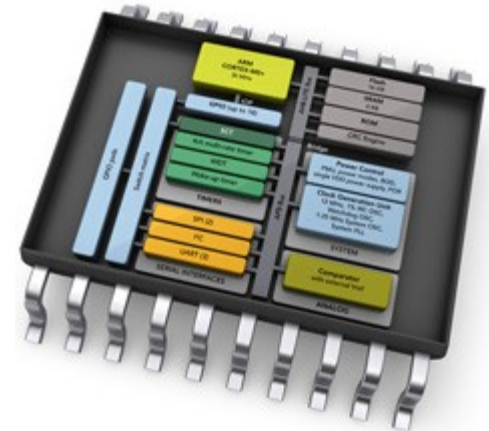


Kommunikationssysteme

(Modulcode 941306)

Prof. Dr. Andreas Terstegge



Einführung in das User Datagram Protocol (UDP)

- IP kann potentiell Pakete verwerfen (Store-and-Forward Prinzip)
- TCP ist sicher, aber komplex
- Häufig wird das nicht benötigt:
 - Kommunikation ist nur lokal
 - Geringer Datenverlust ist okay (Audio/Video Daten)
 - Feste Datenraten (z.B. beim Streaming ↔ TCP Slow-Start ???)
 - Implementierung eigener („leichter“) Sicherungsmechanismen
 - ...

TCP / UDP

IP

| |
|-----------------------------|
| Anwendungs- schicht |
| Darstellungs- schicht |
| Sitzungsschicht |
| Transportschicht |
| Vermittlungs- schicht |
| Sicherungs- schicht |
| Bitübertragungs- schicht |

UDP: User Datagram Protocol

- UDP stellt eine „direkte“ Schnittstelle zur Nutzung von IP dar: Anwendungen können Nachrichten direkt verschicken, ohne Verbindungsaufbau
- Unzuverlässig, verbindungslos
- einfacher und schneller als TCP
- Optionale Prüfsumme
- Sehr viele Multimedia-Anwendungen verwenden UDP, da dort keine zuverlässige Verbindung benötigt wird

UDP Datagram Header Format

| Bit # | 0 | 7 | 8 | 15 | 16 | 23 | 24 | 31 |
|-------|-------------|---|---|----|--------------------------|----|----|----|
| 0 | Source Port | | | | Destination Port | | | |
| 32 | Length | | | | Header and Data Checksum | | | |

Der UDP-Header

Source Port

Identifiziert den sendenden Prozess, also den Prozess, an den gegebenenfalls Rückmeldungen zu senden sind. Die Angabe ist optional; das Feld soll den Wert null haben, wenn die Option nicht genutzt wird.

Destination Port

Identifiziert den Prozess im Zielsystem, an den die Daten abzuliefern sind.

Length

Im Längelfeld wird die Gesamtlänge des UDP-Datagramms in Bytes angegeben; die Mindestlänge beträgt somit 8 (= *Header-Länge*)

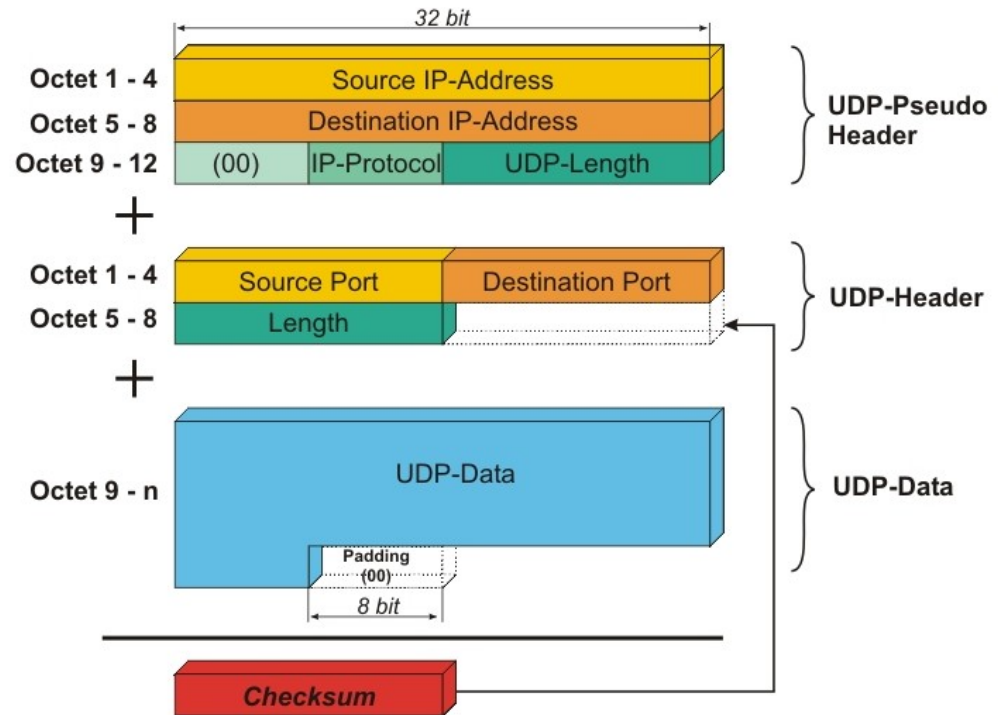
Checksum

Die Angabe ist optional (0 bedeutet: keine Angabe). Für die Berechnung der Längsparität wird dem UDP-Datagramm ein (nicht mitübertragener) *Pseudo-Header* von 12 Bytes Länge vorangestellt, der im wesentlichen *IP-Source Address*, *IP-Destination Address* und die im IP-Datagramm angegebene Protokoll-Nr. für UDP (17) enthält. Da der Datenteil eines IP-Datagramms nicht durch die *IP Header Checksum* geschützt ist, bedeutet ein Verzicht auf die *UDP-Checksum*, dass der Inhalt des UDP-Datagramms (*Header* und Daten) nicht durch eine Prüfsumme gesichert ist.

UDP Checksum

Ziel: Erkennen von Fehlern (z.B. flipped bits) im übertragenen Segment – optionale Nutzung!

- Formal über eine Einerkomplementsumme über ...
 - Pseudo-IP-Header: (Verletzung der Schichtgrenze!)
 - UDP Header
 - Daten
- Details zur Berechnung:
 - Optionales Auffüllen der Daten (wenn ungerade Byte-Anzahl → ‚Padding‘)
 - Berechnung der Prüfsumme durch Interpretation der Daten als 16-Bit Werte. Aufaddieren der 16-Bit-Werte im Einer-Komplement.
 - Am Ende wird das 1-er Komplement der Summe berechnet
- Einfügen der Prüfsumme in den UDP-Header



Wozu UDP? Anwendungen

- **Multimedia:**

Die digitale Übertragung von Audio- und Videodaten besitzt spezifische Anforderungen:

- Geringe Verlustraten stören nicht
- Isochrones Abspielen → schwierig mit TCP ...
- Latenzzeiten müssen insbesondere bei interaktiven Anwendungen gering sein (Telefonie erfordern eine maximale Latenz von 150ms)
- Jitter: Die Variation der Laufzeit sollte ebenso beschränkt sein

- **RPC:** Remote Procedure Calls
- **NFS:** Network File System
- **RTP:** Real-Time Transport Protocol
- **DNS:** Domain Name System
- ...

Weitere wichtige UDP-basierte Anwendung

Das Laden des Betriebssystems (Boot-Vorgang) über das Netzwerk benötigt entsprechende Protokolle

- TCP ist aufwendig, Fähigkeiten der im BIOS verankerten Mechanismen ist begrenzt
- Ethernet als Sicherungsschicht implementiert bereits Mechanismen zur Zuverlässigkeit, wenn der Server im gleichen Netz ist braucht man viele Mechanismen von TCP nicht
- Übertragung von Dateien z.B. über das Trivial File Transfer Protocol (**TFTP**)

BIOS verfügen zumeist bereits über eine UDP-Implementierung, daher wurde mit **BOOTP** und **DHCP** entsprechendes auf UDP-Basis entwickelt Broadcast an alle durch Zieladresse 255.255.255.255.

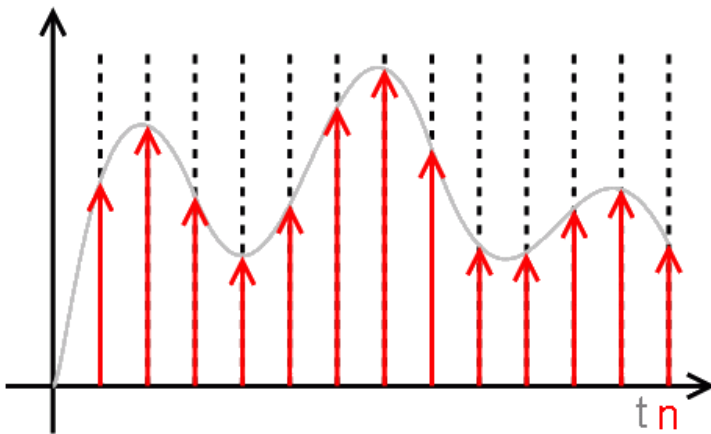
Kleiner Exkurs

Übertragen analoger (streaming)-DATEN

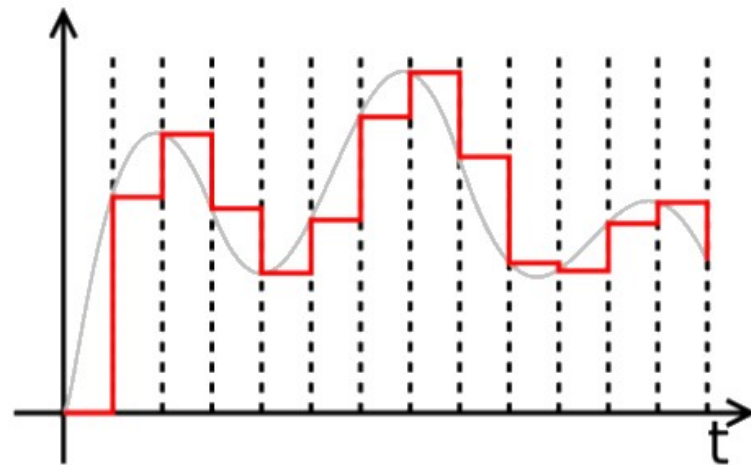
- Abtasten
- Halten
- Quantisieren
- Kodieren



- Signalstärke wird regelmäßig gemessen

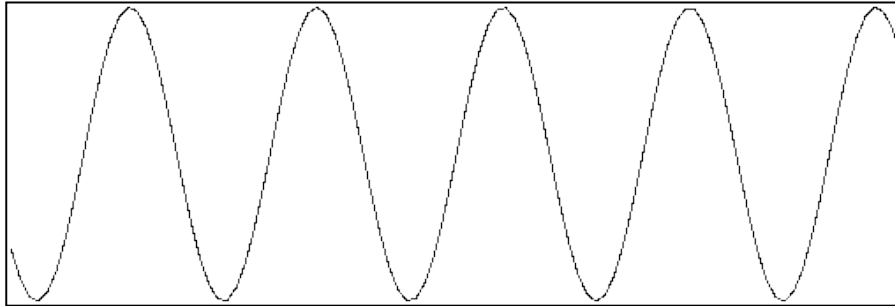


- **Abtastung:**
Messen des analogen
Wertes zu den Zeitpunkten
 $t_n = n * \Delta t$



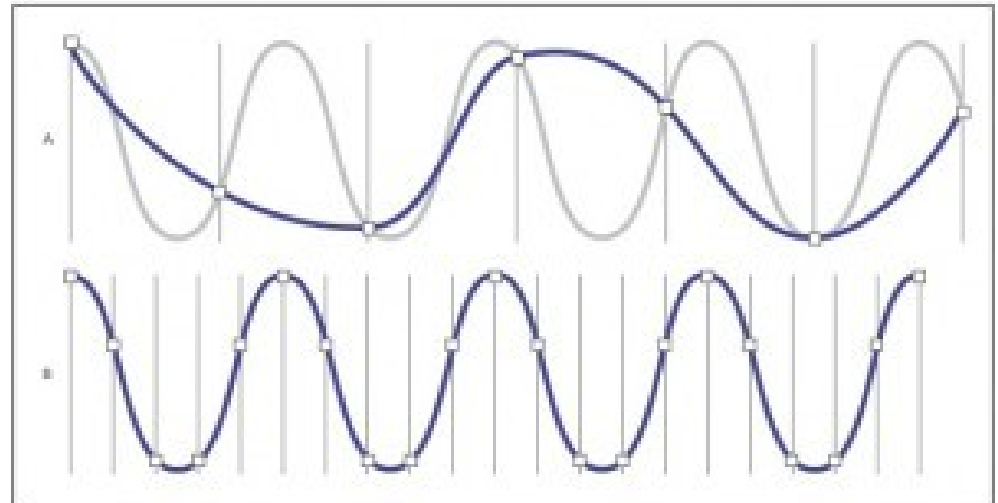
- **Halten des Wertes:**
Festhalten des Mess-
wertes, damit er in der
Zeit Δt digitalisiert
werden kann.

Sampling Rate



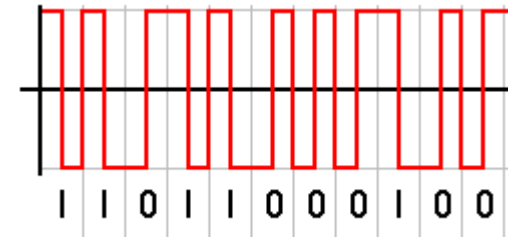
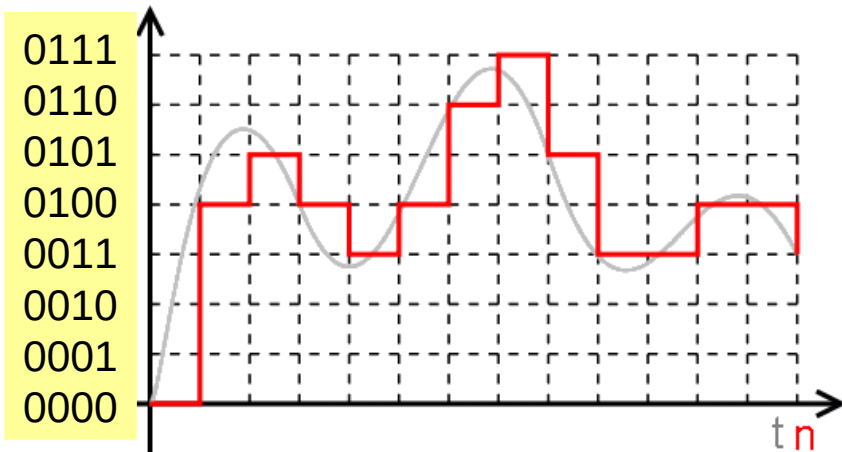
Nyquist Theorem:

Sampling-Frequenz \geq
 $2 \cdot$ maximale Frequenz des
Signals



Quantisierung und Kodierung

- Umwandlung Spannungswerte → Zahlenwerte



- Quantisierung:**
 Umwandlung des Messwertes in den bestmöglichen Wert des digitalen Wertevorrats (Binärzahl)
- Kodierung:**
 Umwandlung der digitalen Werte in z.B. einen Leitungs-Code
 - ggf. Fehlerkorrektur
 - Taktrückgewinnung
 - ...

Qualität des resultierenden Digitalsignals

- Abhängig von
 - Anzahl der Quantisierungsstufen (Auflösung der einzelnen Niveaus / Lautstärken)
→ Quantisierungsrauschen
 - Abtastrate (liefert Bandbreite):
- $\text{Bitrate/s} = \text{Abtastrate/Hz} * \text{Bits/Abtastwert}$
- **Beispiel Audio-CD:**
 $44,1\text{kHz} * 2 \text{ Kanäle} * 16\text{bit} = 1,4 \text{ MBit/s}$
Speicherbedarf bei 60 min : 630 MB
- **Beispiel Telefonie:**
 $8\text{kHz} * 1 \text{ Kanal} * 8\text{bit} = 64\text{kBit/s}$

Multimediale Netzanwendungen

Kategorien:

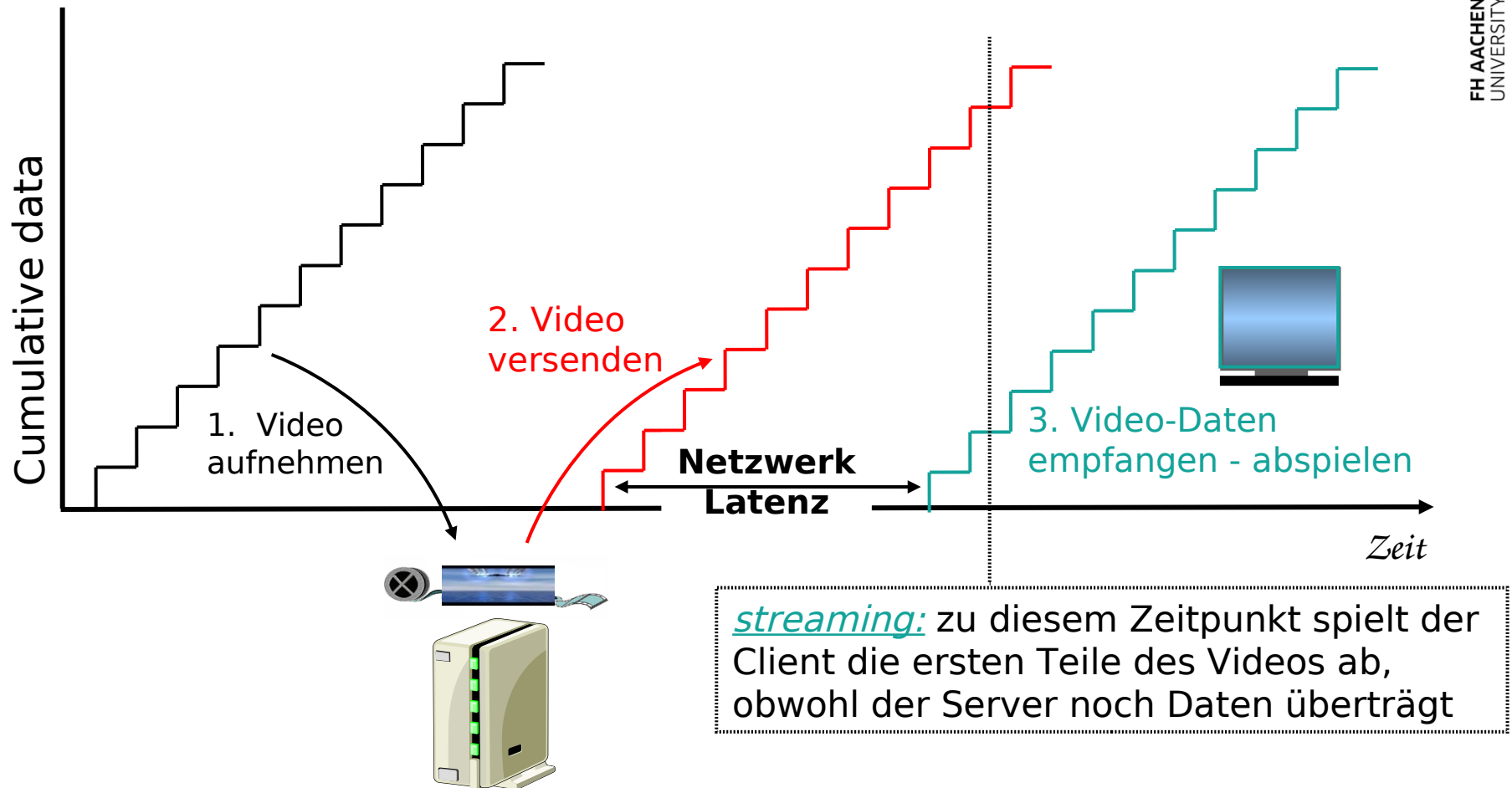
- 1) Streaming gespeicherter Audio- und Video-Daten
- 2) Streaming von live Audio und Video
- 3) Interaktive Audio und Video Nutzung

Streaming:

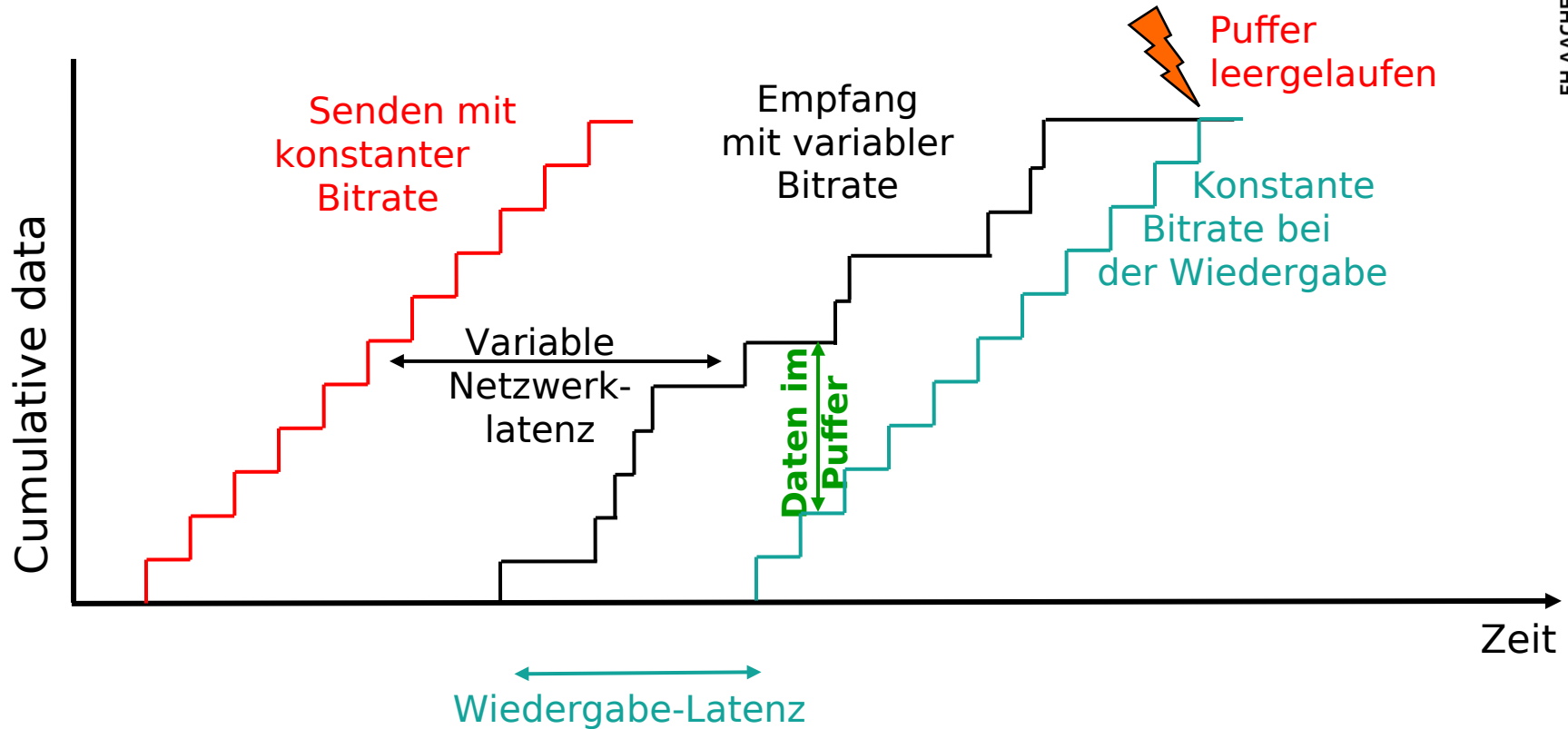
Das Verwenden (ausliefern) von Daten bevor diese vollständig übertragen wurden

→ Interpretation der Daten als (ggf. sehr lange dauernder) Datenstrom

Streaming gespeicherter multimedialer Daten



Streaming gespeicherter multimedialer Daten



Live-Streaming multimedialer Daten

Beispiel:

- Internet-Radio
- IPTV

Streaming:

- Playback-Puffer
- Zeitverschiebung zwischen Wiedergabe und originaler Zeit kann 10 Sekunden sein
- Zeitkritisch: Playback-Puffer darf nicht leerlaufen

Interaktivität

- Vorspulen kann nicht funktionieren
- Rückspulen und pausieren ist möglich

Wie kann das mit UDP realisiert werden?



Real-Time Protocols

RTP/RTCP/RTSP

RTP: Eine UDP-Anwendung

Das **RTP**-Protokoll liefert Transportschnittstellen, die UDP erweitern:

- UDP liefert Port-Nummern
- IP die Adressen der Endpunkte
- RTP liefert u.a.
 - Kodierungskennung
 - Sequenznummern
 - Zeitstempel
- **RTCP** liefert Statistiken (auf UDP)-Basis
- **RTSP** ist unsere Fernbedienung

transport
layer

Application

RTP

UDP

IP

Data Link

Physical

RTP Header:

| Byte 0 | | | | | | | | Byte 1 | | | | | | | | Byte 2 | | | | | | | | Byte 3 | | | | | | | |
|---|---|---|---|----|---|---|---|--------|----|---|---|---|---|---|---|-----------------|---|---|---|---|---|---|---|--------|---|---|---|---|---|---|---|
| Bit 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | Bit 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | Bit 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | Bit 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| V=2 | | P | X | CC | | | | M | PT | | | | | | | Sequence Number | | | | | | | | | | | | | | | |
| Timestamp (in sample rate units) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Synchronization Source (SSRC) identifier | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Contributing Source (CSRC) identifiers (optional) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Header Extension (optional) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

FH Aachen
Fachbereich 9 Medizintechnik und Technomathematik
Prof. Dr.-Ing. Andreas Terstegge
Straße Nr.
PLZ Ort
T +49. 241. 6009 53813
F +49. 241. 6009 53119
Terstegge@fh-aachen.de
www.fh-aachen.de