

# Rechnernetze und verteilte Systeme (BSRvS II)

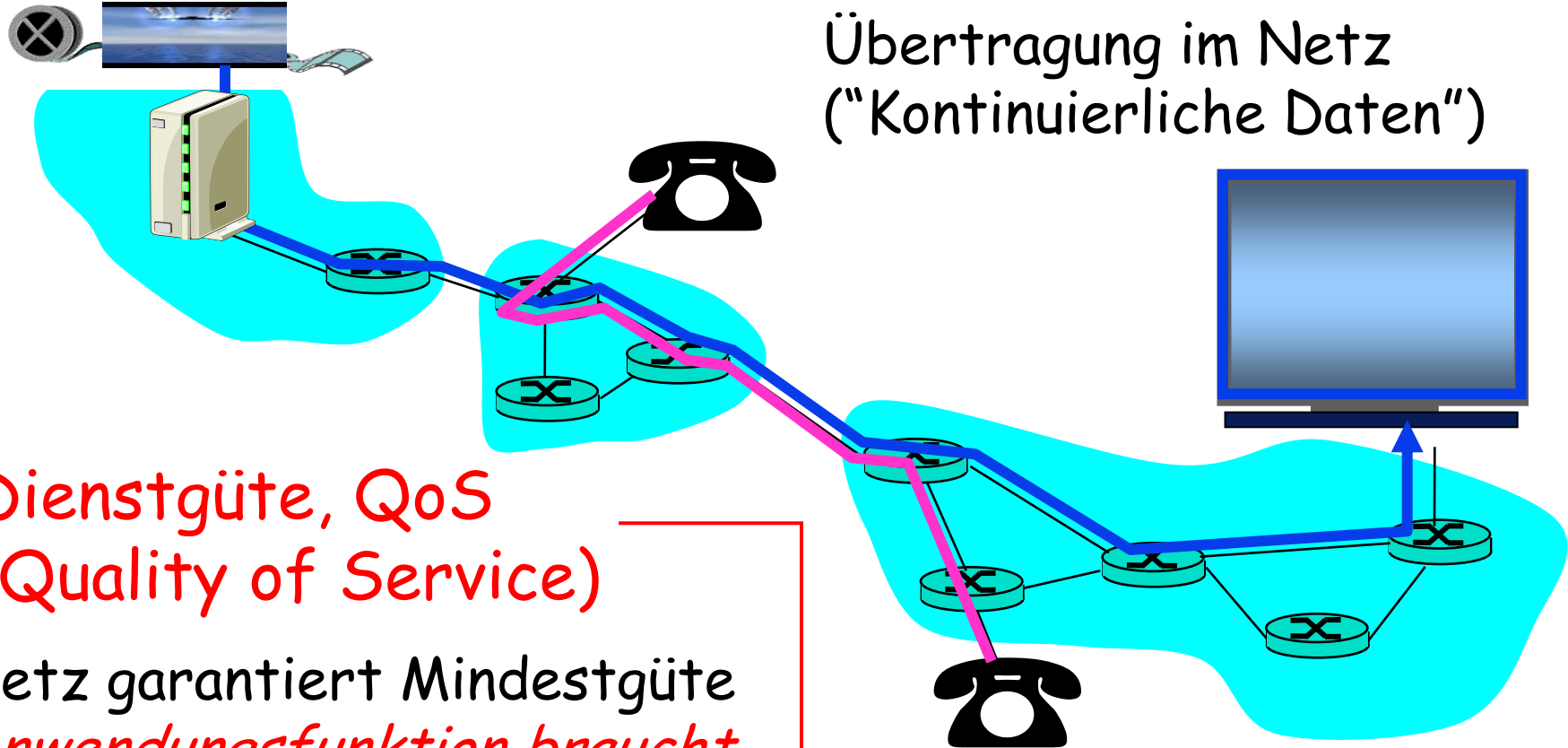
**Prof. Dr. Heiko Krumm**  
**FB Informatik, LS IV, AG RvS**  
**Universität Dortmund**



- **Computernetze und das Internet**
- **Anwendung**
- **Transport**
- **Vermittlung**
- **Verbindung**
- **Multimedia**
- **Sicherheit**
- **Netzmanagement**
- **Middleware**
- **Verteilte Algorithmen**

# Multimedia-Kommunikation: *Dienstgüte - QoS*

**Multimedia-Anwendungen:**  
Audio- und Video-  
Übertragung im Netz  
("Kontinuierliche Daten")



**Dienstgüte, QoS**  
(Quality of Service)

Netz garantiert Mindestgüte  
*Anwendungsfunktion braucht  
bestimmten Leistungspegel.*

# Kapitel 6: Übersicht

---

## 6.1 Multimedia Netzanwendungen

6.2 Audio- und Videostreaming

6.3 Realzeit Multimedia: Voice over IP / Internet-Telephonie

6.4 Protokolle für Realzeit-Anwendungen

RTP,RTCP,SIP

6.5 Multimedia-Verteilung im Netz

6.6 Über Best Effort hinaus

6.7 Scheduling und Policing Mechanismen

6.8 Integrated Services und Differentiated Services

6.9 RSVP

# 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

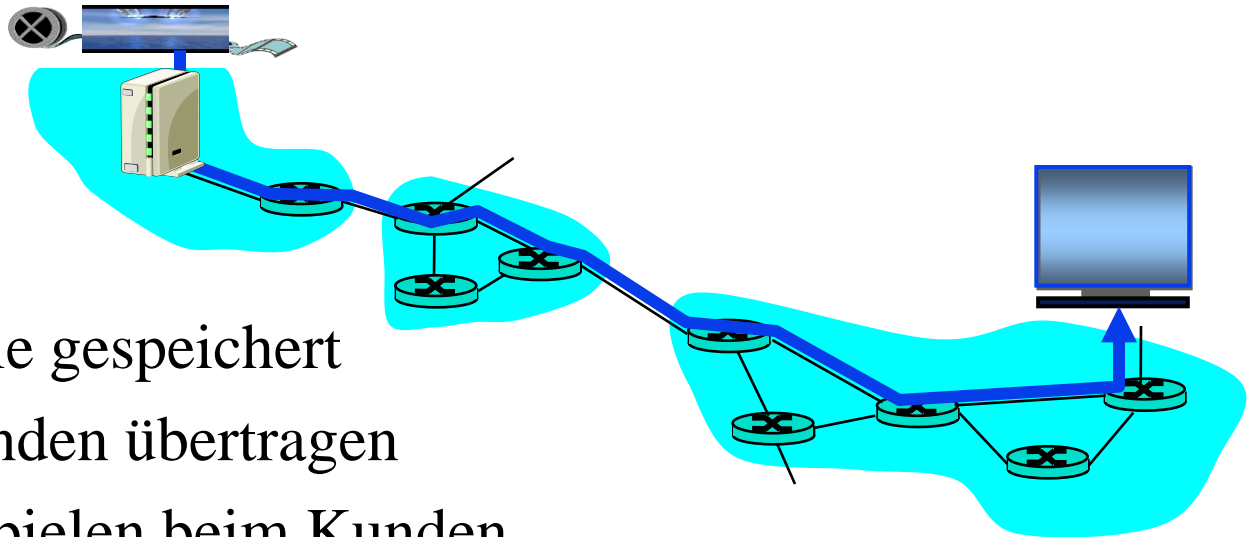
## **Jitter:**

**Veränderungen der Übertragungszeiten der Pakete eines Stroms**

## 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.

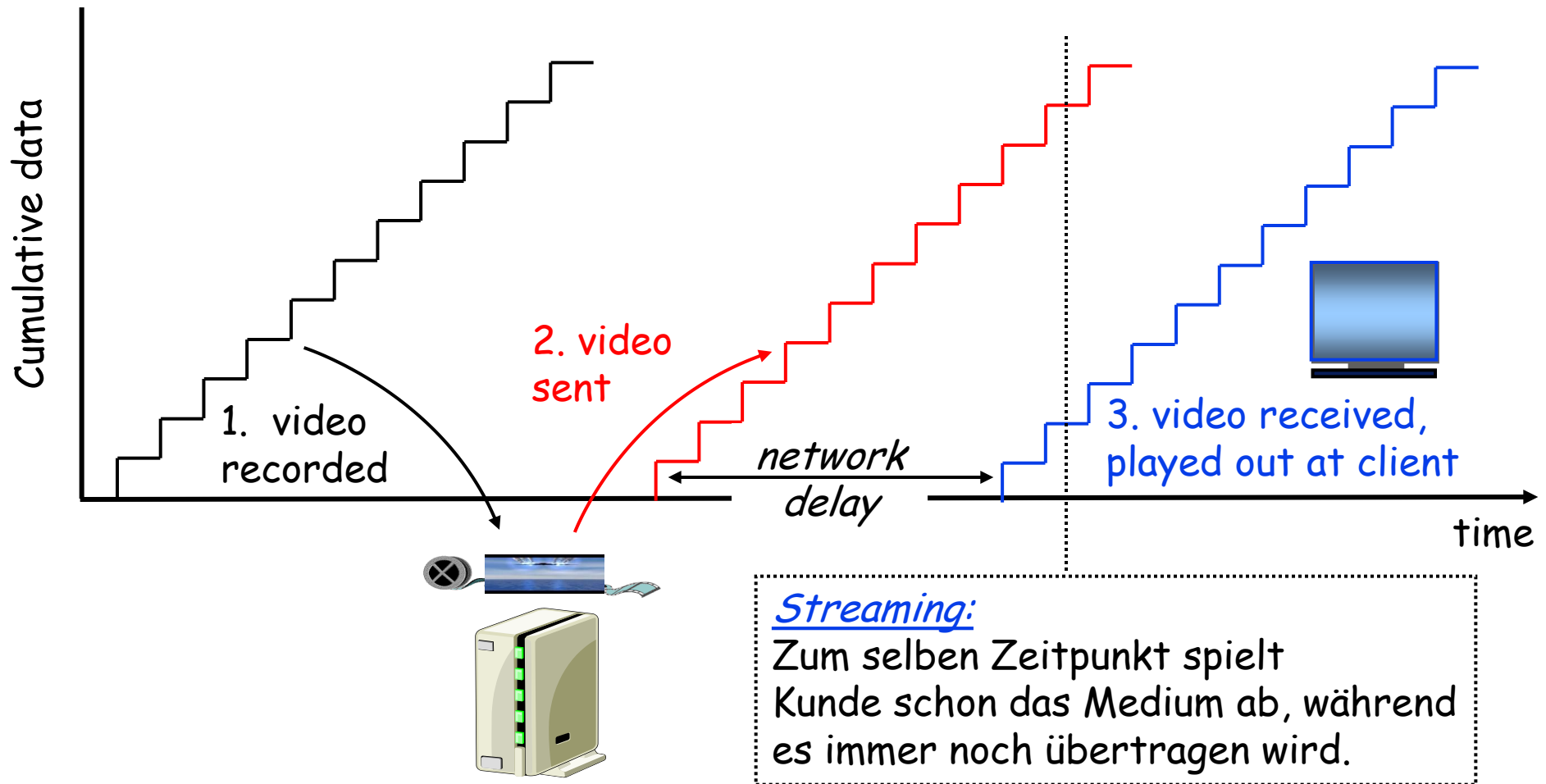
# Streaming gespeicherter Multimediadaten



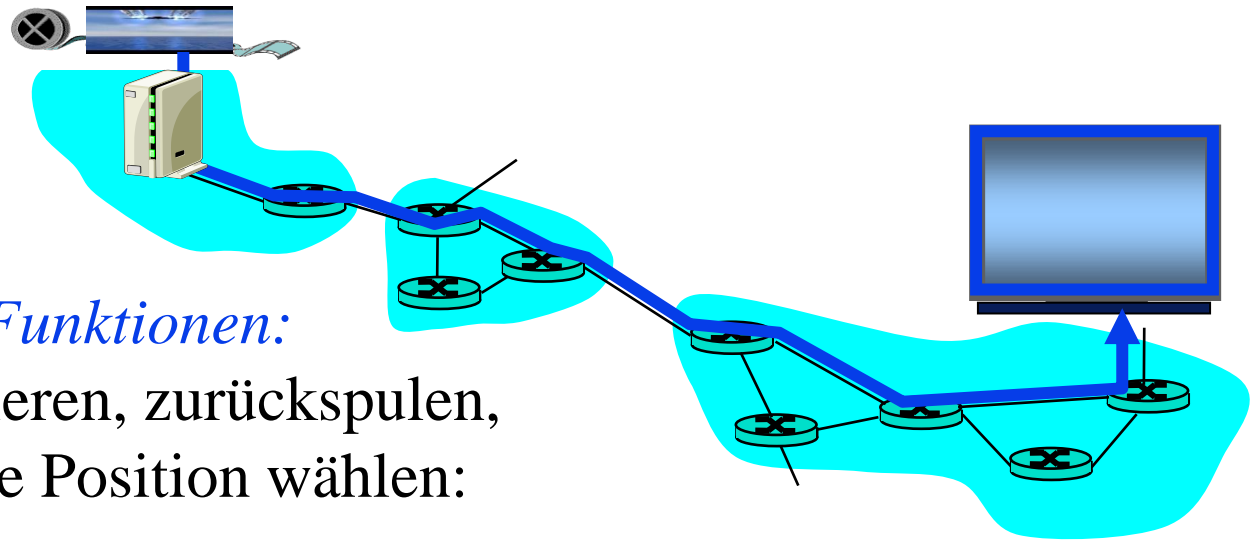
## Streaming:

- ◆ 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 (später)

## ◆ Zeitanforderung für die noch zu übertragenden Daten: Rechtzeitig zum unterbrechungsfreien Abspielen

# Streaming Live Multimedia

---

## Beispiele:

- ◆ Internet Radio Talkshow
- ◆ Live Sportereignis

## Streaming

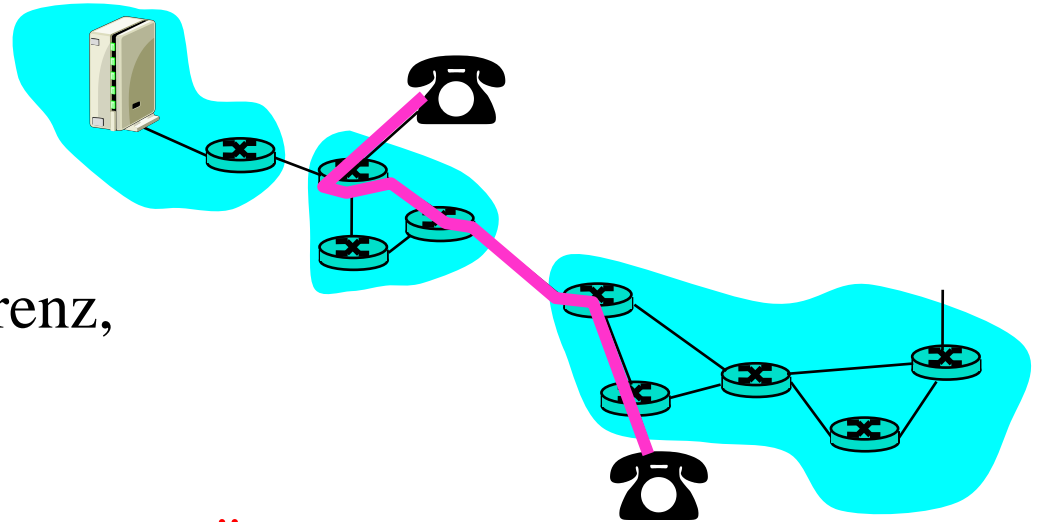
- ◆ Playback Puffer
- ◆ Playback kann um einige 10 sec verzögert werden
- ◆ Auch bei Playback gibt es Rechtzeitigkeitsanforderungen

## 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 als möglich)  
Verzögerungs- und Verlusteffekte zu mildern

# Streaming gespeicherter Multimediadaten im Internet

Application-Level Streaming:

*„Das beste aus*

*Best-Effort-Internet machen“*

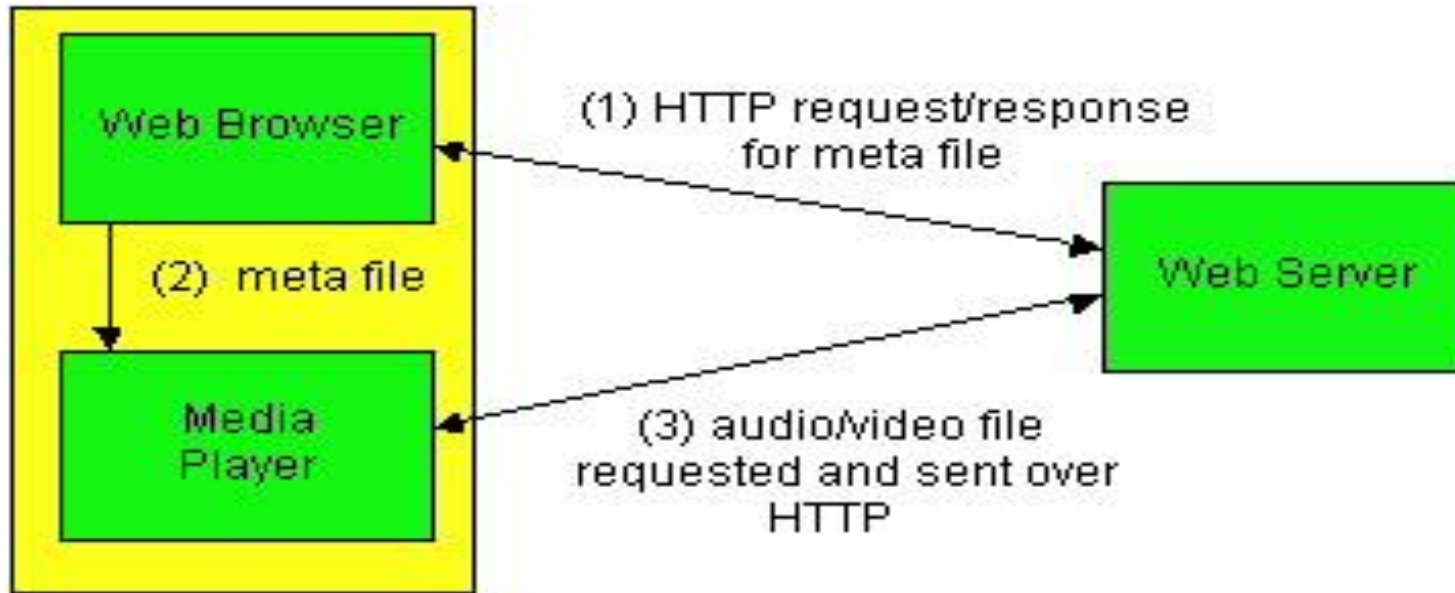
- Pufferung auf Client-Seite
- Benutzung von UDP statt TCP
- Codierung und Kompression

## Media Player

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

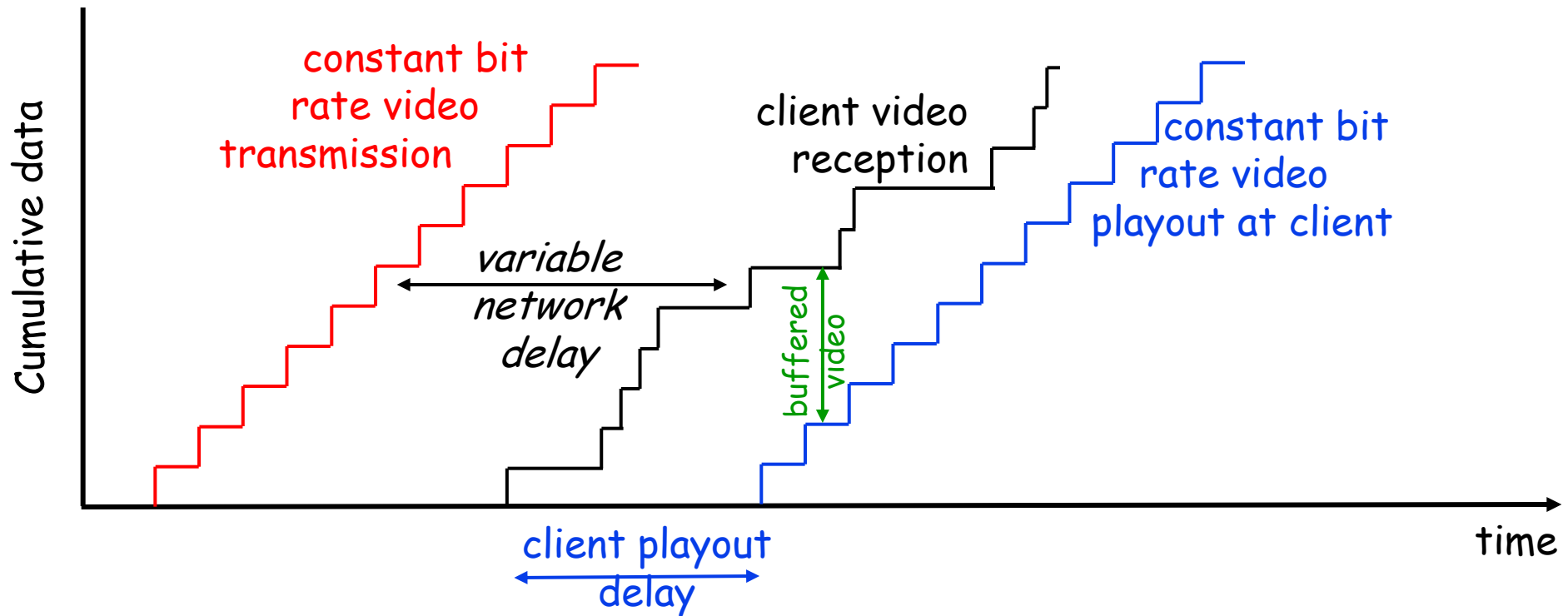


# Internet Multimedia: Streaming



- ◆ Browser GETs **Metafile**
- ◆ Browser startet Player, übergibt Metafile
- ◆ Player kontaktiert Server
- ◆ Server **sendet Strom** zu Audio/Video-Player
- ◆ Nicht-HTTP-Protokoll für Streaming möglich
- ◆ UDP statt TCP möglich

# 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.

## Real-time Streaming Protocol

### RTSP: RFC 2326

- ◆ Client-Server Application Layer Protokoll.
- ◆ Kommandos für Vor- und Zurückspulen, Pause etc.

## Nicht enthalten:

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

*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

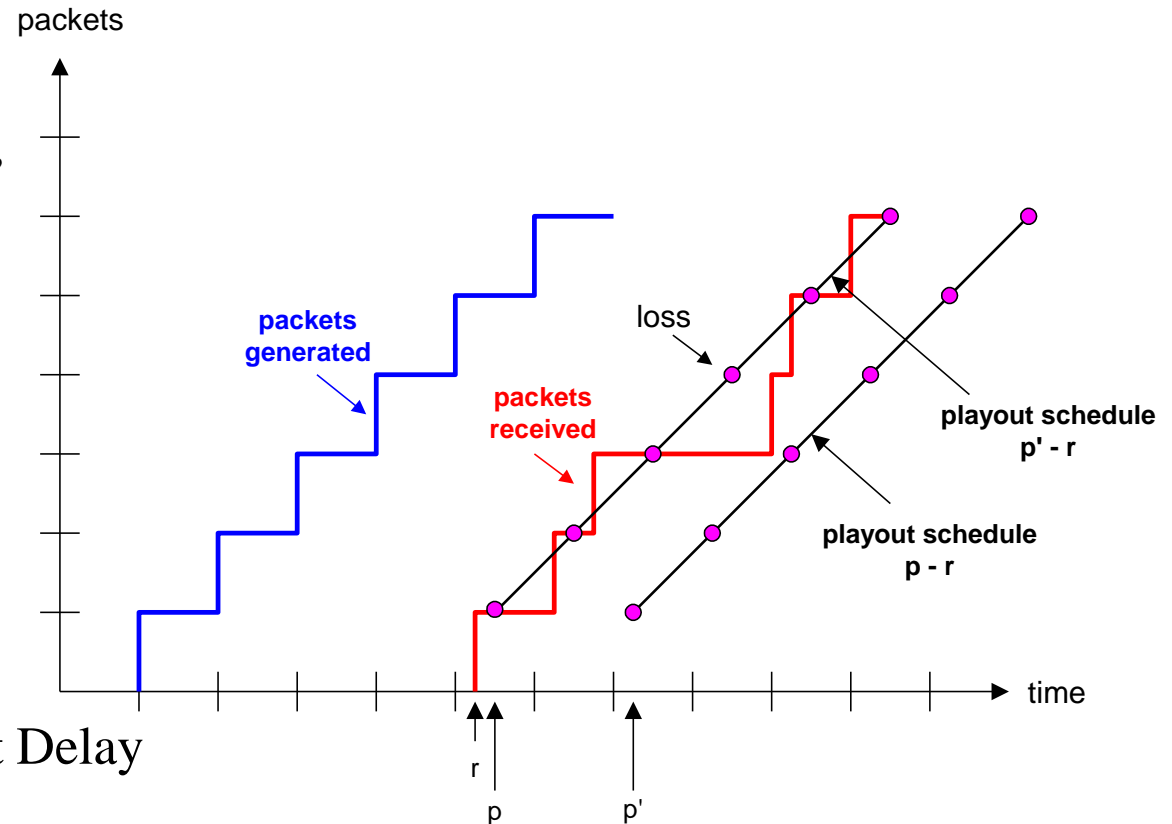
## ◆ UDP Datagramme können:

- 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

## 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 1889)

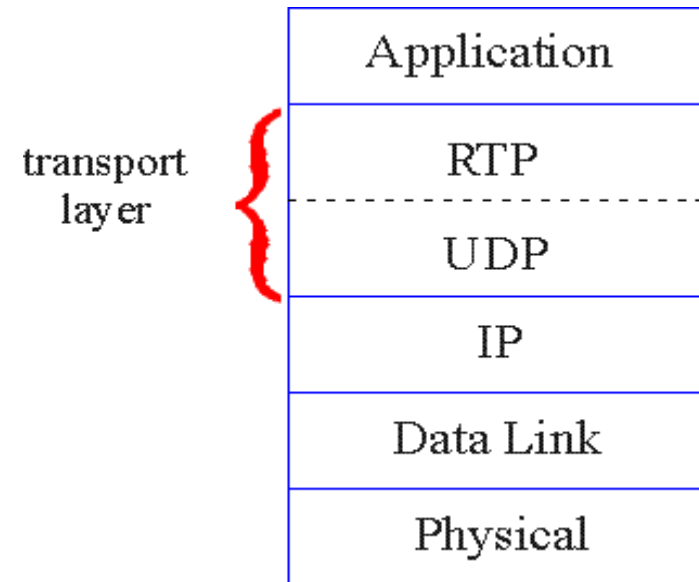
Paketformat für Datenpakete, die Audio- und Videodaten enthalten

- Typkennung für diese Nutzdaten
- Sequenznummer
- Zeitstempel

Transfer in UDP-Datagrammen

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

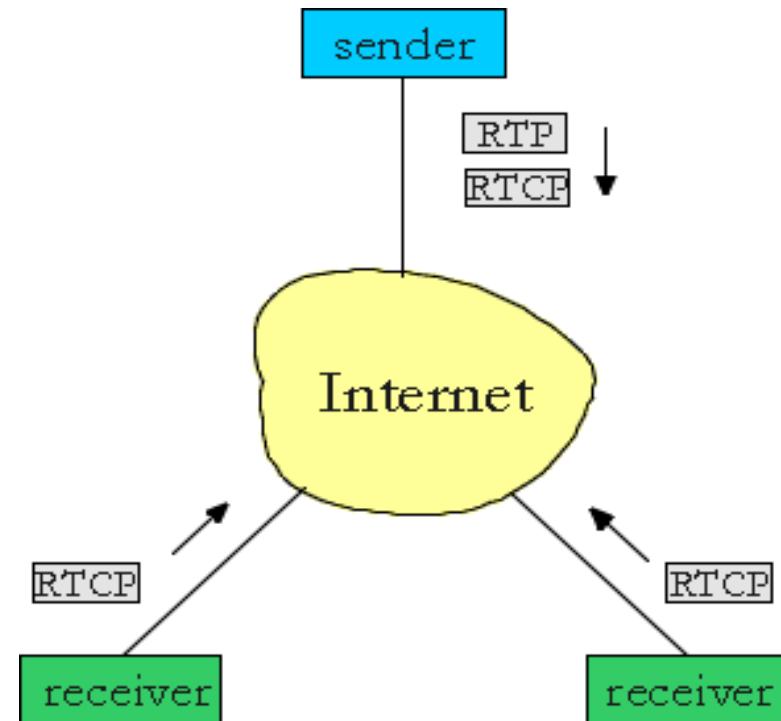
## ◆ **Keine** QoS-Mechanismen enthalten



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)

---

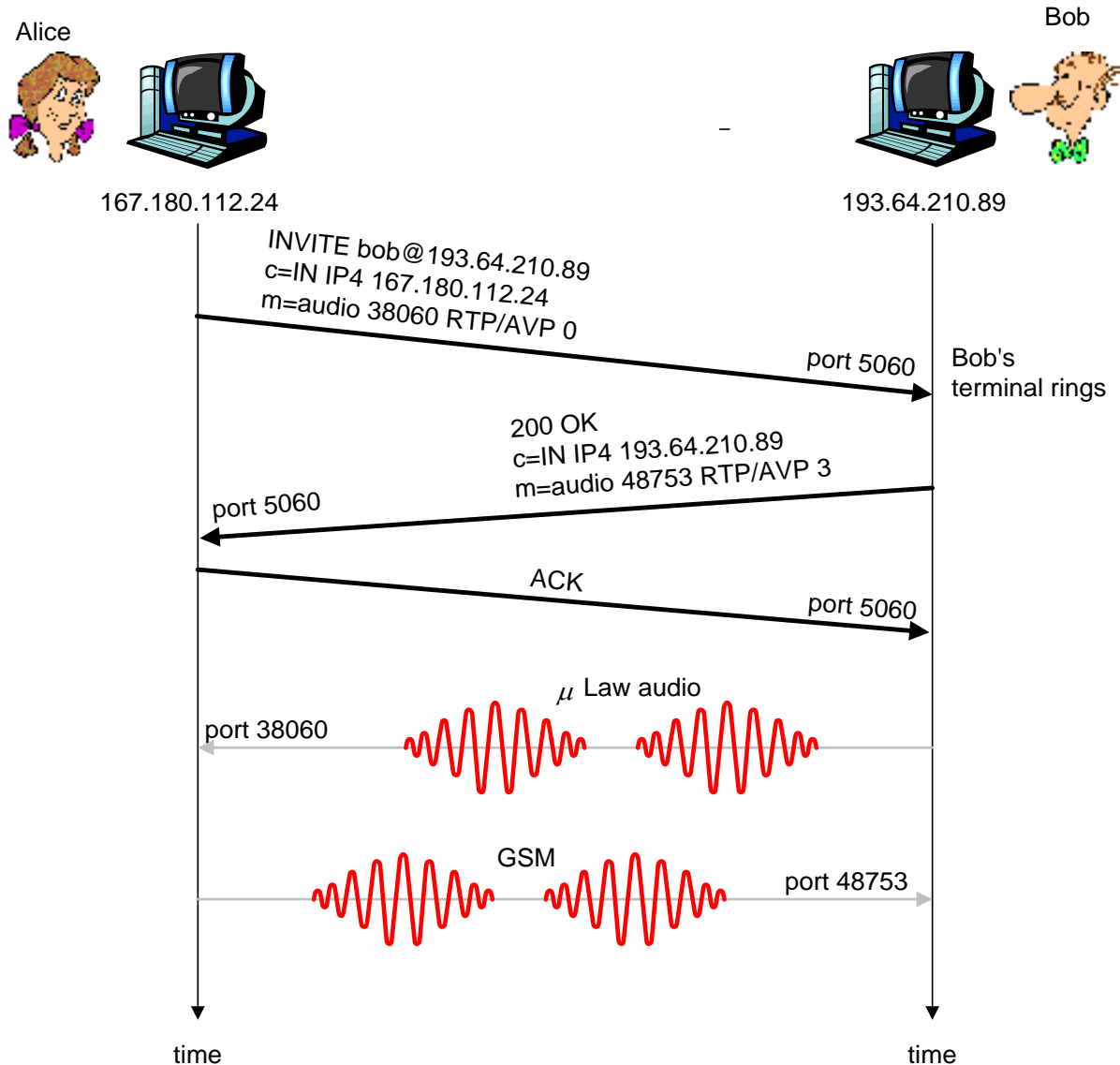
## Vision

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

## Dienste

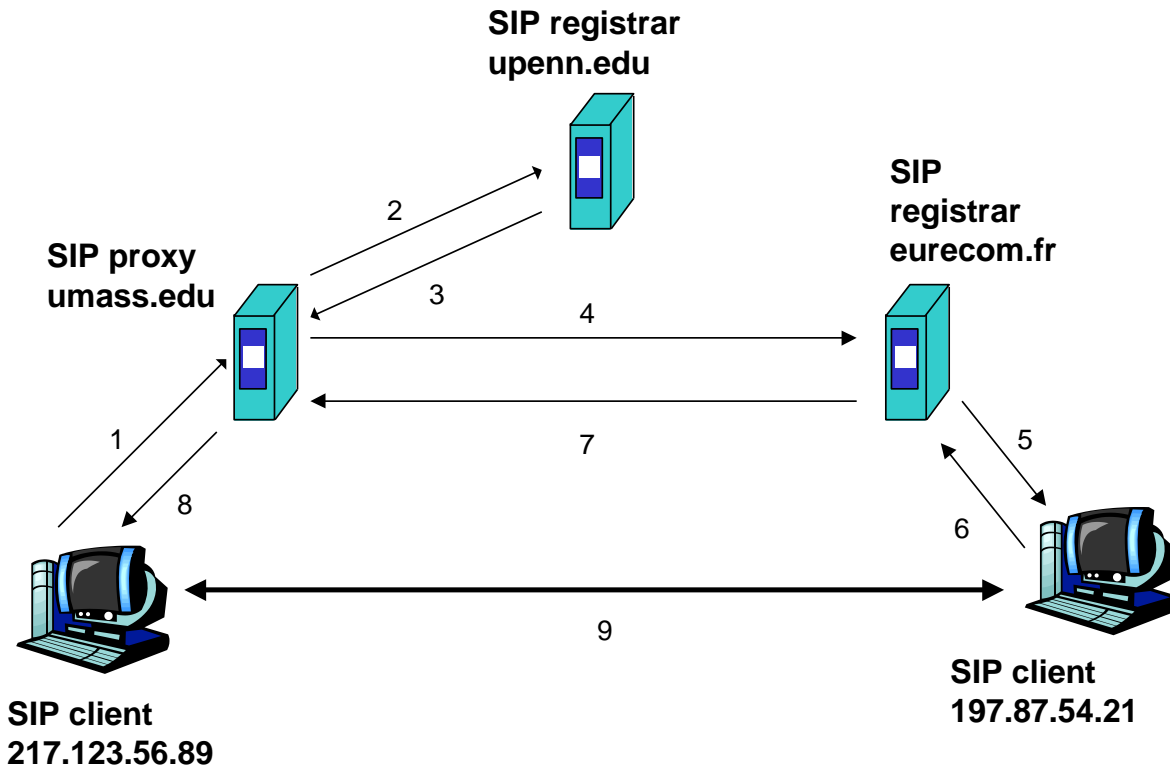
- ◆ Anruf-Erzeugung
  - Rufen des Partners
  - Abstimmen der Medien und der Codierung
  - Beenden der Sitzung
- ◆ Ermittlung der aktuellen IP-Adresse des Partners
- ◆ Verbindungsverwaltung
  - Medien- und Codec-Änderungen
  - Neue Partner dazu
  - Anrufweiterleitung und Pausieren

# Setting up a call to a known IP address



- Alice's SIP invite message indicates her port number & IP address. Indicates encoding that Alice prefers to receive (PCM ulaw)
- Bob's 200 OK message indicates his port number, IP address & preferred encoding (GSM)
- SIP messages can be sent over TCP or UDP; here sent over RTP/UDP.
- Default SIP port number is 5060.

# 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)

Caller **jim@umass.edu** with places a call to **keith@upenn.edu**

- (1) Jim sends INVITE message to umass SIP proxy.
- (2) Proxy forwards request to upenn registrar server.
- (3) upenn server returns redirect response, indicating that it should try keith@eurecom.fr
- (4) umass proxy sends INVITE to eurecom registrar.
- (5) eurecom registrar forwards INVITE to 197.87.54.21, which is running keith's SIP client.
- (6-8) SIP response sent back
- (9) media sent directly between clients.

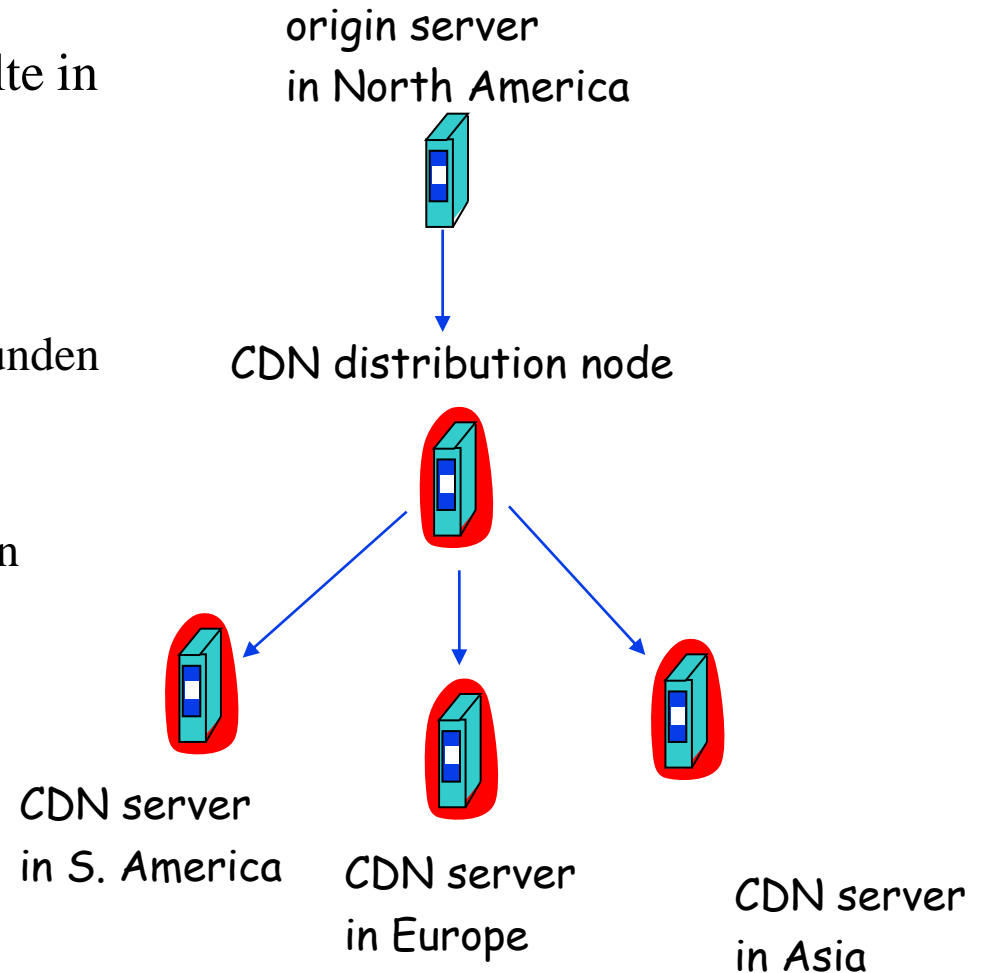
# Content Distribution Networks (CDNs)

## ◆ *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



# Kapitel 6: Übersicht

---

- 6.1 Multimedia Netzanwendungen
- 6.2 Audio- und Videostreaming
- 6.3 Realzeit Multimedia: Voice over IP / Internet-Telephonie
- 6.4 Protokolle für Realzeit-Anwendungen  
RTP,RTCP,SIP
- 6.5 Multimedia-Verteilung im Netz

- 6.6 Über Best Effort hinaus**
- 6.7 Scheduling und Policing Mechanismen
- 6.8 Integrated Services und Differentiated Services
- 6.9 RSVP

# Internet-Evolution für Multimedia

## Integrated Services *IntServ*

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

## Laissez-Faire

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

## Differentiated Services *DiffServ*

- ◆ 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**



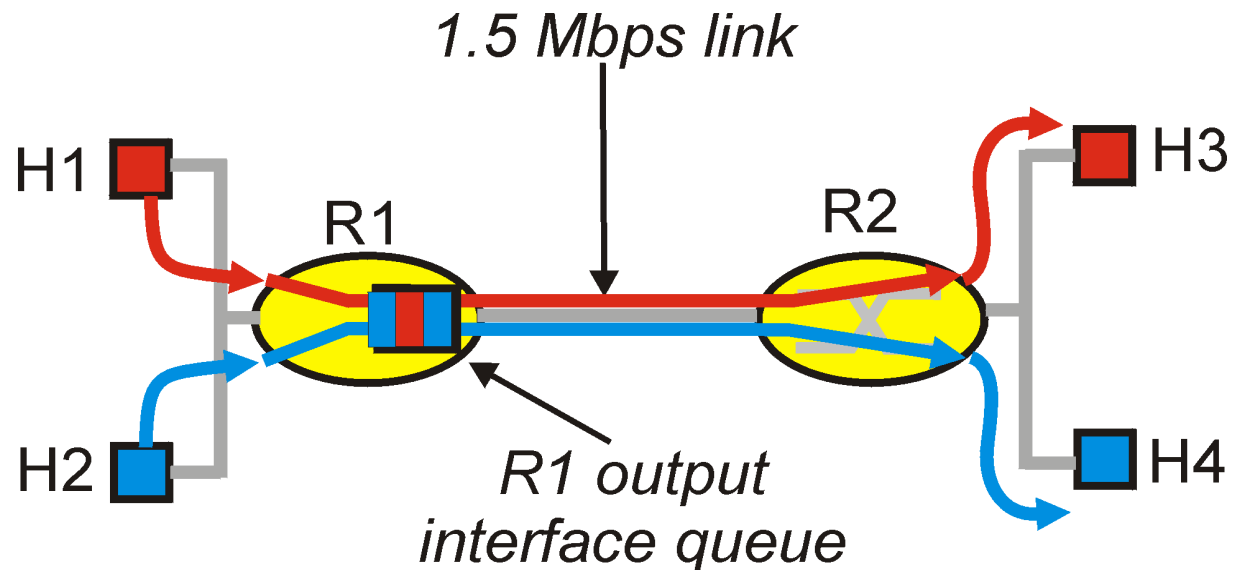
# Verbesserte Dienstgüte in IP Netzen

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

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

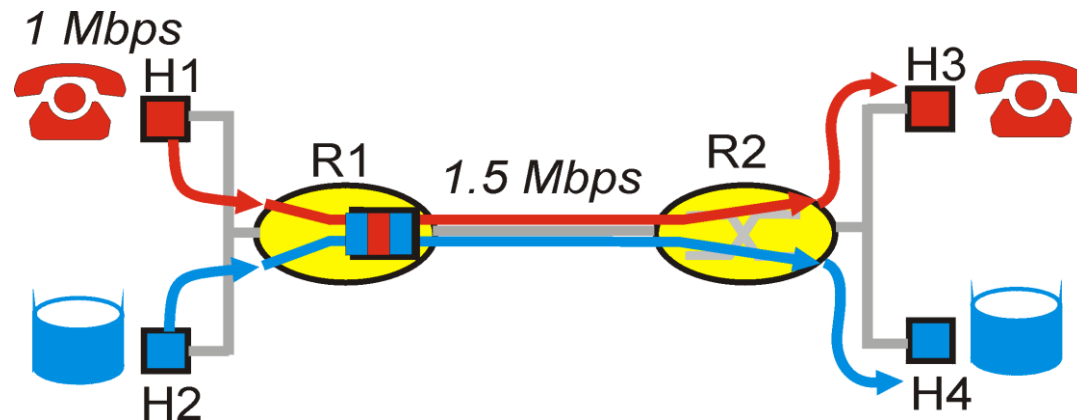
- **RSVP:** Signalisierung für Ressourcenreservierungen
- **Differentiated Services:** Priorisierungen
- **Integrated Services:** Feste Garantien

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



# Prinzipien für QoS-Garantien

- ◆ Beispiel: 1Mbps I P-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

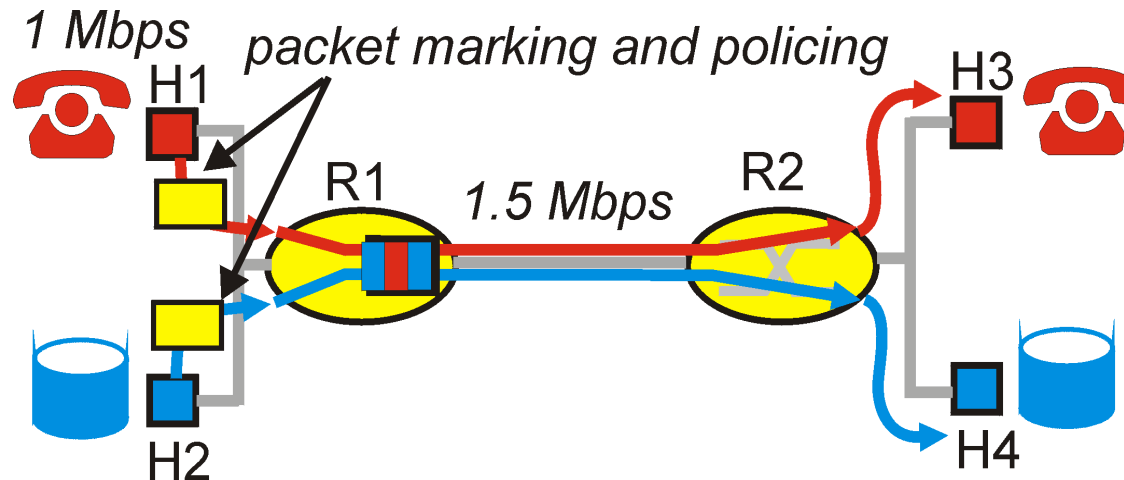


## Prinzip 1

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)

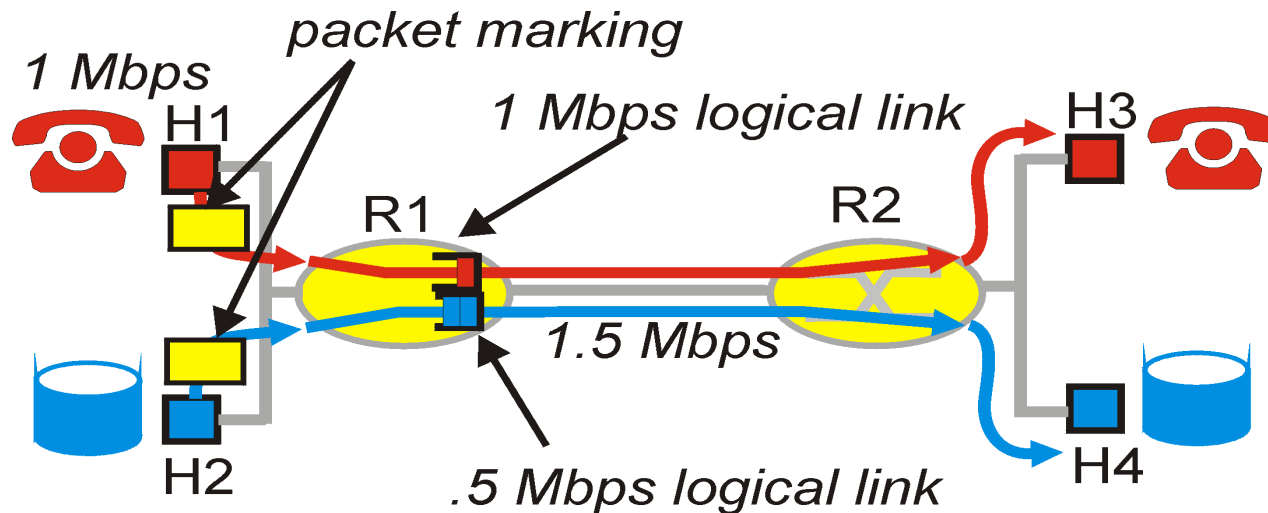


## Prinzip 2

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

# Prinzipien für QoS-Garantien

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



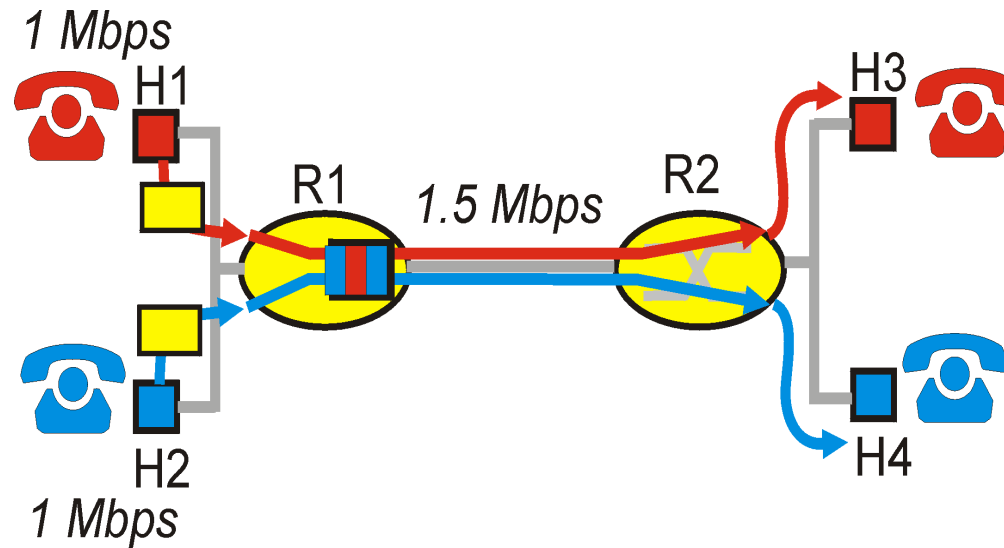
## Prinzip 3

Die Ressourcen sollen trotz Isolation möglichst effizient mehrfach genutzt werden.

# Prinzipien für QoS-Garantien

## ◆ *Der Boden der Tatsachen*

Man kann nicht mehr übertragen, als die Link-Leistung zulässt.



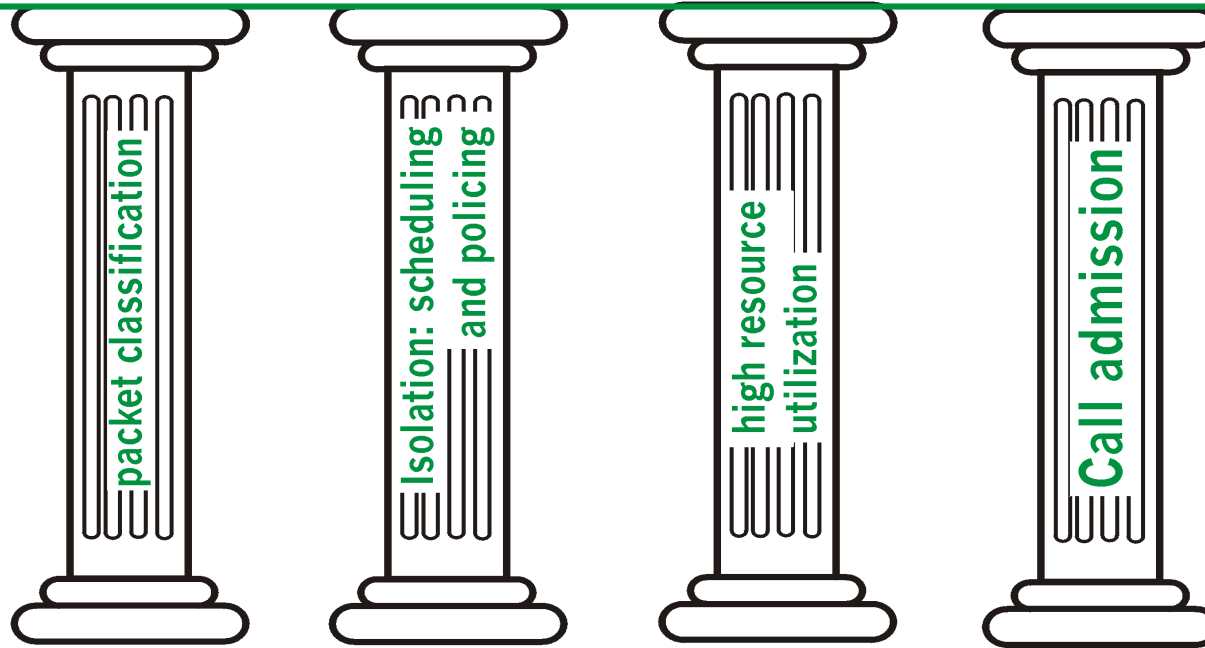
## Prinzip 4

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

# Prinzipien für QoS-Garantien: Zusammenfassung

---

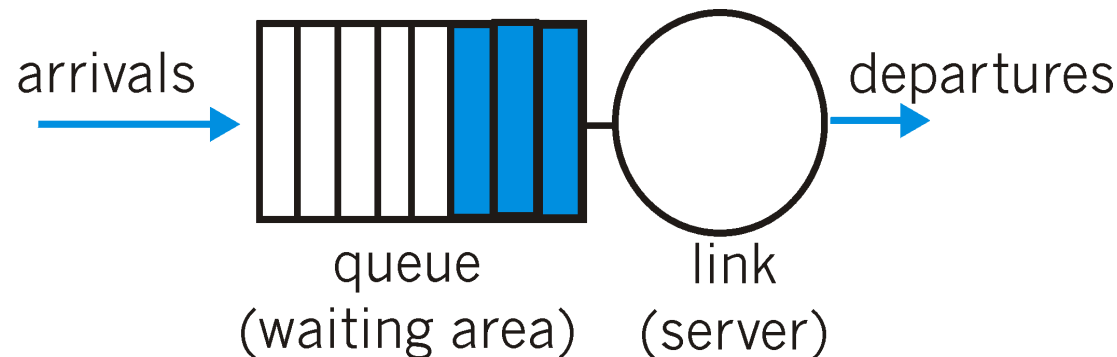
## QoS for networked applications



Im Folgenden: Entsprechende 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, niederprioreres Paket
    - » Random: zufällige Auswahl

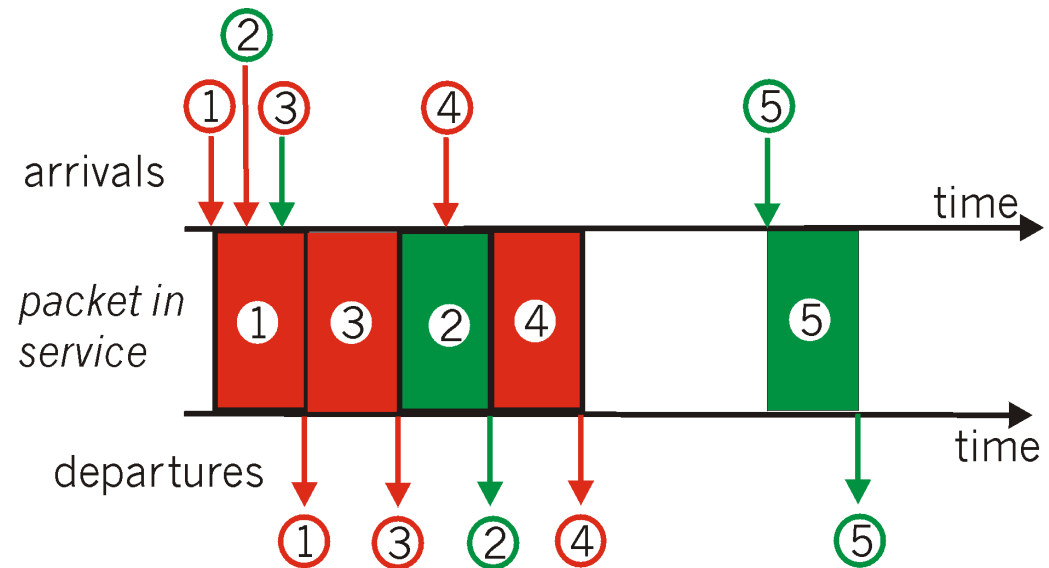
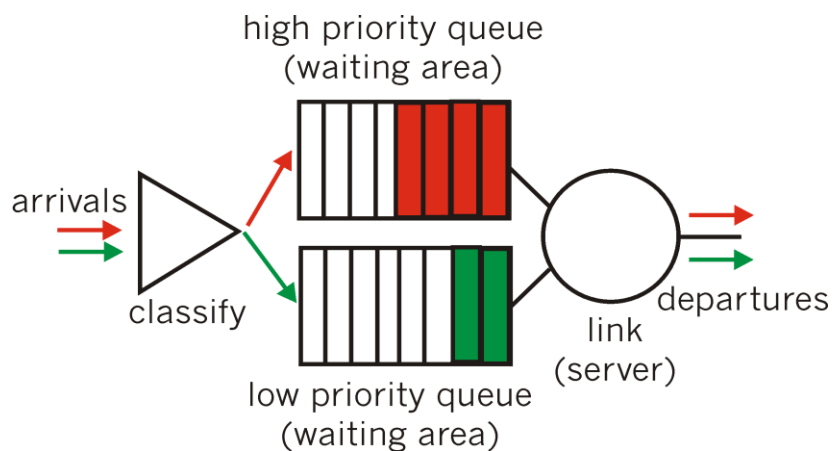


# Scheduling Mechanismen

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

◆ mehrere Prioritätsklassen **Problem: Fairness**

- Prioritätskennung im Paketheader, Portnummer, Protokolltyp, etc.



**Andere Sttrategien (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:

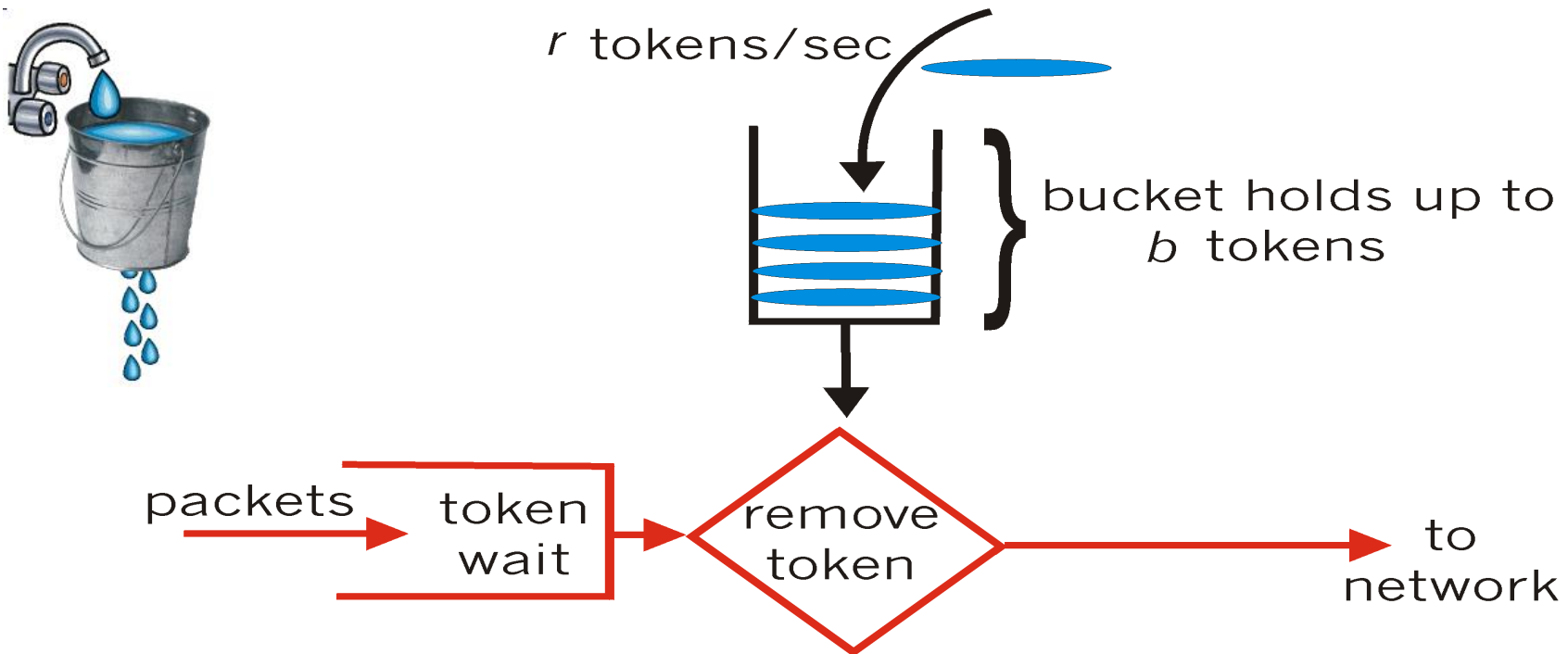
- ◆ *(Langfristige) mittlere Senderate*
- ◆ *Spitzenrate*
- ◆ *(Maximale) Burst-Größe*

Mechanismen sollen für Nutzer nachvollziehbar sein.

# Policing Mechanismen: 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?

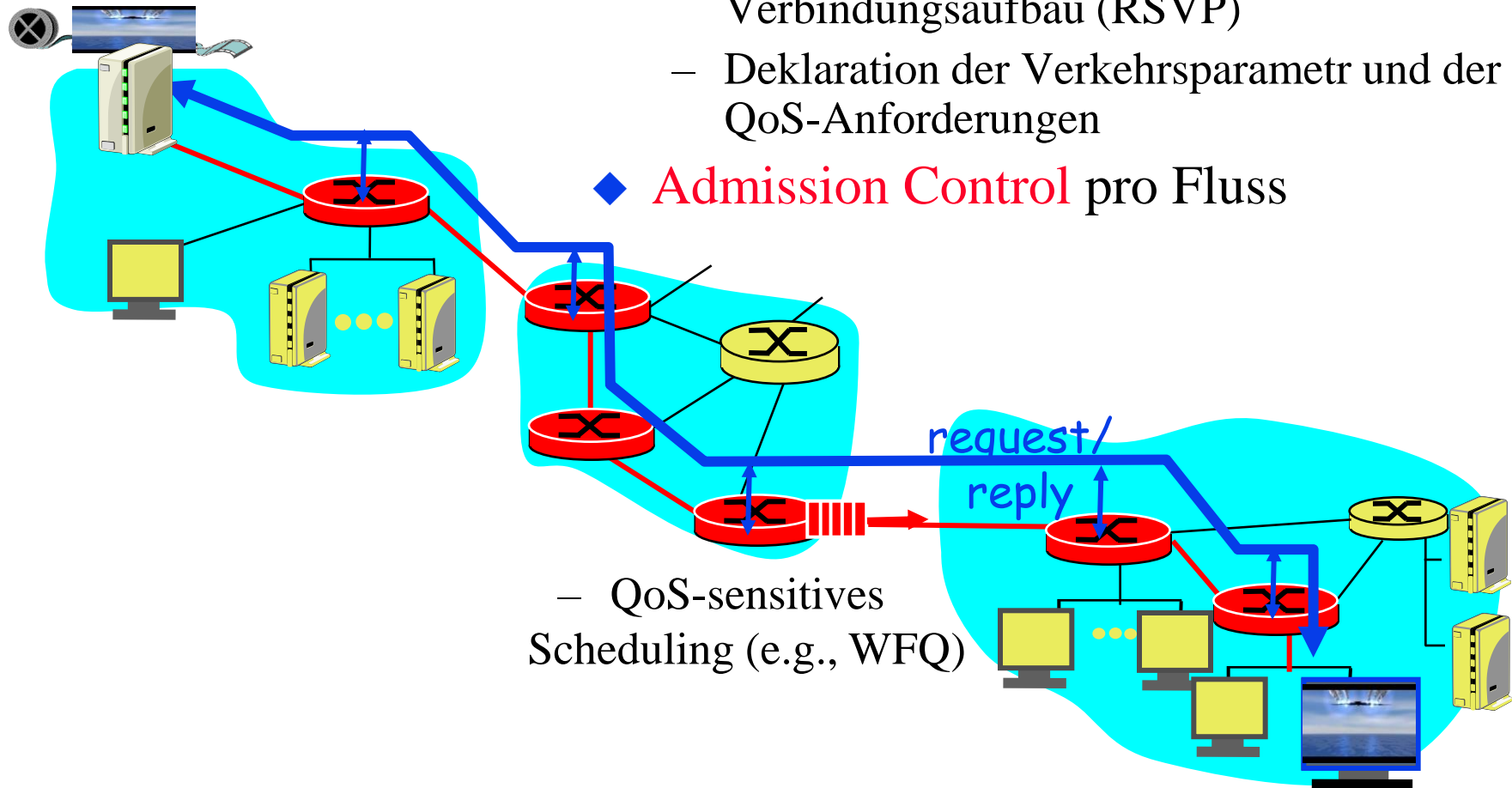
# Intserv: QoS-Garantie-Szenario

## ◆ Ressourcereservierung

- Signalisierung beim Verbindungsaufbau (RSVP)
- Deklaration der Verkehrsparametr und der QoS-Anforderungen

## ◆ Admission Control pro Fluss

- QoS-sensitives Scheduling (e.g., WFQ)



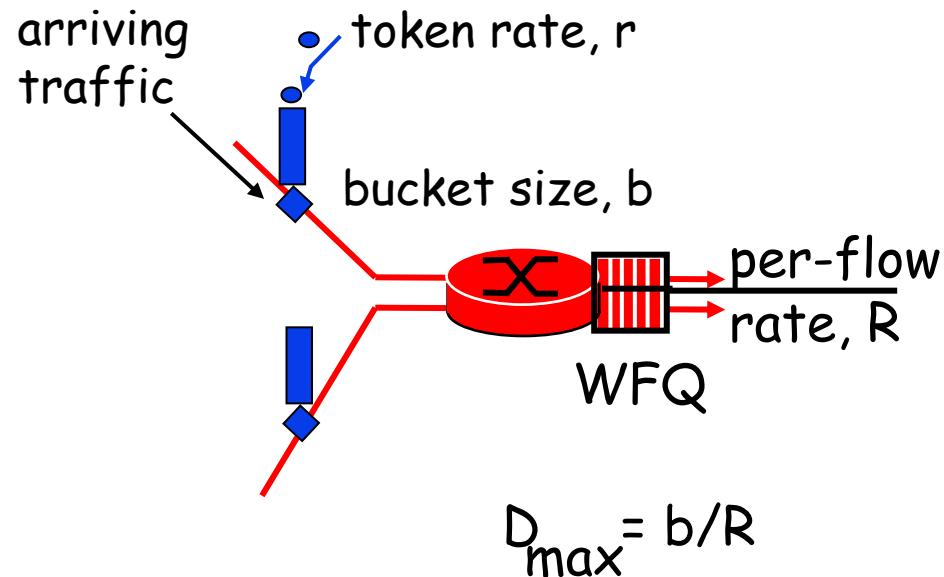
# Intserv QoS: Dienstmodelle [RFC 2211, RFC 2212]

## 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

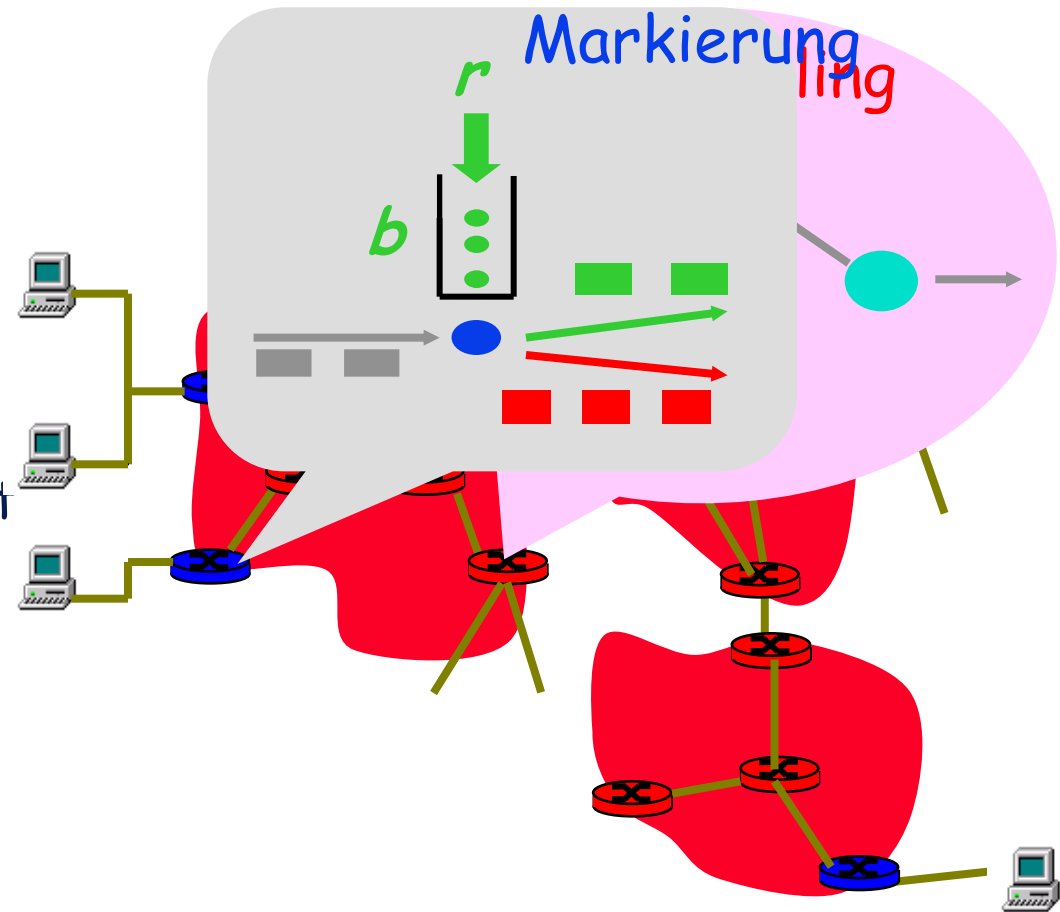
# DiffServ Architektur

## Edge Router:

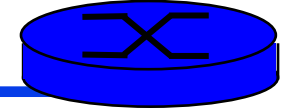
- ❑ Per-Fluss Verkehrsmanagement
- ❑ Markiert Pakete als **in-profile** oder **out-profile**

## Core Router:

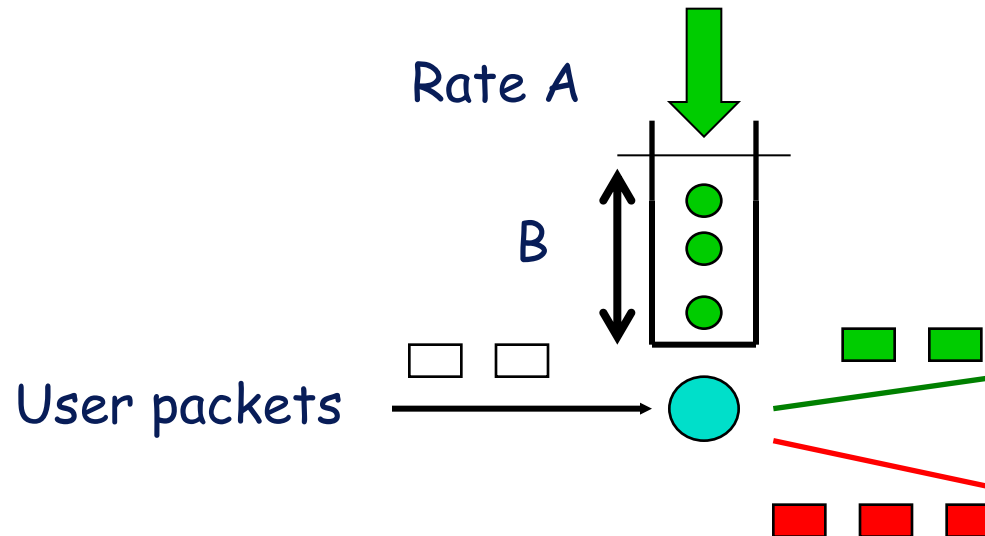
- ❑ **Per-Klasse** Verkehrsmanagement
- ❑ Pufferung und Scheduling entsprechend Markierung
- ❑ **In-profile** Pakete werden vorgezogen
- ❑ Garantierte Weiterleitung



# Edge-Router Paket-Markierung



- ◆ **Profile:** Vorab für Fluss ausgehandelte mittlere Rate **A**, Eimer-Größe **B**
- ◆ Jedes Paket wird **Fluss-bezogen** markiert



## Markierung:

- ◆ Klassen-Zugehörigkeit
- ◆ Innerhalb einer Klasse: Profil-konform / Profil-verletzend

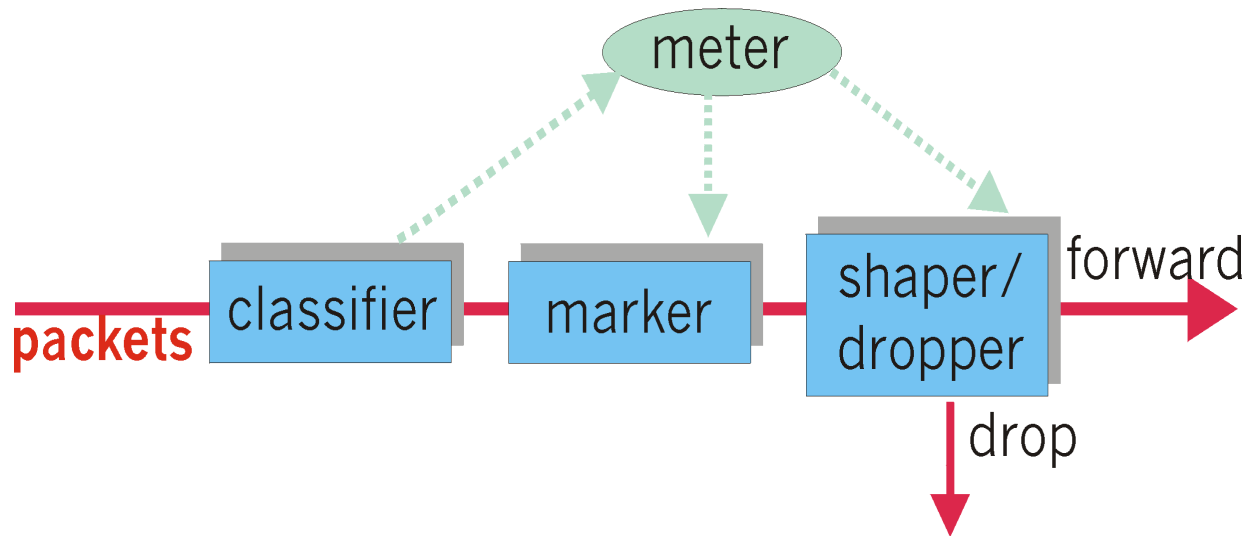
IP V4: Type of Service Header-Feld, IP V6: Traffic Class Header-Feld  
(8 Bit, davon 6 benutzt: Differentiated Service Code Point (DSCP))





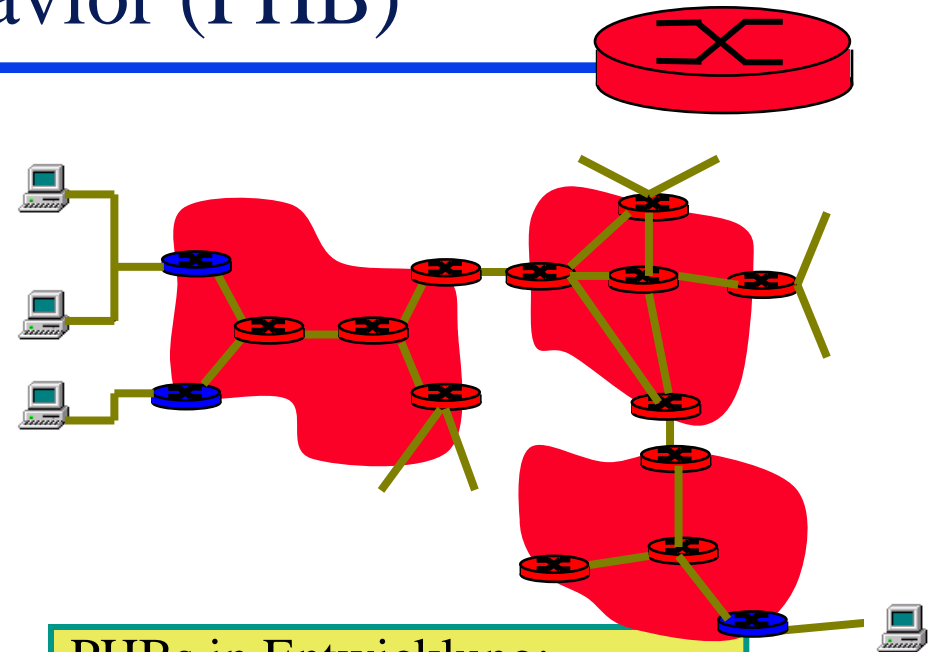
# Konditionierung

- ◆ Nutzer definiert Fluss-Profil (e.g., Rate, Burst-Größe)
- ◆ Verkehr wird gemessen und, falls Profil-verletzend, durch Paket-Verluste geformt



# Weiterleitung – Pro Hop Behavior (PHB)

- ◆ PHB wirkt sich in unterschiedlichen Weiterleitungsleistungsparametern aus
- ◆ PHB definiert Leistungsparameter-Unterschiede als Ziele der einzusetzenden Mechanismen, definiert die Mechanismen aber nicht
- ◆ Beispiele:
  - Klasse A soll je Zeitintervall der Länge 100 msec 22% der Bandbreite des abgehenden Links erhalten
  - Klasse A Pakete werden vor Klasse B Paketen weitergegeben



## PHBs in Entwicklung:

- ◆ **Expedited Forwarding:**  
Mindest-Paket-Weitergaberate einer Klasse (Logische Verbindung mit Mindestbandbreite)
- ◆ **Assured Forwarding:**  
4 Verkehrsklassen
  - Je Klasse bestimmte Mindestbandbreite
  - Unterschiedliche Verlust-Bedingungen

# 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-Signalisierprotokoll: 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:

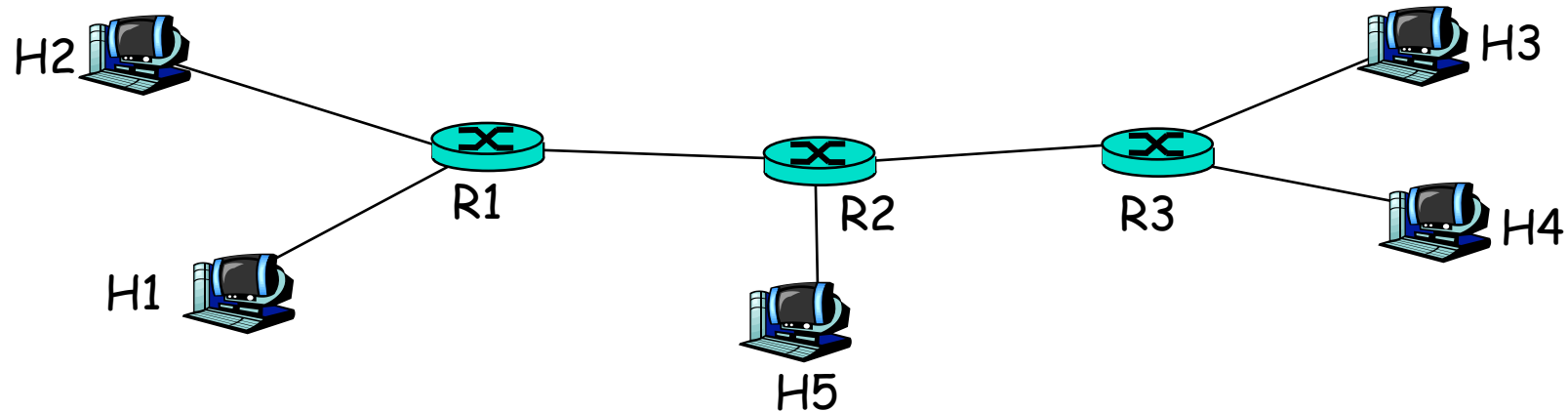
Die Routenermittlung und Broadcast-Gruppen/Adressverwaltung werden außerhalb von RSVP abgewickelt

## ◆ Dynamik: Soft State - Konzept

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

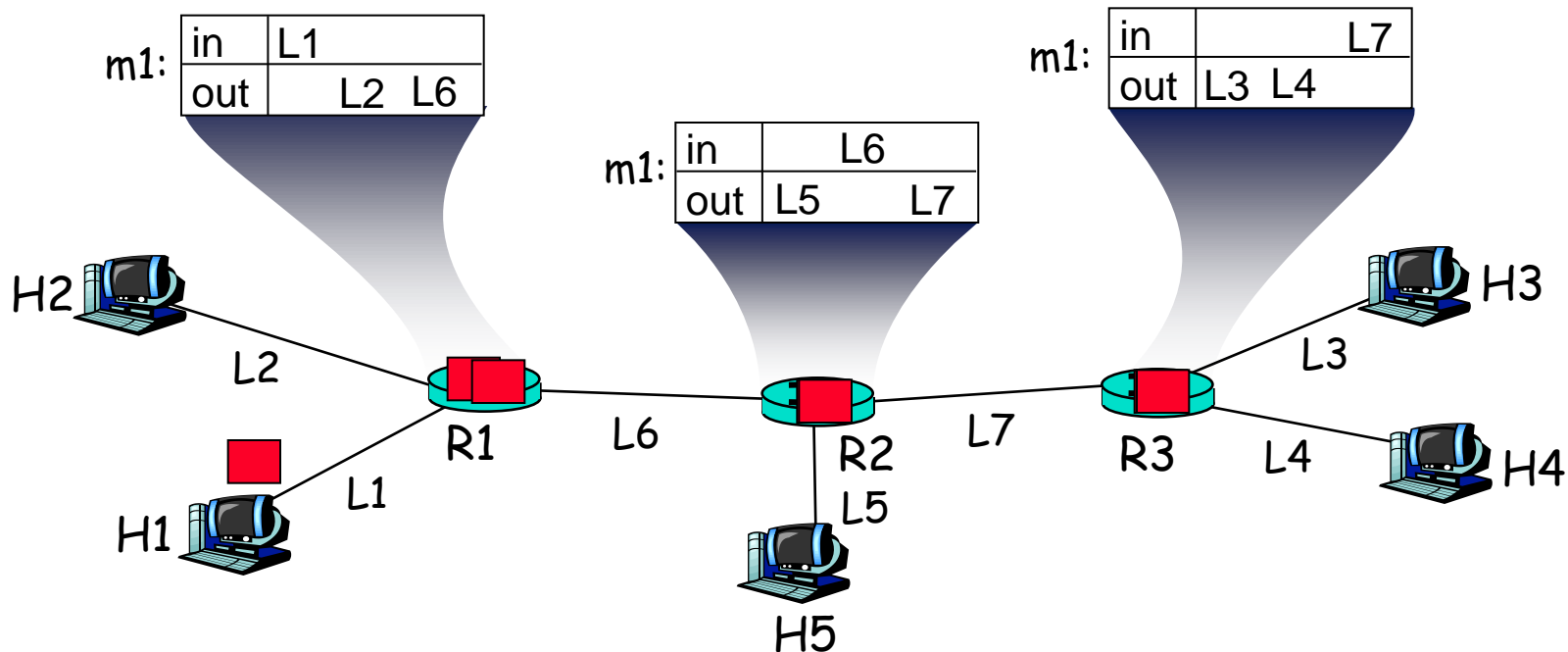
# RSVP: Einfache Audio Konferenz

- ◆ Die Hosts  $H1, H2, H3, H4, H5$  senden und empfangen
- ◆ Multicast-Gruppe  $m1$
- ◆ Keine Filterung: Pakete aller Sender werden weitergeleitet
- ◆ Audio-Rate:  $b$
- ◆ Es wird ein einziger Multicast-Routing-Spannbaum verwendet



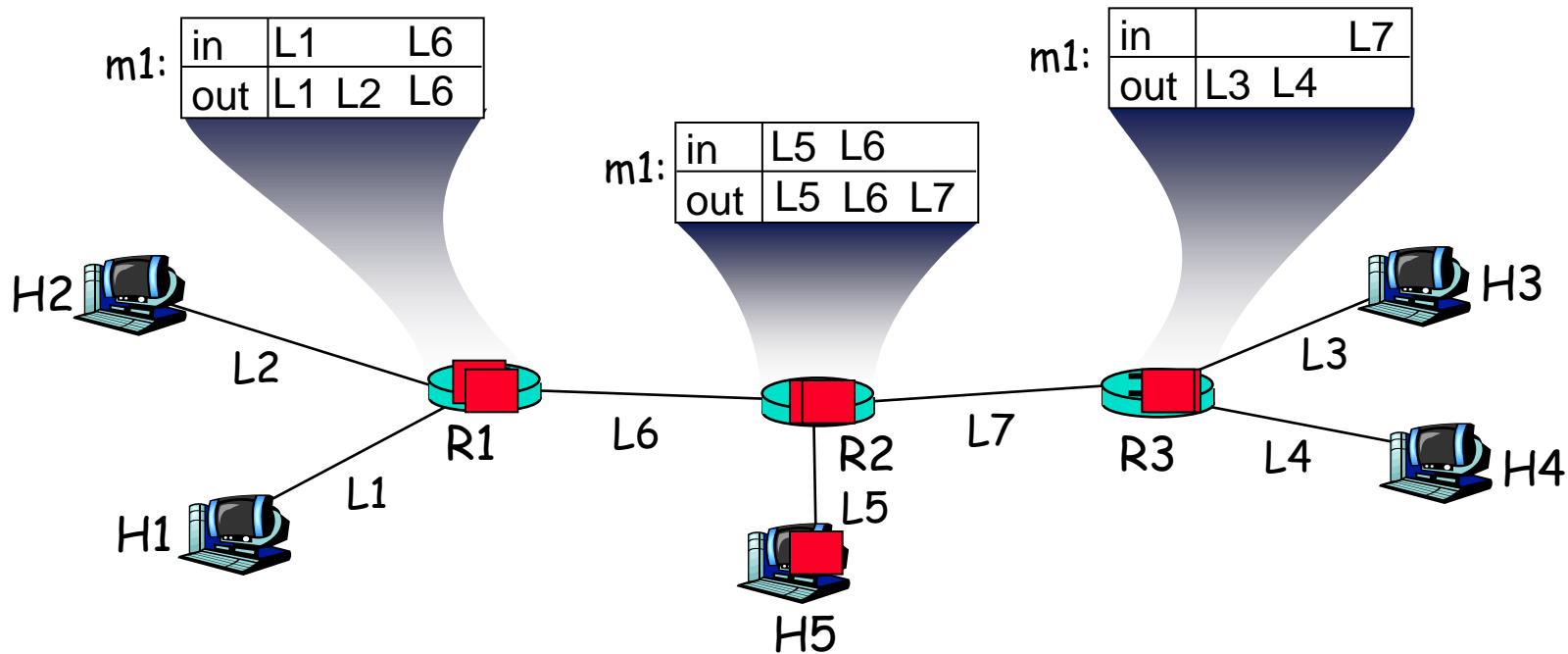
# RSVP: Pfadzustandsinformation in Routern

- ◆ H1, ..., H5 senden alle Pfadnachrichten an *m1*:  
(address=*m1*, Tspec=*b*, filter-spec=no-filter, refresh=100)
- ◆ Annahme: H1 sendet als erster



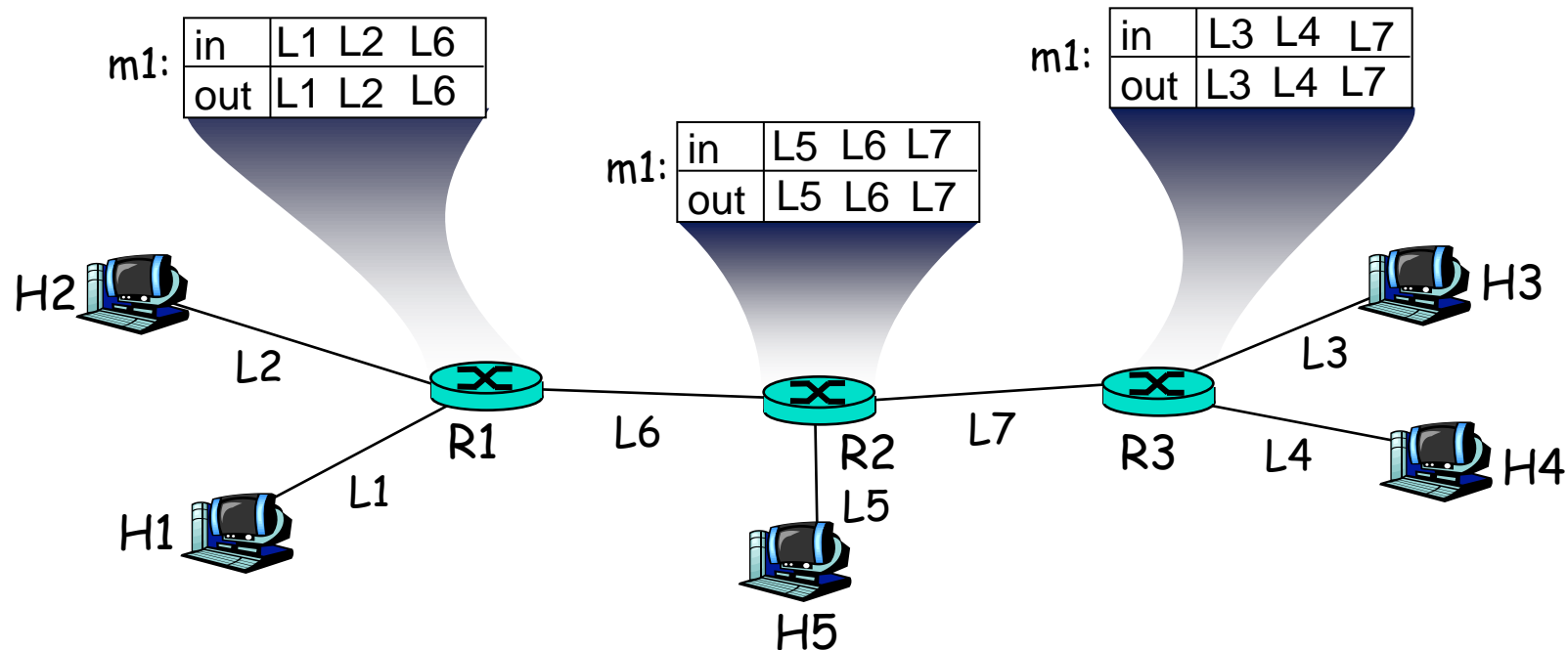
# RSVP: Pfadzustandsinformation

- ◆ als nächstes sendet H5



# RSVP: Pfadzustandsinformation

- ◆ H2, H3, H5 senden jetzt auch
- ◆ Zustandstabellen werden vervollständigt





## ◆ Inhalt der Reservierungsnachrichten

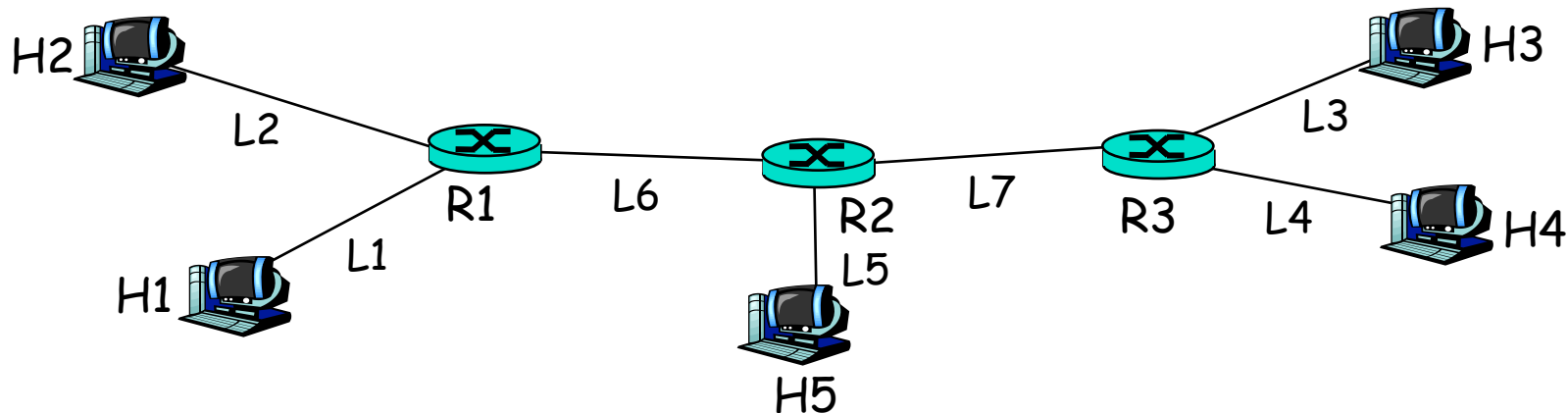
- *Benötigte Bandbreite*
- *Filtertyp:*
  - » no filter: Alle Pakete der Gruppe benutzen die reservierten Ressourcen
  - » fixed filter: Reservierte Ressourcen nur für bestimmte Sender
  - » dynamic filter: Sender-Gruppe kann sich dynamisch ändern
- *Filter-Spezifikation*

## ◆ Die Reservierungsnachrichten werden auf den Pfaden von einem Empfänger hin zu den Sendern verbreitet und erzeugen in den durchlaufenen Routern Empfänger-bezogene Zustandsinformation

# RSVP: *Empfänger*-Ressourcenreservierung

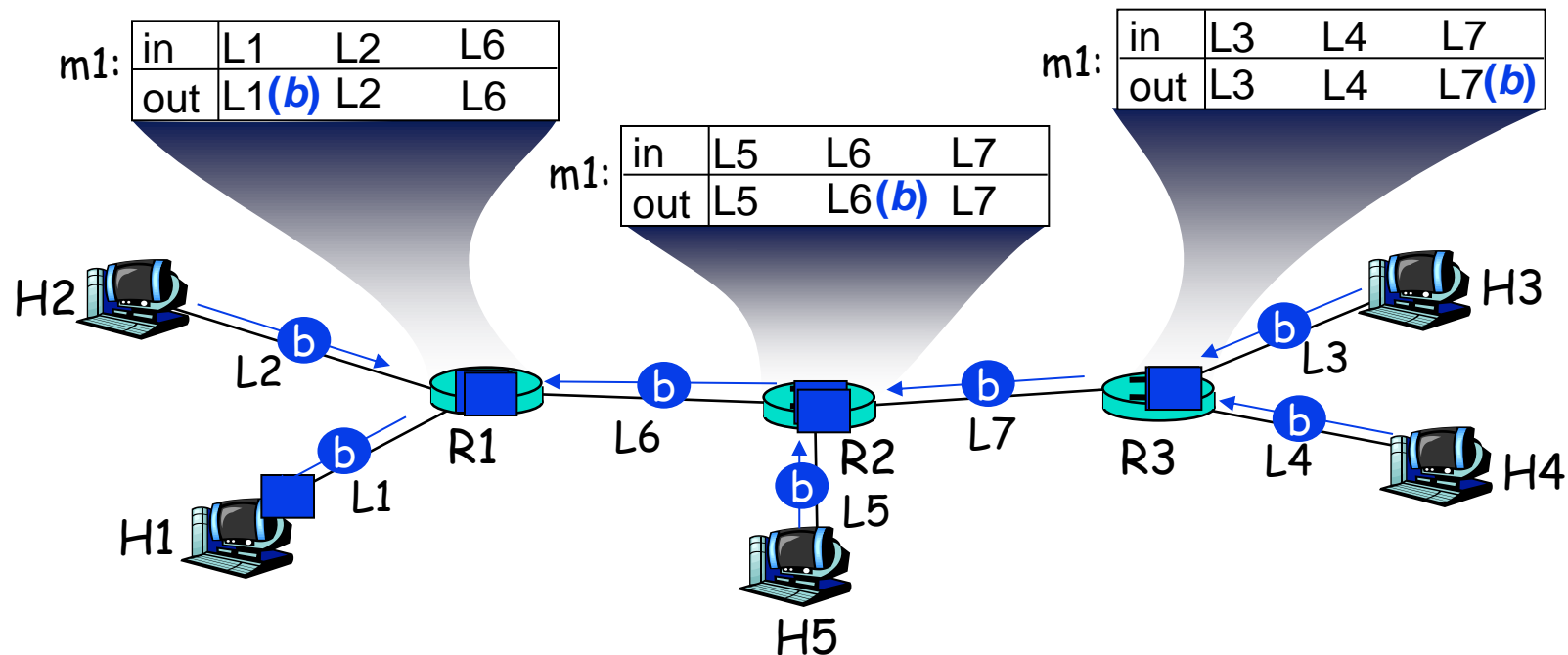
H1 möchte von allen anderen Hosts der Gruppe Audio empfangen

- ◆ H1 Reservierungsnachricht fließt von H1 zu den Sendern
- ◆ H1 reserviert damit Bandbreite für 1 Audio-Strom
- ◆ Reservierungstyp “no filter” – jeder Sender nutzt reservierte Bandbreite



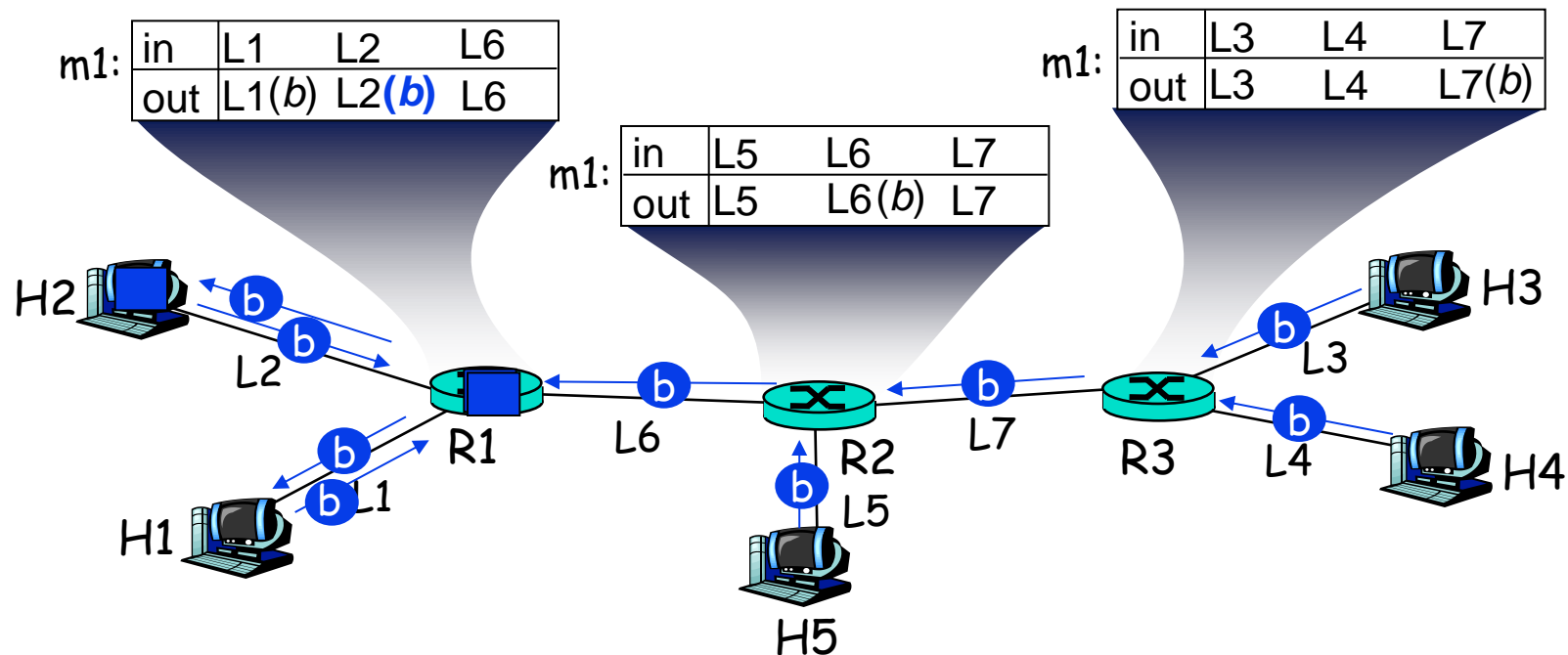
# RSVP: *Empfänger*-Ressourcenreservierung

- ◆ H1 Reservierungsnachricht fließt baumaufwärts zu den Sendern
- ◆ Router und Hosts reservieren Bandbreite  $b$ , die benötigt wird, um Audio zu H1 zu senden



# RSVP: *Empfänger*-Ressourcenreservierung

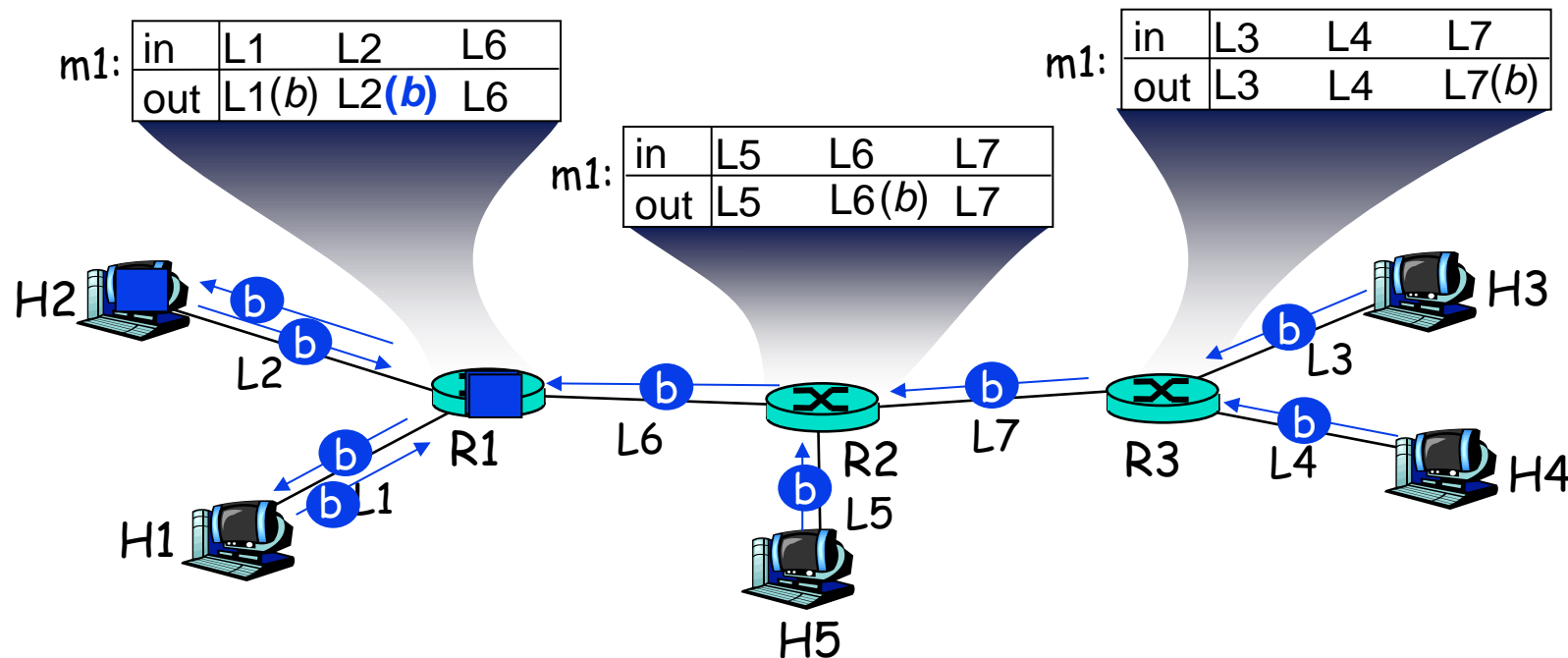
- ◆ Als nächstes reserviert H2 Bandbreite b in Modus “no-filter”
- ◆ H2 gibt an R1 weiter, R1 an H1, aber R2 (?)
- ◆ R2 führt keine Aktion aus, da b auf L6 schon reserviert ist



# RSVP: *Empfänger*-Ressourcenreservierung -- Summenrate

Was passiert, wenn mehrere Sender (e.g., H3, H4, H5) gleichzeitig über einen Link senden (e.g., L6)?

- ◆ Zufällige Überlagerung der Ströme
- ◆ Der Summenfluss über L6 wird per Leaky Bucket reglementiert (Policing): falls die Summenrate  $b$  länger übersteigt, werden Paketverluste auftreten



# Kapitel 6: Übersicht

---

- 6.1 Multimedia Netzanwendungen
- 6.2 Audio- und Videostreaming
- 6.3 Realzeit Multimedia: Voice over IP / Internet-Telephonie
- 6.4 Protokolle für Realzeit-Anwendungen  
RTP,RTCP,SIP
- 6.5 Multimedia-Verteilung im Netz
- 6.6 Über Best Effort hinaus
- 6.7 Scheduling und Policing Mechanismen
- 6.8 Integrated Services und Differentiated Services
- 6.9 RSVP