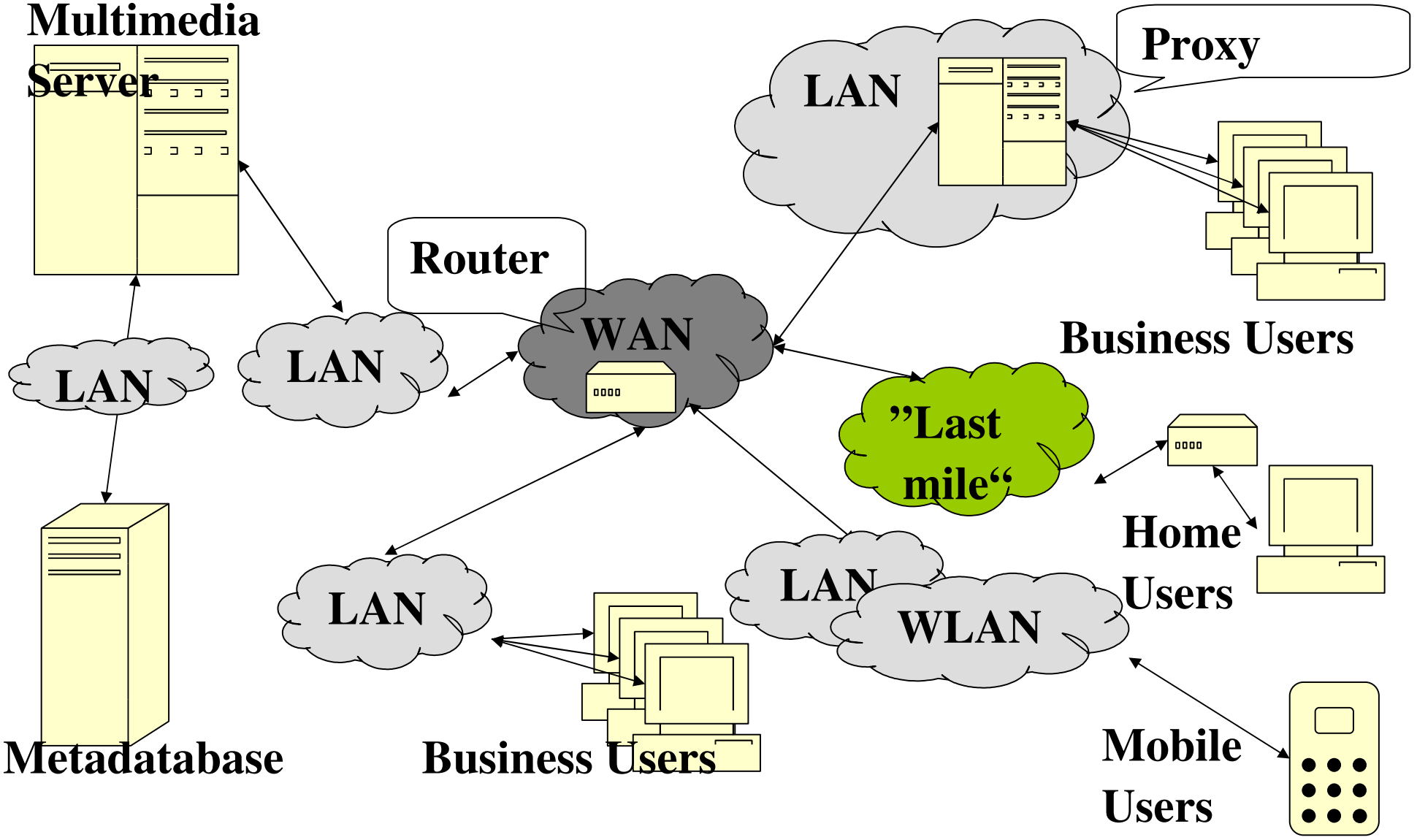
Combining Stream Switching with Fine-grained Intra-stream Adaption for Adaptive Video Streaming

Michael Kropfberger and Hermann Hellwagner
University of Klagenfurt, Austria

VASE - A Video-Adaptive Streaming Environment

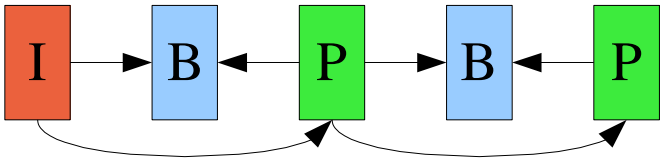


Usage Environment

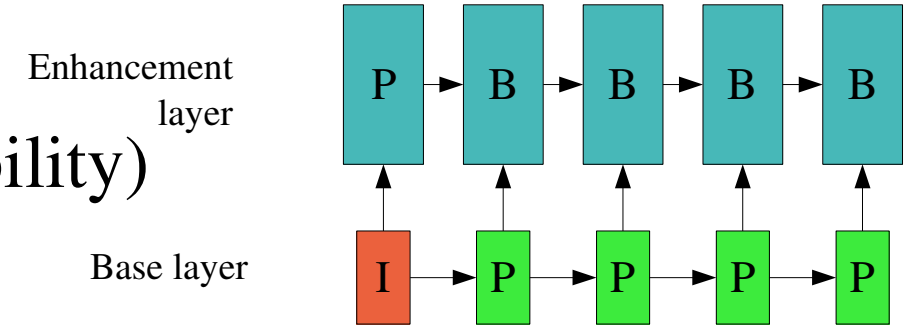
- Unicast Video on Demand (VoD)
- Video+Audio streaming via Cable or xDSL
  - Downstream: max. ~1 Mbps
  - Upstream: ~128 kbps
- Server is somewhere in the Internet
  - Best effort network with multiple users
  - Changing bandwidth
  - Changing latencies (20 ms - 200 ms)

Scalable Codecs

- Temporal



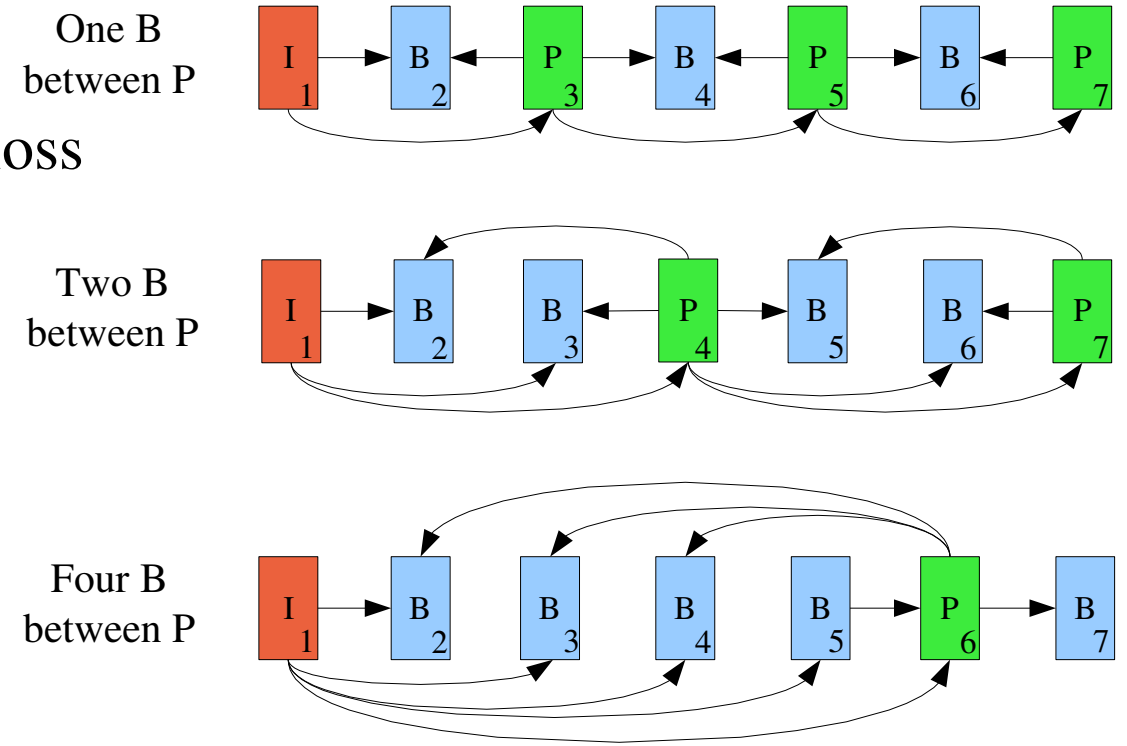
- Signal-to-Noise (SNR)
  - FGS (Fine Granular Scalability)



- Spatial
  - use with re-scaling: special case of SNR
  - different quality up/down-scaling algorithms
- Big goal: The universal scalable video codec

Temporal Scalability

- Frame Patterns
  - Variable patterns
    - IPPPPBPPPBPPBIPIIIIPPBBP....
  - Static patterns
    - I-Frame: sync, loss

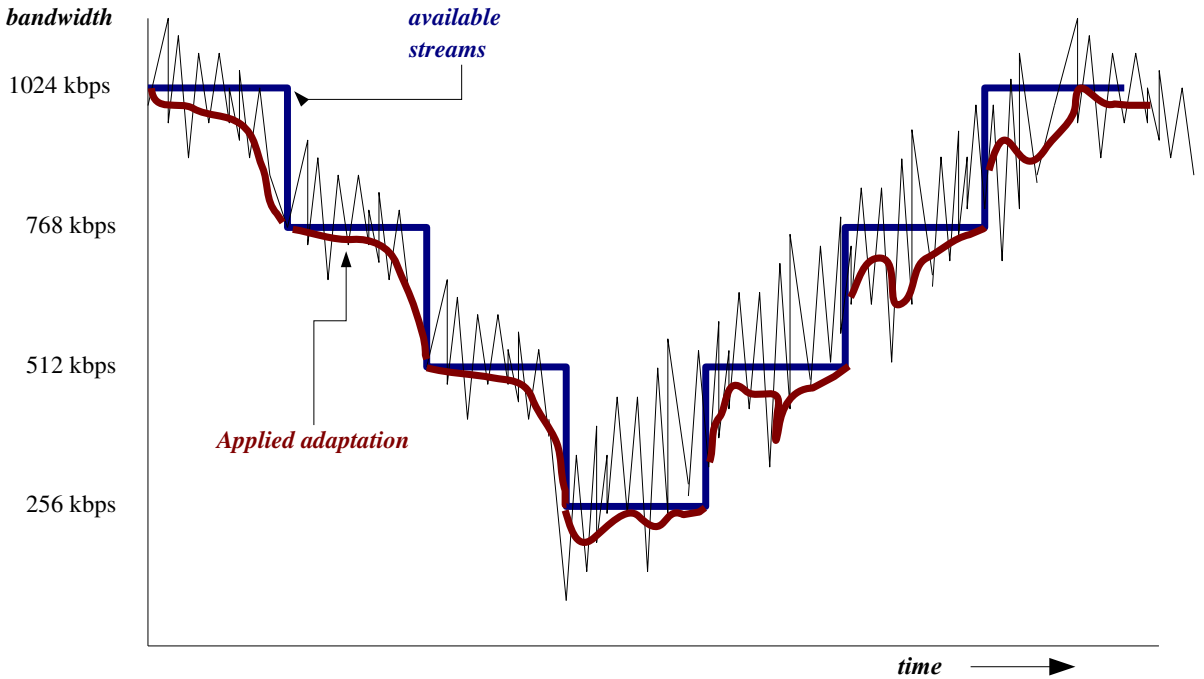


High Number of B-frames

- Pro's
  - Increased scalability (but low fps not wanted?!?)
  - BUT: Larger pool for choice of (intelligent) dropping
- Con's
  - Increased file size (reference distance)
  - Increased encoding/decoding time
  - Increased buffer requirements

Stream Switching

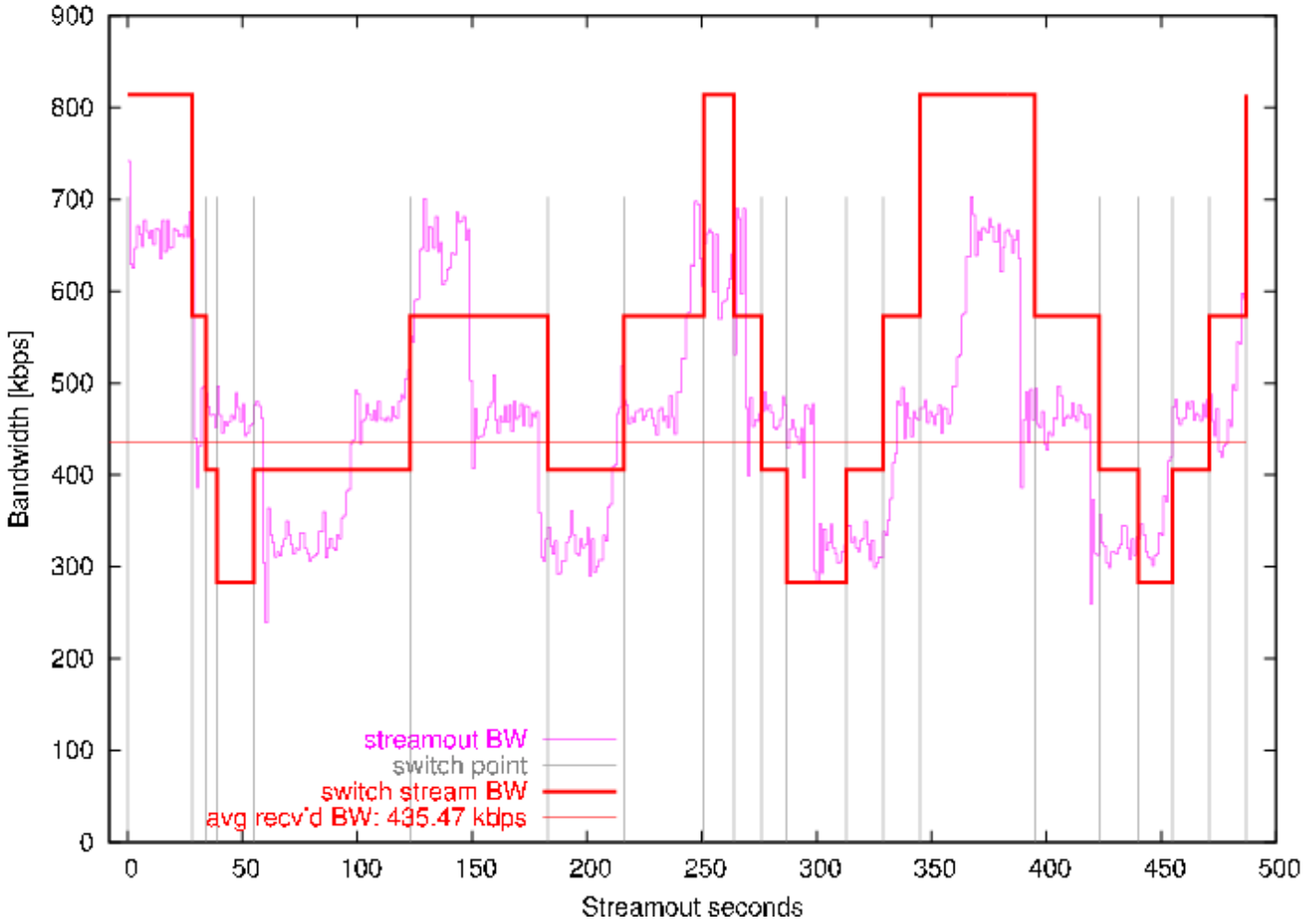
- In-stream adaptation:
  - Scalable codecs are bound to certain ranges
    - dont drop more than 50% of frames!
    - high coding overhead for FGS with small base layer
- Client buffer too high or low
- Switch streams and start adapting again!



Stream Switching and Adaptation

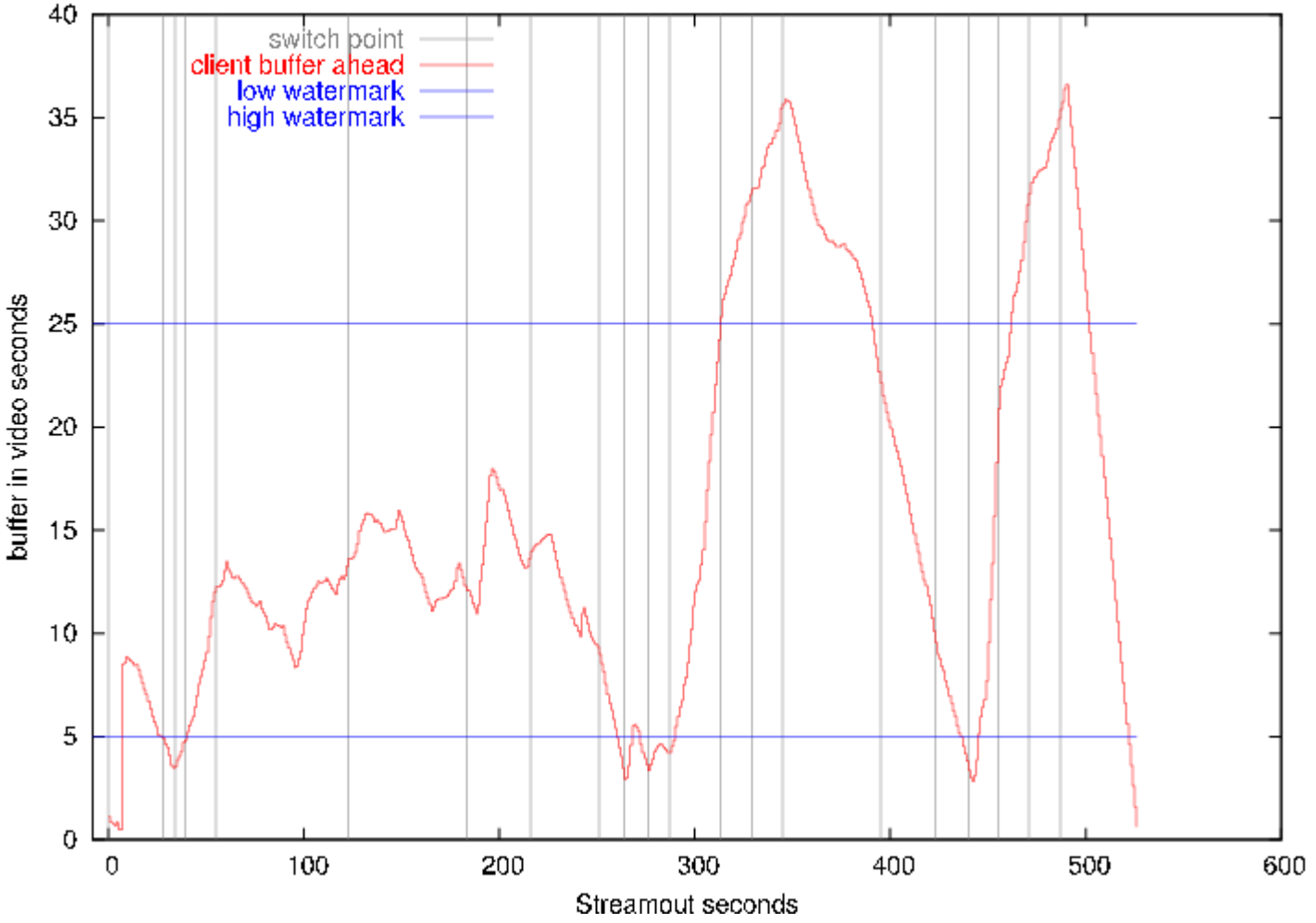
- Multiple streams each -30% BW => -1.5dB PSNR loss
- In-stream Temporal Adaptation
  - ***Helps to overcome buffer underruns or switch-downs***
  - High B-frame dropping leads to up to -6 dB PSNR loss!
  - Less than 15 fps (from 25fps) not acceptable
- More fine grained in-stream SNR adaptation needed!
  - FGS?
  - Wavelet-based SVC?

Stream Switching w/o Temp Adapt



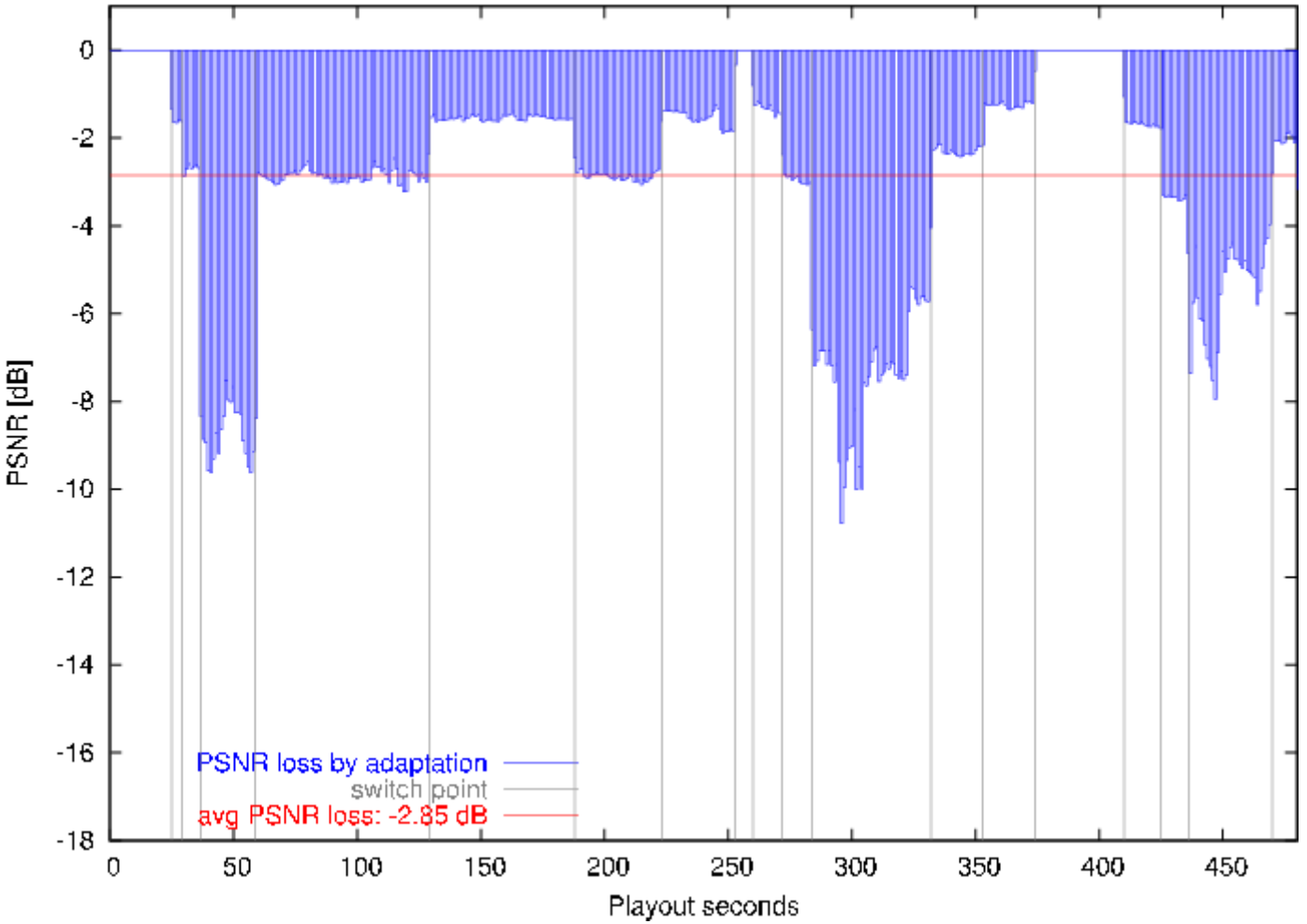
Network bandwidth

Stream Switching w/o Temp Adapt



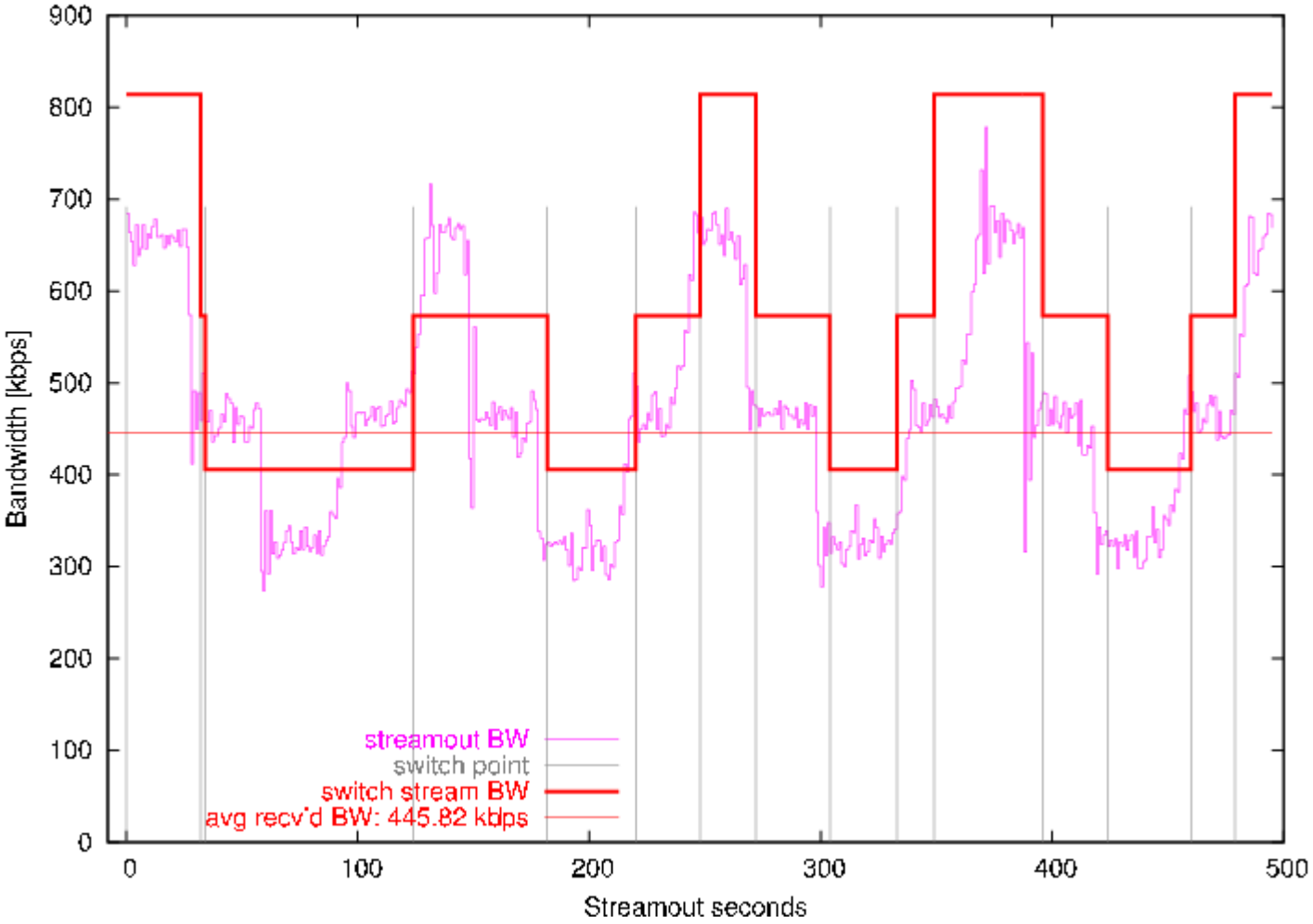
Client buffer fill level of available playout seconds

Stream Switching w/o Temp Adapt



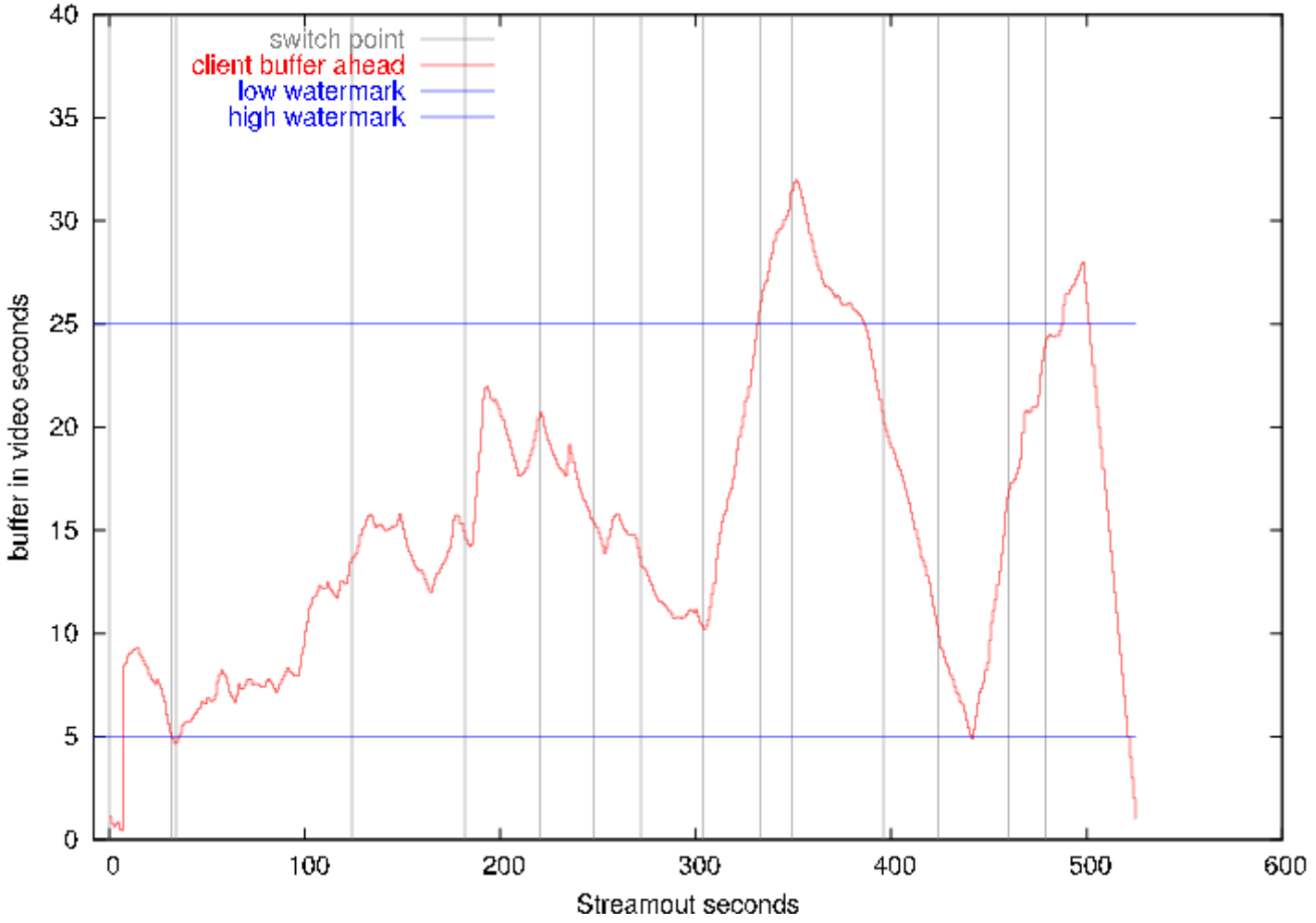
Qualitative reduction caused by adaptation

Stream Switching with Temp Adapt



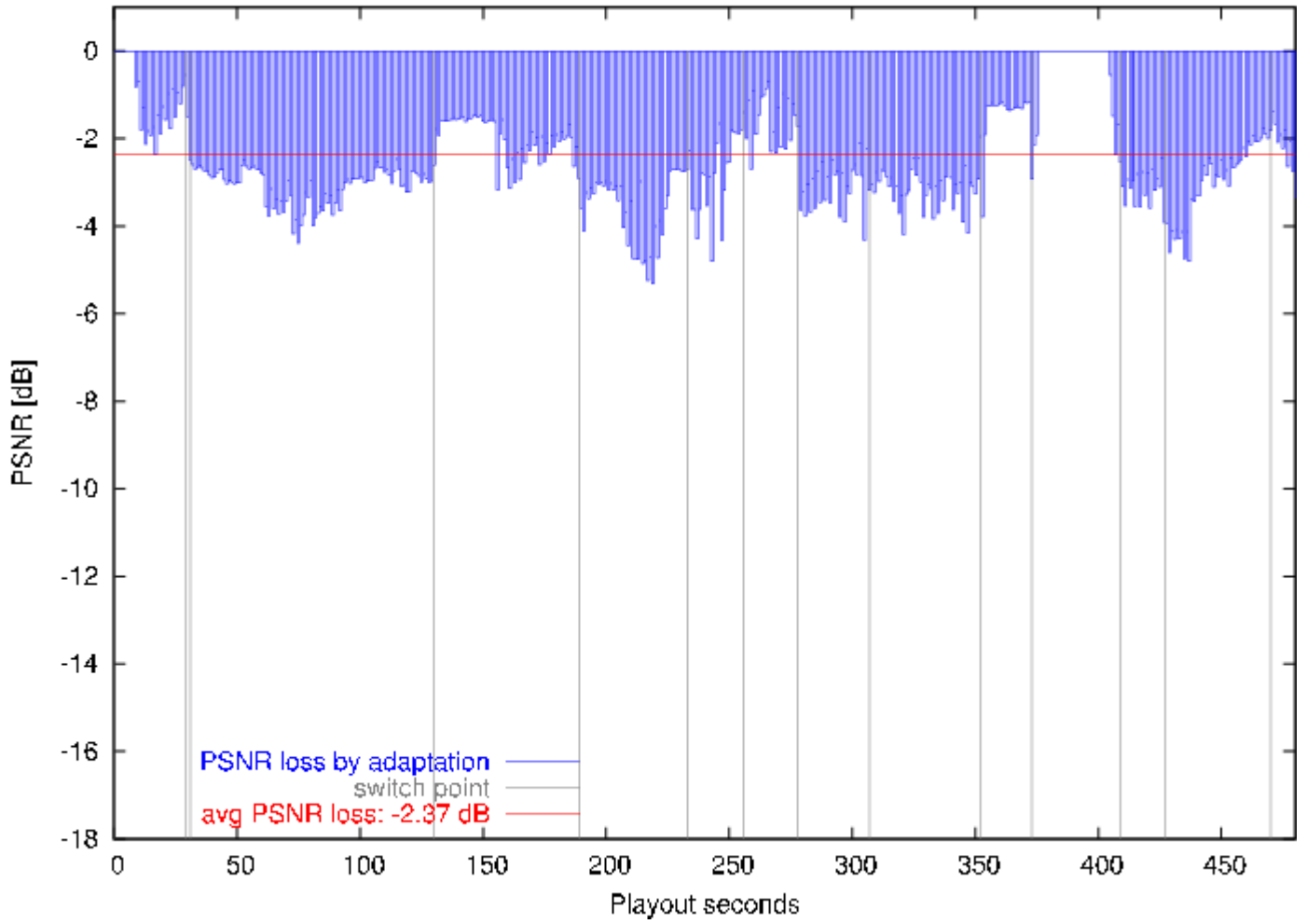
Network bandwidth

Stream Switching with Temp Adapt



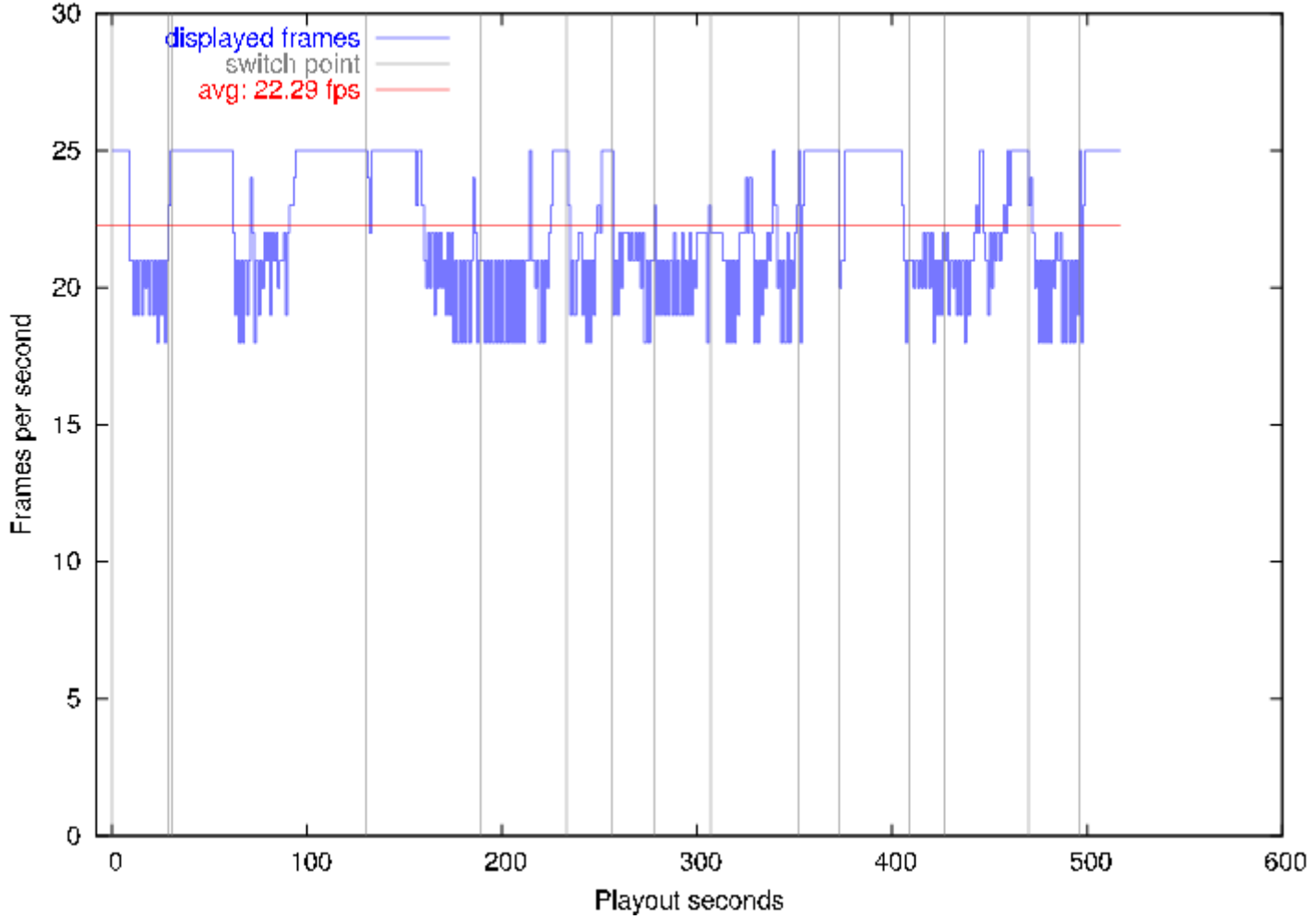
Client buffer fill level of available playout seconds

Stream Switching with Temp Adapt



Qualitative reduction caused by adaptation

Stream Switching with Temp Adapt



Visual frame rate per playout second after temporal adaptation

Conclusion

- In-stream adaptation:
  - Scalable video codecs (Temporal, SNR, Spatial)
  - Specialization&Analysis on temporal (B-frame) adaptation
- Stream switching (+ in-stream adaptation)
- Combination of both
  - stabelizes client buffer
  - prevents major quality loss by unneccessary switching
  - better user satisfaction
- even better results with more fine-grained adaptation (FGS, wavelet,...)

ViTooKi - The Video ToolKit in C++

- Support for MPEG-1/2/4 video codecs (ffmpeg+xvid)
- runs on Win,Linux,Linux-iPAQ, winCE-iPAQ
- MPEG-21 terminal capabilities + live transcoding
- Immediate and fast adaptation and stream switching
- Smoothed streamout for client buffers
- Flow control (TCP-friendliness)
- Transport via RTP
  - extensions: immediate RTCP and NACK+retransmission
- --> avail as open source via CVS (44 kLOC, 1.6MB)
  - The Video ToolKit @ ITEC http://vitooki.sourceforge.net