Mechanisms for Value-Added IP Services

Georg Carle
Fraunhofer FOKUS / University of Tübingen
g.carle@ieee.org
http://www.fokus.gmd.de/usr/carle/

work in collaboration with
Tanja Zseby, Sebastian Zander, Henning Sanneck

Dagstuhl Seminar on Quality of Service in Networks
and Distributed Systems, October 2002

Overview

- Introduction
- AAA-based Quality of Service Control
- Utility/Prize-based Error Control
- Adaptive Streaming
  - Related Work
  - Implementation
  - Testbed Evaluation
- Conclusions
- Future Work

Introduction

- High quality IP based streaming is appealing
  - Video on Demand; Tele-Teaching; TV; Surveillance
- Streaming applications suffer from network congestion
  - Unreliable transport (UDP): Dropouts
  - Reliable transport (TCP): Hangs/Freezes
- Approach 1: “No problem. Just use guaranteed QoS”
  - L3 QoS (Intserv; Diffserv)
- However
  - guaranteed QoS will not come for free!
  - Hard reservations are expensive
  - Need to validate QoS (QoS measurements - also add costs)
- How to reduce cost for high quality streaming?

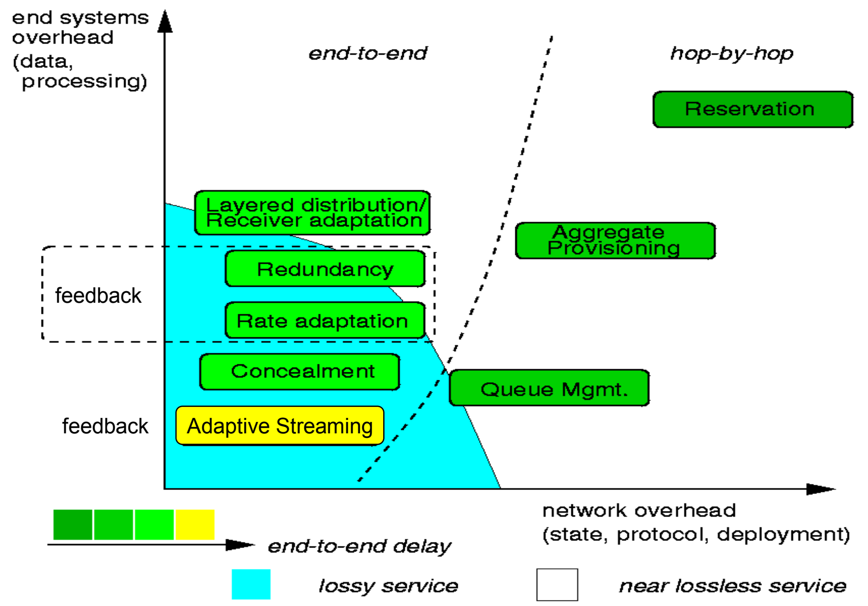
⇨ Utility-based Error Control

➡ Adaptive Streaming (low cost high quality streaming)

QoS enhanced Continuous Media Services

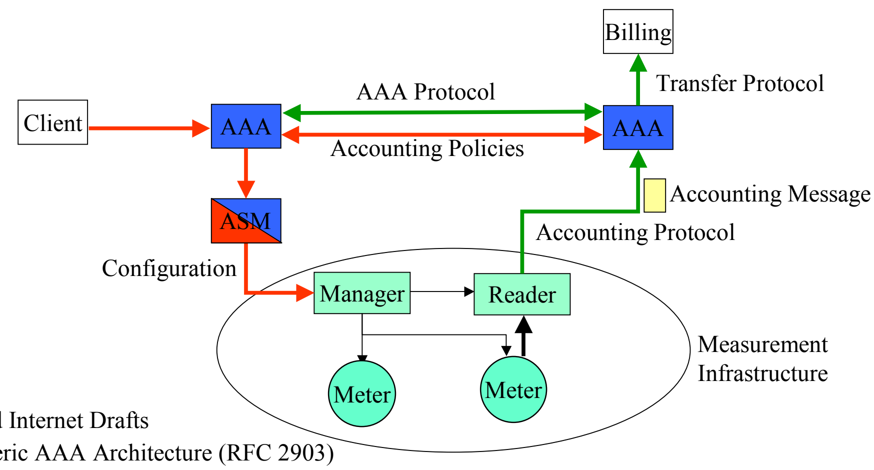
- Provider-viewpoint: AAA-based Quality of Service control
  - Policy-based QoS support
    - QoS provisioning (L3 - DiffServ)
    - QoS validation: QoS measurements
  - Transparent protection of streams on „lossy“ links
    - Error control middleboxes (L4 - FEC).
  - Flexible Unicast/Multicast splitting and merging of streams
  - Adapting streams for bandwidth/resolution requirements according to
    - client connectivity; end system configuration; personal preferences
- Application viewpoint:
  - E2E Error Control
  - Adaptive Streaming

QoS Mechanims:
Cost/quality tradeoffs



AAA-based Quality of Service Control
(Provider Viewpoint)

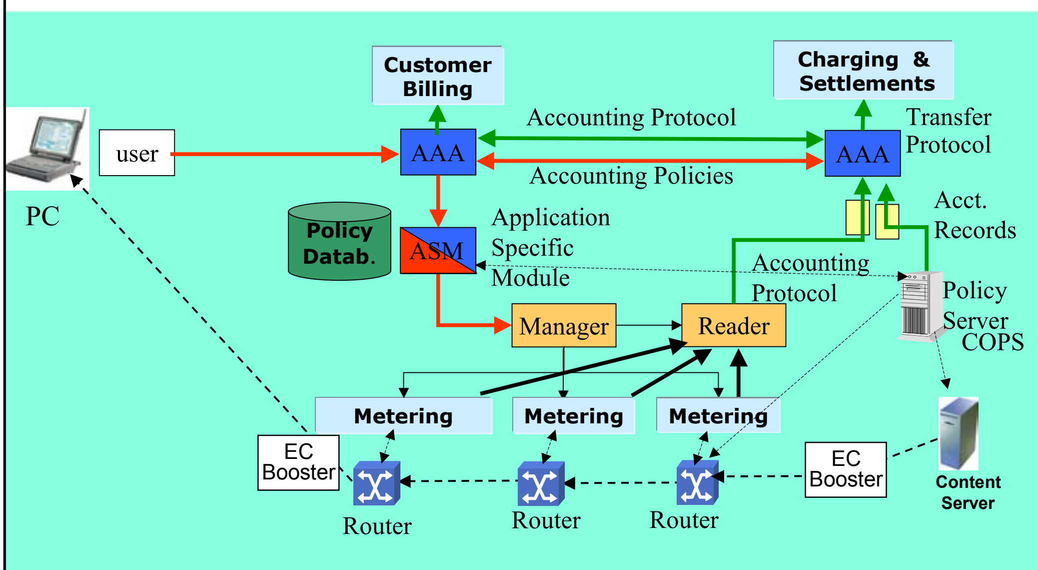
Standardisation of AAA Architecture



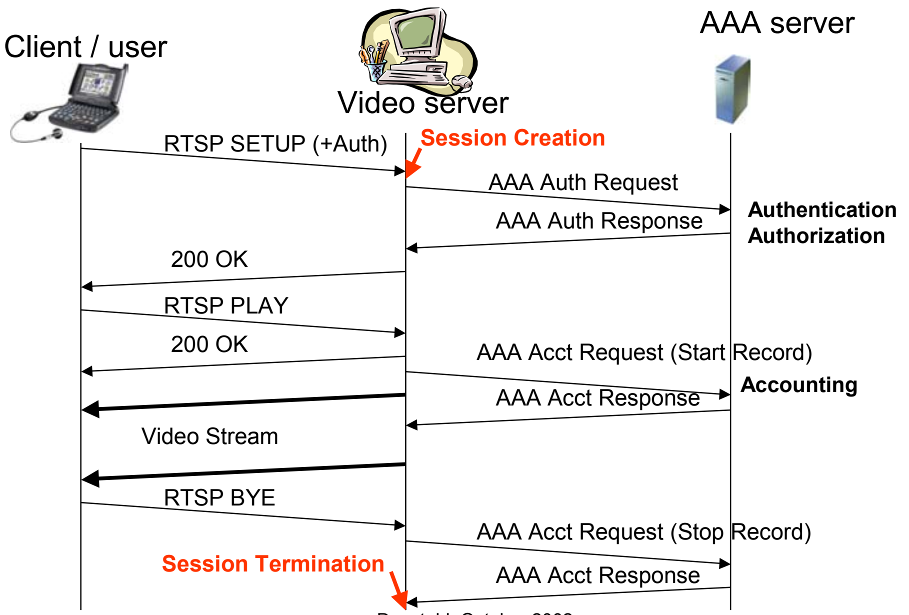
RFCs and Internet Drafts

- Generic AAA Architecture (RFC 2903)
- RTFM (RFC2720-2724)
- Acct. Management (draft-ietf-aaa-acct-06.txt)
- Acct. Attributes (draft-ietf-aaa-accounting-attributes-04.txt)
- Policy-based Accounting (RFC3334 by Zseby/Zander/Carle)

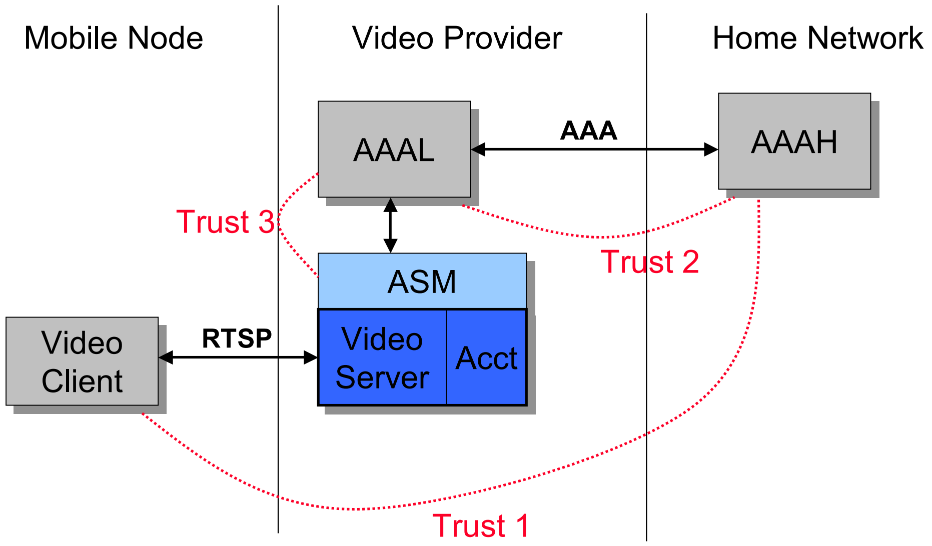
AAA-based Continuous Media Scenario



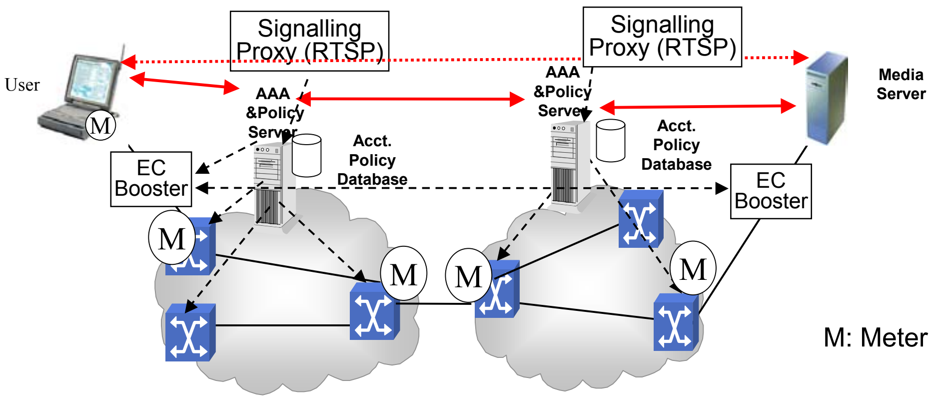
Video Server AAA Interaction



Interdomain Scenario

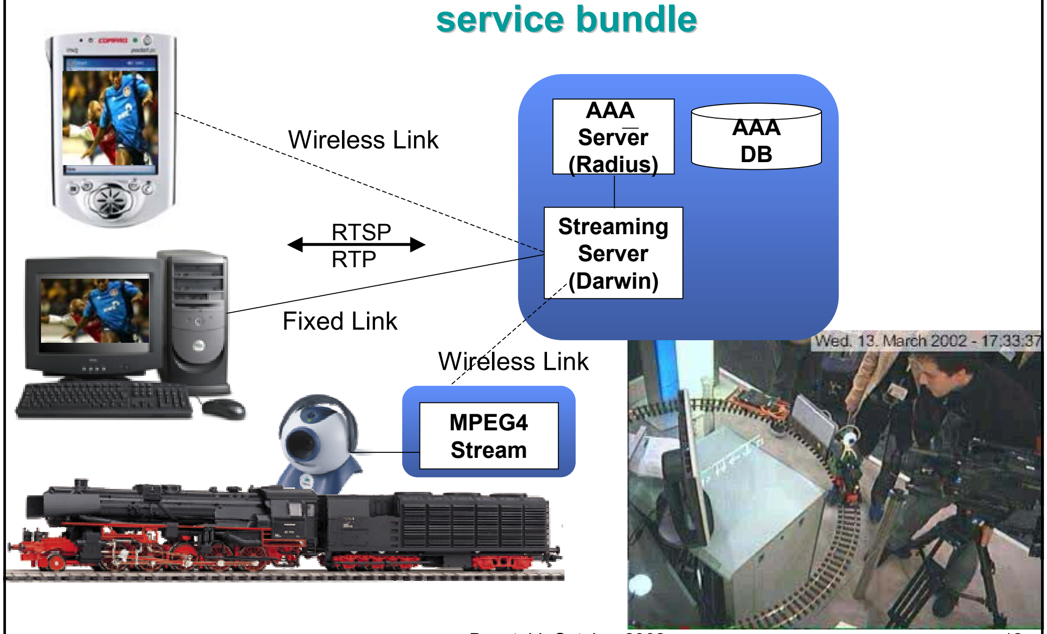


QoS-enhanced Continuous Media Streaming



- Policy-based configuration of QoS compontens (routers, booster)
- Policy-based configuration of measurement infrastructure

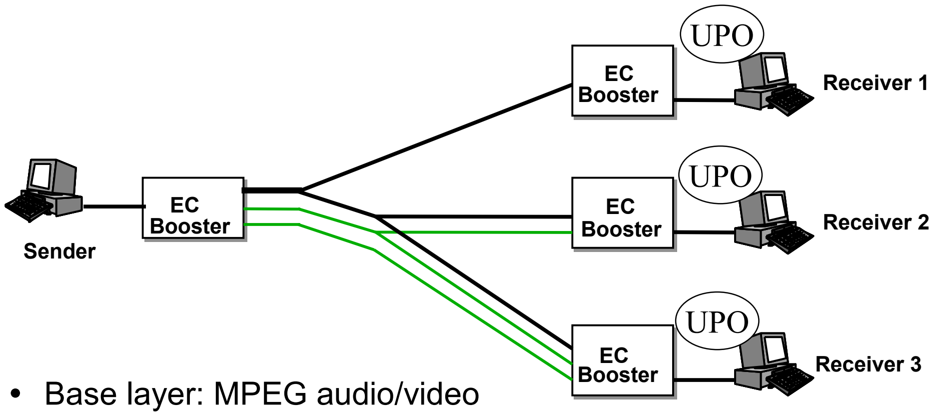
Media Streaming – an example of a service bundle



Media Streaming: User Perspective

Utility/Prize-based Error Control (Application Viewpoint)

Boosters with Tariff-Dependent Service Selection



- Base layer: MPEG audio/video
- Error Control (EC) for enhanced reliability: FEC + optional retransmission
- Utility-Price-Optimiser (UPO) for service selection

Enabler for Service Selection: TFL: Tariff Formula Language

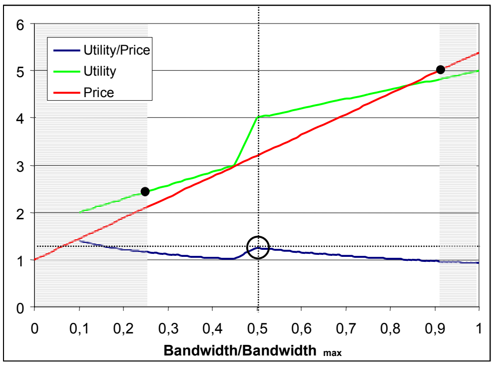
- Description of charging formulas and utility curves
- context-free language
- Mathematical operations (addition, multiplication, etc.)
- Mathematical functions (exponential function, square root, etc.)
- Logical functions (AND, OR, NOT)
- Conditional expressions (if/then/else)
- Pre-defined charging variables
- Example:

Token	Description
D	Date
TD	Time of Day
T	Time/Duration
V	Volume (Bytes)
VP	Volume (Packets)
TR	Token Rate
SR	Service Rate
BN	Normalized Bandwidth

```
# parameter a
a = IF(AND(td>=TIME("00:00:00"), td<TIME("05:00:00")), 0.5,
IF(AND(td>=TIME("05:00:00"), td<TIME("21:00:00"), 0.8, 0.5))
# parameter b
b = IF(AND(td>=TIME("00:00:00"), td<TIME("05:00:00")), 0.2,
IF(AND(td>=TIME("05:00:00"), td<TIME("21:00:00"), 0.4, 0.2))
# tariff formula
p = a*tr + b * (sr-tr)
```

Utility-Price-Optimizer: Example

- Utility and price only dependent on bandwidth
- Simple maximum search



$$UPR(B_n) = \frac{u(B_n)}{p(B_n)}$$

with:

$$p(B_n) = S + \frac{P_V}{U_V} \cdot B_n \cdot B_{\max}$$

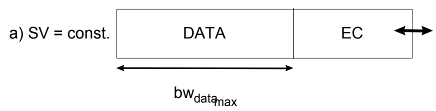
$$u(B_n) \geq u_{\min}$$

$$p(B_n) \leq p_{\max}$$

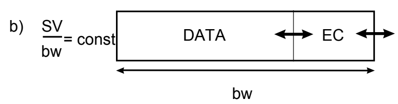
Error Control Optimization Policies

Goal: choose best distribution of error control (proactive/reactive redundancy) for different transmission rounds

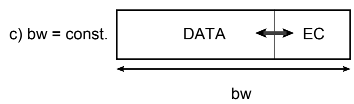
Optimization policies:



Smallest BW for const. SV (low network load)



Best ratio of SV vs. BW



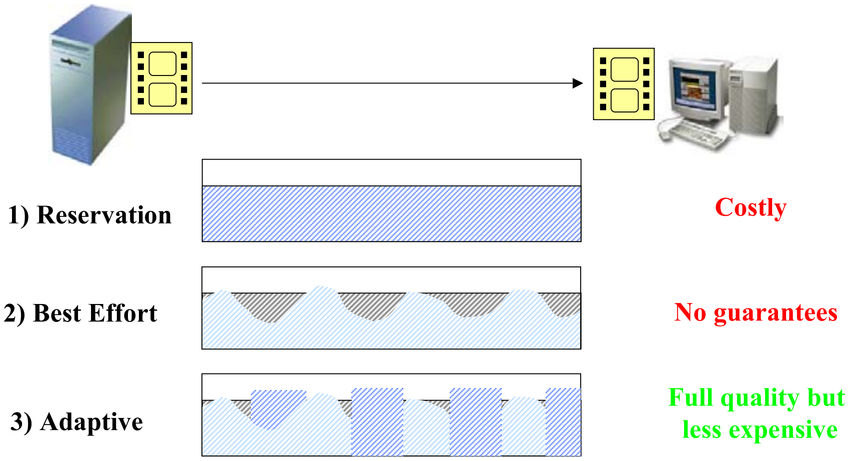
Best SV for given BW (high network load, or assured bandwidth)

Adaptive Streaming
(Application Viewpoint)

Adaptive Streaming - Goals

- Assumptions:
  - Best effort bandwidth will be cheap (flat rate)
  - Guaranteed bandwidth will be more expensive (charge per volume - sent/reserved)
  - ISPs or companies have long term SLAs covering different service classes (no dynamic re-negotiation)
  - Important business case: pre-recorded video (large end-to-end delay budget)
- Goals
  - Use the expensive guaranteed classes as little as possible
  - Utilize best effort class as good as possible
  - Provide high quality streaming (DVD-like quality)
  - Provide 100% quality (no dropouts, no freezes)

Adaptive Streaming - Approach

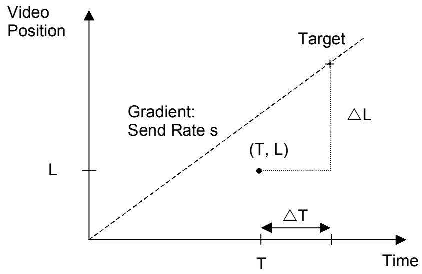


Adaptive Streaming - Approach

- Transport the stream reliably and TCP friendly using Best Effort service (BE)
- Use bandwidth from a Guaranteed service (G) if BE bandwidth is insufficient (smaller than video bandwidth)
- Dynamically adjust G bandwidth according to available BE bandwidth, so that there are no buffer underuns or overflows at the receiver
- If G bandwidth is insufficient sender rate (and quality) must be decreased
- The algorithm consists of two parts
  – Sender Rate Control
  – Adaptive Marking

Adaptive Streaming - Sender Rate Control

- The sender adjust its rate in intervals $\triangle T$ according to the rate required by the video
- The server always knows its position in the video stream
- A receiver playout buffer compensates network bandwidth fluctuations
- The receiver periodically informs the server on playout buffer fill status
- The server adjusts the send rate according to the desired video position and receiver buffer fill status



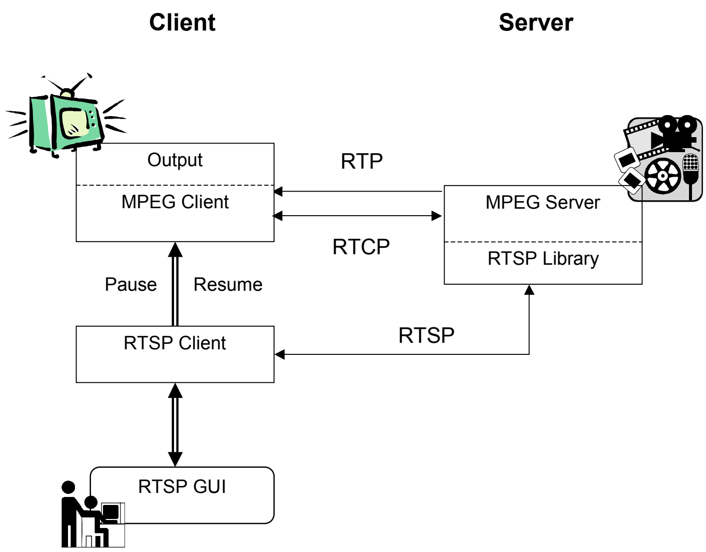
Adaptive Streaming- Adaptive Marking

- Two service classes: guaranteed (G), best effort (BE)
- A packet is marked with probability *p* as G class and *1-p* as BE class
- *p* depends on ratio of the real and optimal sending rate (*p* increases for decreasing receiver buffer level)
- A video frame based marking scheme would lead to better performance if G bandwidth is insufficient
- If there is sufficient G bandwidth the probabilistic scheme is simpler to implement and produces the same result (100% quality)

Implementation - Features

- MPEG-2 over RTP (transport, program)
- RTCP fast feedback
- New RTCP Application Feedback Messages
  - Buffer fill, Throughput measured for classes, …
- Basic RTSP implementation
  - play, pause, stop
- Text based interface to RTSP engine enables GUI flexibility
  - Java GUI, Web GUI, Shell Scripts

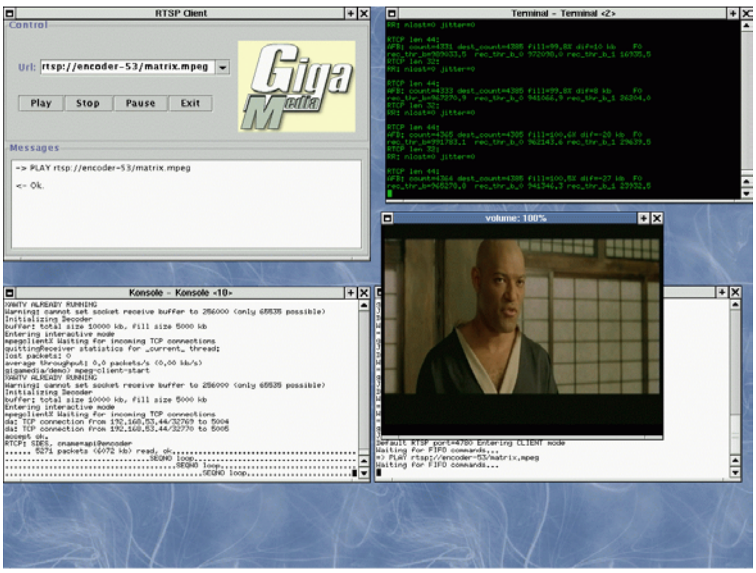
Implementation - Overview



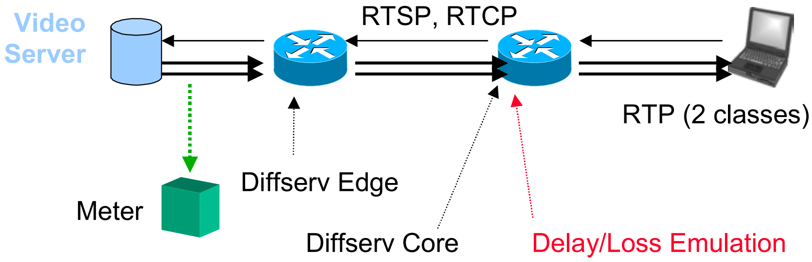
Implementation – Adaptive Streaming

- Client sends extended RTCP feedback messages to the server
- Server assigns each RTP packet to one class (via the RTP m-bit)
- Kernel classifier implemented which classifies packets based on RTP m-bit
  (no control I/F needed)
- Linux Diffserv is used for marking and scheduling
- RTP over TCP provides reliability and TCP-friendliness

Implementation - Screenshot

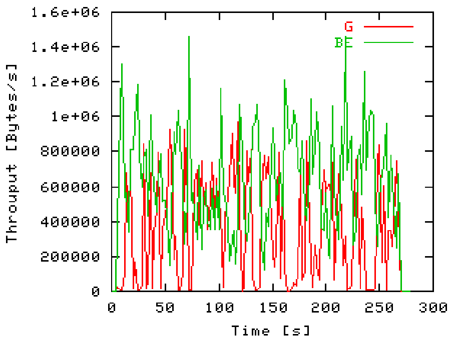


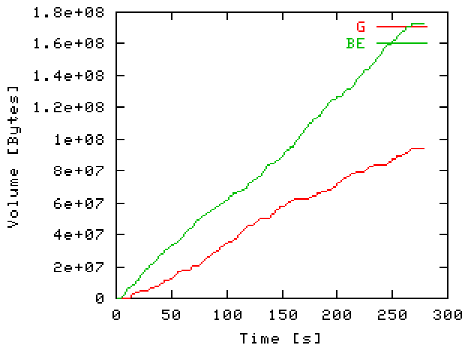
Evaluation - Testbed



- MPEG-2 video stream: 8 Mbit/s constant bit rate (DVD)
- Routers run Linux 2.4 (DiffServ enabled)
- Edge router marks with a DSCP according to the RTP m-bit
- 2 Classes: Expedited Forwarding (EF) and a best effort (BE)
- Congestion emulated by dropping BE packets

Evaluation - Algorithm Behaviour

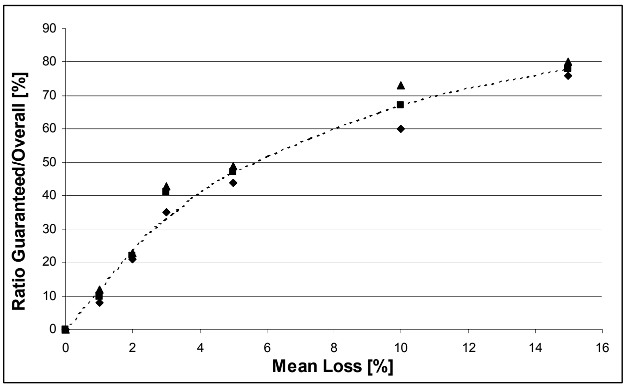




- Mean packet loss: 3%, maximum packet loss: 6%
- 8 MByte receiver buffer

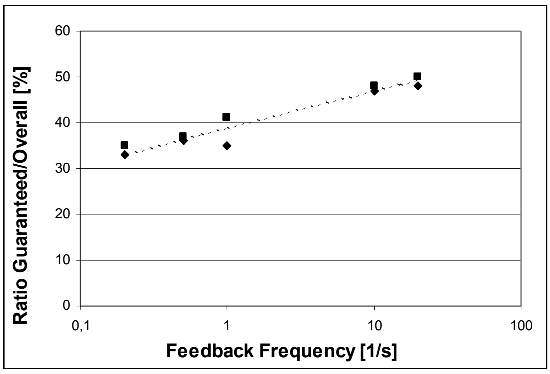
➡ 64% of the video sent over BE (only 36% over G)
No application layer losses (100% quality)

Evaluation – Loss Rate Impact



For small loss rates the gain is quite high while for large loss rates (>10%) the ratio is still over 70% but moving to 100%.

Evaluation – Feedback Frequency Impact



Algorithm looses performance in case of too frequent feedback. The smallest usable frequency is the playout time of half of the receiver buffer size.

Conclusions - Adaptive Streaming

- Lightweight flexible experimental MPEG-2 Video Server & Client
  - RTSP, RTP/RTCP over UDP/TCP
  - MPEG-2 transport/program payload
- Adaptive Streaming Algorithm & Proof of Concept Implementation
- Evaluation shows that for mean loss rates up to 10% a substantial amount of bandwidth can be obtained from a best effort service and thus saving guaranteed bandwidth

Future Work - Adaptive Streaming

- Improve the algorithm
- Test the behaviour with real TCP background traffic
- Integrate marking schemes which are based on the video frames to improve performance when overall bandwidth is insufficient
- Open loop solution (without receiver feedback)
- Smooth the usage of the guaranteed bandwidth
- Derive rules for dimensioning receiver buffer size, feedback interval
- Investigate how much guaranteed bandwidth is needed to satisfy a certain number of clients to be able to create admission control rules
- Combinations with application-level (MPEG4/H.26L) retransmissions and/or FEC