universität wien

Chair for Future Communication
Prof. Dr. K. Tutschku
Institute for Distributed and Multimedia Systems
Faculty for ComputerScience

050069 VO Netzwerktechnologie für Multimedia Anwendungen

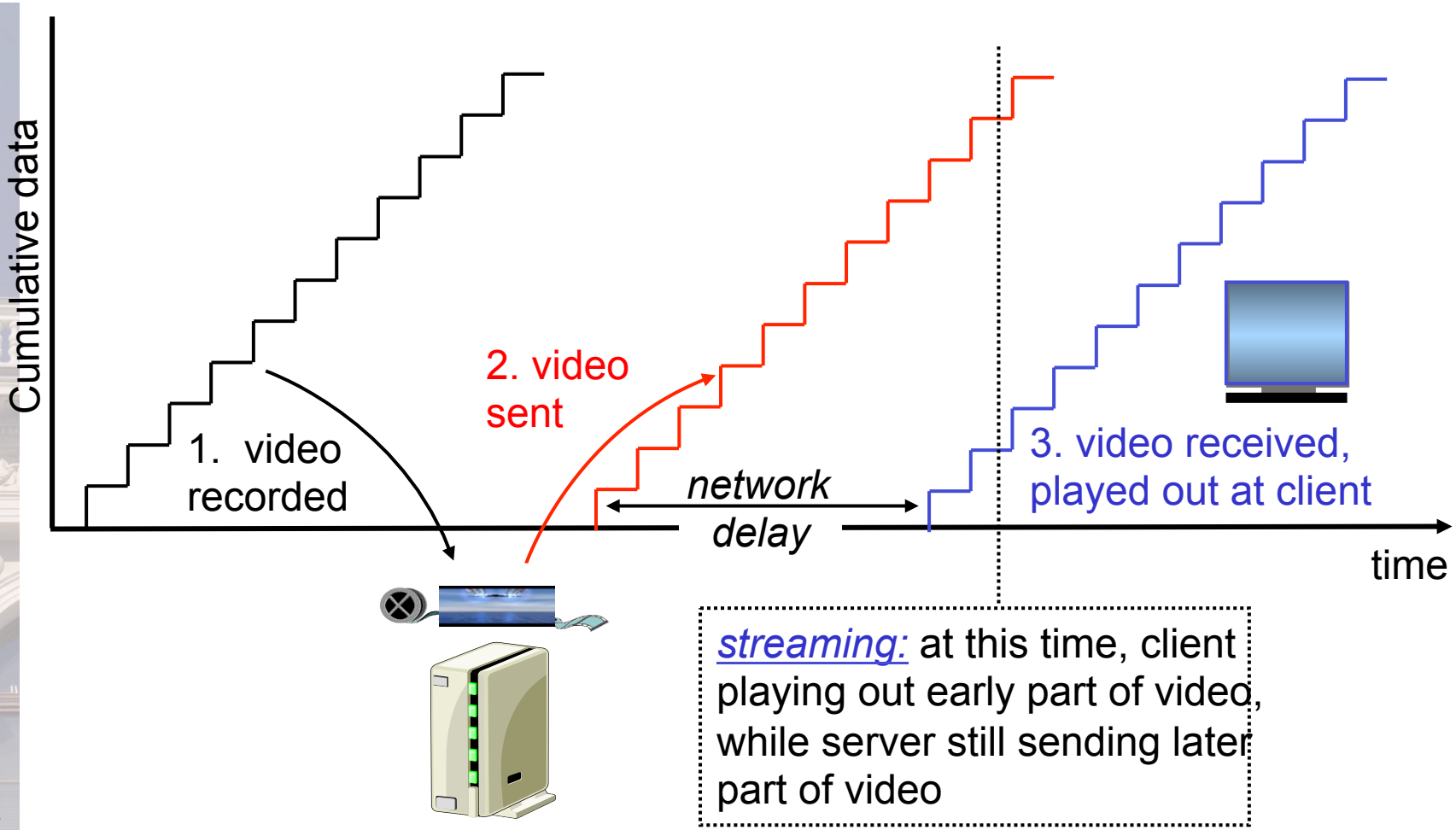
Lecture 5: Multimedia-Networking

Prof. K. Tutschku (kurt.tutschku@univie.ac.at)

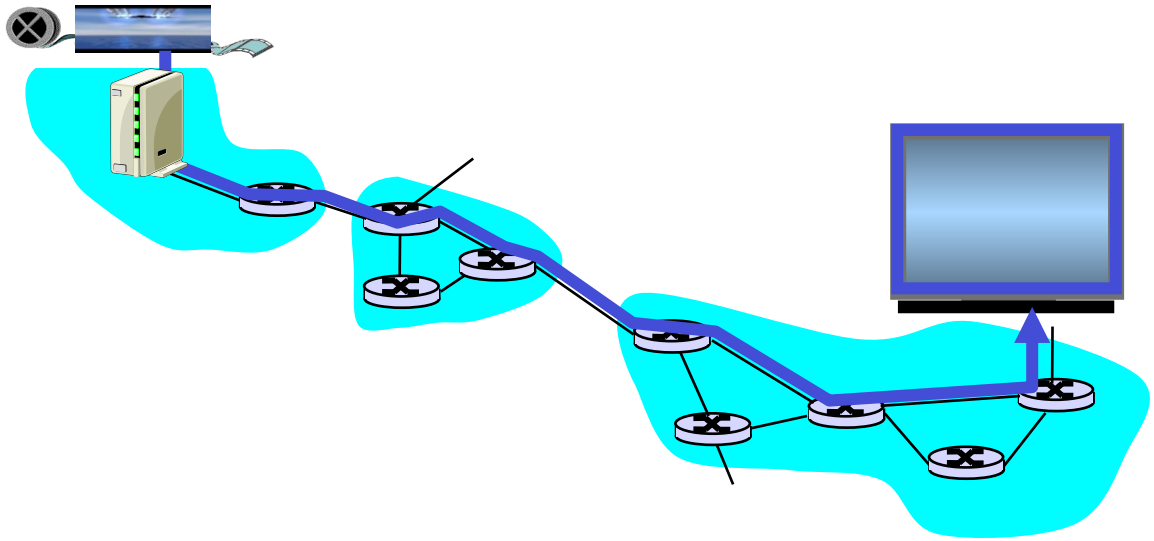
BachelorInformatik (Medieninformatik)
WS 2010/11

Endowed by
A1
TELE KOM AUS TRIA

Streaming Stored Multimedia: What is it?



Streaming Stored Multimedia: Interactivity



- ***VCR-like functionality:*** **client can pause, rewind, FF, push slider bar**
  - 10 sec initial delay OK
  - 1-2 sec until command effect OK
  - RTSP often used (more later)
- **timing constraint for still-to-be transmitted data: in time for playout**

Streaming Live Multimedia

Examples:

- Internet radio talk show
- Live sporting event

Streaming

- playback buffer
- playback can lag tens of seconds after transmission
- still have timing constraint

Interactivity

- fast forward impossible
- rewind, pause possible!

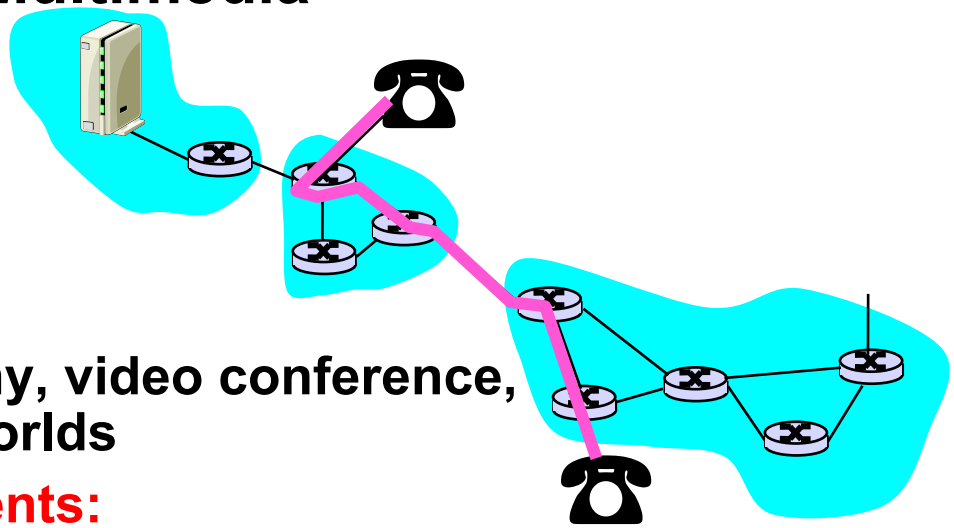
universität wien

Chair of Future Communication

endowed by

A1 TELEKOM AUSTRIA

Interactive, Real-Time Multimedia



- **applications: IP telephony, video conference, distributed interactive worlds**
- **end-end delay requirements:**
  - audio: < 150 msec good, < 400 msec OK
    - includes application-level (packetization) and network delays
    - higher delays noticeable, impair interactivity
- **session initialization**
  - how does callee advertise its IP address, port number, encoding algorithms?

Multimedia Over Today's Internet

TCP/UDP/IP: "best-effort service"

- ***no*** **guarantees on delay, loss**

But you said multimedia apps requires QoS and level of performance to be effective!

Today's Internet multimedia applications use application-level techniques to mitigate (as best possible) effects of delay, loss

How should the Internet evolve to better support multimedia?

Integrated services philosophy:

- **Fundamental changes in Internet so that apps can reserve end-to-end bandwidth**
- **Requires new, complex software in hosts & routers**

Laissez-faire

- **no major changes**
- **more bandwidth when needed**
- **content distribution, application-layer multicast**
  - application layer

Differentiated services philosophy:

- **Fewer changes to Internet infrastructure, yet provide 1st and 2nd class service.**

What's your opinion?

A few words about audio compression

- **Analog signal sampled at constant rate**
  - telephone: 8,000 samples/sec
  - CD music: 44,100 samples/sec
- **Each sample quantized, i.e., rounded**
  - e.g., 2^8=256 possible quantized values
- **Each quantized value represented by bits**
  - 8 bits for 256 values
- **Example: 8,000 samples/sec, 256 quantized values --> 64,000 bps**
- **Receiver converts it back to analog signal:**
  - some quality reduction

Example rates

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

A few words about video compression

- **Video is sequence of images displayed at constant rate**
  - e.g. 24 images/sec
- **Digital image is array of pixels**
- **Each pixel represented by bits**
- **Redundancy**
  - spatial
  - temporal

Examples:

- **MPEG 1 (CD-ROM) 1.5 Mbps**
- **MPEG2 (DVD) 3-6 Mbps**
- **MPEG4 (often used in Internet, < 1 Mbps)**

Research:

- **Layered (scalable) video**
  - adapt layers to available bandwidth

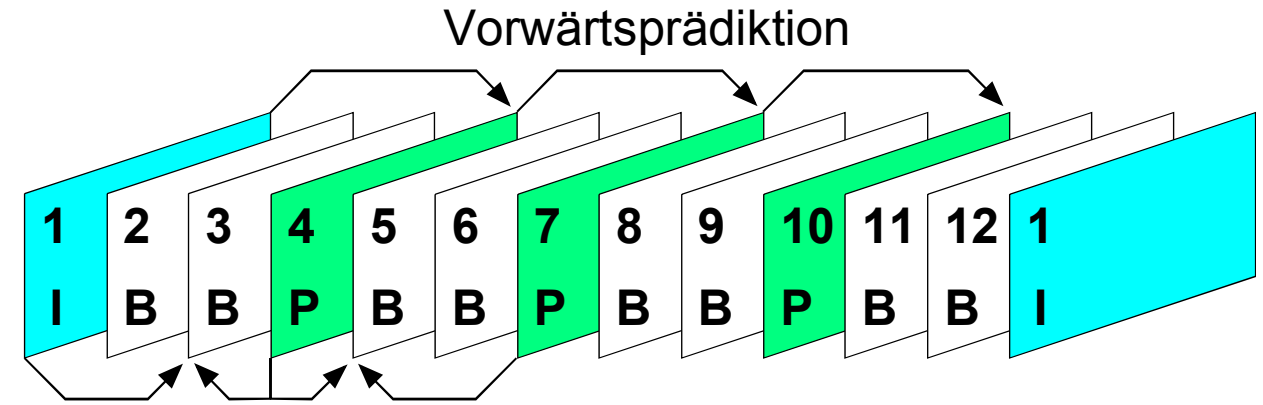
Generation of MPEG Video Sequences

- **Given: sequence of digital images**
- **MPEG compression is combination of**
  - Intra-frame compression (spatial redundancy reduction)
    - Discrete-Consine Transformation (DCT):
      8x8 pixel blocks ⇒ DCT ⇒ 8x8 DC coefficients
    - Quantization
    - Zig-zag entropy encoding
  - Inter-frame compression (temporal redundancy reduction)
    - Block-based motion compensation

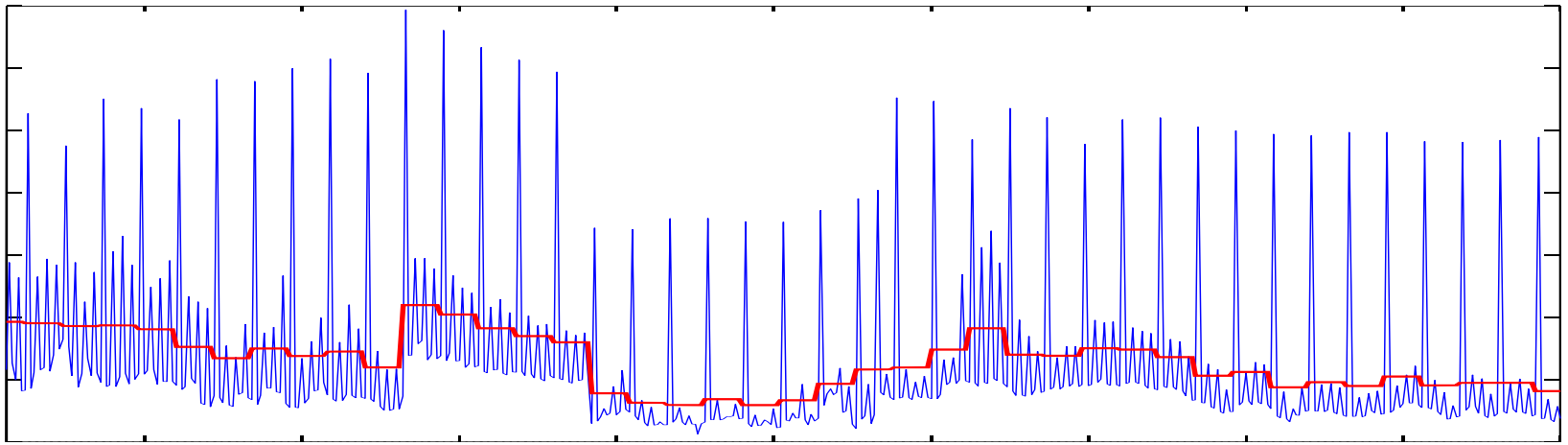
Inter-Frame Dependencies

- **Frame types**
  - Intracoded frames (I-frame)
    - Does not depend on any other frames
    - Most important information, largest frame size
  - Predicted frames (P-frame)
    - Depends on preceding I- or P-frame
    - Medium frame size
  - Bidirectional frames (B-frame)
    - Depends on preceding and succeeding I- or P-frame
    - Small frame size, no other frame depends on it
- **Group of Pictures (GoP)**
  - All frames following and depending on a specific I-frame

Measured Time Series of MPEG Frame Sizes

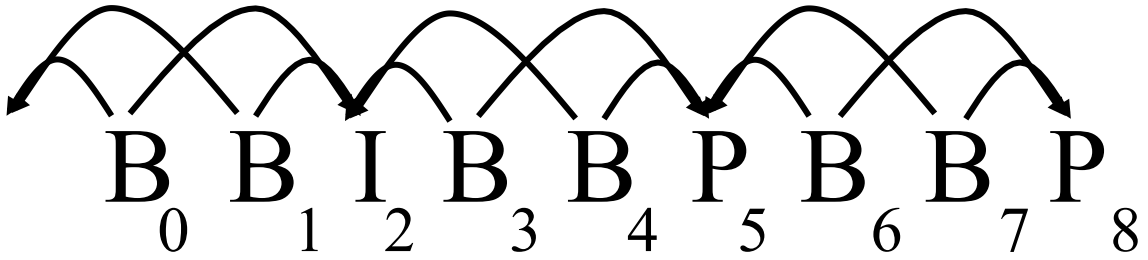


Bidirektionale Prädiktion

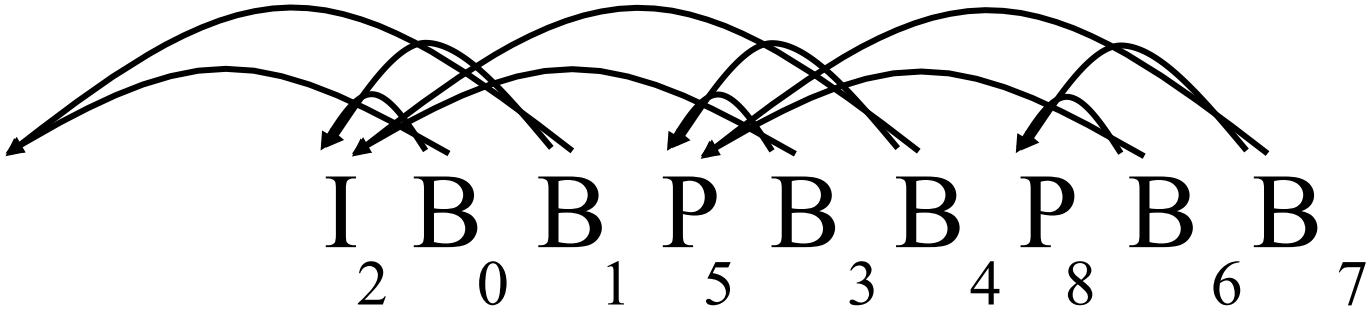


Display and Bitstream Order

- **Display order**



- **Bitstream order**



universität wien

Chair of **Future Communication**

endowed by

A1 TELEKOM AUSTRIA

Chapter 3: Multimedia Networking

Overview:

- 2.1 Multimedia Networking Applications
- 2.2 Streaming stored audio and video
- 2.3 Real-time Multimedia: Internet Phone study
- 2.4 Protocols for Real-Time Interactive Applications
  - RTP,RTCP
- 2.5 IP Telefony, SIP, and H.323
- 2.6 Distributing Multimedia: content distribution networks

Streaming Stored Multimedia

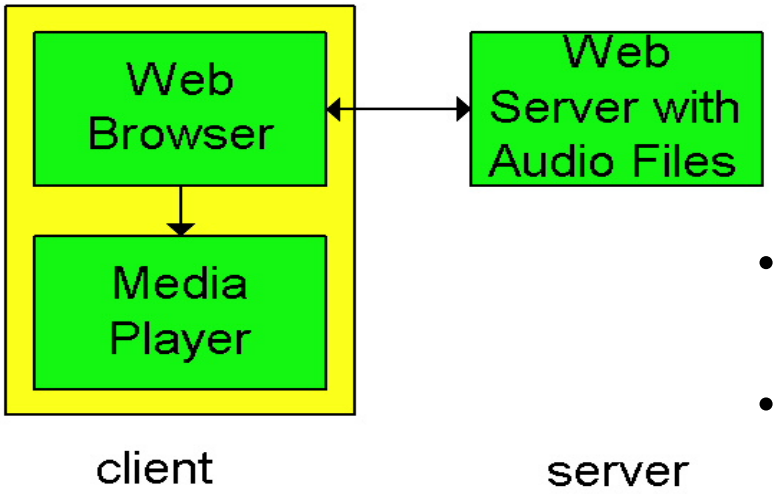
Application-level streaming techniques for making the best out of best effort service:

- client side buffering
- use of UDP versus TCP
- multiple encodings of multimedia

Media Player

- jitter removal
- decompression
- error concealment
- graphical user interface w/ controls for interactivity

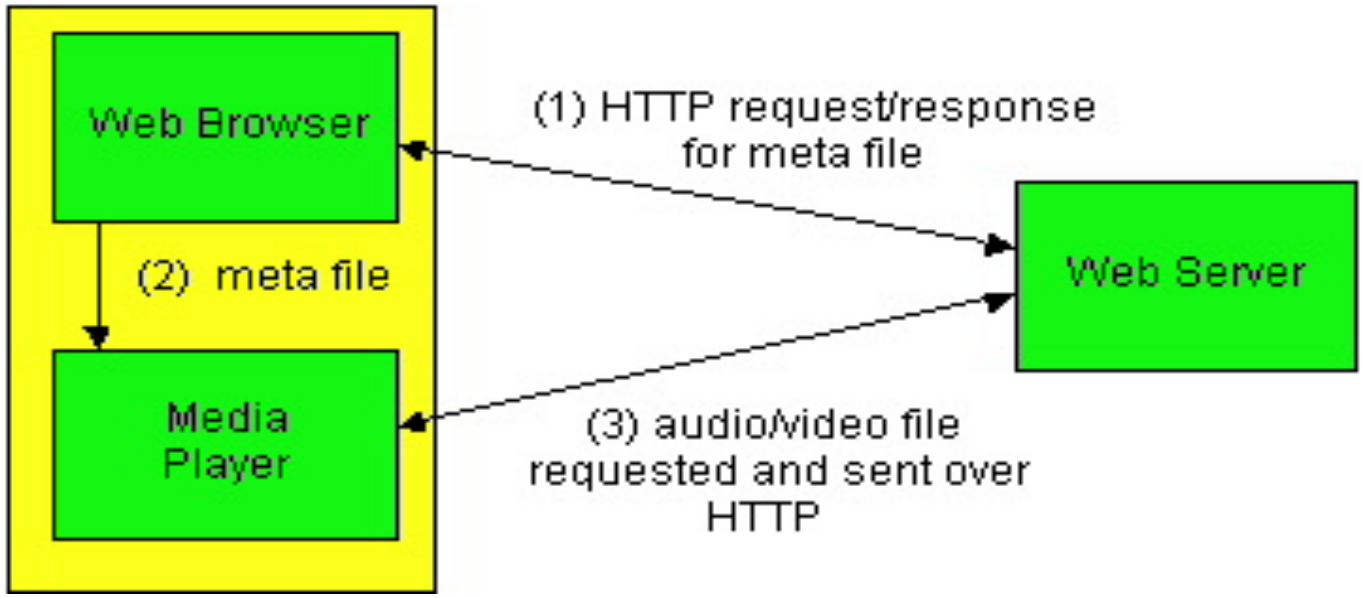
Internet multimedia: simplest approach



- **audio or video stored in file**
- **files transferred as HTTP object**
  - received in entirety at client
  - then passed to player

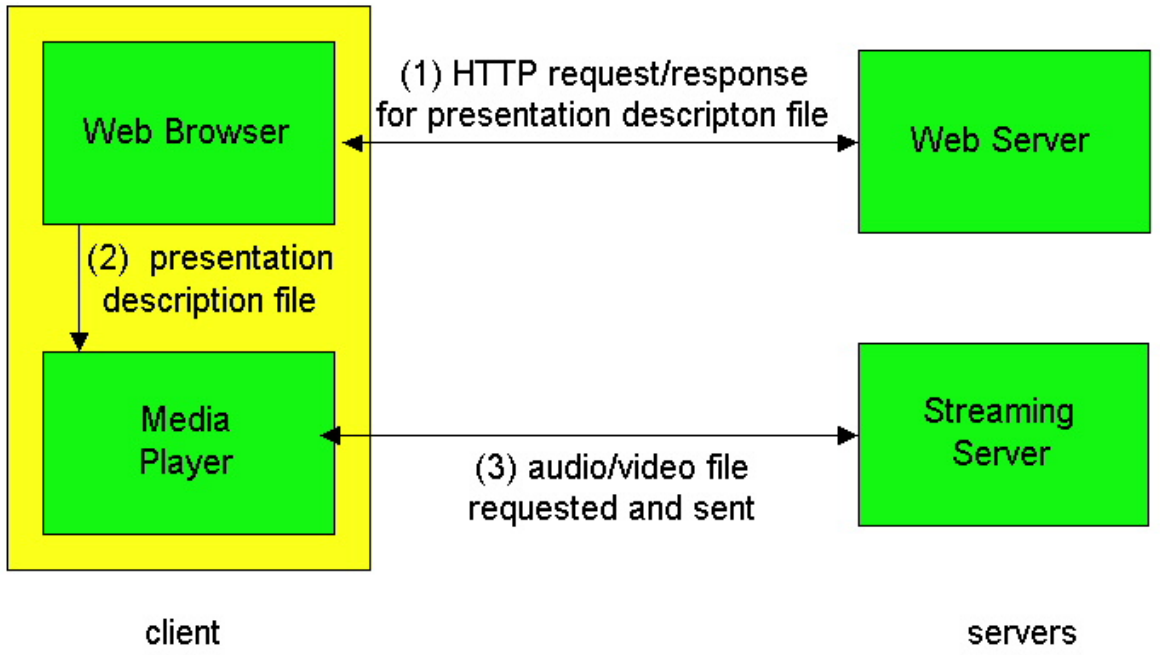
audio, video not streamed:
no, “pipelining,” long delays until playout!

Internet multimedia: streaming approach



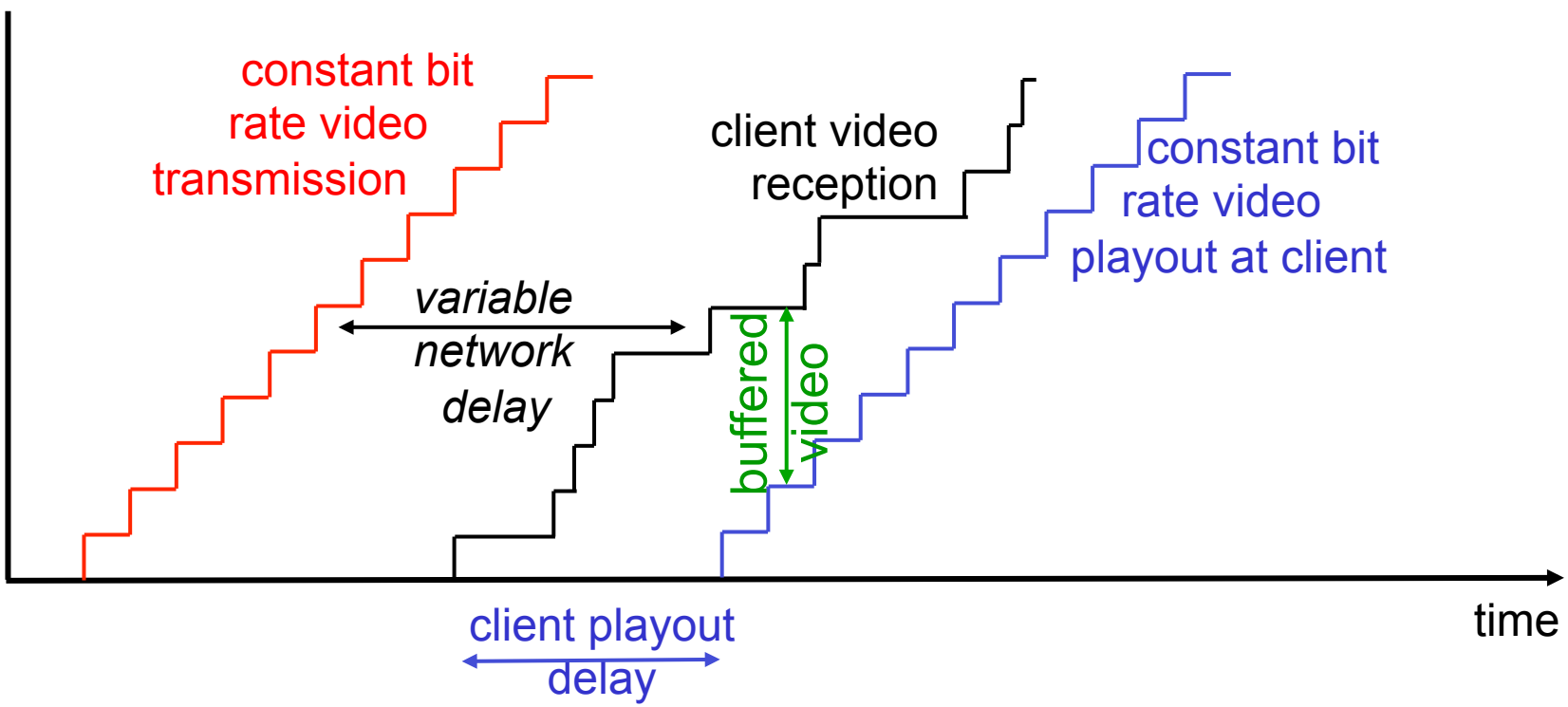
- **browser GETs metafile**
- **browser launches player, passing metafile**
- **player contacts server**
- **server streams audio/video to player**

Streaming from a streaming server



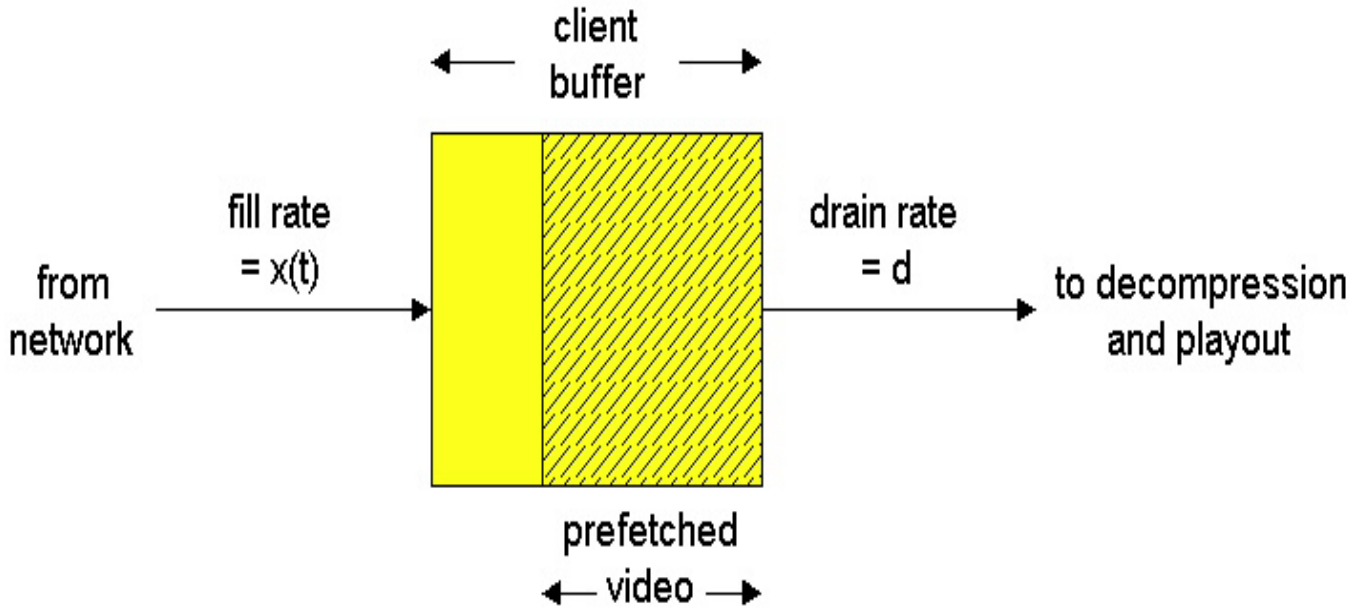
- **This architecture allows for non-HTTP protocol between server and media player**
- **Can also use UDP instead of TCP.**

Streaming Multimedia: Client Buffering



- **Client-side buffering, playout delay compensate for network-added delay, delay jitter**

Streaming Multimedia: Client Buffering



- **Client-side buffering, playout delay compensate for network-added delay, delay jitter**

Streaming Multimedia: UDP or TCP?

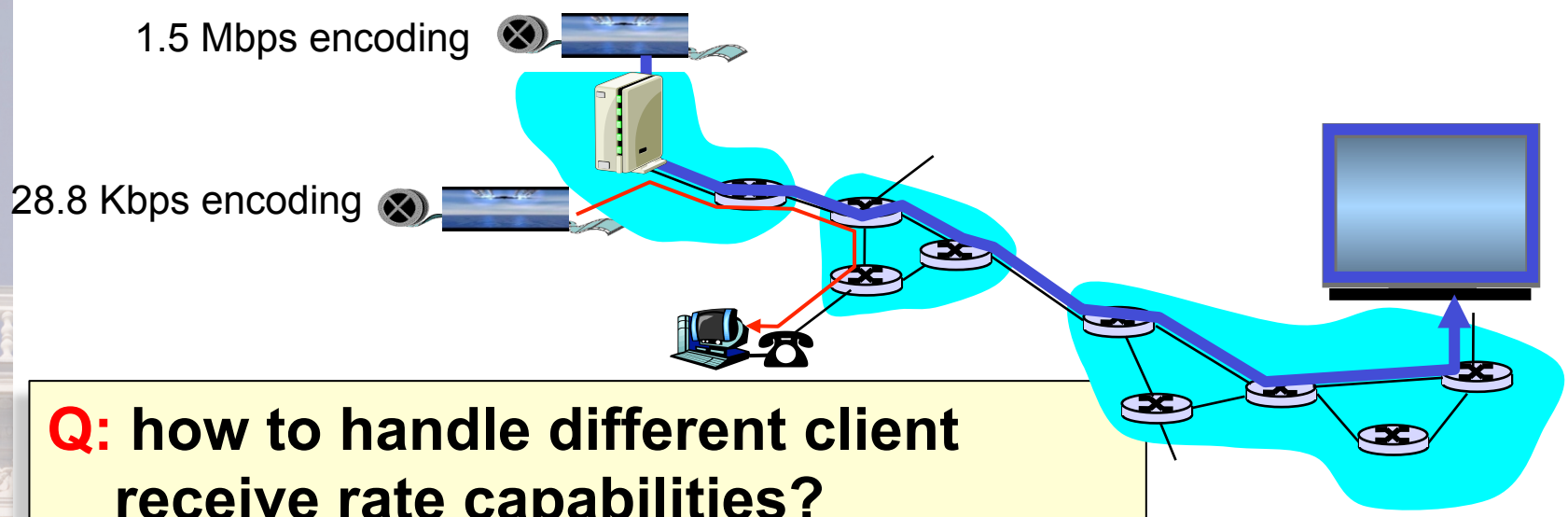
UDP

- **server sends at rate appropriate for client (oblivious to network congestion !)**
  - often send rate = encoding rate = constant rate
  - then, fill rate = constant rate - packet loss
- **short playout delay (2-5 seconds) to compensate for network delay jitter**
- **error recovery if time is permitting**

TCP

- **send at maximum possible rate under TCP**
- **fill rate fluctuates due to TCP congestion control**
- **larger playout delay: smooth TCP delivery rate**
- **HTTP/TCP passes more easily through firewalls**

Streaming Multimedia: client rate(s)



Q: how to handle different client receive rate capabilities?

- 28.8 Kbps dialup
- 100Mbps Ethernet

A: server stores, transmits multiple copies of video, encoded at different rates

User Control of Streaming Media: RTSP

HTTP

- Does not target multimedia content
- No commands for fast forward, etc.

RTSP: RFC 2326

- Realtime Streaming Protocol
- Client-server application layer protocol.
- For user to control display: rewind, fast forward, pause, resume, repositioning, etc…

What it doesn't do:

- does not define how audio/ video is encapsulated for streaming over network
- does not restrict how streamed media is transported; it can be transported over UDP or TCP
- does not specify how the media player buffers audio/ video

RTSP: out of band control

FTP uses an "out-of-band" control channel:

- **A file is transferred over one TCP connection.**
- **Control information (directory changes, file deletion, file renaming, etc.) is sent over a separate TCP connection.**
- **The "out-of-band" and "in-band" channels use different port numbers.**

RTSP messages are also sent out-of-band:

- **RTSP control messages use different port numbers than the media stream: out-of-band.**
  - Port 554
- **The media stream is considered "in-band".**

RTSP Example

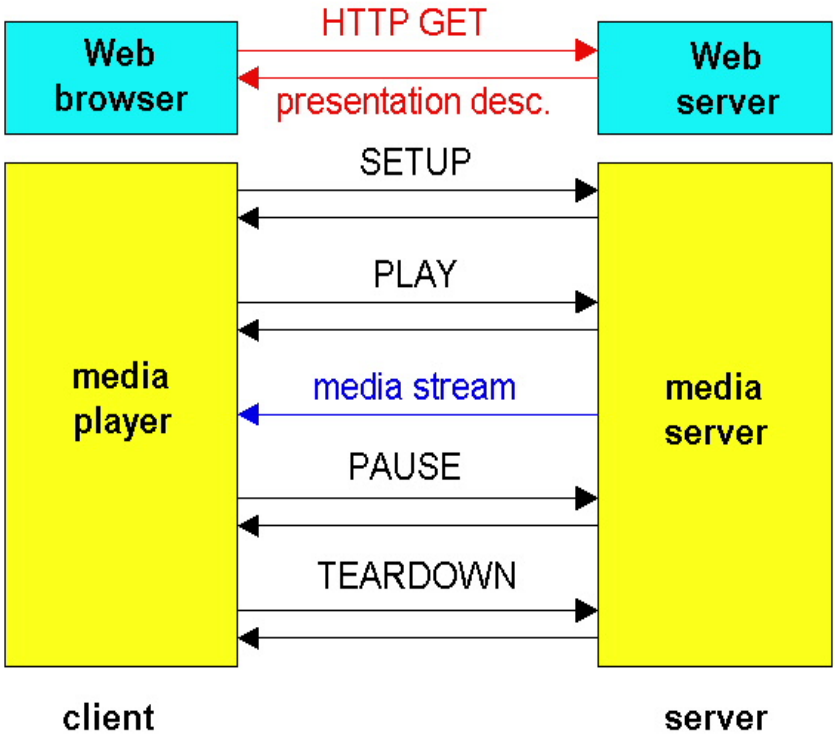
Scenario:

- metafile communicated to web browser
- browser launches player
- player sets up an RTSP control connection, data connection to streaming server

Metafile Example

```
<title>Twister</title>
<session>
    <group language=en lipsync>
        <switch>
            <track type=audio
                e="PCMU/8000/1"
                src = "rtsp://audio.example.com/twister/audio.en/
lofi">
            <track type=audio
                e="DVI4/16000/2" pt="90 DVI4/8000/1"
                src="rtsp://audio.example.com/twister/audio.en/
hifi">
        </switch>
        <track type="video/jpeg"
                src="rtsp://video.example.com/twister/video">
    </group>
</session>
```

RTSP Operation



RTSP Exchange Example

C: SETUP rtsp://audio.example.com/twister/audio RTSP/1.0
Transport: rtp/udp; compression; port=3056; mode=PLAY

S: RTSP/1.0 200 1 OK
Session 4231

C: PLAY rtsp://audio.example.com/twister/audio.en/lofi RTSP/1.0
Session: 4231
Range: npt=0-

C: PAUSE rtsp://audio.example.com/twister/audio.en/lofi RTSP/1.0
Session: 4231
Range: npt=37

C: TEARDOWN rtsp://audio.example.com/twister/audio.en/lofi RTSP/ 1.0
Session: 4231

S: 200 3 OK

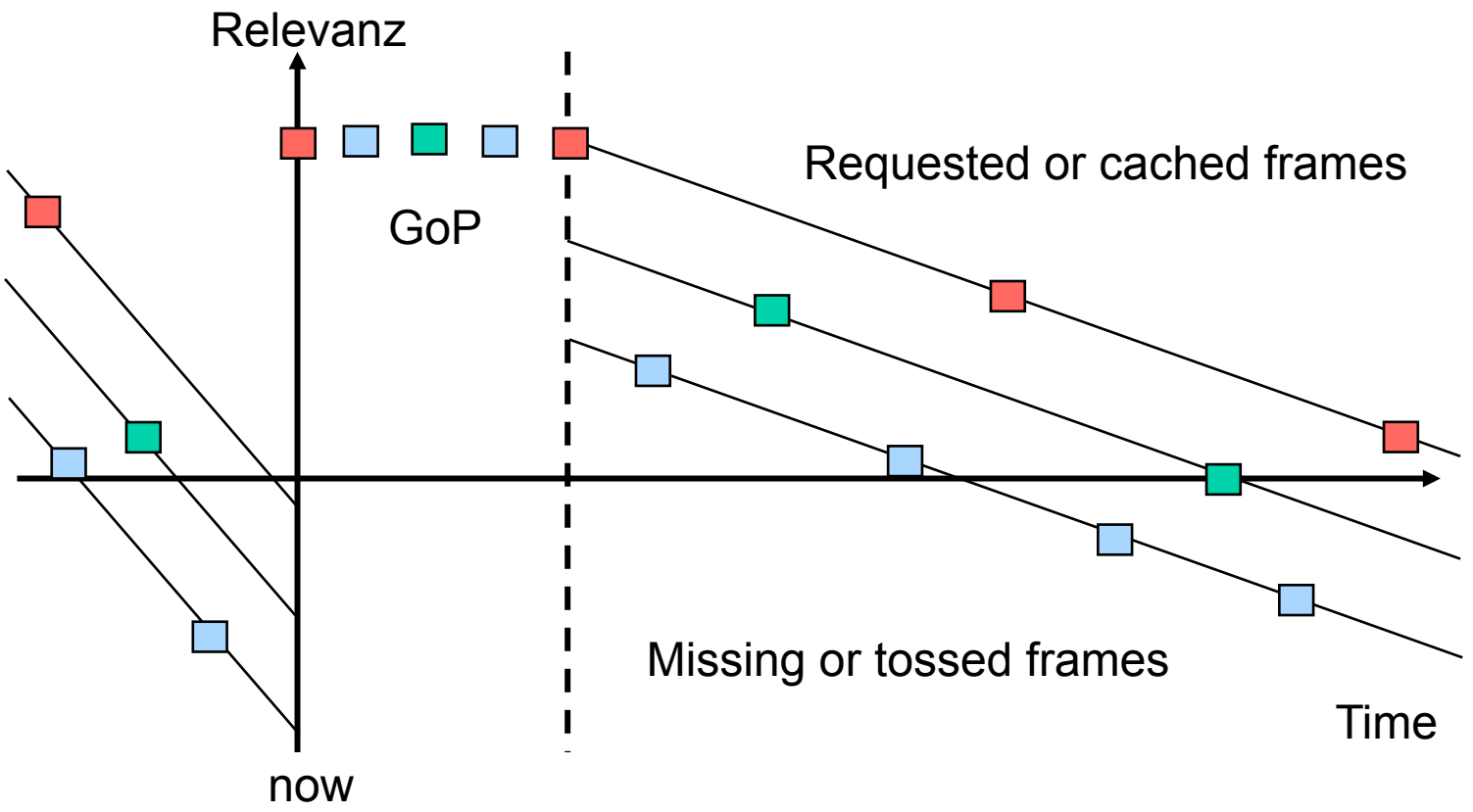
Problems with Interactive Streaming Control

- **VCR-control-like interactions desired**
  - Play forward
  - Play backward
  - Fast forward
  - Fast backward
  - Set bookmark
  - Jump to bookmark
- **Frames required for these actions might be missing**
  - ⇒ request and transmission latency
  - ⇒ Buffer and prefetch strategy required
- **Assumption**
  - Client-pull architecture: clients request frames explicitly

L/MRP Buffer Management Algorithm

- **Problems**
  - Buffer space limited
  - MPEG frames have different importance
  - Importance of streamed frames also depends on current playback time
  - Bitstream order vs. display order
- **Buffer management algorithm**
  - Idea: relevance function depends on
    - Frame type
    - Time distance to playback point / bookmark
  - Least/most relevant for presentation:
    - Request most important frames for presentation
    - Toss least important frames for presentation from buffer

Typical Relevance Function and Memory Occupation



Implementation of an Interactive Streaming Application

- **Problems**
  - Requesting too many frames in advance increases reaction time, e.g., to bookmark jumps etc.
    - Already requested frames will be delivered by the server before sending „bookmark“ frames
  - Request only most important frames if bandwidth is insufficient
    - Transmit only I-frames and save bandwidth by retaining P- and B-frames
- **Solution: controlled prefetching**
  - Requested number of frames (bytes) must not exceed a dynamic threshold
  - Consequences
    - Short response times from server
    - Automatic rate reduction of the stream

Chapter 3: Multimedia Networking

Overview:

- 2.1 Multimedia Networking Applications
- 2.2 Streaming stored audio and video
- 2.3 Real-time Multimedia: Internet Phone study
- 2.4 Protocols for Real-Time Interactive Applications
  - RTP,RTCP
- 2.5 IP Telefony, SIP, and H.323
- 2.6 Distributing Multimedia: content distribution networks

Real-time interactive applications

- **PC-2-PC phone**
  - instant messaging services are providing this
- **PC-2-phone**
  - Dialpad
  - Net2phone
- **videoconference with Webcams**

Going to now look at a PC-2-PC Internet phone example in detail

Interactive Multimedia: Internet Phone

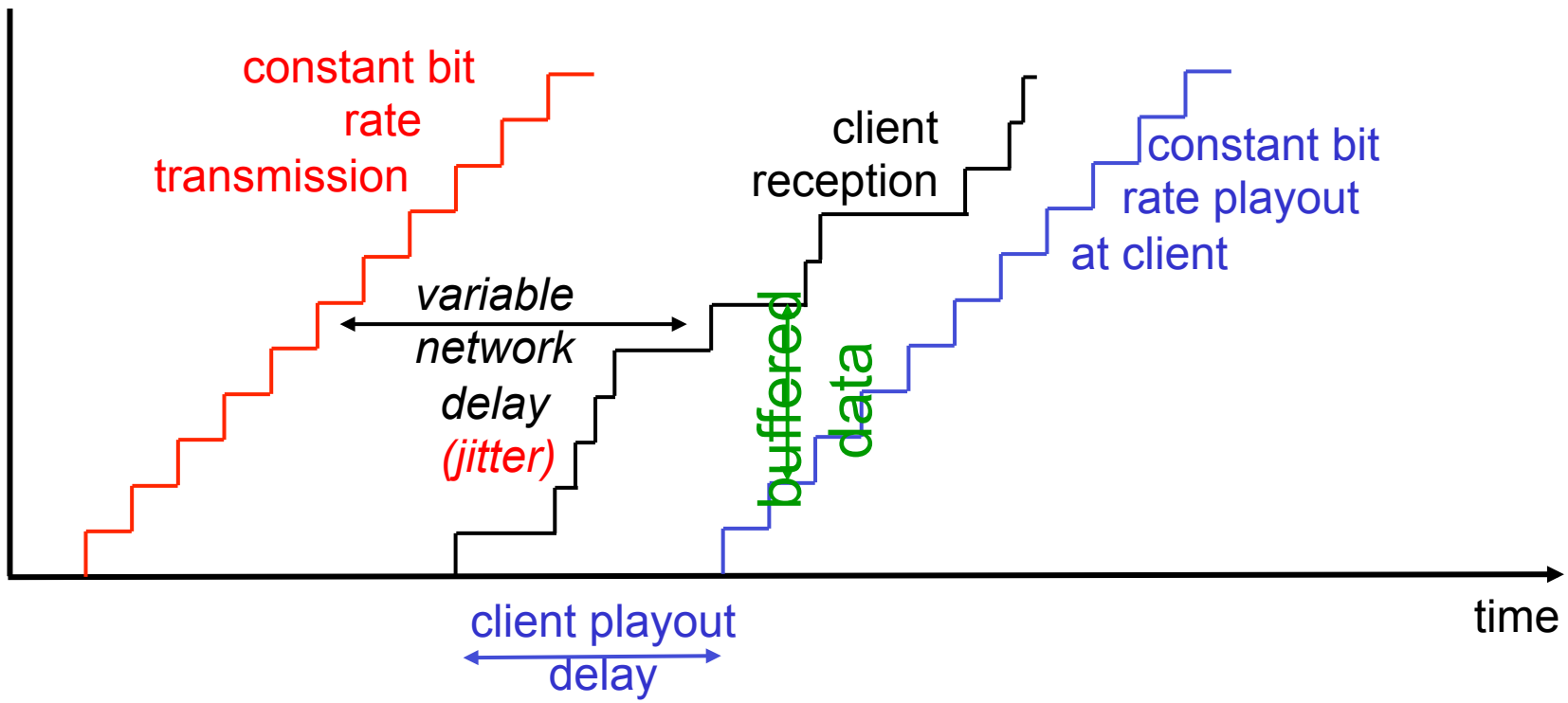
Introduce Internet Phone by way of an example

- **speaker's audio: alternating talk spurts, silent periods.**
  - 64 kbps during talk spurt
- **pkts generated only during talk spurts**
  - 20 msec chunks at 8 Kbytes/sec: 160 bytes data
- **application-layer header added to each chunk.**
- **Chunk+header encapsulated into UDP segment.**
- **application sends UDP segment into socket every 20 msec during talkspurt.**

Internet Phone: Packet Loss and Delay

- **network loss: IP datagram lost due to network congestion (router buffer overflow)**
- **delay loss: IP datagram arrives too late for playout at receiver**
  - delays: processing, queueing in network; end-system (sender, receiver) delays
  - typical maximum tolerable delay: 400 ms
- **loss tolerance: depending on voice encoding, losses concealed, packet loss rates between 1% and 10% can be tolerated.**

Delay Jitter



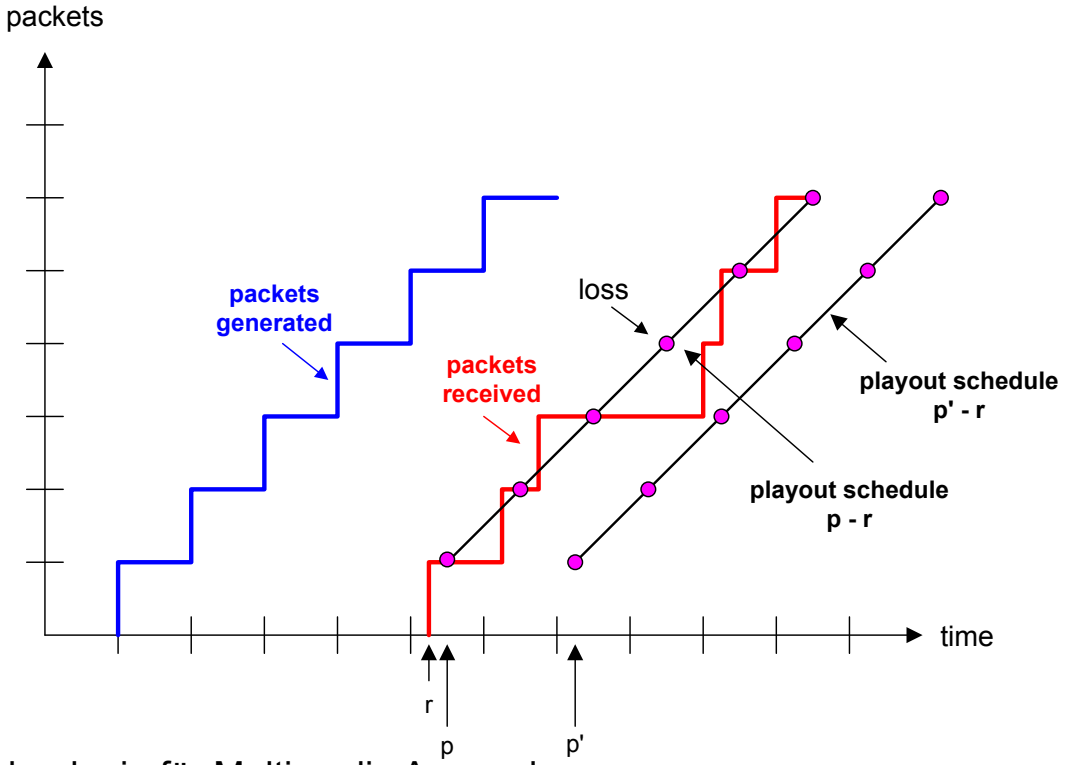
- **Consider the end-to-end delays of two consecutive packets: difference can be more or less than 20 msec**

Internet Phone: Fixed Playout Delay

- **Receiver attempts to playout each chunk exactly q msecs after chunk was generated.**
  - chunk has time stamp t: play out chunk at t+q .
  - chunk arrives after t+q: data arrives too late for playout, data “lost”
- **Tradeoff for q:**
  - large q: less packet loss
  - small q: better interactive experience

Fixed Playout Delay

- Sender generates packets every 20 msec during talk spurt.
- First packet received at time r
- First playout schedule: begins at p
- Second playout schedule: begins at p'



Adaptive Playout Delay, I

- **Goal: minimize playout delay, keeping late loss rate low**
- **Approach: adaptive playout delay adjustment:**
  - Estimate network delay, adjust playout delay at beginning of each talk spurt.
  - Silent periods compressed and elongated.
  - Chunks still played out every 20 msec during talk spurt.

t_i=timestamp of the ith packet

r_i=the time packet i is received by receiver

p_i=the time packet i is played at receiver

$r_i - t_i$=network delay for ith packet

d_i=estimate of average network delay after receiving ith packet

Dynamic estimate of average delay at receiver:

$$d_i = (1-u)d_{i-1} + u(r_i - t_i)$$

where u is a fixed constant (e.g., u = .01).

Adaptive playout delay II

Also useful to estimate the average deviation of the delay, *vi* :

$$v_i = (1-u)v_{i-1} + u\,|\,r_i - t_i - d_i\,|$$

The estimates *di* and *vi* are calculated for every received packet, although

they are only used at the beginning of a talk spurt.

For first packet in talk spurt, playout time is:

$$p_i = t_i + d_i + Kv_i$$

where K is a positive constant.

Remaining packets in talkspurt are played out periodically

Adaptive Playout, III

Q: How does receiver determine whether packet is first in a talkspurt?

- **If no loss, receiver looks at successive timestamps.**
  - difference of successive stamps > 20 msec -->talk spurt begins.
- **With loss possible, receiver must look at both time stamps and sequence numbers.**
  - difference of successive stamps > 20 msec and sequence numbers without gaps --> talk spurt begins.

Recovery from packet loss (1)

forward error correction (FEC): simple scheme

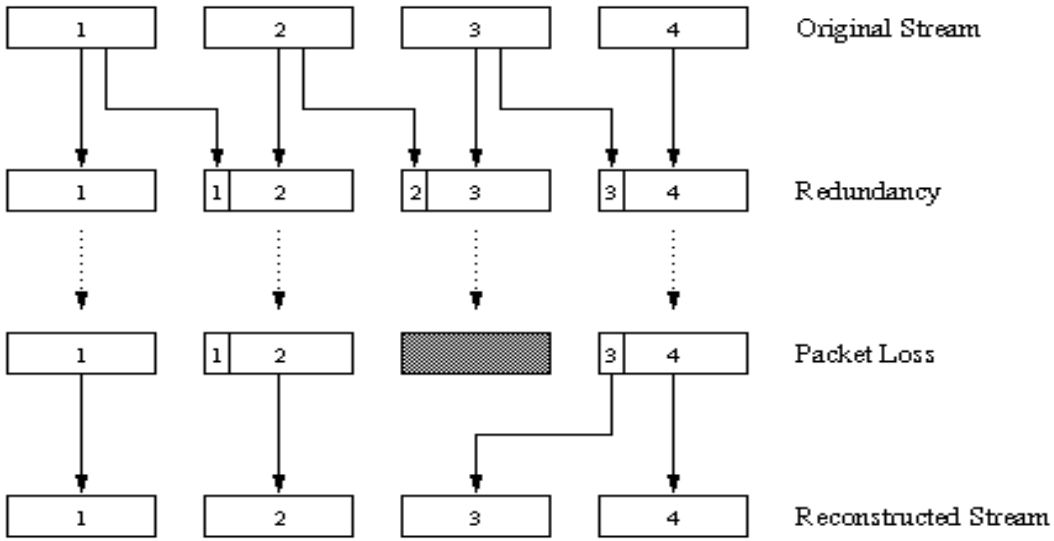
- **for every group of n chunks create a redundant chunk by exclusive OR-ing the n original chunks**
- **send out n+1 chunks, increasing the bandwidth by factor 1/n.**
- **can reconstruct the original n chunks if there is at most one lost chunk from the n+1 chunks**

- **Playout delay needs to be fixed to the time to receive all n+1 packets**
- **Tradeoff:**
  - increase n, less bandwidth waste
  - increase n, longer playout delay
  - increase n, higher probability that 2 or more chunks will be lost

Recovery from packet loss (2)

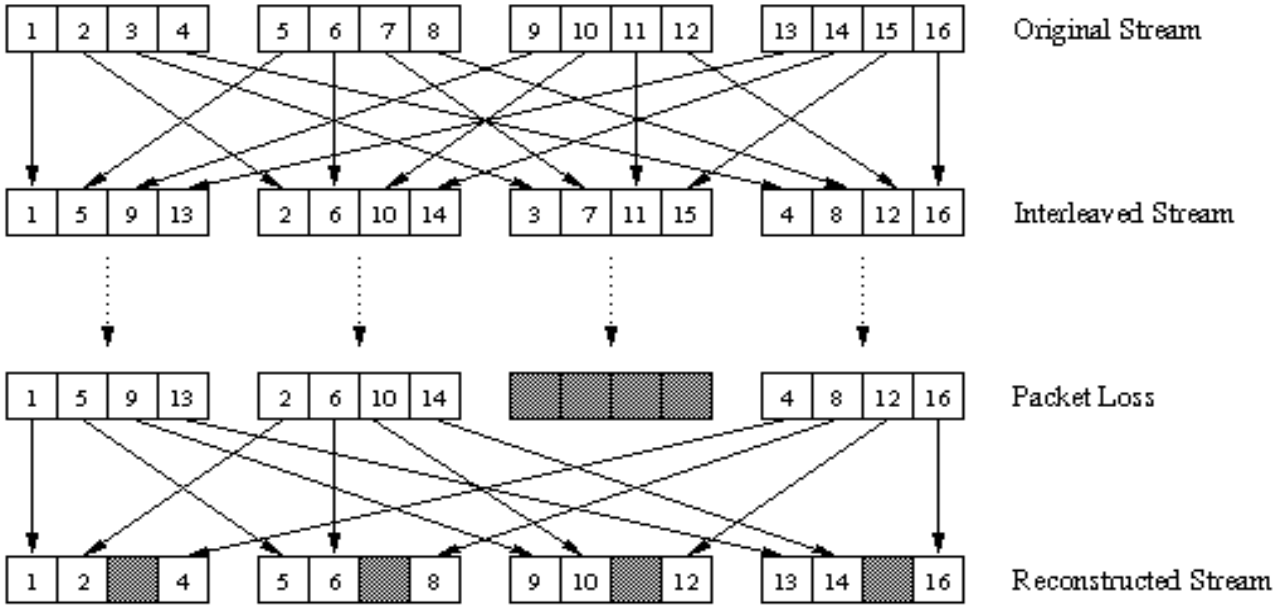
2nd FEC scheme

- "piggyback lower quality stream"
- send lower resolution audio stream as the redundant information
- for example, nominal stream PCM at 64 kbps
  and redundant stream GSM at 13 kbps.



- Whenever there is non-consecutive loss, the receiver can conceal the loss.
- Can also append (n-1)st and (n-2)nd low-bit rate chunk

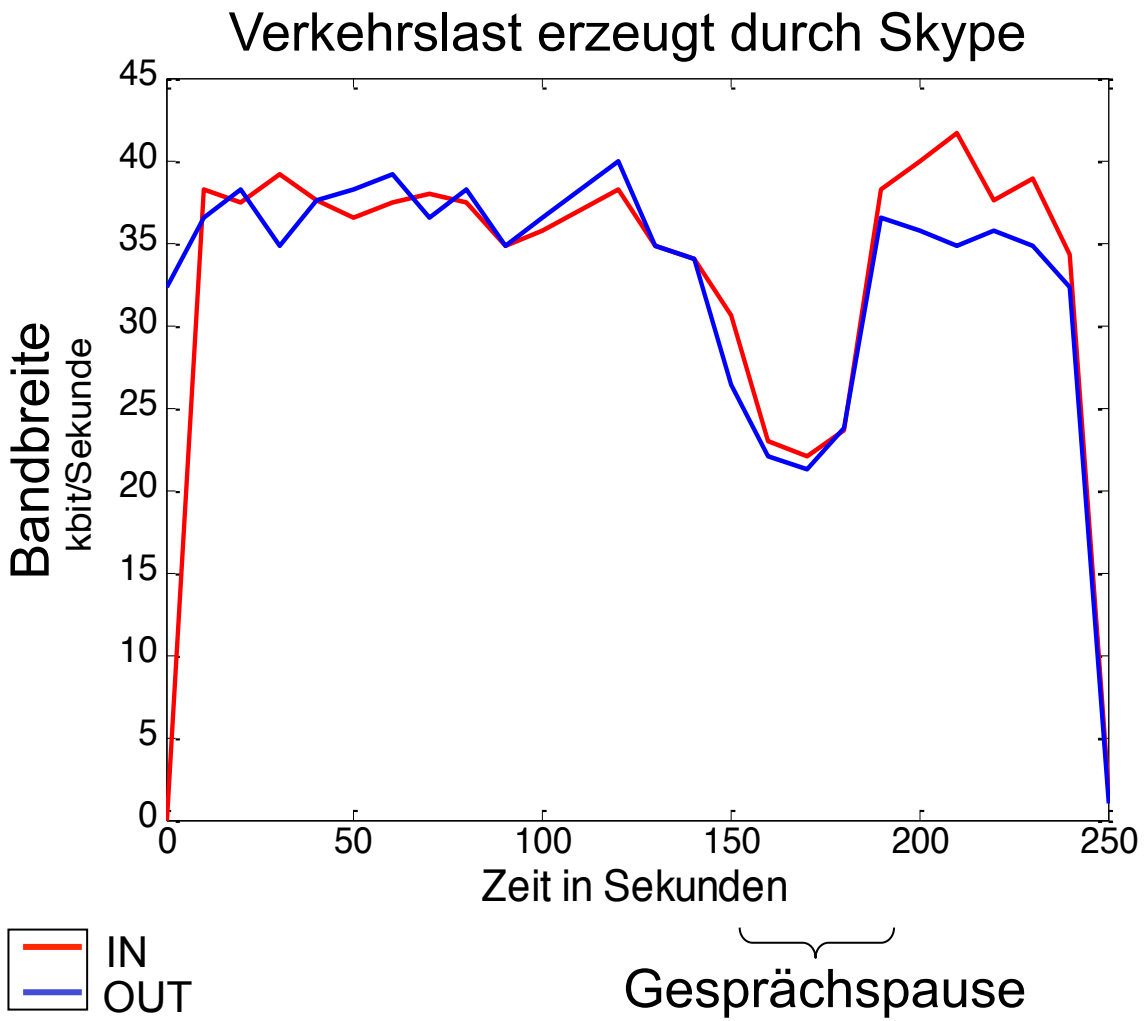
Recovery from packet loss (3)



Interleaving

- **chunks are broken up into smaller units**
- **for example, 4 5 msec units per chunk**
- **Packet contains small units from different chunks**
- **if packet is lost, still have most of every chunk**
- **has no redundancy overhead**
- **but adds to playout delay**

Bandbreitenmessung VoIP (Skype)



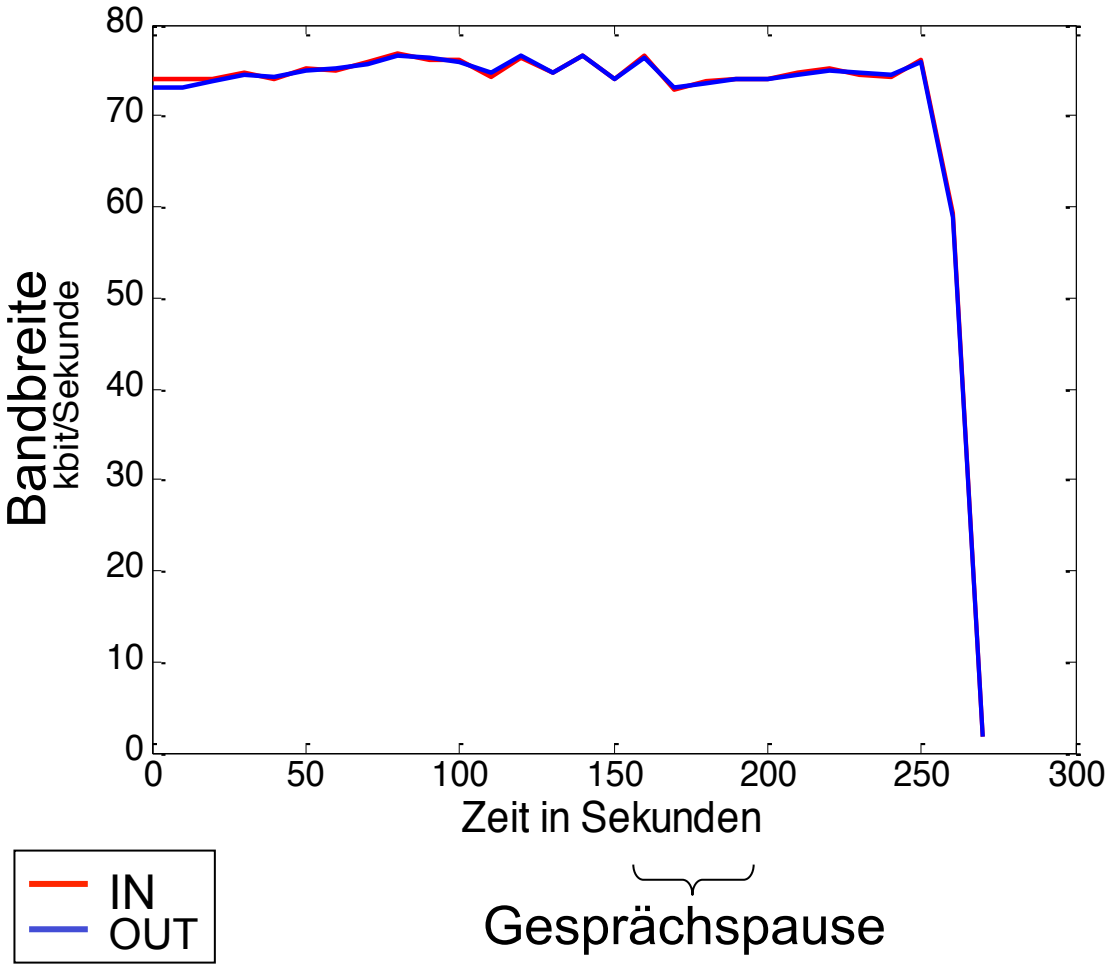
Mittelwert:
IN: 32,4 kbit/s
OUT: 32,6 kbit/s

Chair of Future Communication

endowed by

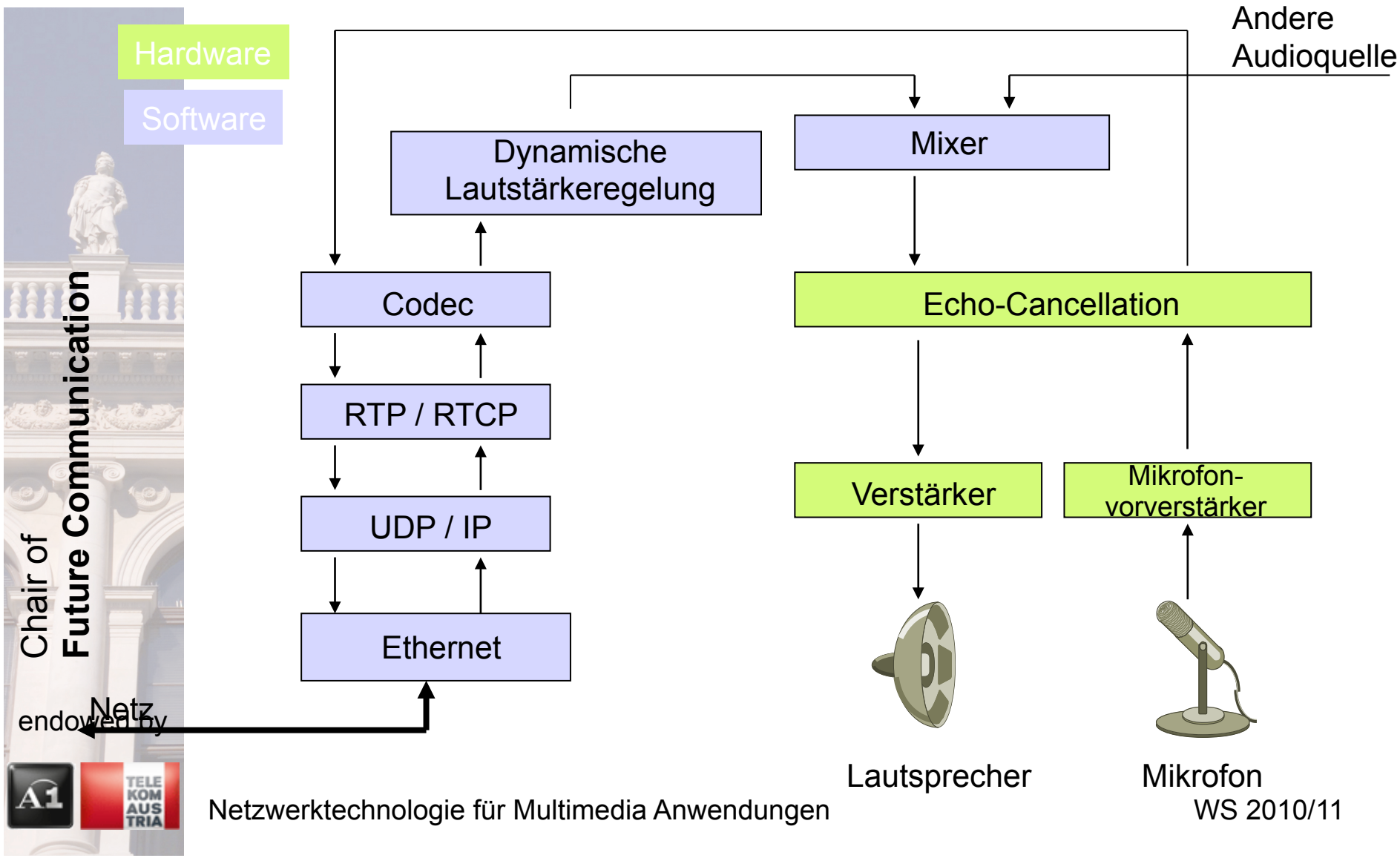
Bandbreitenmessung VoIP (SIP)

Verkehrslast erzeugt durch SIP (Codec: G711u)



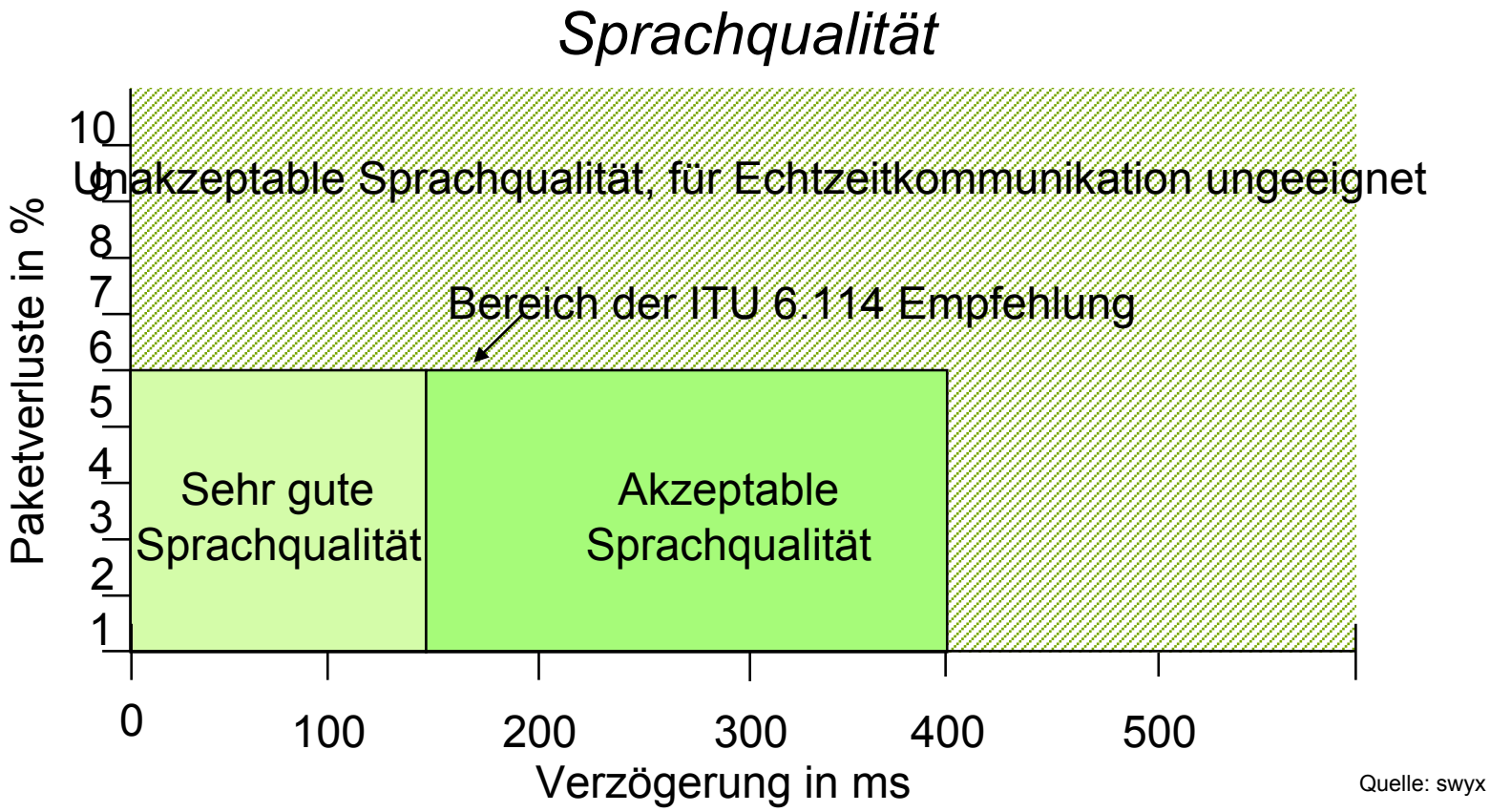
Mittelwert:
IN: 71,8 kbit/s
OUT: 71,6 kbit/s

Lokale Enflussfaktoren - Beispiel PC



Anforderungen Sprachqualität

- **Die Qualität der Sprache in IP-Netzen hängt wesentlich von den Paketverlusten und von der Verzögerung ab.**



Quelle: swyx

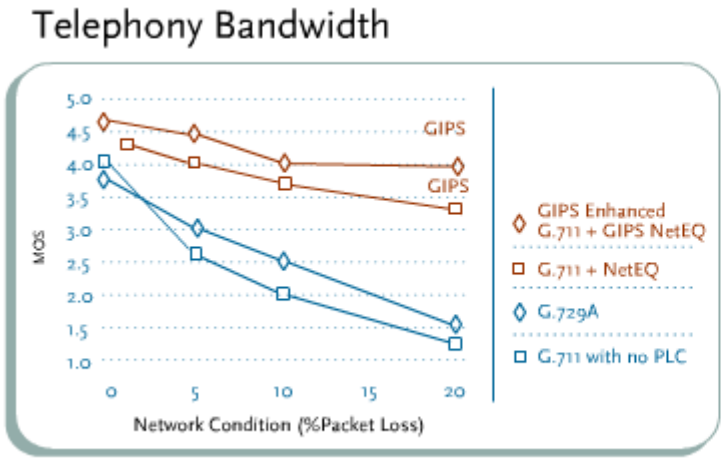
Bewertung der Übertragungsgüte

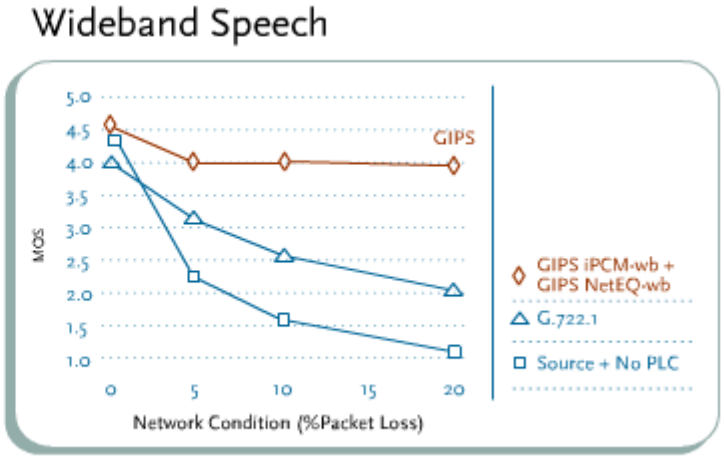
- **Bewertung der subjektiven Übertragungsqualität**
  - Übertragung einer vorgegebenen Audiodatei
  - Aufzeichnung am Empfänger
  - Vergleich mittels eines standardisierten Algorithmus ergibt:
    - Mean Opinion Score (MOS)

Sehr gute Sprachqualität, in leiser Umgebung	Excellent	5.0
Natürliche Sprachqualität, wie digitales Telefon	Good	4.0
Akzeptabel, erfordert aber teilweise Konzentration	Fair	3.0
Schwer zu verstehende Sprache	Poor	2.0
Kaum zu verstehen, Unterbrechungen	Bad	1.0

Skype - Sprachqualität

- **Einsatz eines speziellen Voicecodecs**
  - Global IP Sound





  - Reagiert adaptiv auf Bandbreitenveränderungen
  - Kann das Verhältnis Prozessor-/Bandbreitenlast optimieren

Quelle: www.globalipsound.co

universität wien

Summary: Internet Multimedia: bag of tricks

- **use UDP to avoid TCP congestion control (delays) for time-sensitive traffic**
- **client-side adaptive playout delay: to compensate for delay**
- **server side matches stream bandwidth to available client-to-server path bandwidth**
  - chose among pre-encoded stream rates
  - dynamic server encoding rate
- **error recovery (on top of UDP)**
  - FEC, interleaving
  - retransmissions, time permitting
  - conceal errors: repeat nearby data