Chair of Future Communication
Faculty of Computer Science
Prof. Dr. K. Tutschku

050069

VO Netzwerktechnologie für Multimedia Anwendungen

Lecture 5: Multimedia Networking

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

Bachelor Informatik (Medieninformatik)
WS 2011/12

Endowed by

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

universität wien

Chapter 2: Multimedia Networking

Chair of **Future Communication**

endowed by A1

Overview:

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

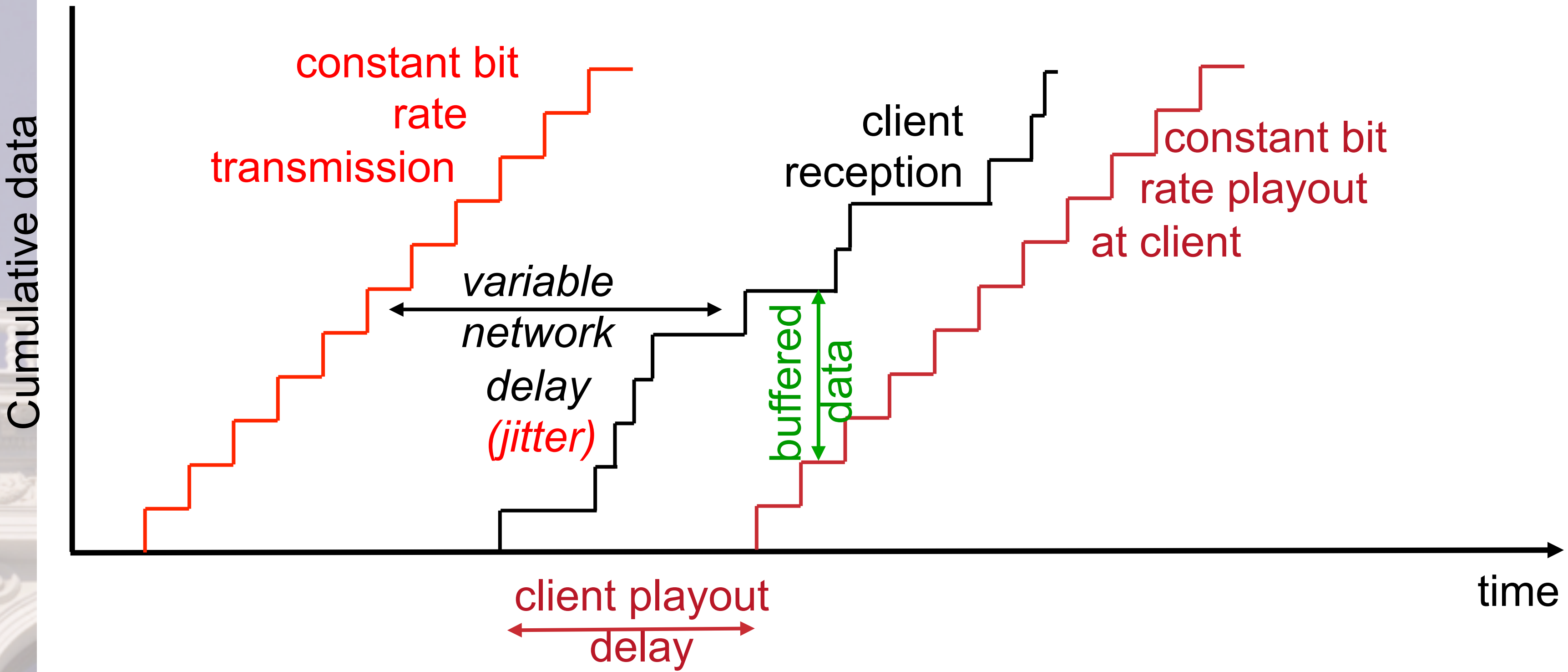
Introduce Internet Phone by way of an example

- 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



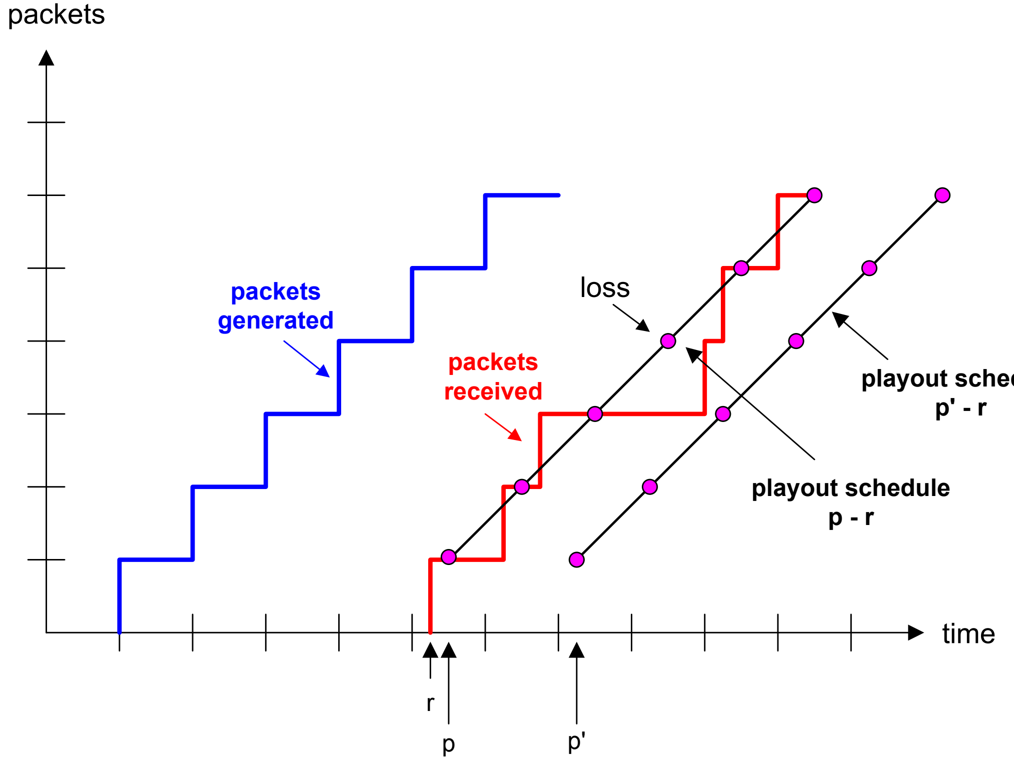
- Consider the end-to-end delays of two consecutive packets: difference can be more or less than 20 msec

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'



Chair of
Future Communication
endowed by

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

Chair of **Future Communication**

endowed by

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

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

The estimates *di* and *vi* are calculated for every received packet, altho they are only used at the beginning of a talk spurt.

For first packet in talk spurt, playout time is:

$$\boxed{p_i = t_i + d_i + Kv_i}$$

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

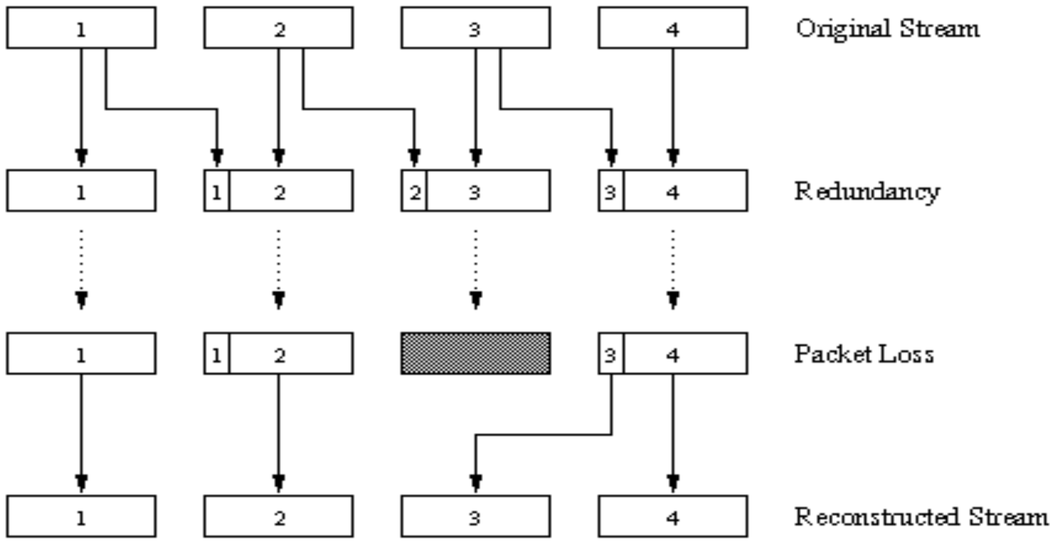
Recovery from packet loss (2)

Chair of **Future Communication**

endowed by

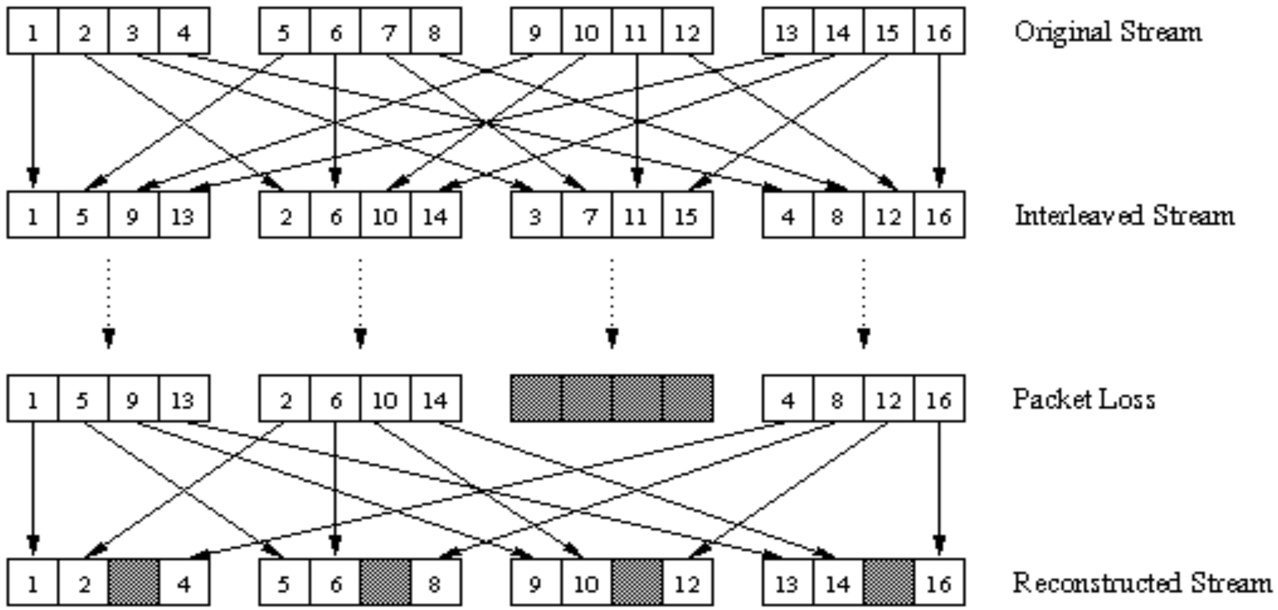
2nd FEC scheme

- "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

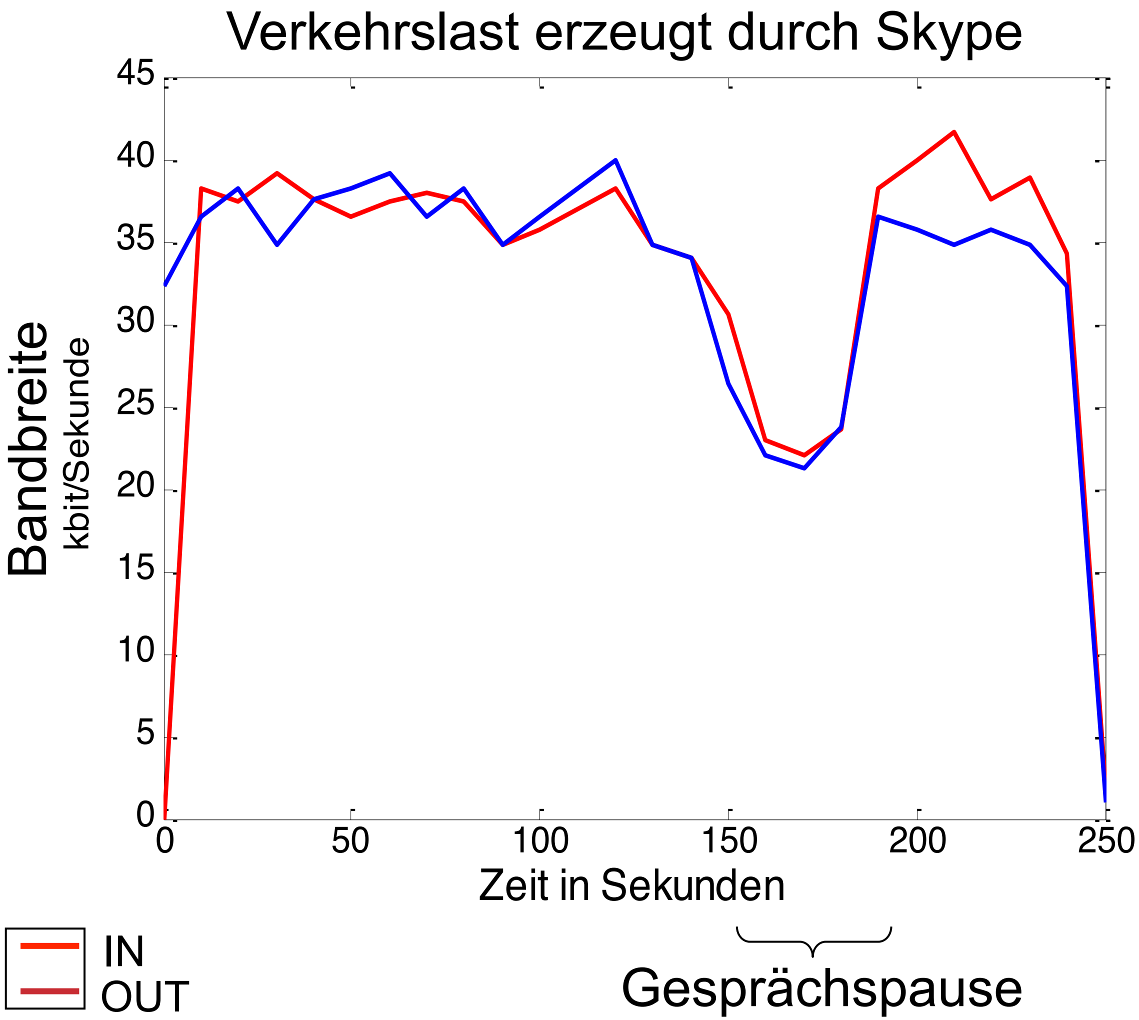
- if packet is lost, still have most of every chunk
- has no redundancy overhead
- but adds to playout delay

Chair of **Future Communication**

endowed by

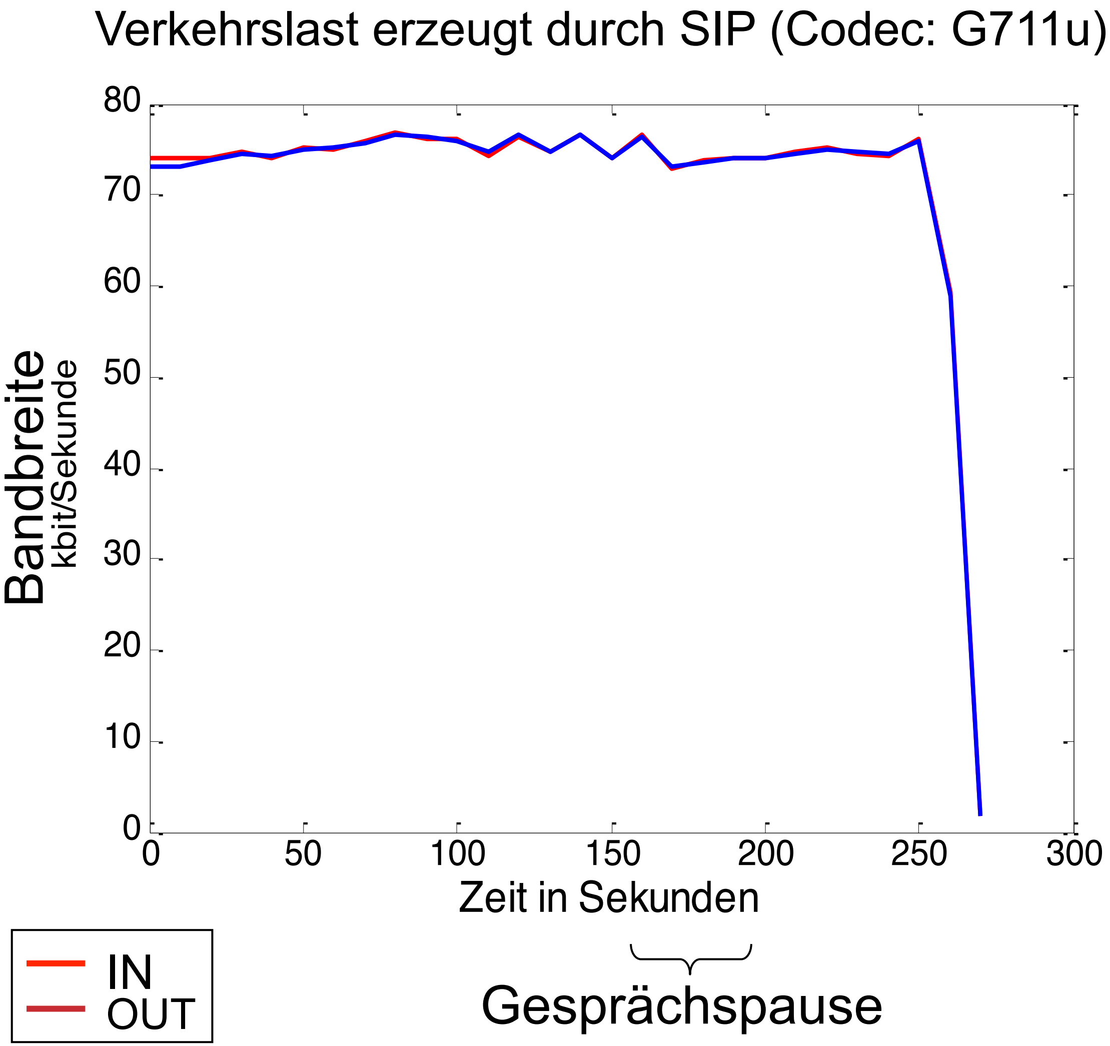
universität wien

Bandbreitenmessung VoIP (Skype)



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

Bandbreitenmessung VoIP (SIP)

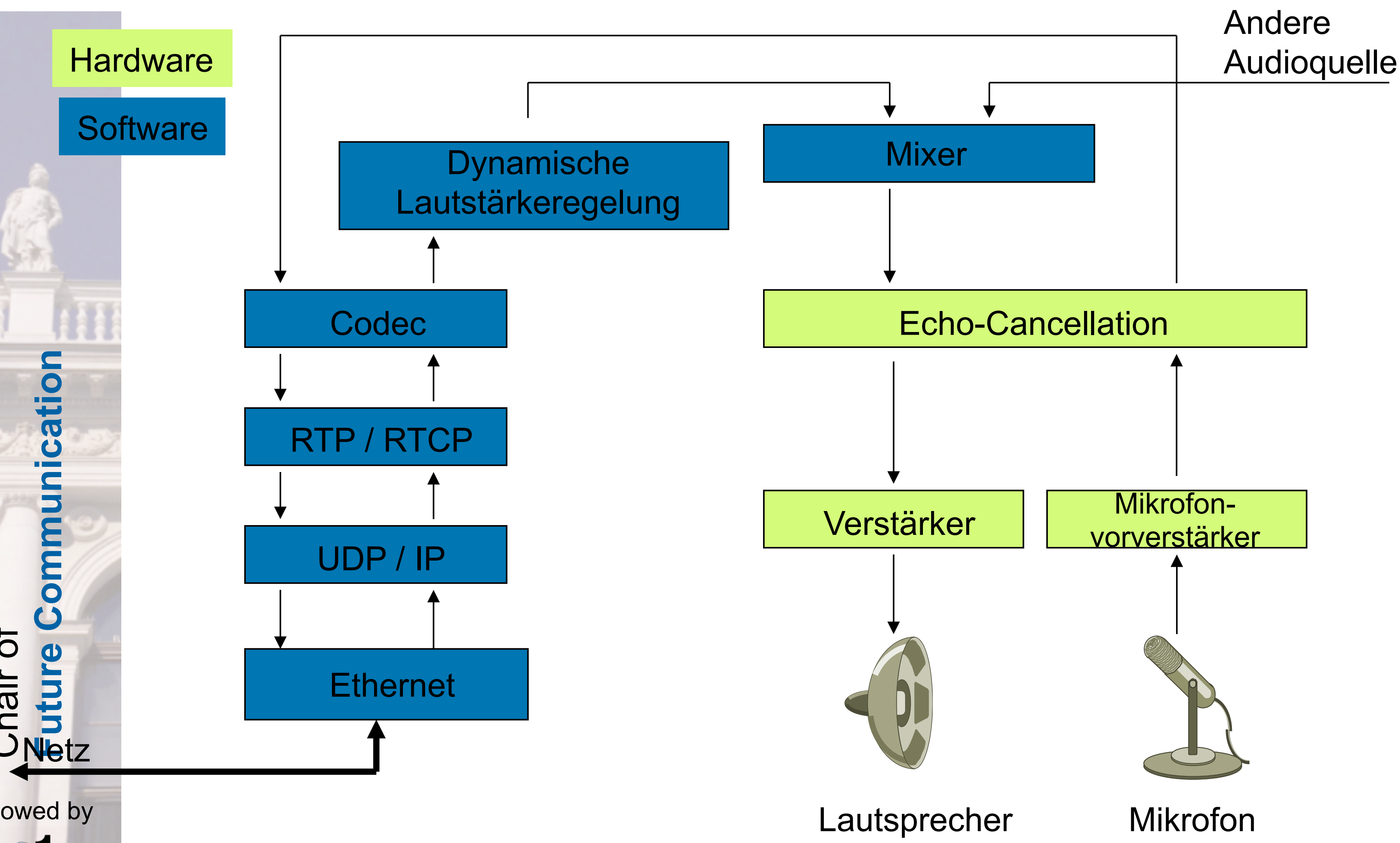


Mittelwert:
IN: 71,8 kbit/s
OUT: 71,6 kbit/s

Chair of **Future Communication**

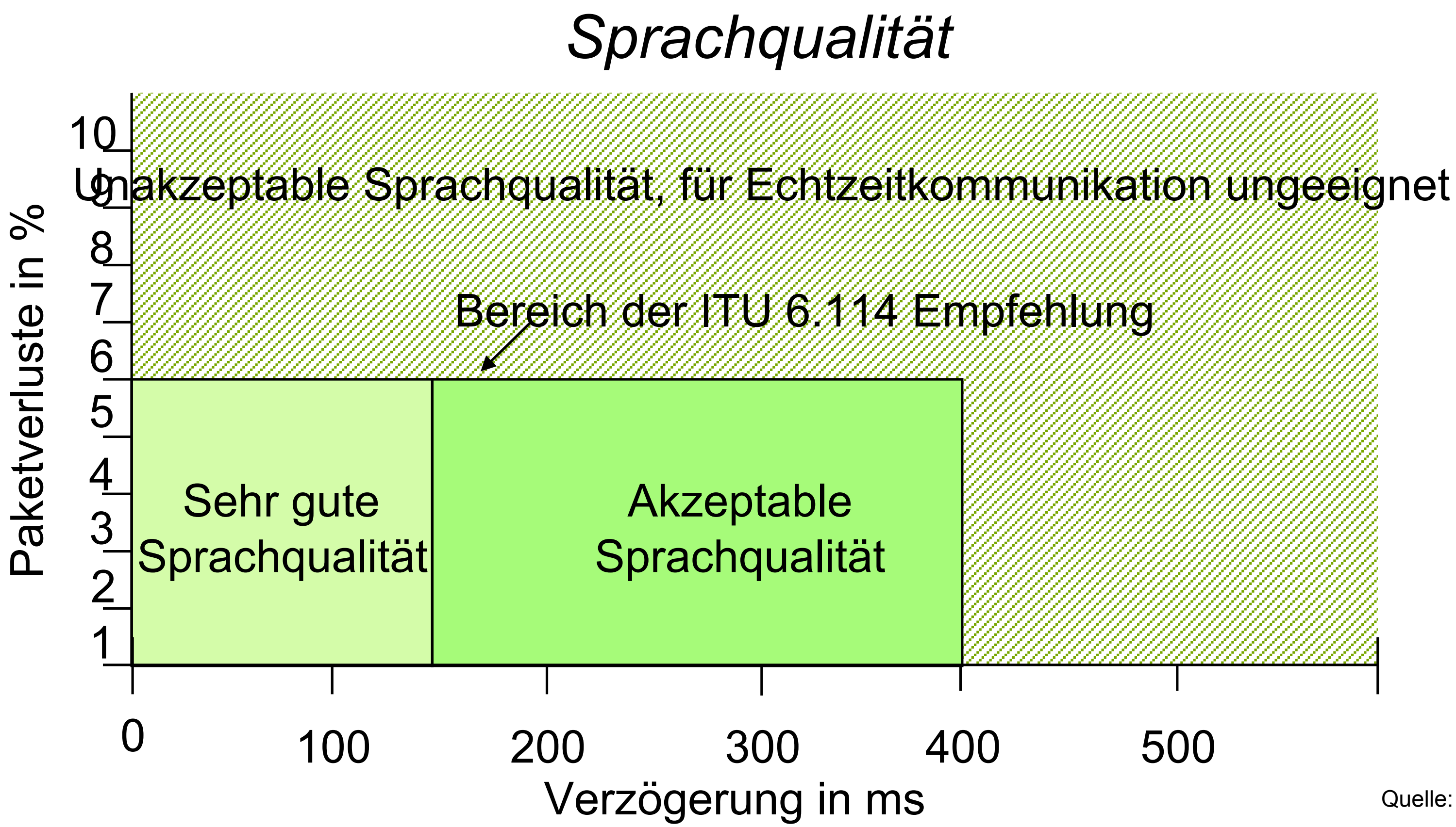
endowed by A1

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

Chair of **Future Communication**

endowed by

Bewertung der Übertragungsgüte

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

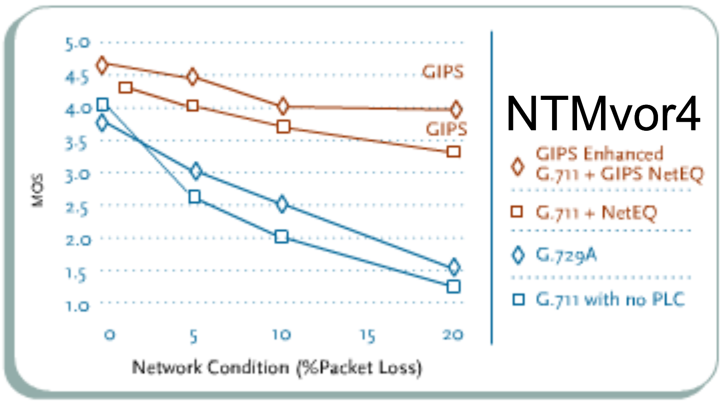
Sehr gute Sprachqualität in leiser Umgebung	Excellent	5.0
Natürliche Sprachqualität wie digitales Telefon	Good	4.0
Akzeptabel, erfordert aber teilweise Konzentration	Fair	3.0
Schwer zu verstehende Sprache	Poor	2.0
Kaum zu verstehen, Unterbrechungen	Bad	1.0

universität wien

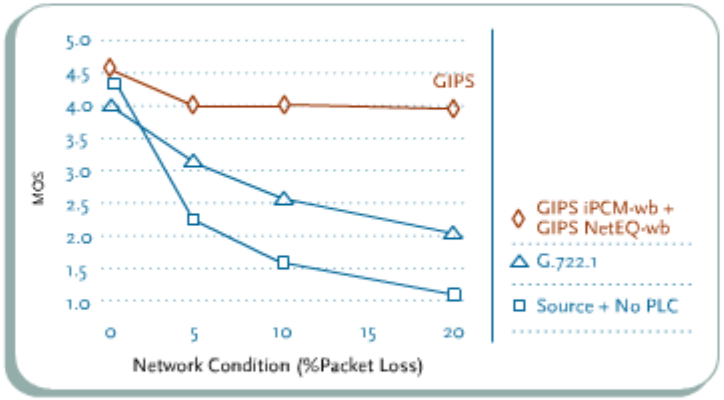
Skype - Sprachqualität

- Einsatz eines speziellen Voicecodecs
  - Global IP Sound

Telephony Bandwidth



Wideband Speech



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

Quelle: www.globalipsound.co

Chair of Future Communication

endowed by A1

Our Research: What do I have to do when I want to apply QoE?

1. How can QoE be measured in general?

2. What metric can be applied and how can this metric be matched with the providers view of the networks?

3. How can a scalable, adaptive, inter-domain measurement concept be implemented?

However:

- Applications require different metrics!
- Is a generic measurement concept feasible at all?
- What happens if domains do not collaborate?

Our Research: The IQX Hypothesis

Chair of **Future Communication**

endowed by A1

Example:

If we dined in a five-star restaurant, a single spot on the clean white table cloth strongly disturbs the experience. The same incident appears much less severe in a beer pub.

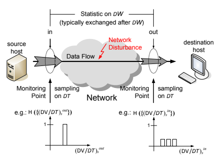
The IQX Hypothesis [Hossfeld, Fiedler, Tutschku et al. '07/08]

- QoE is function QoS, i.e. $QoE = f(QoS)$
- The subjective sensibility on QoE is more sensitive, the higher the experienced quality is.
- If the QoE is very high, a small disruption (i.e. decrease of QoS) will decrease strongly the QoE.
- Mathematical description: $\frac{\partial QoE}{\partial QoS} = -\tilde{\beta} \cdot (QoE - \gamma).$

→ Only possible solution: exponential relationship (IQX Hypothesis)

$$\boxed{QoE = \alpha \cdot \exp(-\beta \cdot QoS) + \gamma.}$$

End-to-End Comparative Monitoring



Possible application : audio/video quality

MOS (Mean Opinion Score; ITU-T. rec. P.800.1): numerical indication of the perceived quality of received media after compression and/or transmission

MOS range:

MOS	quality	Impairment
5	excellent	imperceptible
4	good	perceptible but not annoying
3	fair	slightly annoying
2	poor	annoying
1	bad	very annoying

Multiplicative relationship model

$$(MOS_{Out}-1) = f_{Utility}(MOS_{In}) \approx U_{Netw,MOS} \cdot (MOS_{In}-1)$$

Chair of **Future Communication**

endowed by

Test of the IQX Hypothesis

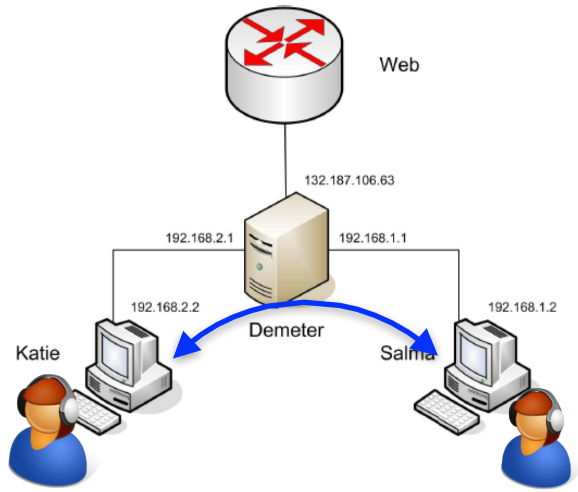
Chair of
Future Communication

endowed by

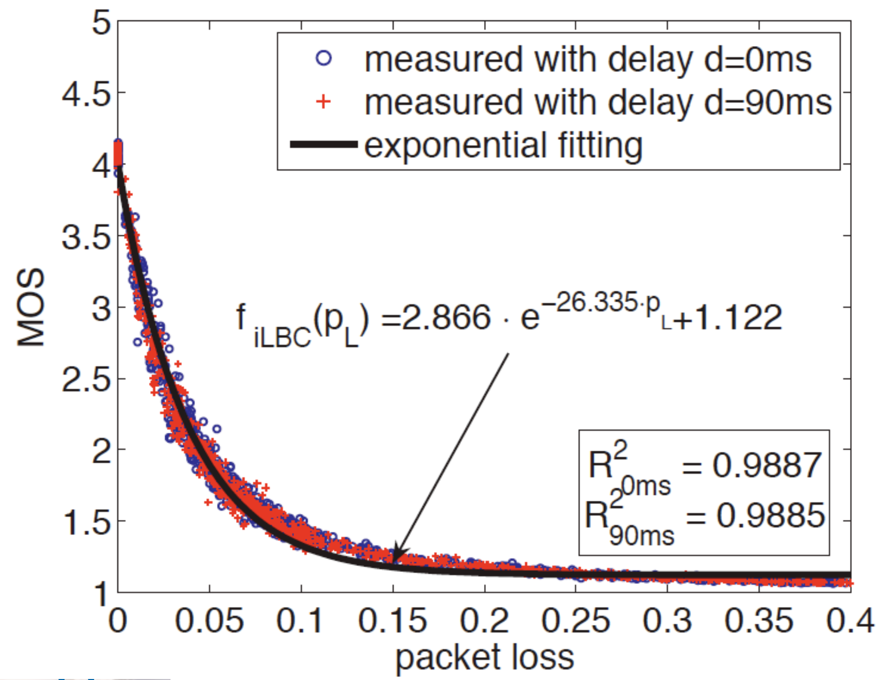
Case Study: Audio Codecs
- iLBC
- G.711

Setup:
- use of comparative measurements
- computation of PESQ values mapped to MOS at input and output
- automatic initiation of test calls
- variation of QoS, i.e., of packet delay/packet loss, at inter-connecting router (Demeter)
- non-linear regression for fitting QoS onto the QoE mapping function:

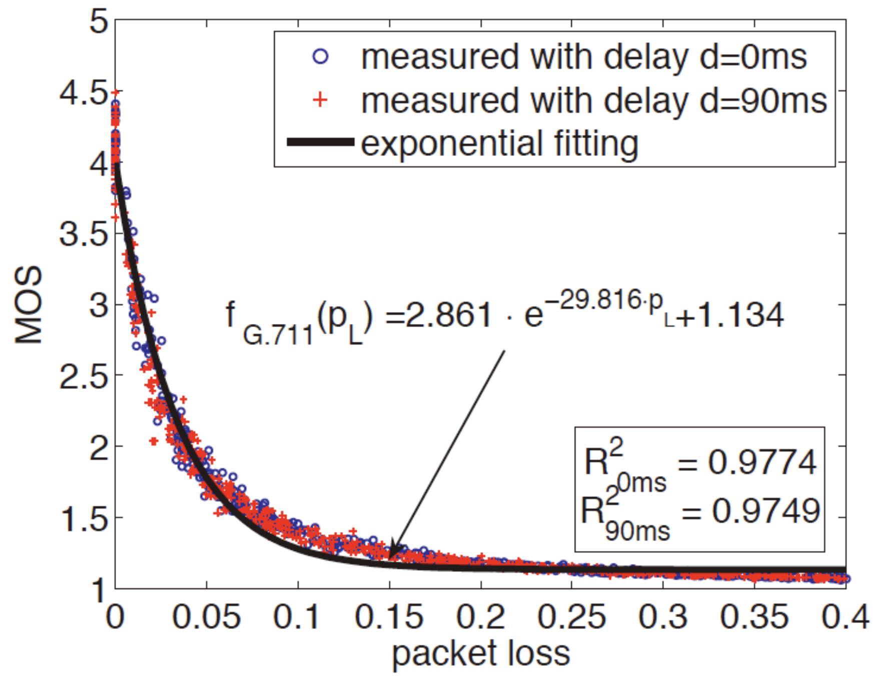
$$QoE = \alpha \cdot \exp(-\beta \cdot QoS) + \gamma.$$



Test of the IQX Hypothesis: Results for Packet Loss

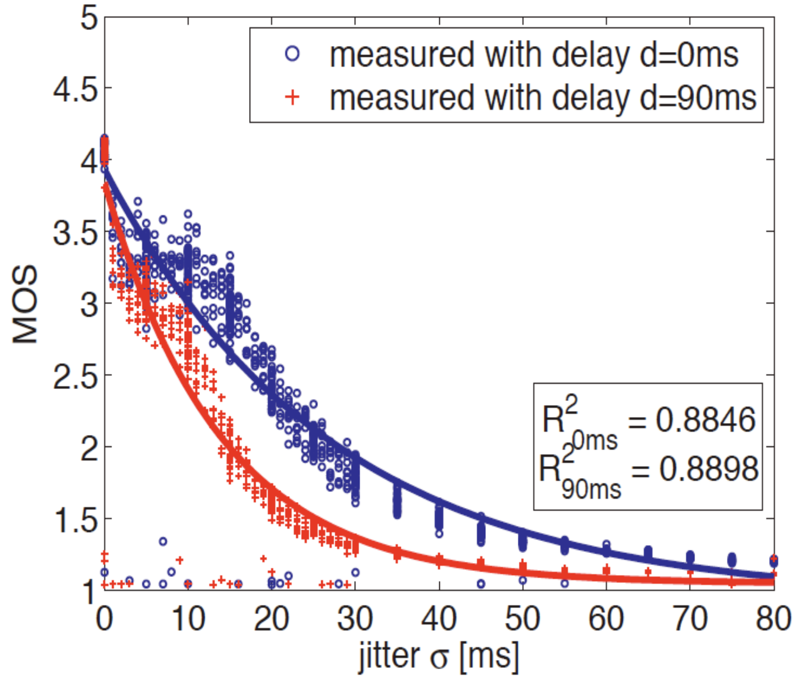


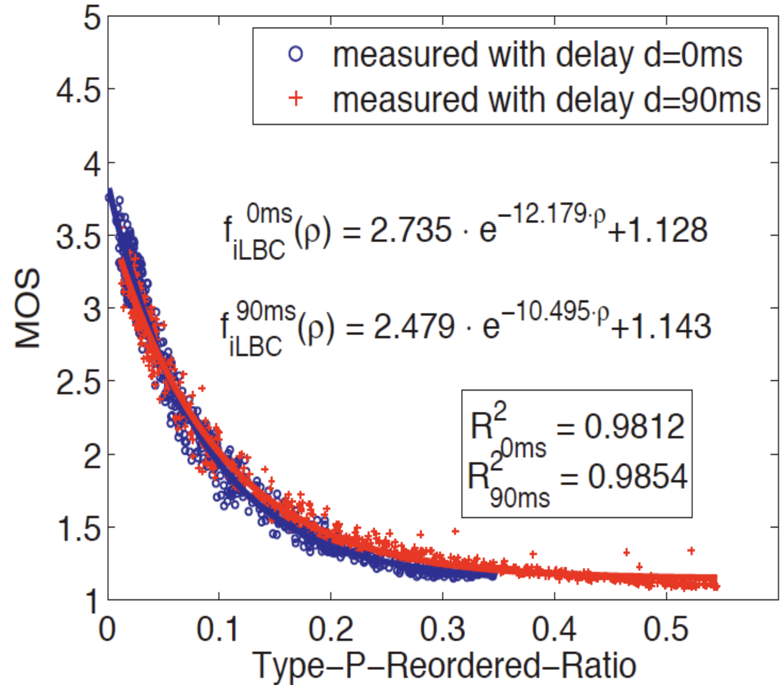
iLBC Codec



G.711 Codec

Test of the IQX Hypothesis: Results For Jitter I



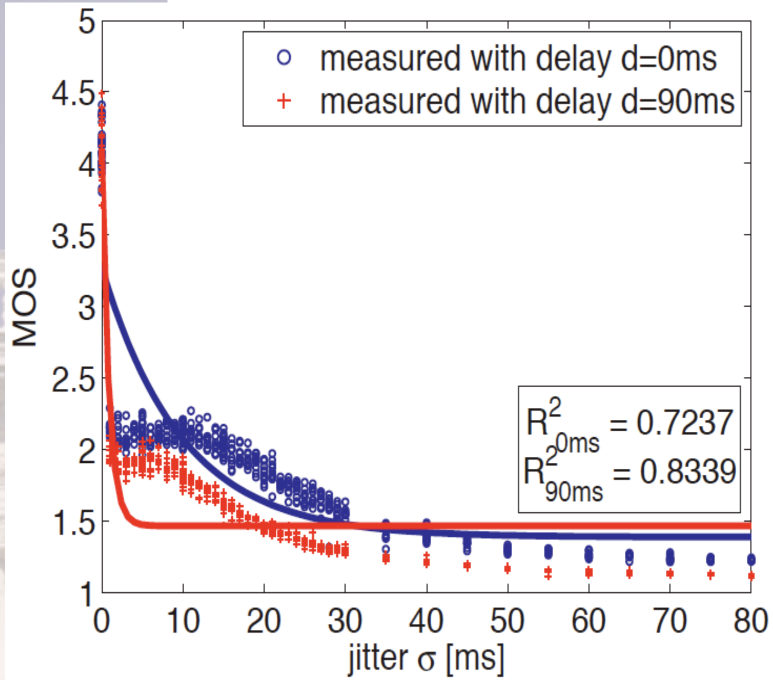


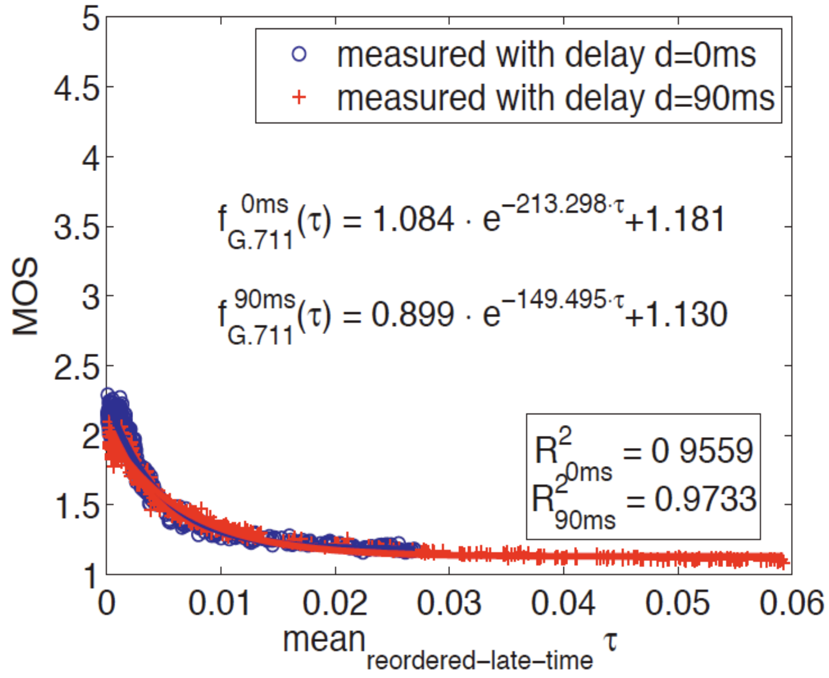
iLBC Codec

Chair of Future

endowed by A1

Test of the IQX Hypothesis: Results For Jitter II





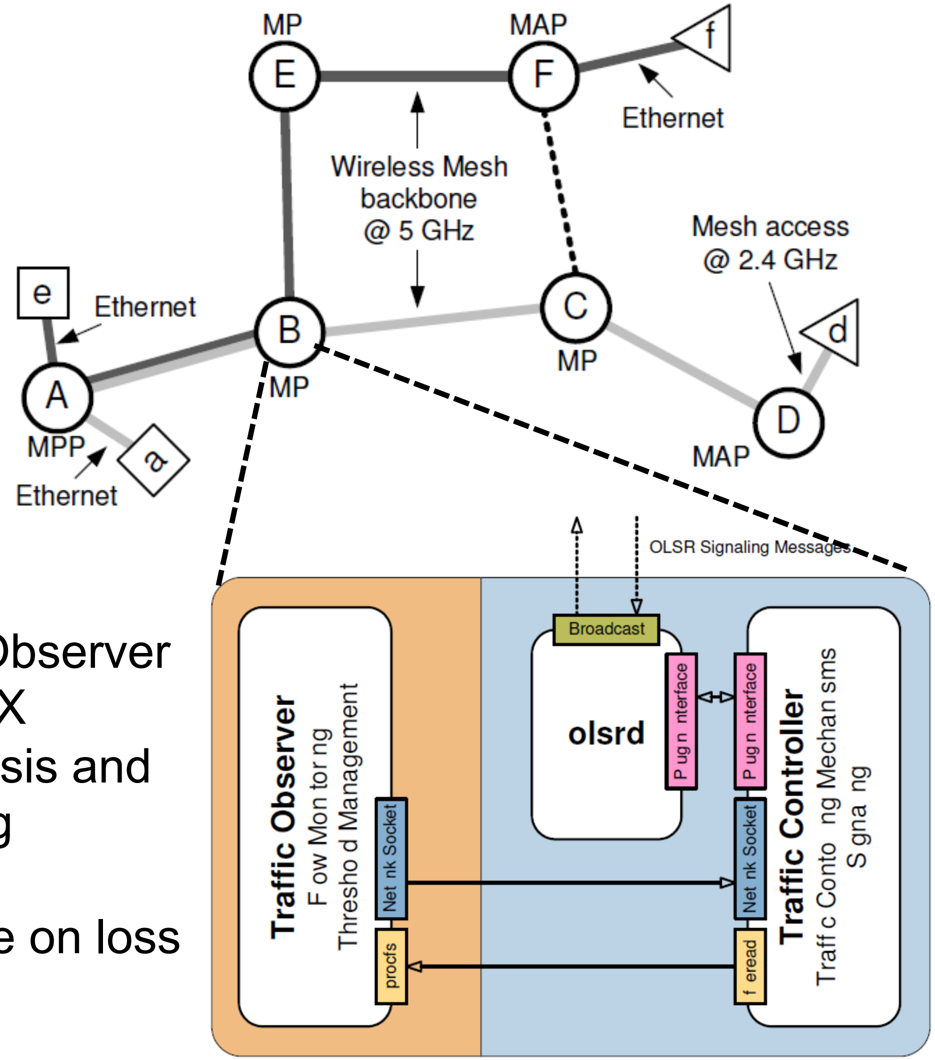
G.711 Codec

Chair of Future C

endowed by

Adaptive Traffic Management Based on QoE and the IQX Hypothesis

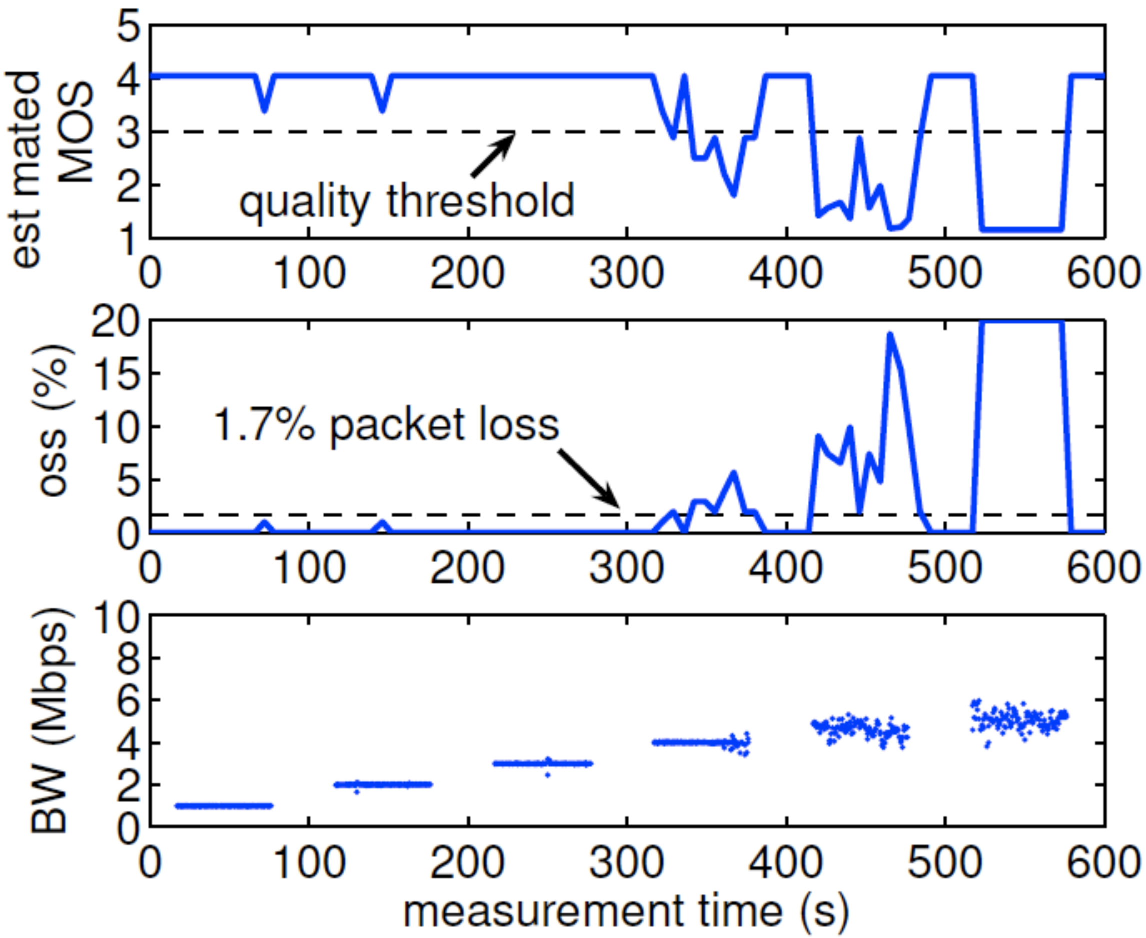
Dynamic Bandwidth Control in Wireless Mesh Networks
[Pries et al. ‘08]



Traffic Observer uses IQX hypothesis and mapping function sensitive on loss

Routing Controller

Adaptive Traffic Management Based on QoE and the IQX Hypothesis -- Results



Chair of
Future Communication

endowed by

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

Chapter 3: Multimedia Networking

universität wien

Chair of Future Communication

endowed by A1

Overview:

Real-Time Protocol (RTP)

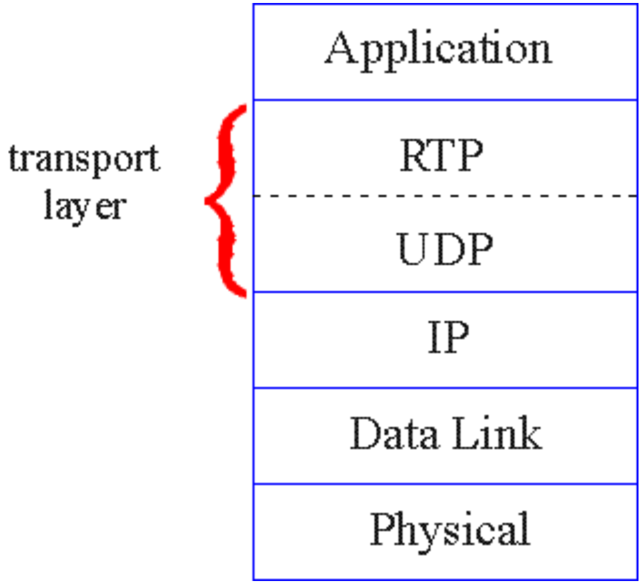
- RTP specifies a packet structure for packets carrying audio and video data
- RFC 1889.
- RTP packet provides
  - payload type identification
  - packet sequence numbering
  - timestamping
- RTP runs in the end systems.
- RTP packets are encapsulated in UDP segments
- Interoperability: If two Internet phone applications run RTP, then they may be able to work together

RTP and QoS

- RTP does not provide any mechanism to ensure timely delivery of data or provide other quality of service guarantees.
- RTP encapsulation is only seen at the end systems: it is not seen by intermediate routers.
  - Routers providing best-effort service do not make any special effort to ensure that RTP packets arrive at the destination in a timely matter.

RTP runs on top of UDP

- RTP libraries provide a transport-layer interface
- that extend UDP:
  - port numbers, IP addresses
  - payload type identification
  - packet sequence numbering
  - time-stamping



RTP Example

- Consider sending 64 kbps PCM-encoded voice over RTP.
- Application collects the encoded data in chunks, e.g., every 20 msec = 160 bytes in a chunk.
- The audio chunk along with the RTP header form the RTP packet, which is encapsulated into a UDP segment.
- RTP header indicates type of audio encoding in each packet
  - sender can change encoding during a conference.
- RTP header also contains sequence numbers and timestamps

RTP Header

RTP Header

- Payload Type (7 bits)
  - Indicates type of encoding currently being used.
  - If sender changes encoding in middle of conference, sender informs the receiver through this payload type field.
    - 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
- Sequence Number (16 bits)
  - Increments by one for each RTP packet sent
  - May be used to detect packet loss and to restore packet sequence.

RTP Header (2)

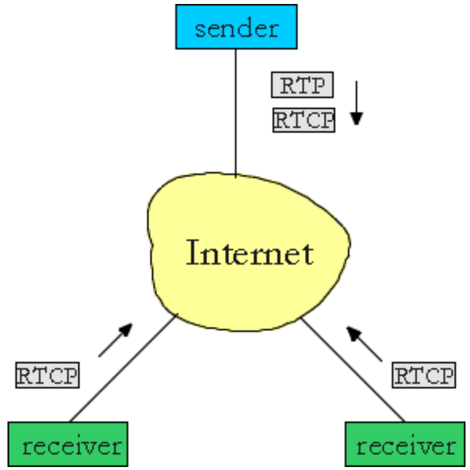
- Timestamp field (32 bits long)
  - Reflects the sampling instant of the first byte in the RTP data packet.
  - For audio, timestamp clock typically increments by one for each sampling period (for example, each 125 usecs for a 8 KHz sampling clock)
  - If application generates chunks of 160 encoded samples, then timestamp increases by 160 for each RTP packet when source is active. Timestamp clock continues to increase at constant rate when source is inactive.

- SSRC field (32 bits long)
  - Identifies the source of the RTP stream.
  - Each stream in a RTP session should have a distinct SSRC.

Real-Time Control Protocol (RTCP)

- Works in conjunction with RTP.
- Each participant in RTP session periodically transmits RTCP control packets to all other participants.
- RTCP packets contain sender and/or receiver reports
  - report statistics useful to application
- Statistics include
  - number of packets sent,
  - number of packets lost,
  - interarrival jitter, etc.
- Feedback can be used to control performance
  - Sender may modify its transmissions based on feedback

RTCP - Continued



- For an RTP session there is typically a single multicast address; all RTP and RTCP packets belonging to the session use the multicast address.
  - RTP and RTCP packets are distinguished from each other through the use of distinct port numbers.
  - To limit traffic, each participant reduces his RTCP traffic as the number of conference participants increases.

RTCP Packets

Receiver report packets:

- Fraction of lost packets,
- last sequence number,
- average interarrival jitter.

Sender report packets:

- SSRC of the RTP stream,
- the current time,
- the last time stamp
- the number of sent packets,
- and the number of sent bytes.

Source description packets:

- e-mail address of sender, sender's name, SSRC of associated RTP stream.
- Provide mapping between the SSRC and the user/host name

Synchronization of Streams

- RTCP can synchronize different media streams within a RTP session.
- Consider videoconferencing app for which each sender generates one RTP stream for video and one for audio.
- Timestamps in RTP packets tied to the video and audio sampling clocks
  - not tied to the wall-clock time
- Each RTCP sender-report packet contains (for the most recently generated packet in the associated RTP stream):
  - timestamp of the RTP packet
  - wall-clock time for when packet was created.
- Receivers can use this association to synchronize the playout of audio and video.

RTCP Bandwidth Scaling

- RTCP attempts to limit its traffic to 5% of the session bandwidth.

Example

- Suppose one sender, sending video at a rate of 2 Mbps. Then RTCP attempts to limit its traffic to 100 Kbit/s.
- RTCP gives 75% of this rate to the receivers; remaining 25% to the sender

- The 75 kbps is equally shared among receivers:
  - With R receivers, each receiver gets to send RTCP traffic at 75/R kbps.
- Sender gets to send RTCP traffic at 25 kbps.
- Participant determines RTCP packet transmission period by calculating avg RTCP packet size (across the entire session) and dividing by allocated rate.