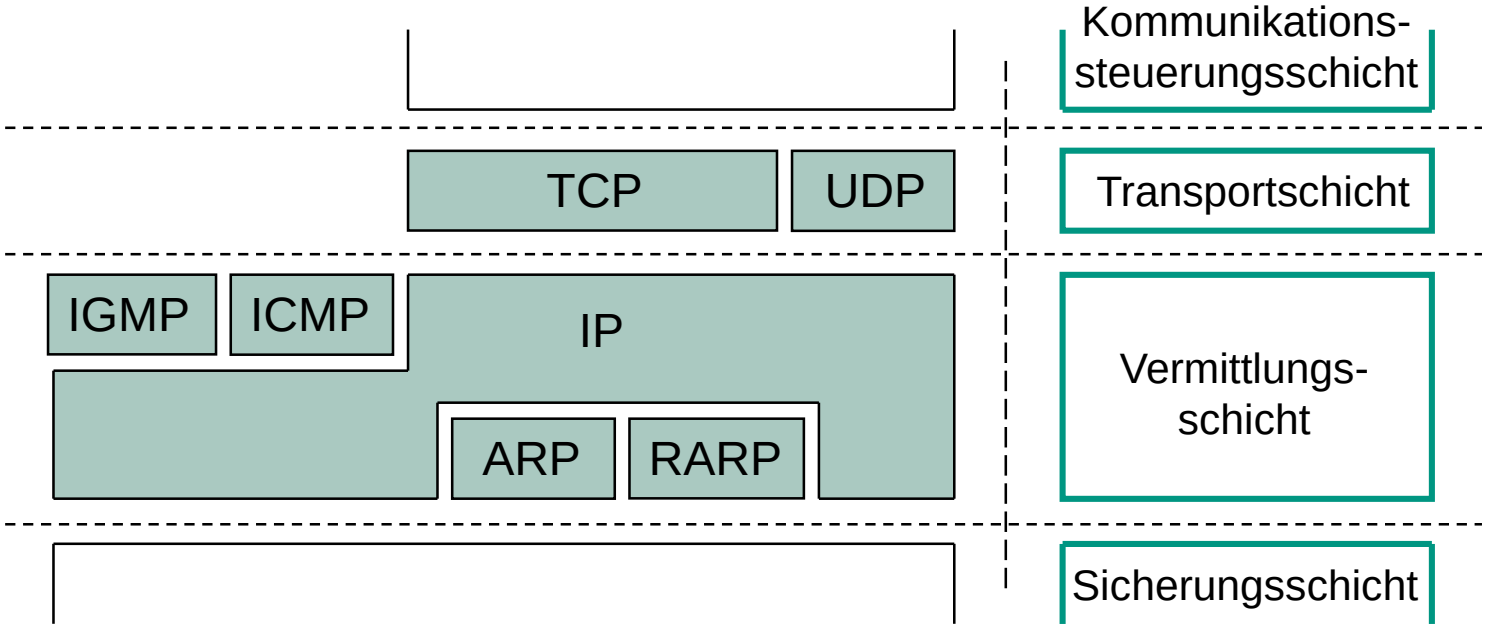


1. Einführung
2. Netzwerkarchitekturen
3. Physikalische Grundlagen
4. Protokollmechanismen
5. Die Sicherungsschicht: HDLC
6. Die Sicherungsschicht: Lokale Netze
7. Netzkopplung und Vermittlung
8. Die Transportschicht
9. Sicherheit
10. Anwendungssysteme

1. Einführung
2. Transportprotokolle im Internet
3. User Datagram Protocol (UDP)
4. Transmission Control Protocol (TCP)
5. Zusammenspiel der Protokolle

8.5 Zusammenspiel der Protokolle

- Die TCP/IP-Protokollfamilie
  - Bezeichnung TCP/IP häufig als Synonym für gesamte Protokollfamilie verwendet
- Einordnung der Internetprotokolle in das ISO/OSI-Referenzmodell



- Obwohl ICMP und IGMP den IP-Dienst nutzen, werden sie dennoch der Vermittlungsschicht zugeordnet

Die TCP/IP-Protokollfamilie: Protokollaufgaben

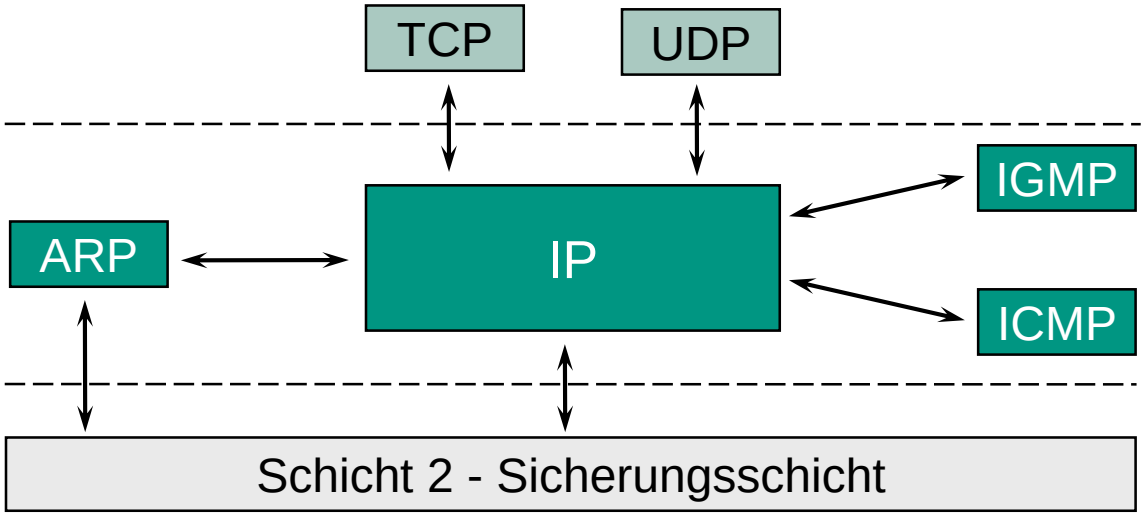
- *TCP* (Transmission Control Protocol)
  Stellt zuverlässigen Transportdienst bereit
- *UDP* (User Datagram Protocol)
  Stellt unzuverlässigen Transportdienst bereit
- *IP* (Internet Protocol)
  Wegewahl und unzuverlässige Übertragung von Datagrammen
- *ICMP* (Internet Control Message Protocol): Unterstützt den Austausch von Kontrollinformationen innerhalb der Vermittlungsschicht
- *IGMP* (Internet Group Management Protocol)
  Unterstützt die Verwaltung von Kommunikationsgruppen
- *ARP* (Address Resolution Protocol): Unterstützt die Zuordnung von IP-Adressen zu den entsprechenden Adressen der Sicherungsschicht
- *RARP* (Reverse Address Resolution Protocol)
  Stellt die Umkehrfunktion von ARP zur Verfügung

Internet-Anwendungen und Transportprotokolle

Anwendung	Anwendungs-protokoll	Verwendetes Transportprotokoll
E-Mail	SMTP	TCP
Remote Terminal Access	Telnet	TCP
Web	HTTP	TCP
Dateitransfer	FTP	TCP
Entfernter Fileserver	NFS	i.d.R. UDP
Streaming Multimedia	RTSP, …	i.d.R. UDP
Internet-Telefonie	SIP/H.323 & RTP	i.d.R. UDP
Netzmanagement	SNMP	i.d.R. UDP
Intra-Domain-Routing	RIP	i.d.R. UDP
Inter-Domain-Routing	BGP	TCP
Namensübersetzung	DNS	i.d.R. UDP

Zusammenspiel der Protokollinstanzen

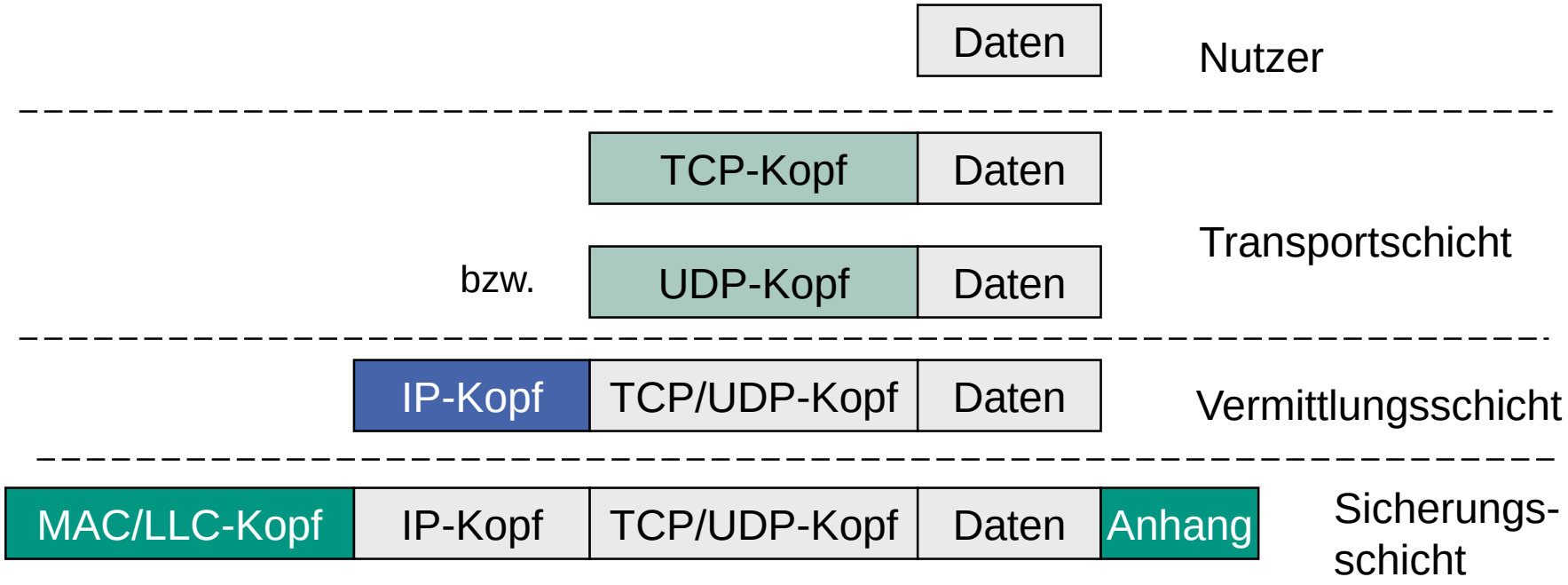
- Senden
  - TCP- bzw. UDP-Instanz übergibt Daten mit IP-Adresse des Empfängers zur Übertragung an IP-Instanz
  - IP-Instanz beauftragt ARP-Instanz mit Ermittlung der entsprechenden Schicht-2-Adresse
  - IP-Instanz übergibt Dateneinheiten mit ermittelter Schicht-2-Adresse an Instanz der Sicherungsschicht
- Empfangen
  - Unterscheidung zwischen den verschiedenen Protokollen (TCP, UDP, ICMP, …) anhand des „Protocol“-Feldes im IP-Kopf
  - IP-Instanz reicht empfangene Daten an TCP- bzw. UDP-Instanzen weiter



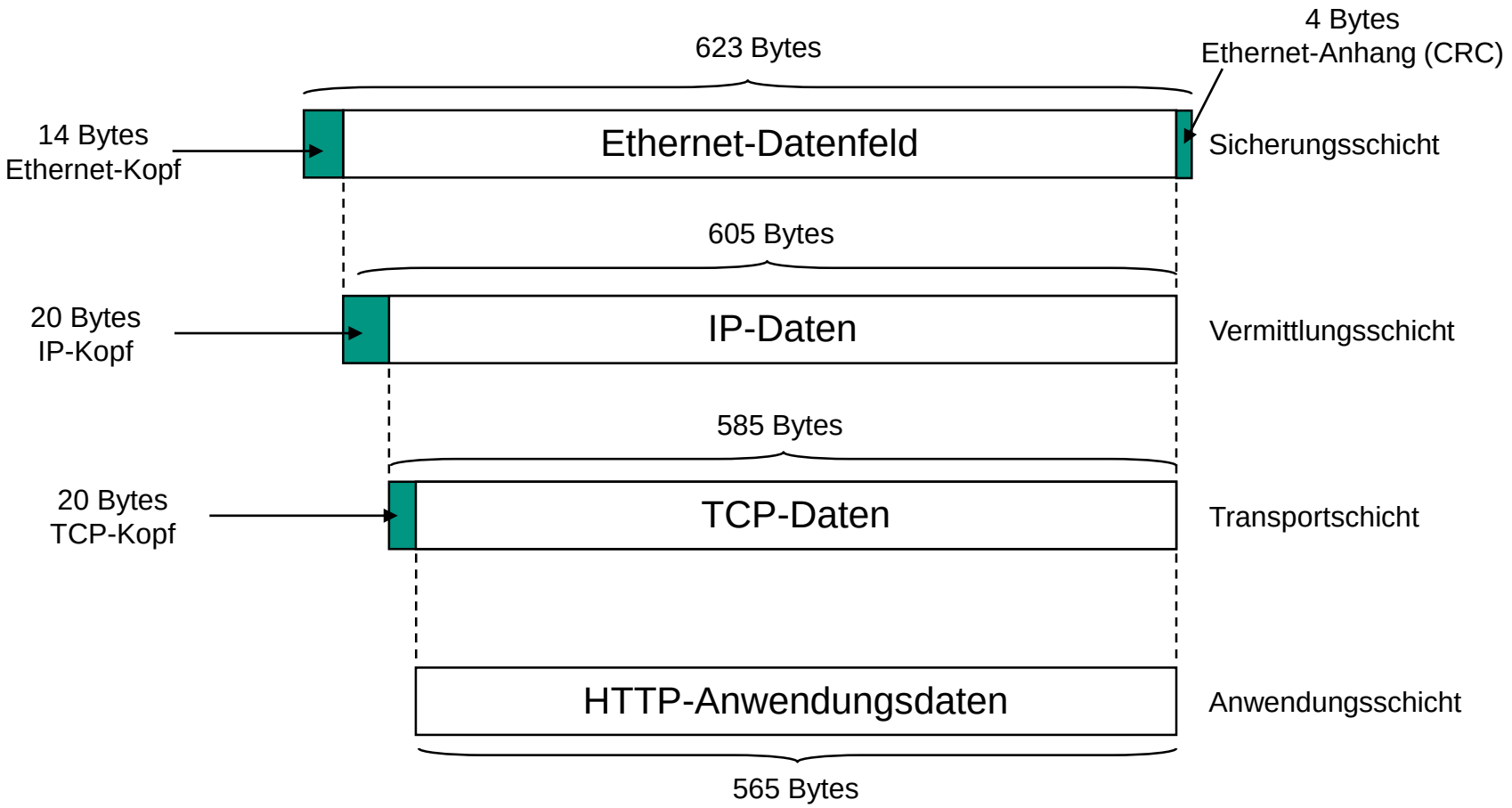
Zusammenspiel der Schichten

- Grob gesehen
  - IP leitet Dateneinheiten durch das Netz zum Empfänger
  - TCP/UDP fügen Anwendungsadressierung hinzu
  - TCP stellt darüber hinaus zuverlässigen Dienst bereit

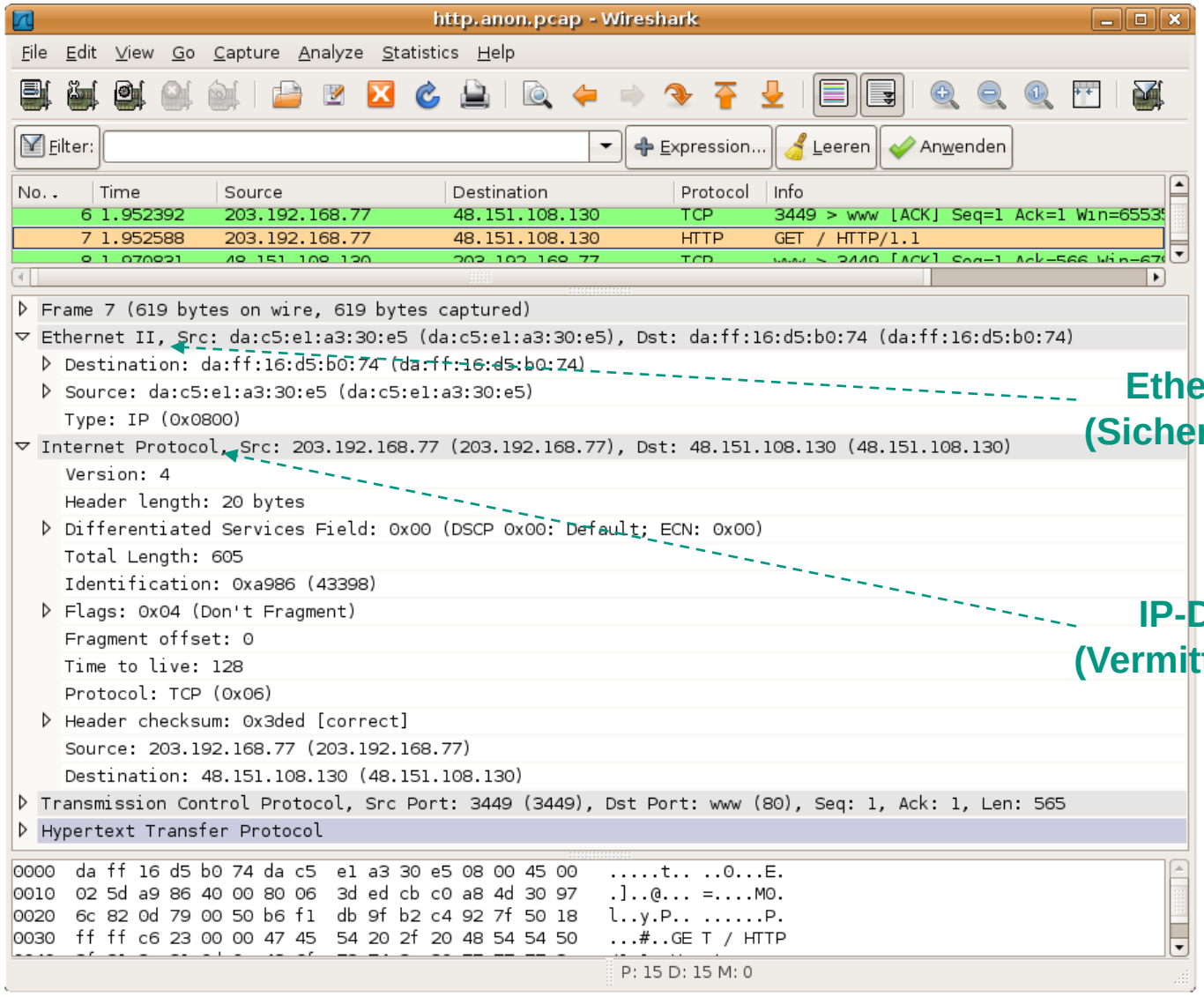
- Konstruktion der Dateneinheiten



Beispiel: HTTP-Get-Request



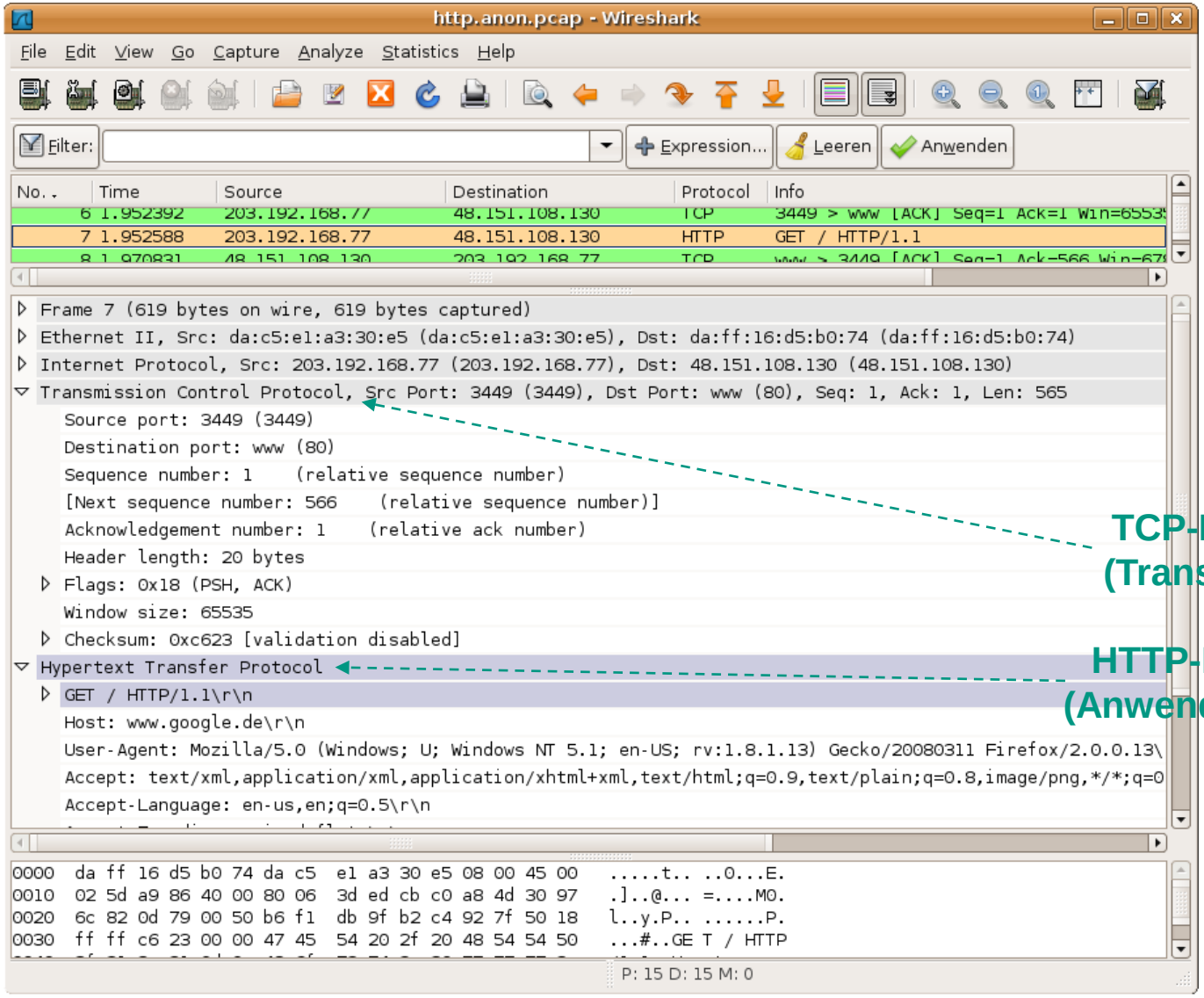
Beispiel: Ethernet- und IP-Dateneinheiten



Ethernet-Frame (Sicherungsschicht)

IP-Datagramm (Vermittlungsschicht)

Beispiel: TCP- und HTTP-Dateneinheiten



TCP-Dateneinheit (Transportschicht)

HTTP-Dateneinheit (Anwendungsschicht)

Anzahl & Charakteristik geöffneter Verbindungen bei Webseitenaufruf

- Was passiert beim Aufruf einer Webseite?
  - Früher waren Webseiten einfach strukturiert
    - Eine zentrale HTML-Seite mit einigen Grafiken auf demselben Server
- Untersuchung der Verbindungscharakteristika bei Webseitenaufrufen
  - Studie vom Frühjahr 2009
  - Messgegenstände: 100 „beliebteste“ Homepages der USA
  - Messwerte von Interesse
    - Übertragene Datenmenge [Bytes]
    - Anzahl geöffneter TCP-Verbindungen
    - Anzahl unterschiedlicher Server, zu denen Verbindungen aufgebaut werden
    - Anzahl geladener Elemente (nach HTTP GET und POST Requests)
    - Anzahl verschiedener Netzwerkpräfixe (NP) und Autonomer System (AS), die involviert sind
    - Anzahl verschiedener DNS „Second Level Domains“
      - www.google.com, www.kit.edu
      - Können Rückschlüsse auf beteiligte Unternehmen bieten

[Char09]

Anzahl & Charakteristik geöffneter Verbindungen bei Webseitenaufruf (2)

	Bytes	Verbind.	Server	Elemente	AS	DSN SLDs
Minimum	943	1	1	2	1	1
Durchschn.	600 kB	22,94	8,24	62,18	5,04	5,15
Maximum	11,7 MB	107	21	284	13	15

- Um eine komplette Homepage zu laden müssen durchschnittlich …
  - etwa 23 TCP-Verbindungen geöffnet werden
  - mehr als acht verschiedene Server vom Client kontaktiert werden, die in mehr als fünf autonomen Systemen liegen
    - „Helper Services“ insbesondere von akamai.net, google.com, doubleclick.net
  - mehr als 60 Elemente geladen werden
  - Verbindungen zu Servern mit mehr als fünf verschiedenen Second-Level-Domains aufgebaut werden

Ausblick: Protokoll-Engineering

- Detailliertes Protokoll-Engineering am Beispiel TCP in der Vorlesung Telematik
  - Dynamik und TCP
    - „Conservation of Packets“
    - Aktives Warteschlangenmanagement
  - Evaluierung von TCP
    - Periodisches Modell
    - Detailliertes Paketverlustmodell
  - TCP und Fairness
  - TCP in unterschiedlichsten „Einsatzgebieten“
    - Webanwendungen, Datenzentren …

Aufgaben

8.1 Welche Aufgaben hat die Transportschicht?

8.2 Zwischen was genau transportiert die Transportschicht Daten?

8.3 Wie werden TCP- bzw. UDP-Anwendungen adressiert?

8.4 Durch welche Eigenschaften zeichnet sich das *User Datagram Protocol* aus und welche Vor- bzw. Nachteile besitzt es?

8.5 Erläutern Sie die Eigenschaften des *Transmission Control Protocols*

8.6 Geben Sie einige Beispiele für Anwendungen, bei denen Sie den Einsatz von UDP bzw. TCP bevorzugen würden und begründen Sie Ihre Wahl

8.7 Welches Verfahren kommt beim TCP-Verbindungsaufbau zum Einsatz?

8.8 Wozu wird für den Dienst, den TCP zur Verfügung stellt, überhaupt eine Verbindung benötigt?

8.9 Welche Art Quittungsverfahren kommt bei TCP zum Einsatz?

8.10 Worin liegt der Unterschied zwischen Fluss- und Staukontrolle? Welche Mechanismen von TCP spielen in diesem Zusammenhang eine Rolle?

Literatur

[Char09] J. Charzinski, Traffic, Structure and Locality Characteristic of the Web‘s Most Popular Services‘ Home Pages, Kommunikation in Verteilten Systemen (KiVS) 2009, Kassel, Germany

[FlFa99] S. Floyd, K. Fall, Promoting the Use of End-to-End Congestion Control in the Internet, IEEE/ACM Transactions on Networking, August 1999

[Hals05] F. Halsall, Computer Networking and the Internet, Addison-Wesley, 2005, 5th Edition

[KuRo10] J. Kurose, K. Ross, Computer Networking, Addison Wesley 2010, 5th Edition

[RFC768] J. Postel, User Datagram Protocol, RFC 768, August 1980

[RFC793] J. Postel (Ed.), Transmission Control Protocol, RFC 793, September 1981

[Stev94] W. R. Stevens, TCP/IP Illustrated, Volume 1, Addison-Wesley Professional, 1994