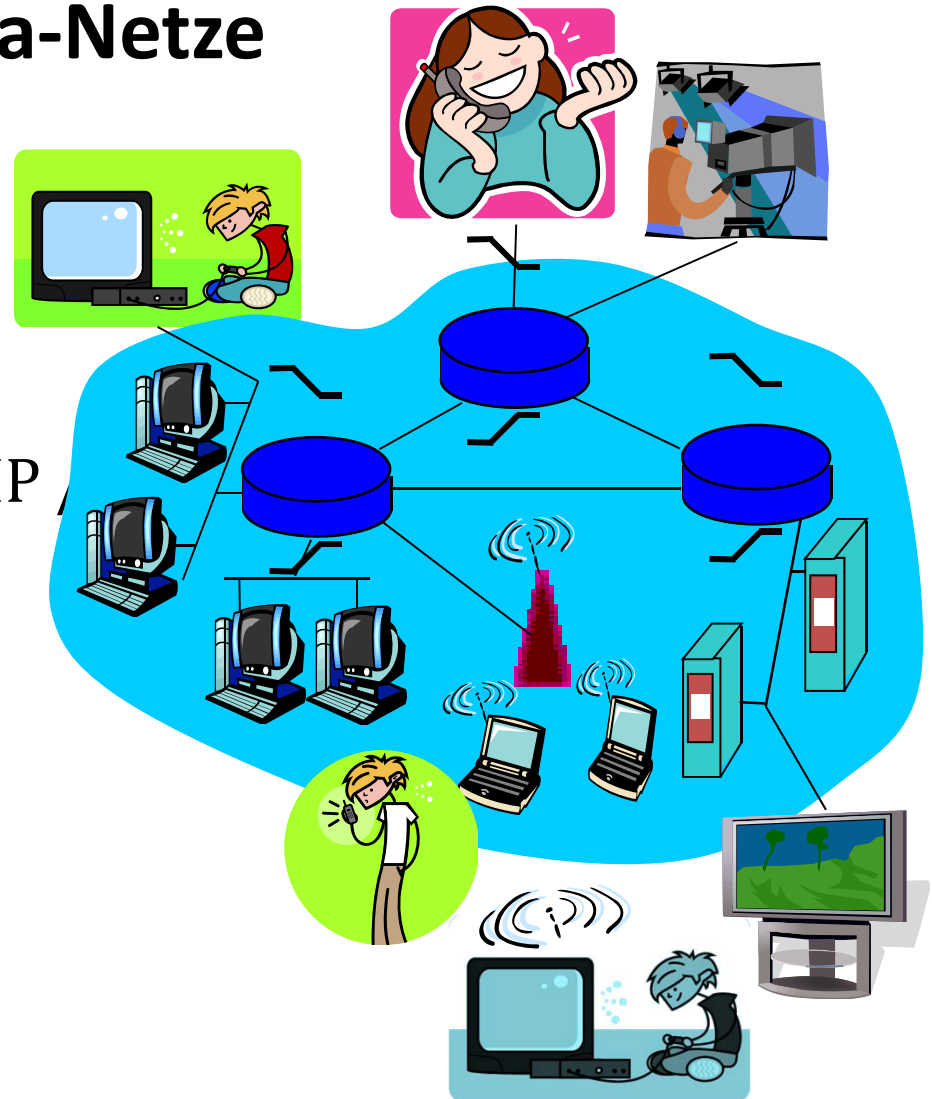


# Multimedia-Netze

## Gliederung

- Multimedia-Anwendungen
- Audio- und Videostreaming
- Realzeit Multimedia: Voice over IP  
Internet-Telephonie
- Protokolle für Realzeit-Anwendungen  
RTP, RTCP, SIP
- Über best effort hinaus
- Scheduling und Policing



# Multimedia Netz-Anwendungen

## Anwendungsklassen:

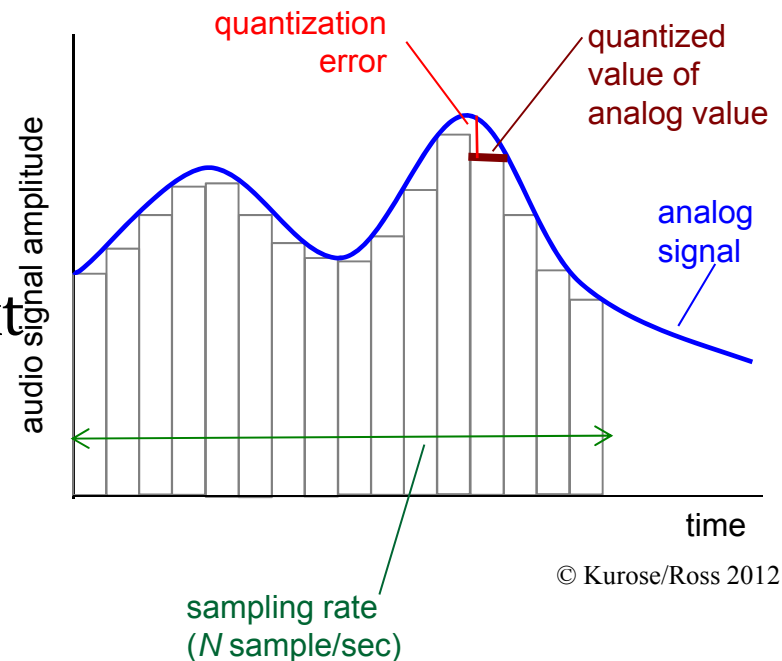
- 1) Streaming gespeicherter Audio- und Video-Daten
- 2) Streaming aktueller Audio- und Video-Daten (live)
- 3) Interaktive Realzeit-Audio und Video-Kommunikation

## Grundlegende Eigenschaften:

- ◆ Typisch: **Verzögerung ist kritisch**
  - Ende-zu-Ende-Verzögerung
  - **Jitter**  
(Verzögerungsschwankungen)
- ◆ Aber **Verluste sind akzeptabel**: seltene Paketverluste werden kaum bemerkt
- ◆ Unterschied zu klassischem Datentransfer, wo Verluste nicht akzeptabel, aber Verzögerungen unkritisch sind.

# Multimedia Daten Audio

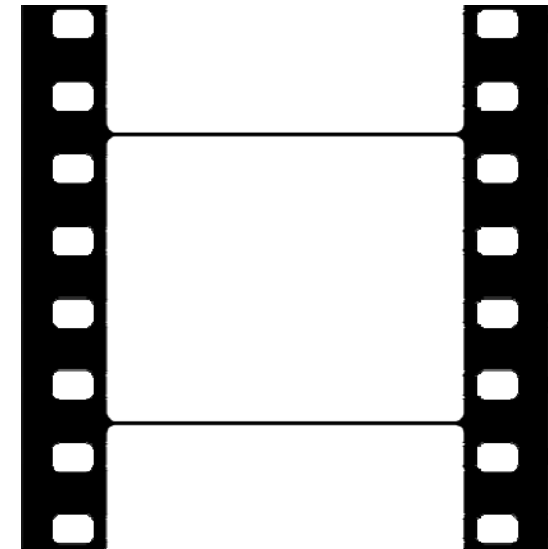
- Analoges Signal wird mit konstanter Rate abgetastet (digitales Signal)
- Digitales Signal wird u.U. kodiert (Reduktion der Größe)
- Digitales Signal wird verschickt
- Empfänger transformiert digitales Signal in analoges Signal
- Typische Raten  
CD 1.411 Mbps, MP3 128-160 kbps, IP-Telefonie 5.3 kbps



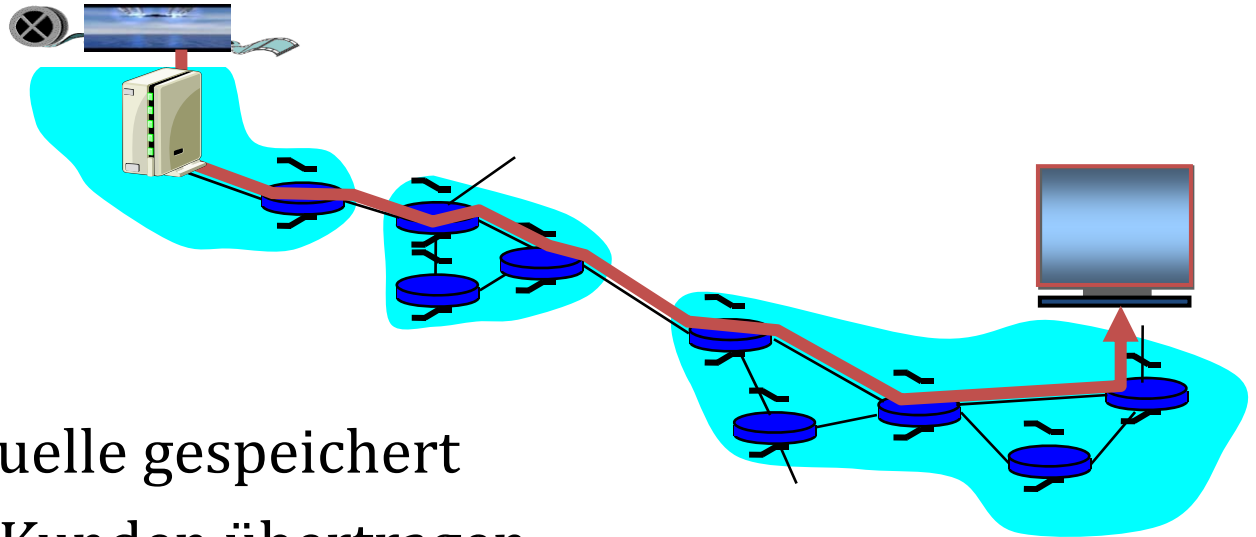
# Multimedia Daten Video

---

- Sequenz von Bildern, die mit konstanter Rate (z.B. 24 Bilder/sec) abgespielt werden
- Bild als Feld von Pixeln (durch mehrere Bits repräsentiert)
- Kodierung und Größenreduktion durch Nutzung der Redundanz im Bild und zwischen Bildern
- Typische Raten  
MPEG 1 (CD) 1.5 Mbits,  
MPEG 2 (DVD) 4-6 Mbps,  
MPEG 4 (Internet) < 1 Mbps  
(unkodiertes Fernsehbild (HDTV) 1080x1920 Pixel mit 24 Bit Farbinformation pro Pixel bei 24 Bildern pro Sekunde > 1 Gbps)



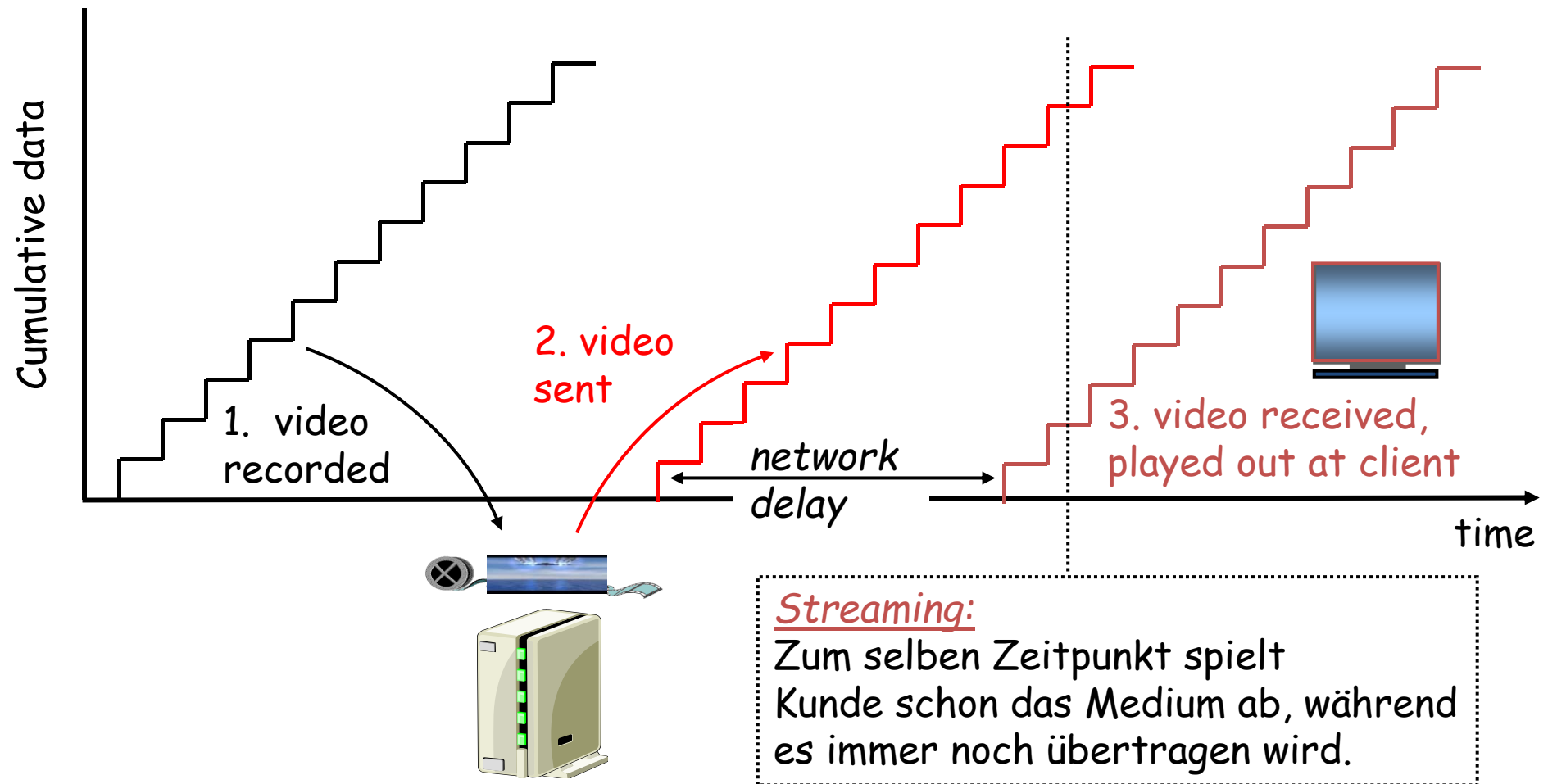
# Streaming gespeicherter Multimediadaten



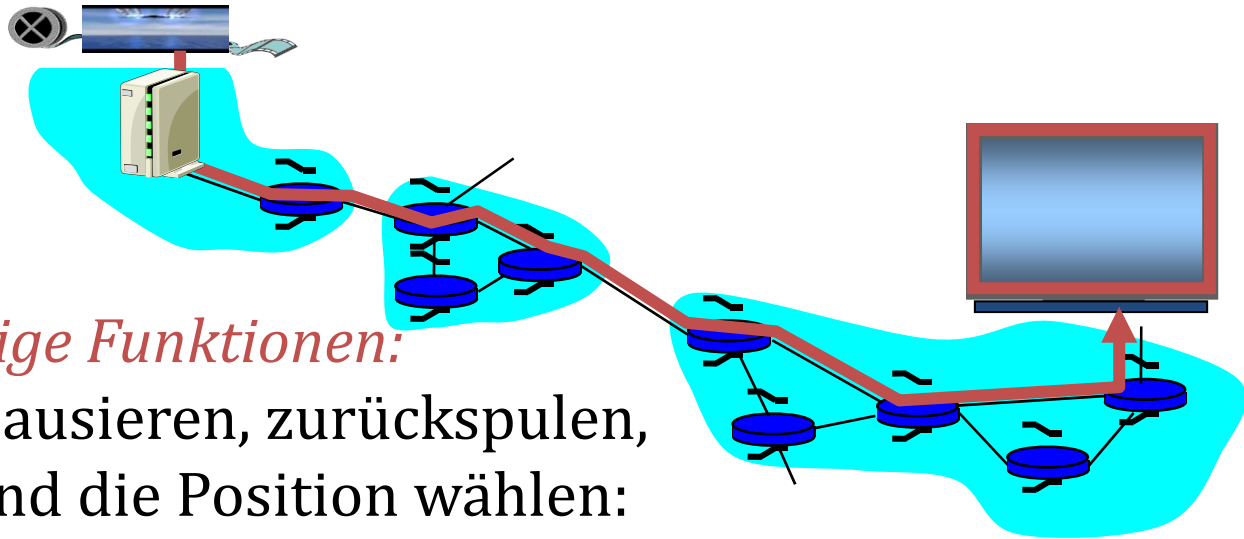
## Streaming:

- Daten sind bei Quelle gespeichert
- Sie werden zum Kunden übertragen
- Streaming: Das Abspielen beim Kunden beginnt, noch bevor die gesamte Datei übertragen ist
- Zeitanforderung für die noch zu übertragenden Daten:  
Rechtzeitig zum lückenlosen Abspielen!

# Streaming



# Streaming: Interaktivität



- *Videorecorder-artige Funktionen:*  
Der Kunde kann pausieren, zurückspulen, vorwärtsspulen und die Position wählen:
  - 10 sec Anfangsverzögerung OK
  - 1-2 sec bis Kommando wirkt OK
  - Protokoll *RTSP* wird dazu oft benutzt
- Zeitanforderung für die noch zu übertragenden Daten:  
Rechtzeitig zum unterbrechungsfreien Abspielen

# Streaming Live Multimedia

## Beispiele:

- ◆ Internet Radio Talkshow
- ◆ Live Sportereignis

## Streaming

- ◆ Playback Puffer
- ◆ Playback kann um einige 10 sec verzögert werden
- ◆ Auch bei Playback gibt es Rechtzeitigkeitsanforderungen

## Interaktivität

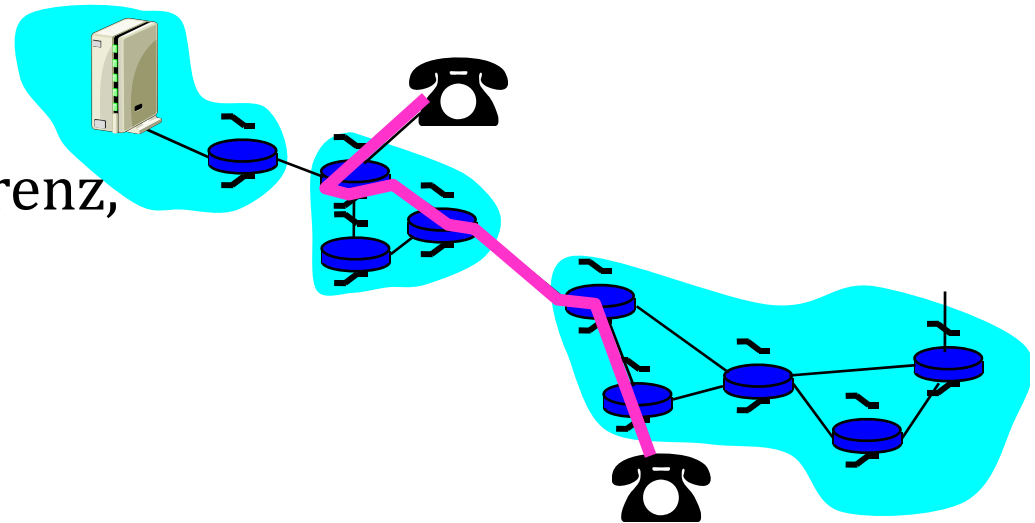
- ◆ Vorwärtsspulen nicht möglich
- ◆ Pause und Rückwärtsspulen möglich



# Interaktive Realzeit-Multimediadaten

## ◆ Anwendungen:

IP Telephonie, Video-Konferenz,  
Verteilte interaktive Welten



## • Anforderungen an Übertragungsverzögerung:

– Audio: < 150 msec gut, < 400 msec OK

- Muss Anwendungsbearbeitung und Transferzeit umfassen
- Höhere Verzögerung stören die Interaktivität

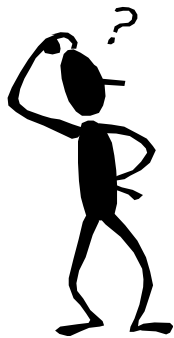
## • Sitzungsaufbau

– Wie veröffentlicht der Angerufene seine IP Adresse, Port-Nummer und Codieralgorithmen?

# Multimedia über das heutige Internet

## TCP/UDP/IP: “Best-Effort Service”

- *keine* Garantien zu Verzögerungszeiten und Verlustfreiheit



ABER: Anwendungen brauchen  
Mindestgüte, um adäquat  
zu funktionieren



Heutige Anwendungen nutzen Techniken  
auf Anwendungsebene, um (so gut wie möglich)  
Verzögerungs- und Verlusteffekte zu mildern

# Streaming gespeicherter Multimediatechniken im Internet

Application-Level Streaming:

*„Das beste aus  
Best-Effort-Internet machen“*

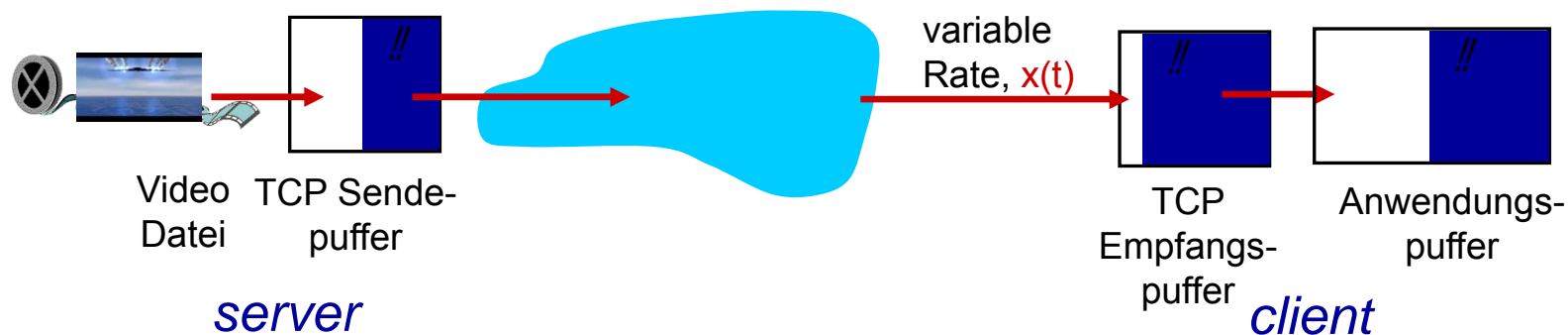
- Pufferung auf Client-Seite
- Benutzung von UDP statt TCP
- Codierung und Kompression

- Jitter entfernen
- Dekompression
- Fehler-Verschleierung
- GUI-Bedienknöpfe



# Internet Multimedia: Streaming

Streaming über HTTP (und TCP):



- Ankunftsrate schwankt durch TCP Fluss- und Überlastkontrolle, Übertragungswiederholungen
- Längere Verzögerung bei der Ausgabe glättet den Verkehr
- Kombination HTTP/TCP wird durch Firewalls zugelassen

Alternative:

- UDP mit konstanter Senderate (aber Jitter im Netz)

# Dynamic Adaptive Streaming over HTTP (DASH)

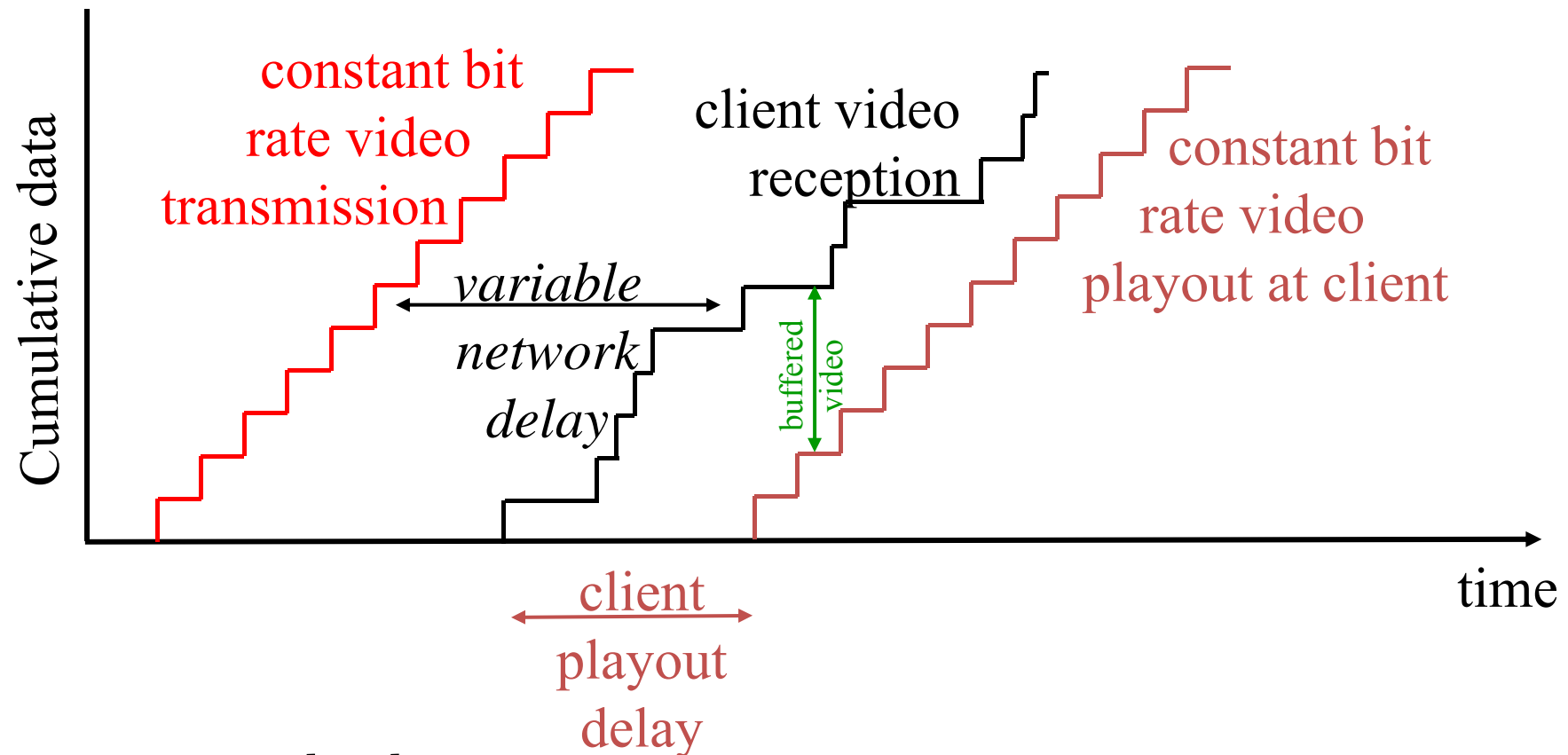
## ➤ *Server:*

- Unterteilt die Video-Datei in mehrere *chunks*
- *chunks* werden in verschiedenen Formaten (Kodierungen) vorgehalten
- *manifest Datei*: enthält URLs für die *chunks*

## ➤ *Client:*

- Misst die Bandbreite zwischen Server und Client periodisch
- Fordert Chunks nacheinander in bestimmter Qualität an
  - Auswahl der maximalen mit gegebener Bandbreite möglichen Kodierungsrate
  - Kodierung kann während der Übertragung wechseln
- Bestimmt wann, in welcher Form und von woher ein Chunk angefordert wird

# Streaming Multimedia: Client-seitige Pufferung



- Jitter-Ausgleich

# Nutzerkontrolle von Streaming Media: RTSP

---

## HTTP

- ◆ Nicht für Multimedia-Austausch gedacht
- ◆ Keine Kommandos für Vor- und Zurückspulen, Pause etc.
- ◆ Durch DASH Anpassung an Multimedia-Anwendungen

## Nicht enthalten:

- ◆ Keine Codierungs- und Kompressionsfestlegungen
- ◆ Keine Multimedia-Transfer Festlegungen (z.B. UDP, TCP)
- ◆ Keine Festlegungen zur Pufferung

## Real-time Streaming Protocol

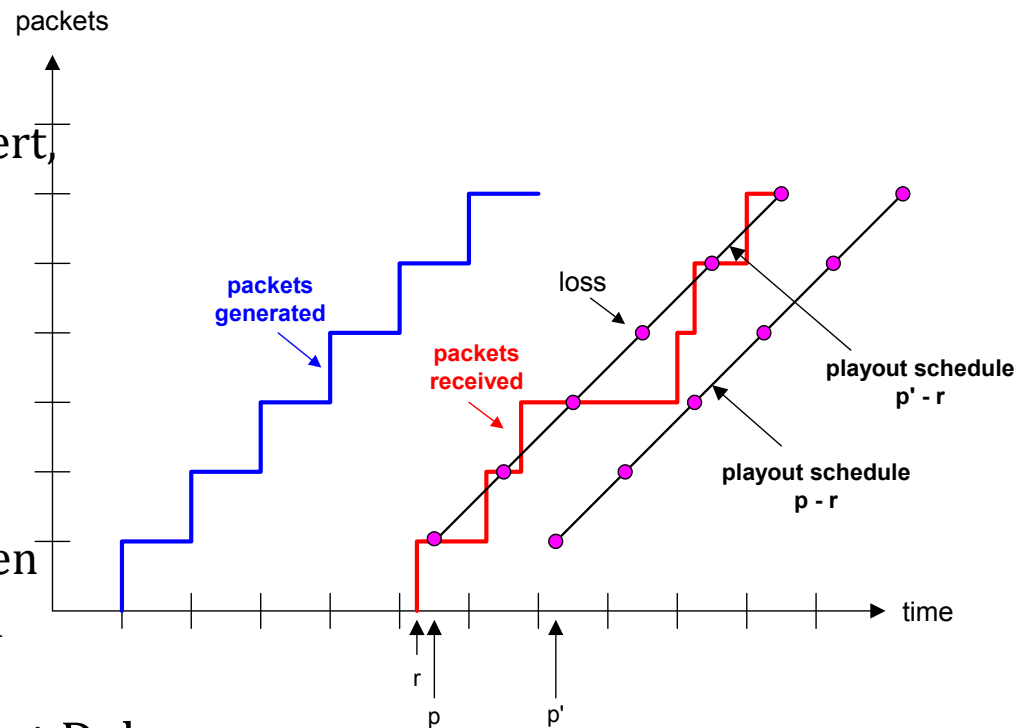
### RTSP: RFC 2326

- ◆ Client-Server Protokoll der Anwendungsschicht.
- ◆ Kommandos für Vor- und Zurückspulen, Pause etc.

*RTSP-PDUs werden in separater Verbindung ("Out of Band") übertragen*

# Interaktive Realzeit-Anwendung: Internet-Telephonie

- Je Richtung gibt es Sprech- und Pausenphasen
  - In den Sprechphasen werden alle 20 msec ein Paket generiert, das 160 Datenbyte enthält (entsprechend 8KByte/sec)
  - Jedes Paket wird als UDP-Datagramm gesendet
- UDP Datagramme können:
  - verloren gehen
  - zu langsam transferiert werden
  - 1-10% Verluste sind tolerabel



- Jitter-Behandlung: Fixed Playout Delay
  - Zeitstempel je Paket
  - Abspielen nach konstanter Verzögerungszeit
  - je größer diese Zeit, umso weniger Pakete kommen zu spät
  - je größer diese Zeit, umso weniger kommt ein Gespräch zustande
- Verbesserung: Adaptiver Playout Delay



# Behandlung von Paketverlusten

## Forward Error Correction (FEC):

### Einfaches Schema

- ◆ Für je  $n$  Pakete wird  $(n+1)$ -tes Paket als Parity-Vektor gesendet
  - Redundanz erhöht Bandbreite
  - ermöglicht Rekonstruktion eines verlorenen Pakets, wenn je  $n$ -Gruppe höchstens ein Paket verloren geht

Auswahl des Parameters  $n$ :  
Vergrößerung von  $n$  führt zu

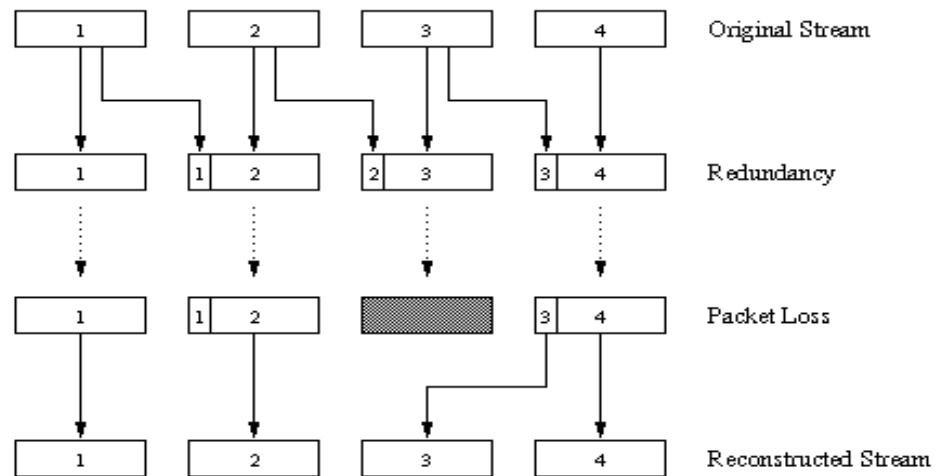
- geringeren Bandbreitenverlusten
- längeren Verzögerungszeiten bis zum Abspielen
- erhöhter Wahrscheinlichkeit, dass 2 oder mehr von  $n$  Paketen verloren gehen

# Behandlung von Paketverlusten

## Forward Error Correction (FEC):

### Flexibleres Schema

- ◆ Dem Datenstrom, der den Audiostrom mit guter Qualität codiert wird ein zweiter Datenstrom überlagert, der den Audiostrom mit schlechter aber kurzzeitig akzeptabler Qualität codiert



# Transfer mit dem Real-Time Protokoll (RTP)

## ◆ RTP (RFC 3550)

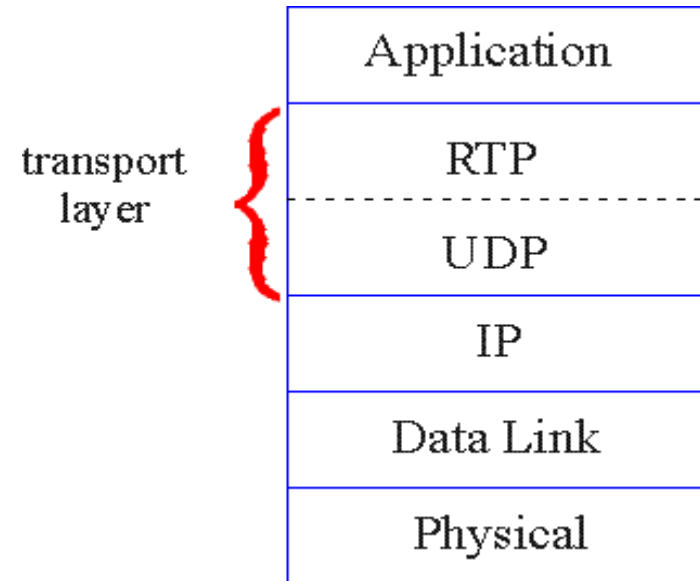
Paketformat für Datenpakete, die Audio- und Videodaten enthalten

- Typkennung für diese Nutzdaten
- Sequenznummer
- Zeitstempel

Transfer in UDP-Datagrammen

Interoperabilität zwischen zwei Anwendungsprozessen, die beide RTP benutzen und dieselben Codierungen verstehen.

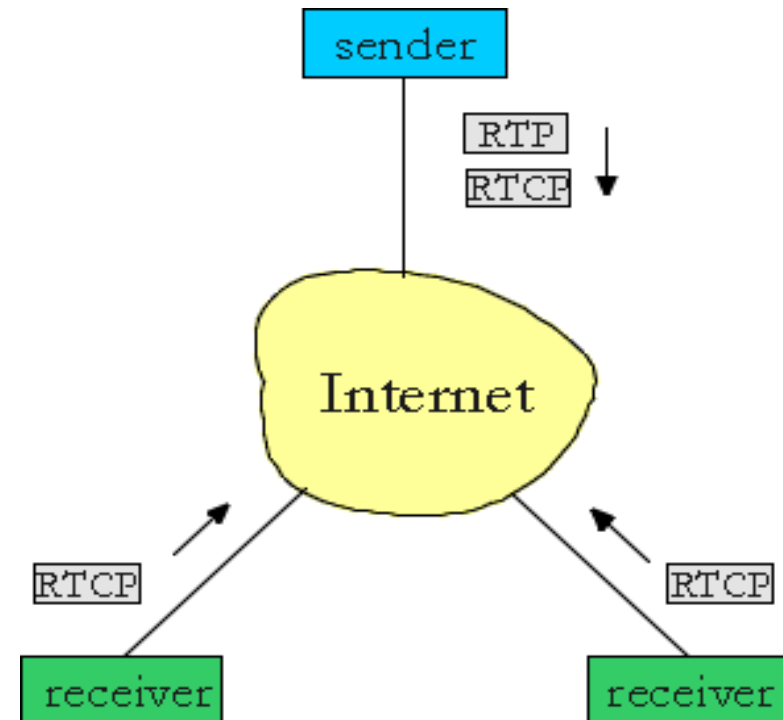
## ◆ **Keine** QoS-Mechanismen enthalten



Payload type 0: PCM mu-law, 64 kbps  
Payload type 3, GSM, 13 kbps  
Payload type 7, LPC, 2.4 kbps  
Payload type 26, Motion JPEG  
Payload type 31. H.261  
Payload type 33, MPEG2 video

# Real-Time Control Protokoll (RTCP)

- RTP: Medientransfer
- RTCP: Jeder RTP-Anwendungsprozess sollte periodisch RTCP-PDUs zu seinen entfernten Partnern senden, um Anpassungen zu ermöglichen:
  - Sender bzw. Empfänger-Report: Statistische Daten (Paketanzahl, Verlustanzahl, Jitter, ..)
  - Paare aus RTP-Stromzeitstempel und Paketerzeugungszeitstempel zur wechselseitigen Synchronisation von Strömen
- Adressierung typischerweise über Multicast-Adressen
  - RTP und RTCP benutzen dieselbe Gruppenadresse, aber verschiedene Port-Nummern



# Session Initiation Protokoll (SIP)

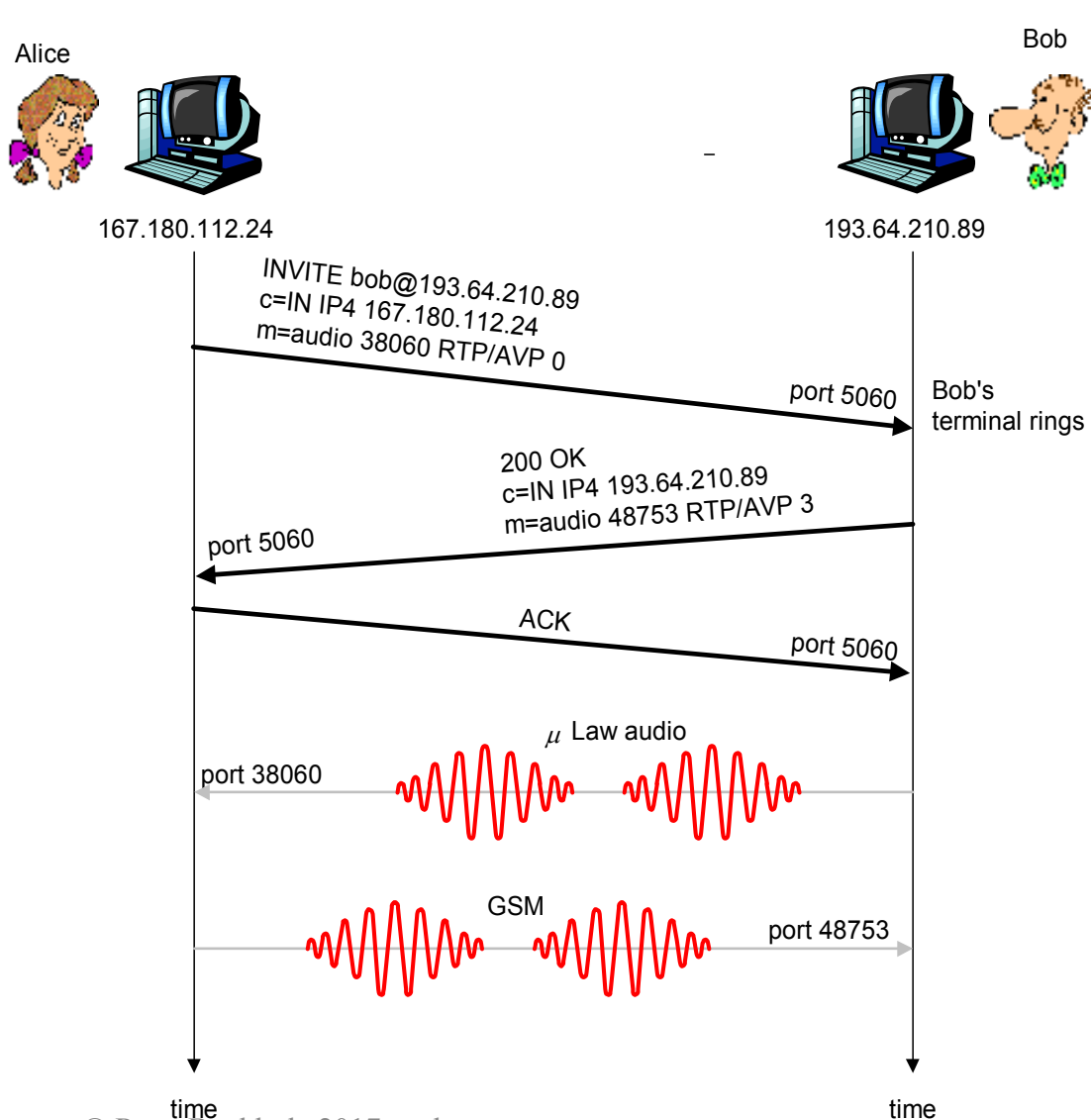
## **Vision**

- Jede Form von Telekommunikation (Telefonie, Videokonferenzen, ..) wird über das Internet abgewickelt.
- Adressaten werden durch Namen oder E-Mail-Adressen identifiziert, nicht mehr durch Telefonnummern
- Der Angerufene kann unabhängig davon erreicht werden, ob er momentan am Arbeitsplatz-PC sitzt, auf Reisen ist, oder ..

## **⇒ Session Initiation Protocol mit den Diensten**

- Anruf-Erzeugung
  - Rufen des Partners
  - Abstimmen der Medien und der Codierung
  - Beenden der Sitzung
- Ermittlung der aktuellen IP-Adresse des Partners
- Verbindungsverwaltung
  - Medien- und Codec-Änderungen
  - Neue Partner dazu nehmen
  - Anrufweiterleitung und Pausieren

# Rufaufbau bei bekannter IP-Adresse



- ❑ SIP-Nachricht von Alice beinhaltet ihre Portnummer, IP-Adresse, bevorzugte Codierung (PCM)
- ❑ Bob antwortet 200 OK mit seiner Portnummer, IP-Adresse, bevorzugter Codierung (GSM)
- ❑ SIP-Nachrichten können über TCP oder UDP gesendet werden (hier RTP/UDP).
- ❑ default SIP Portnummer ist 5060.

# SIP – weitere Aspekte

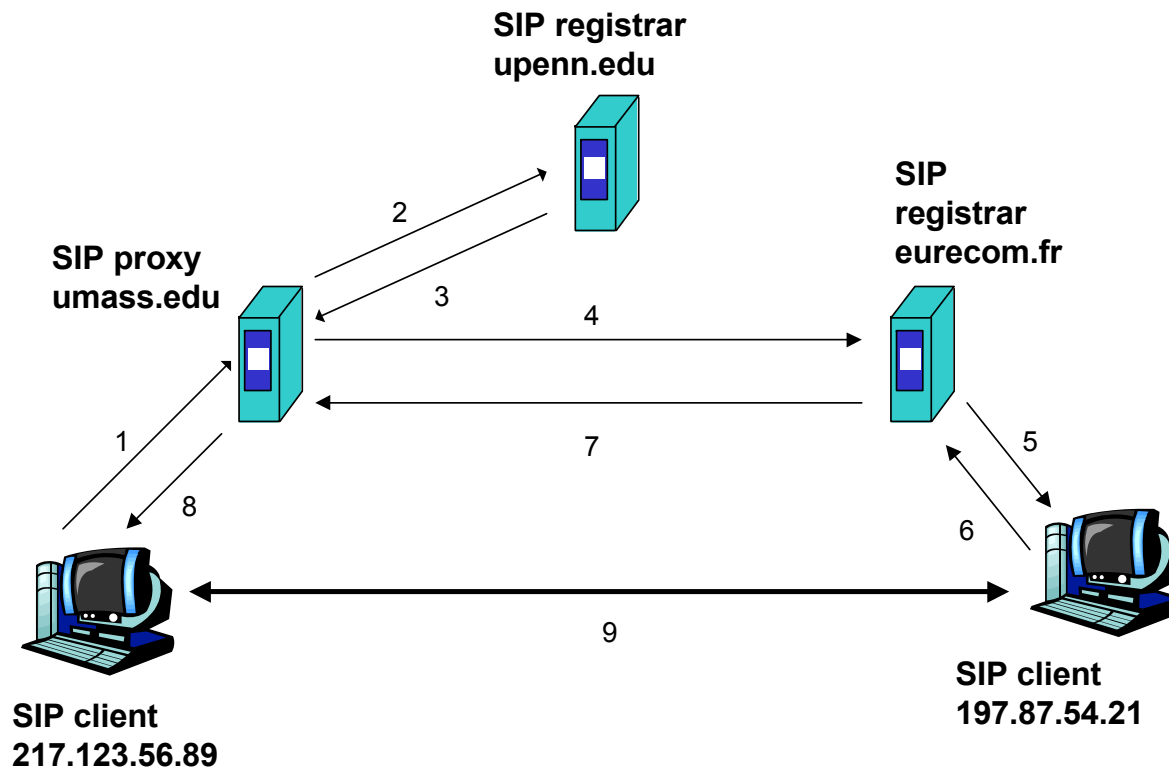
## HTTP-artiges Nachrichtenformat

```
INVITE sip:bob@domain.com
  SIP/2.0
Via: SIP/2.0/UDP
  167.180.112.24
From: sip:alice@hereway.com
To: sip:bob@domain.com
Call-ID:
  a2e3a@pigeon.hereway.com
Content-Type:
  application/sdp
Content-Length: 885

c=IN IP4 167.180.112.24
m=audio 38060 RTP/AVP 0
```

- Aushandlung des Codecs:
  - Wenn Bob über keinen PCM-Encoder verfügt, sendet er eine Fehlermeldung (606), auf die Alice mit einer neuen Anfrage mit geändertem Encoder antworten kann
  - Weitere mögliche Fehlermeldungen:  
503 service unavailable  
600 busy  
....

# Namensübersetzung und Nutzerlokation



## ➤ SIP Registrar Server

- Nutzer melden sich dort jeweils aktuell an

## ➤ SIP Proxy Server

- Übernimmt die Weiterleitung der SIP-Nachrichten für einen Nutzer (u.U. über eine Kette von Proxies)

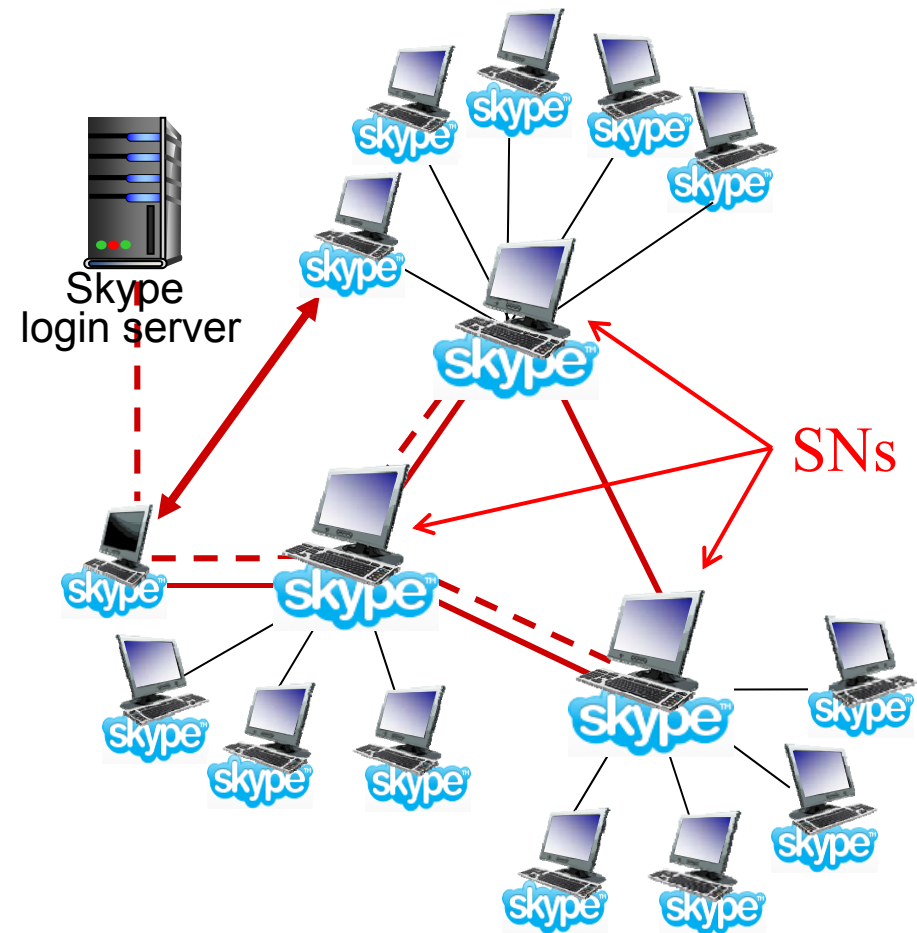
**Aufrufer jim@umass.edu möchte eine Verbindung mit keith@upenn.edu herstellen**

- (1) Jim sendet eine Nachricht INVITE zum SIP Proxy umass.edu.
- (2) Proxy leitet die Nachricht an den SIP Proxy upenn.edu weiter.
- (3) Sip Registrar upenn.edu antwortet mit der aktuellen Adresse keith@eurecom.fr.
- (4) Umass.edu sendet INVITE an Sip Registrar eurecom.fr.
- (5) Eurecom.fr leitet die Nachricht INVITE an 197.87.54.21 weiter (Keith's client läuft dort)
- (6-8) SIP Antwort wird zurückgeschickt (9) Medium wird direkt zwischen Clients ausgetauscht.



# Proprietärer Ansatz zur Internettelefonie: Skype

1. Eintritt in das Skype-Netz durch Verbindung zu einem SN (super Node)
2. Login beim Skype Login Server
3. Anfrage nach der IP-Adresse des Anzurufenden vom SN
4. Direkte Verbindung zwischen den beiden Gesprächspartnern



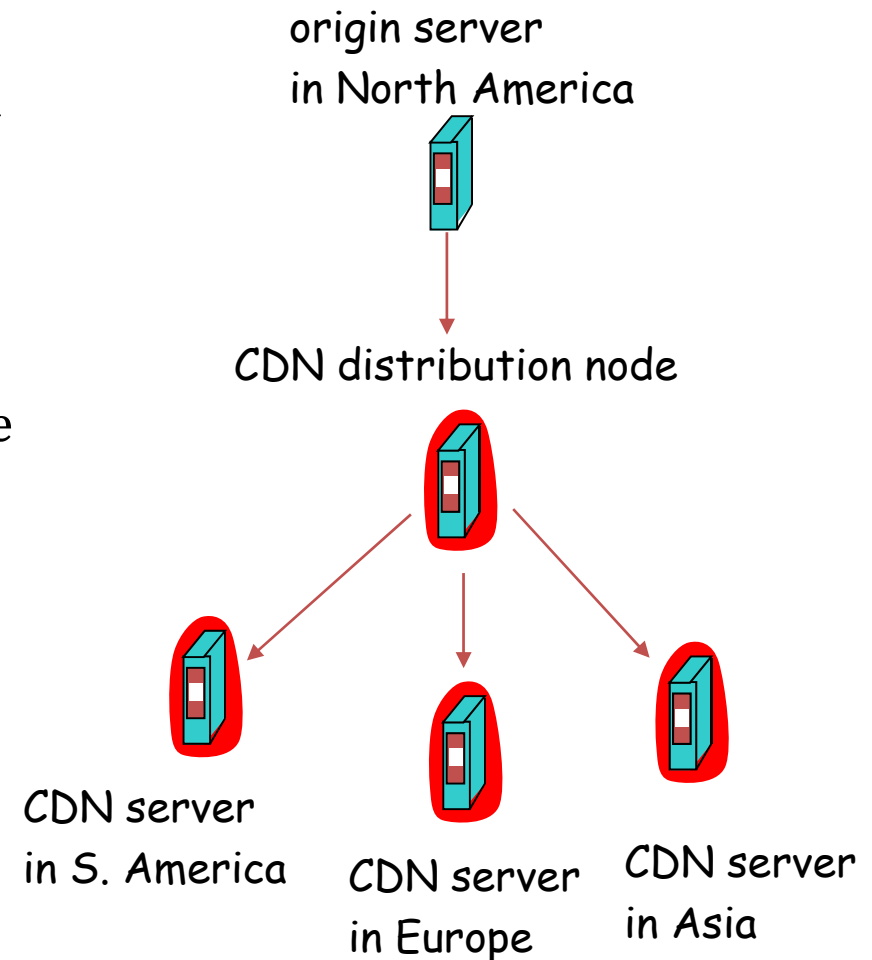
# Content Distribution Networks (CDNs)

## ➤ **Replikation**

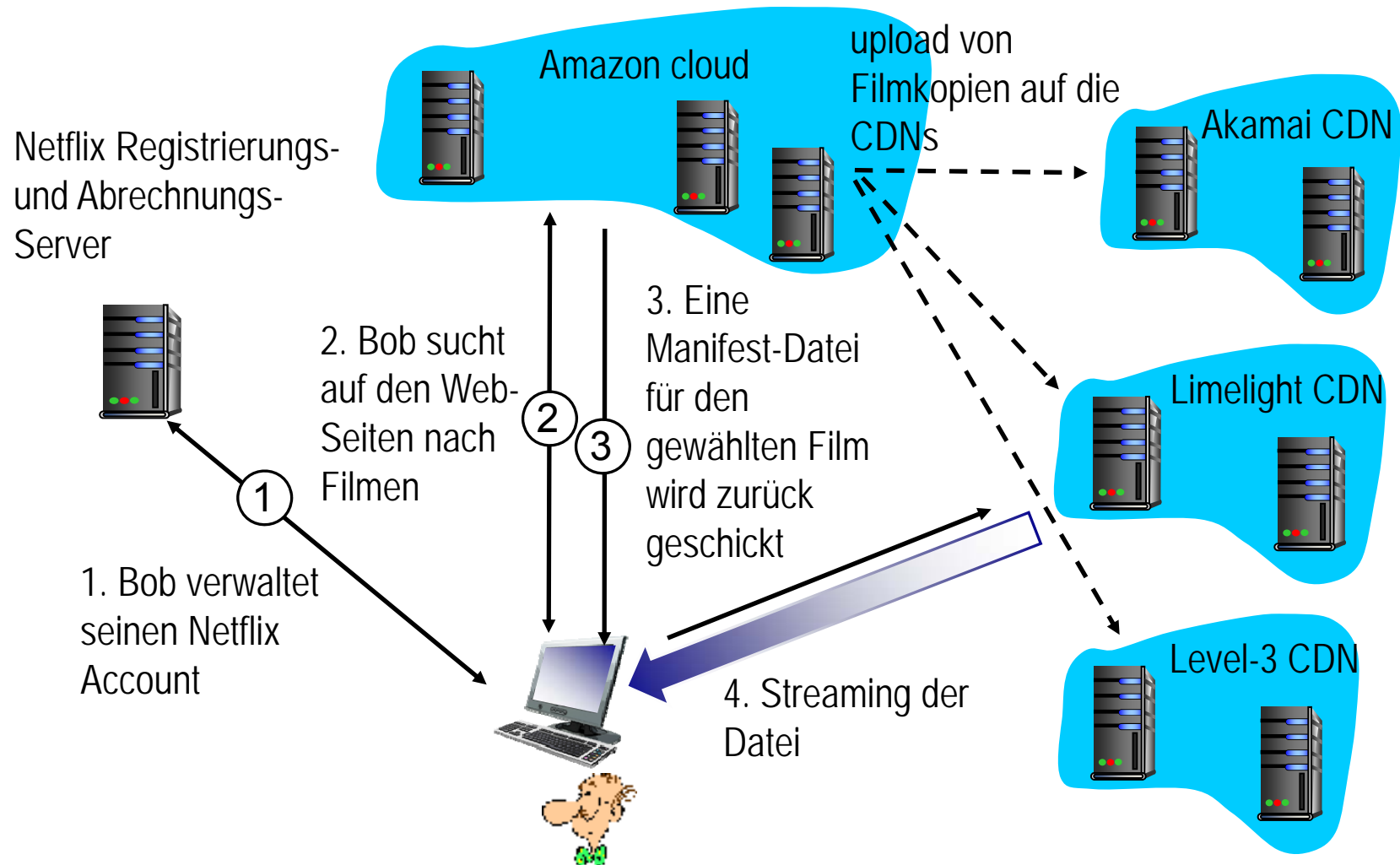
um Transfers zu sparen, werden die Inhalte in Kopien auf vielen Servern gespeichert

## ➤ **Interessante Aspekte**

- Auswahl und Verteilung der Inhalte
- Finden des nächsten Servers für einen Kunden
- Aktualisierung der Server bei Updates
- Gemeinsame Teilwege beim Ausliefern derselben Inhalte an verschiedene Kunden



# Beispiel für ein CDN: Netflix



# Evolution des Internets für Multimedia-Anwendungen

## Integrated Services

- ◆ Grundlegende Änderungen im Internet, so dass Anwendungen Bandbreite reservieren können
- ◆ Neue, komplexe Software in Hosts und Routern

## Laissez-Faire

- ◆ Keine besonderen Änderungen
- ◆ Ausbau, wenn Bandbreite benötigt
- ◆ Multimedia und Gruppenkommunikation über Anwendungssysteme
  - Application Layer

## Differentiated Services

- ◆ Wenige Änderungen im Internet
- ◆ Dienste
  - Erste Klasse
  - Zweite Klasse

### ◆ **Audio-Übertragungsrate**

- CD: **1.411 Mbps**
- MP3: **96, 128, 160 kbps**
- Internet telephony: **5.3 - 13 kbps**

### ◆ **Video-Übertragungsrate**

- MPEG 1 (CD-ROM) **1.5 Mbps**
- MPEG2 (DVD) **3-6 Mbps**
- MPEG4 (oft im Internet verwendet)  
**< 1 Mbps**

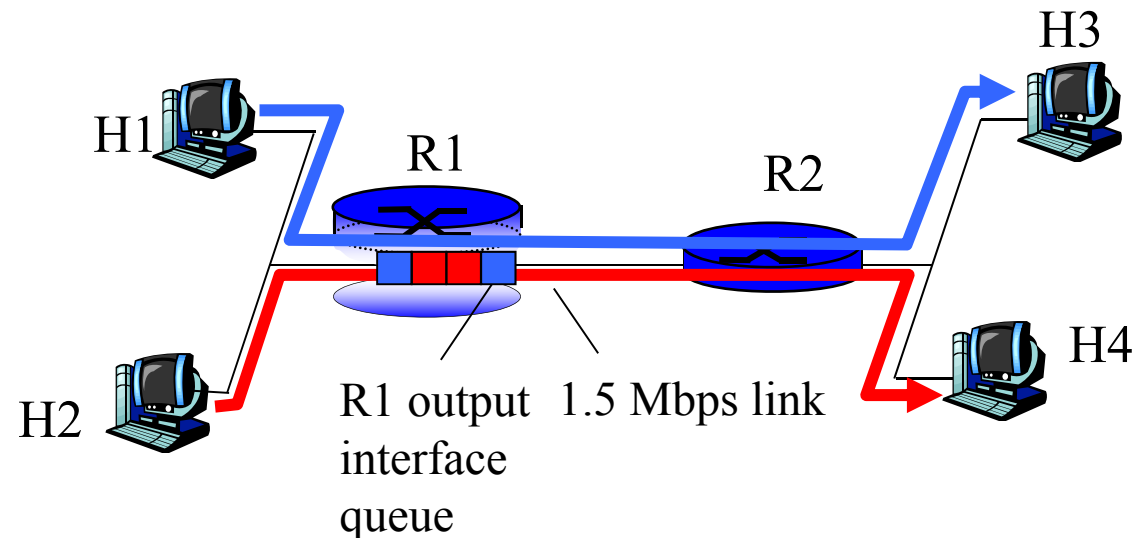
# Verbesserte Dienstgüte in IP Netzen

**Internet bisher:** “Best Effort – das Beste draus machen”

**Zukünftig:** Next Generation Internet mit QoS Garantien

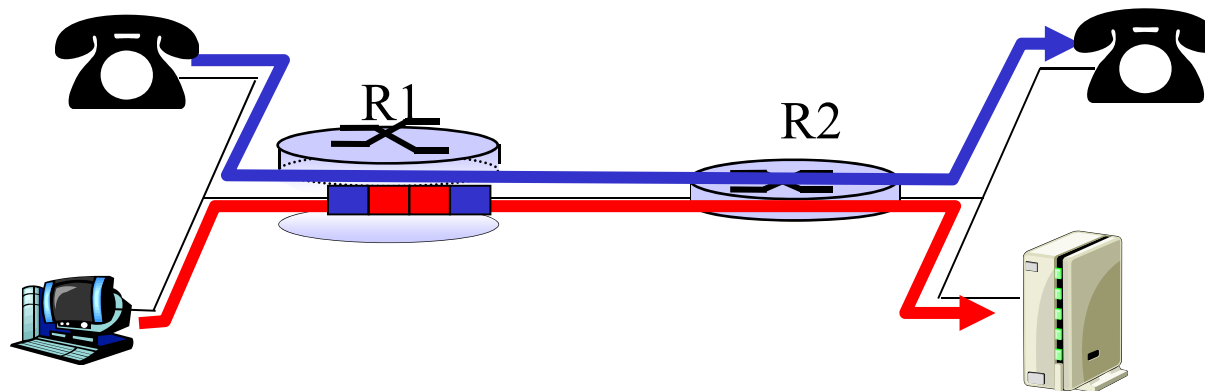
- **RSVP:** Signalisierung für Ressourcenreservierungen
- **Differentiated Services:** Priorisierungen
- **Integrated Services:** Feste Garantien

◆ Grundprobleme des Ressourcensharings und der Staubildung sind schon sichtbar an:



# Prinzipien für QoS-Garantien

- ◆ Beispiel: 1Mbps IP-Telephonie und FTP nutzen einen 1.5 Mbps Link gemeinsam
  - FTP-Burst können Router verstopfen und Audio-Verluste bewirken
  - Priorität für Audio vor FTP wäre eine Lösung

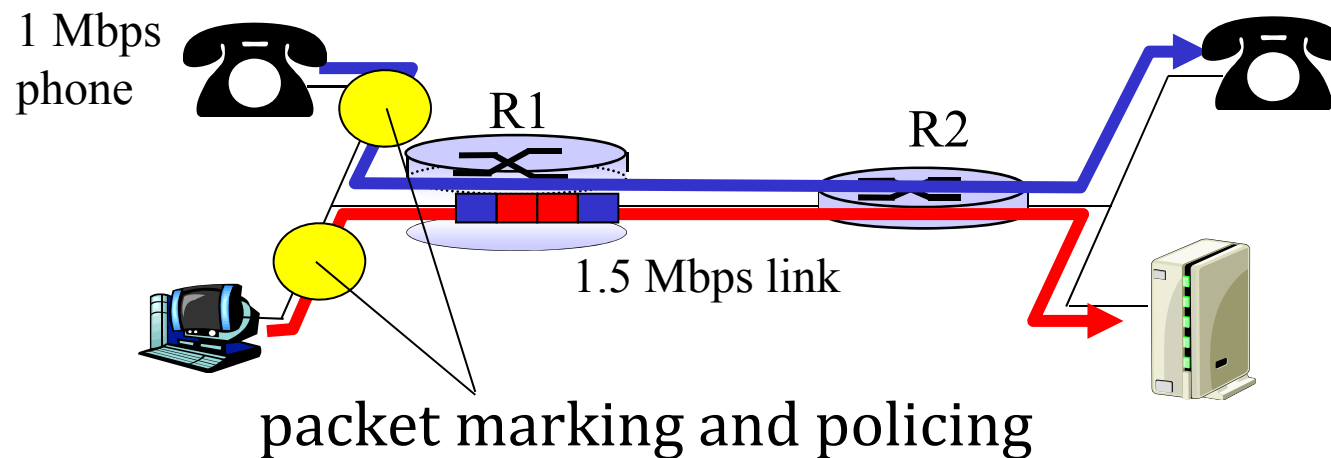


## Prinzip 1

Pakete werden markiert, damit die Router zwischen verschiedenen Verkehrsklassen unterscheiden können

# Prinzipien für QoS-Garantien

- ◆ Anwendung weist Fehlverhalten auf (z.B. Audio sendet mit mehrfacher Rate)
  - Policing (Reglementierung): Setze durch, dass die Audioquelle ihre maximale Rate nicht überschreitet
- ◆ Markieren und Policing an der Netz-Grenze (ähnlich ATM Netzinterface)

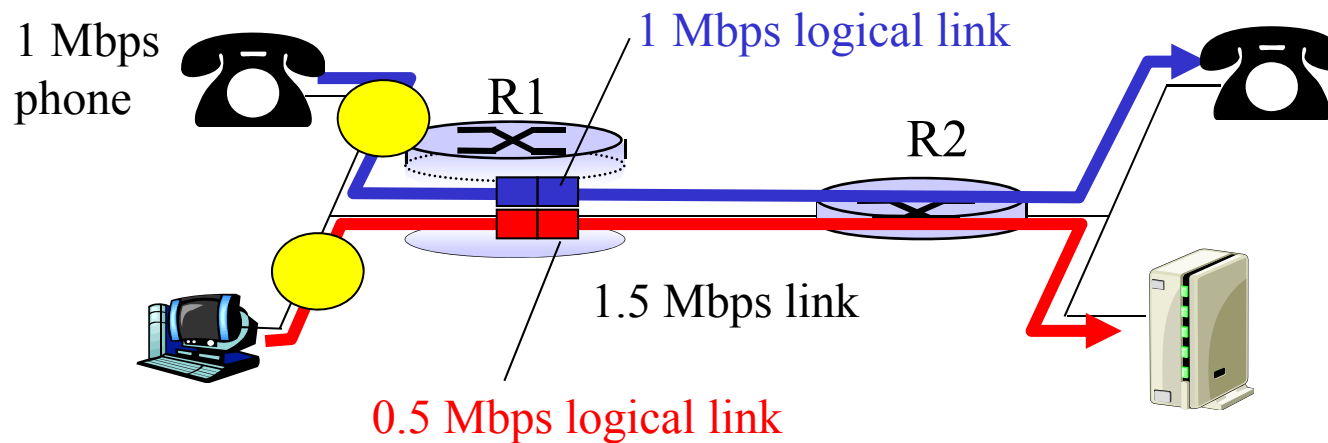


## Prinzip 2

Schütze eine Klasse vor Fehlverhalten (Überlastung des Netzes) durch andere: **Isolation**

# Prinzipien für QoS-Garantien

- ◆ Feste Bandbreiten-Reservierung ist keine gute Lösung: Ineffizienz



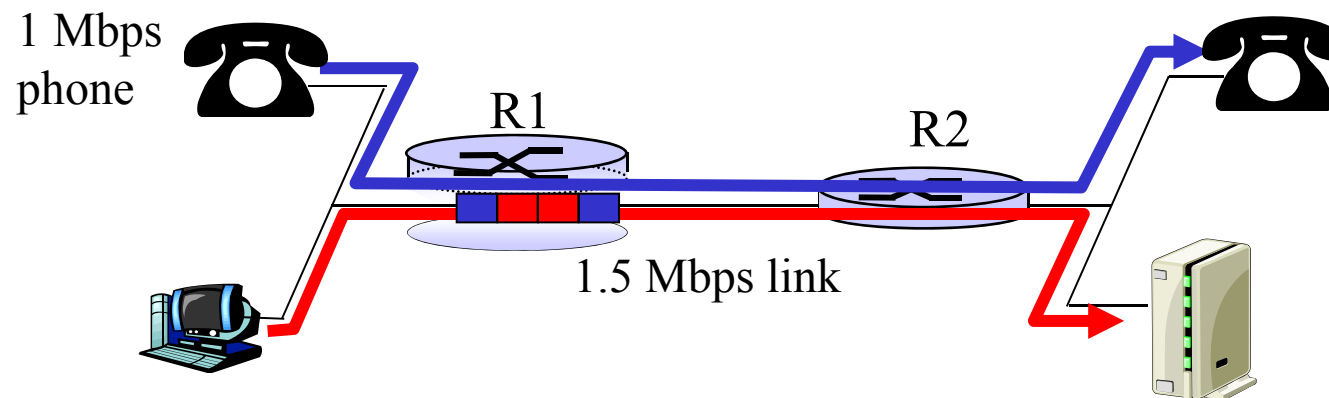
## Prinzip 3

Die Ressourcen sollen trotz Isolation möglichst effizient mehrfach genutzt werden.



# Prinzipien für QoS-Garantien

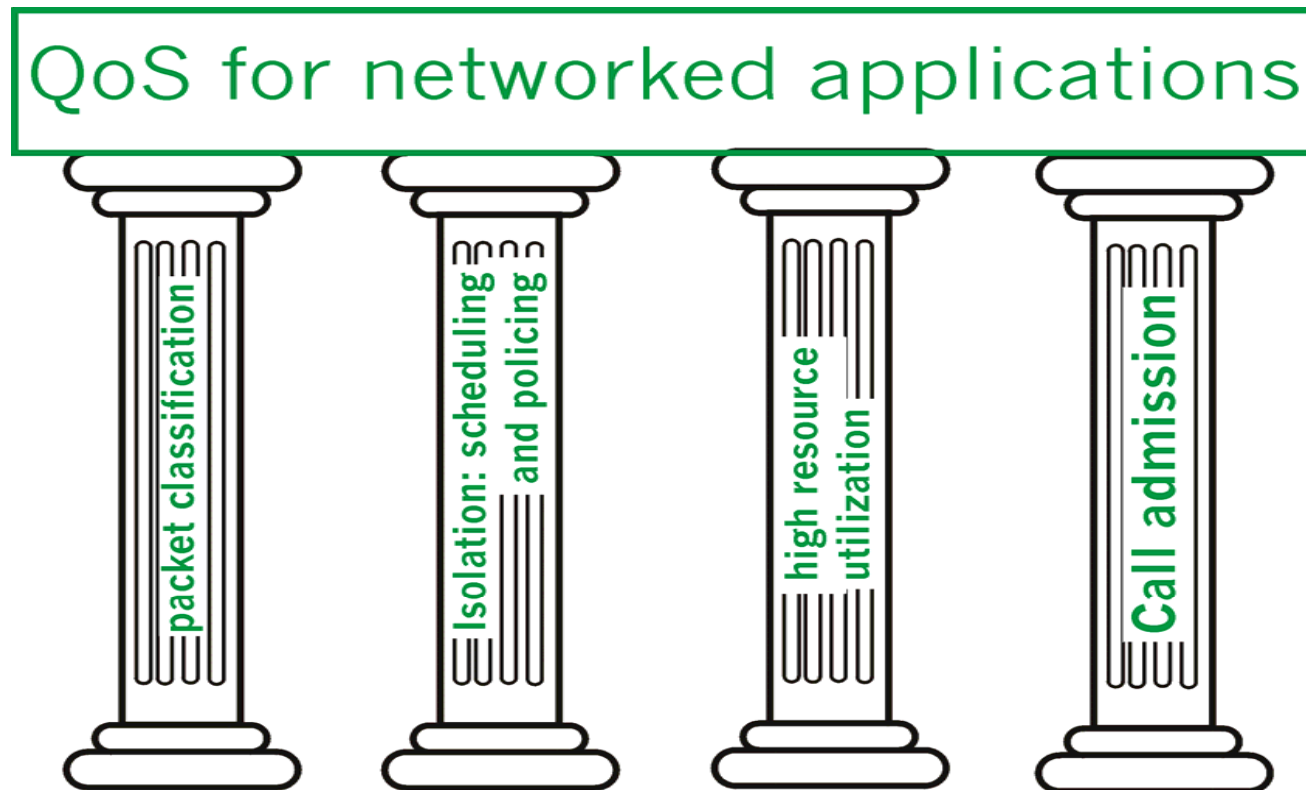
- ◆ *Auf den Boden der Tatsachen:*  
Man kann nicht mehr übertragen, als die Verbindung zulässt.



## Prinzip 4

Call Admission: Ein Fluss deklariert seinen Bedarf.  
Das Netz entscheidet, ob es den Fluss zulassen kann.

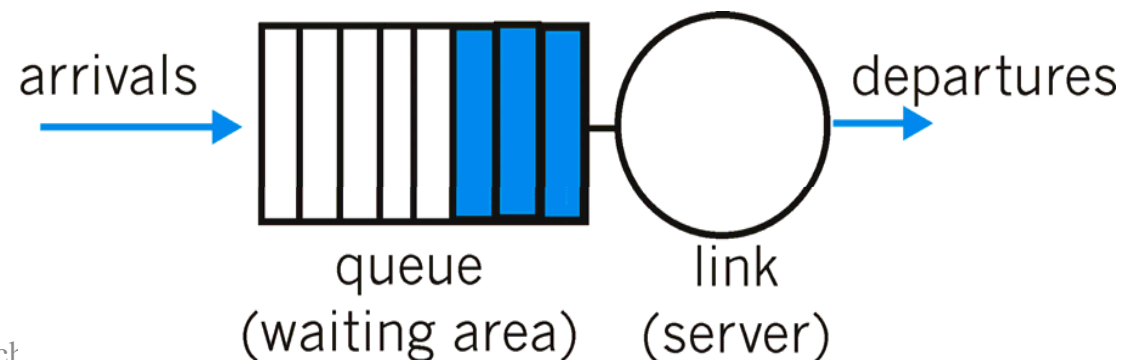
# Prinzipien für QoS-Garantien: Zusammenfassung



Im Folgenden: Entsprechene Mechanismen

# Scheduling und Policing Mechanismen

- **Scheduling:** Einplanung und Auswahl des nächsten auf Link zu sendenden Pakets
- **FIFO (first in first out) Scheduling:** Senden in Empfangsreihenfolge
  - **Discard Policy:** Falls ein ankommendes Paket auf eine volle Queue trifft: Welches Paket soll gelöscht werden?
    - Tail Drop: ankommendes Paket
    - Priorität: Prioritätskennungen, nieder priorres Paket
    - Random: zufällige Auswahl



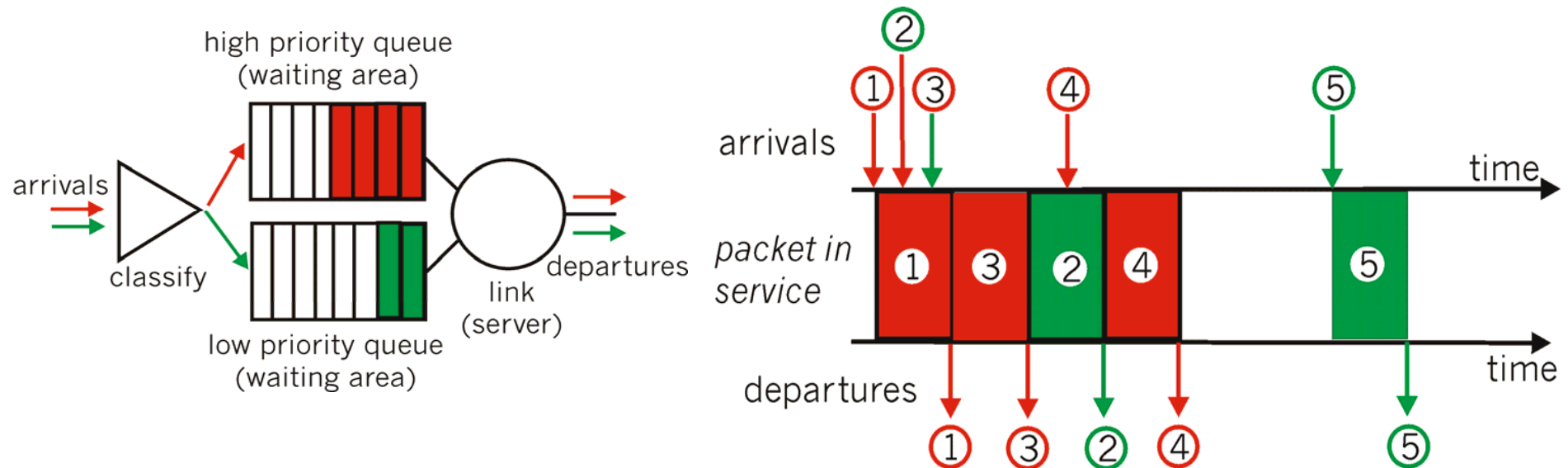
# Scheduling Mechanismen

**Priority Scheduling:** Sende höchstprioreres Paket als nächstes

➤ mehrere Prioritätsklassen

**Problem: Fairness**

- Prioritätskennung im Paketheader, Portnummer, Protokolltyp, etc.



**Andere Strategien (vgl. Prozessorscheduling)**

- Round Robin
- Weighted Fair Queuing

# Policing Mechanismen

Ziel: Zur Laufzeit soll der Paketstrom so begrenzt werden, dass ausgemachte Schranken nicht überschritten werden

Schranken für:

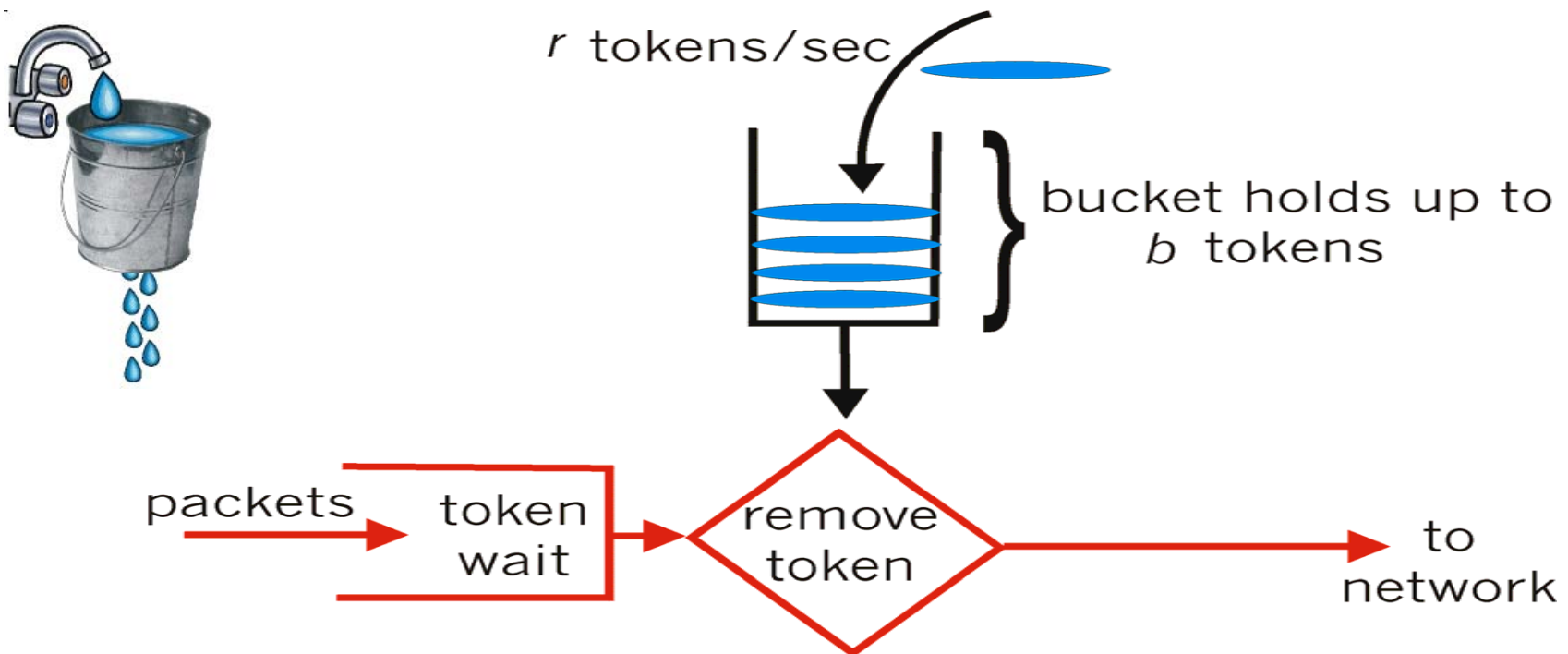
- *(Langfristige) mittlere Senderate*
- *Spitzenrate*
- *(Maximale) Burst-Größe*

Mechanismen sollen für Nutzer nachvollziehbar sein.

# Policing Mechanisms: Leaky Bucket Verfahren

## Begrenze Burst-Größe und mittlere Rate

(Idee: Der lecke Eimer – Zufluss und Abfluss, Zufluss darf, solange Eimer nicht überläuft, größer als Abfluss sein (Burst), muss aber im Mittel kleiner gleich Abfluss sein)



# IETF – Internet: Integrated Services (IntServ)

- Architektur, um QoS-Garantien für individuelle Anwendungsanforderungen in IP-Netzen zu unterstützen
- Mittel: Ressourcen-Vorabreservierung, Router verwalten “Virtuelle Verbindungen”
- Neue Verbindungen müssen zugelassen und können abgelehnt werden:

## ***Call Admission***

### Fragestellung:

Kann ein neuer Fluss zugelassen werden, ohne die Leistungsgarantien an bestehende Flüsse zu gefährden?

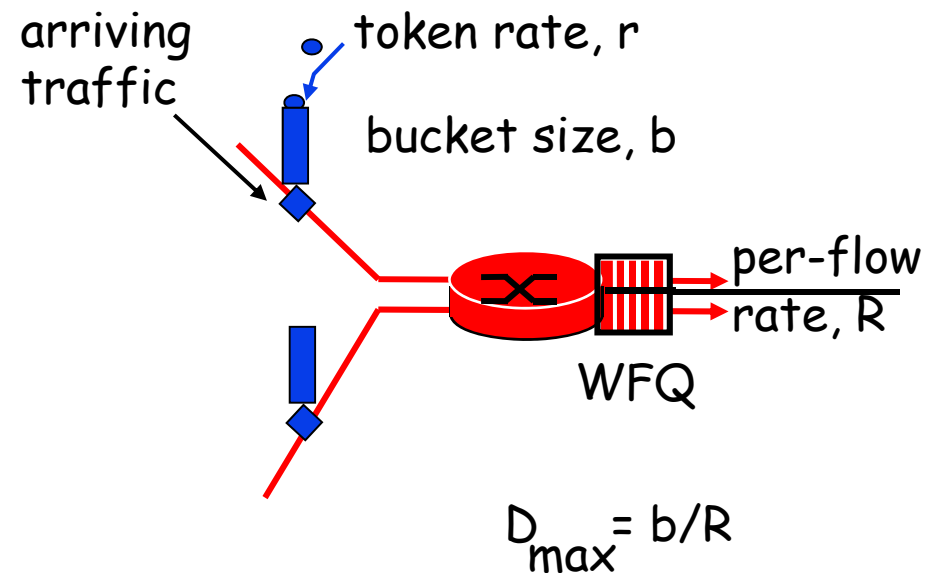
# Intserv QoS: Dienstmodelle [RFC 2211, RFC 2212]

## Guaranteed Service:

- Worst Case Verkehrslast durch Source Policing begrenzt (Leaky Bucket)
- Paketverzögerung ist begrenzt

## Controlled Load Service:

- Netz stellt eine QoS zur Verfügung, die derselbe Fluss annähernd auch von einem unbelasteten Netz bekäme





# IETF – Internet: Differentiated Services (DiffServ)

## Probleme bei Intserv:

- ❖ **Skalierbarkeit:** Bei großer Flussanzahl werden Router durch die Verwaltung der Flüsse übermäßig belastet
- ❖ **Flexible Dienstmodelle:** Intserv bietet nur 2 Klassen an.

Man möchte gerne “qualitative” Dienstklassen

- Relative Dienst-Unterscheidung: Platin-, Gold- und Silber-Dienste

## DiffServ approach:

- ❖ Im Inneren des Netzes nur einfache Funktionen
- ❖ Komplexe Funktionen nur am Rand (Edge Router o. Host)
- ❖ Keine Service-Klassen direkt definiert, nur Funktionseinheiten gegeben, mit denen Services gebildet werden können

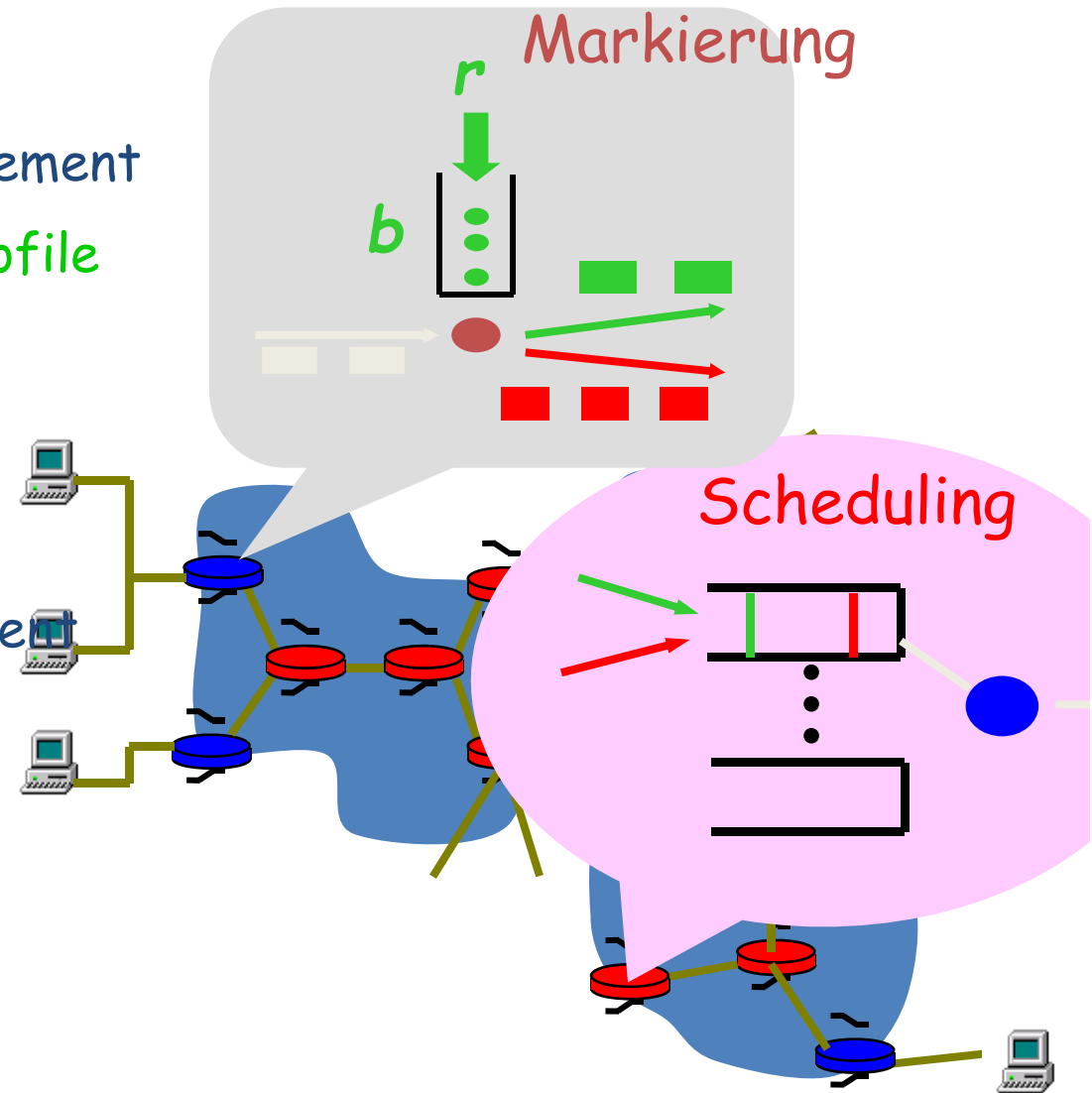
# IETF – Internet: Differentiated Services (DiffServ)

## Edge Router:

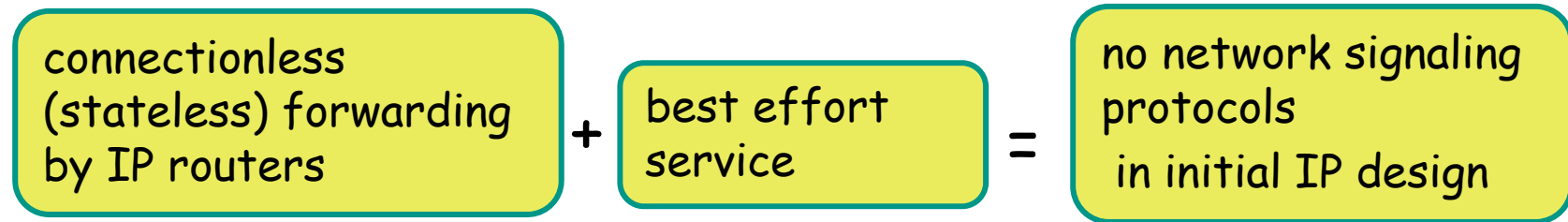
- ❑ Per-Fluss Verkehrsmanagement
- ❑ Markiert Pakete als **in-profile** oder **out-profile**

## Core Router:

- ❑ Per-Klasse Verkehrsmanagement
- ❑ Pufferung und Scheduling entsprechend Markierung
- ❑ **In-profile** Pakete werden vorgezogen
- ❑ Garantierte Weiterleitung



# Signalisierung im Internet



- ❖ **Signalisierung:** Austausch von Kontrollinformation im Telekommunikationsnetz, Beispiel: Wählzeichen beim Telefon
- ❖ **Neue Anforderung:** Reserviere Ressourcen entlang eines Ende-zu-Ende-Pfades, um Dienstgüte zu gewährleisten
- ❖ **RSVP:** Resource Reservation Protocol [RFC 2205]
  - ❖ “ ... allow users to communicate requirements to network in robust and efficient way.” i.e., signaling !
- ❖ Vorläufer als Internet-Signalisierungsprotokoll: ST-II [RFC 1819]

# RSVP: Funktion – Multimedia-Multicast-Verwaltung

---

## ❖ Signalisierung Sender → Netz

- *Path Message*: Router werden über Sender und seine Route informiert
- Path Teardown: Router löschen die Informationen zum Pfad

## ❖ Signalisierung Empfänger → Netz

- *Reservation Message*: Reserviere Ressourcen für Pfade zum Empfänger
- Reservation Teardown: Ziehe Reservierungen zurück

## ❖ Signalisierung Netz → Host: Fehlermeldungen (Pfad / Reservierung)

### Anmerkung:

Die Routenermittlung und Broadcast-Gruppen/Adressverwaltung werden außerhalb von RSVP abgewickelt

## ❖ Dynamik: Soft State - Konzept

- Bei Routern gespeicherte Zustandsinformationen verfallen nach Zeitintervall
- Sie müssen durch periodische RVSP-PDUs wieder aufgefrischt werden