universität wien

Chair of Future Communication
Prof. Dr. K. Tutschku

Modeling and Evaluation of Video Streaming

Florian Metzger, Albert Rafetseder

HTTP Streaming Primer

Differences of RTP and HTTP streaming

RTP streaming
- Classical "textbook" approach, well researched behavior
- UDP and loss tolerant encoding enables implicit quality adaptation to current network conditions
- UDP transport prevents usage in the WWW
- Server-side "push-based" control scheme

HTTP streaming
- Reality today, in wide use (up to 50% of peak traffic)
- Multitude of different protocol incarnations, defined only by server and player behavior
- Client-centric "pull-based" control scheme
- Use of TCP means reliabilty, e.g., no intrinsic frame dropping

Measuring HTTP streaming

- RTP measuring metrics not applicable
- Network layers could influence new streaming approaches differently
- Hard to compare specific implementations
- Find generic mechanisms common to all
  → Buffering and playback strategies
  - Incorporate every possible behavior into a single testbed emulation
- Find suitable comparison metrics
  - Stalling duration
  - Number of stalls
  - Inter-arrival time of stalls
- Derive user quality from basic metrics

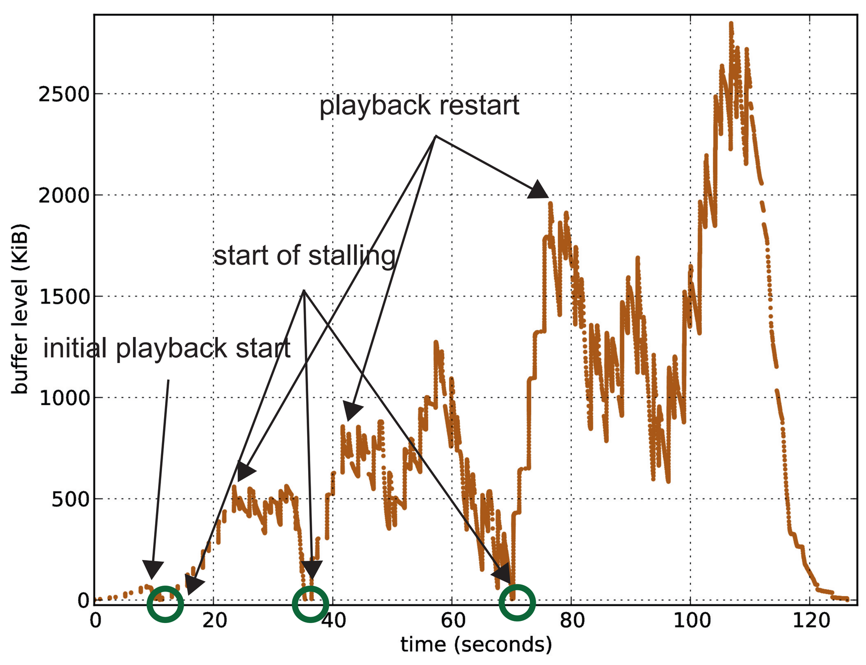
Necessity of buffering

- Video decoder would need only milliseconds of data at once
  - Network jitter and VBR cause variations in the received data rate
- Playback stalls when buffer runs out of data → large buffer!
- Playback should start as soon as possible → small buffer!
- Model playback as a simple buffer fill level equation:

$$buffer(t) = \sum_{0}^{t} data_{\text{received}} - \sum_{0}^{t} data_{\text{played}}$$

- Initial playback start time and restart time after empty buffer
  → Governing factors in any non-skipping streaming playback strategy

Playback Models & Strategies



YouTube Flash player strategy

- Start playing when buffer contains more than two seconds of video data
- If stalled, buffer at least five seconds video data before restarting
- Compromise between small waiting times and number of stalls

Theoretical Strategies

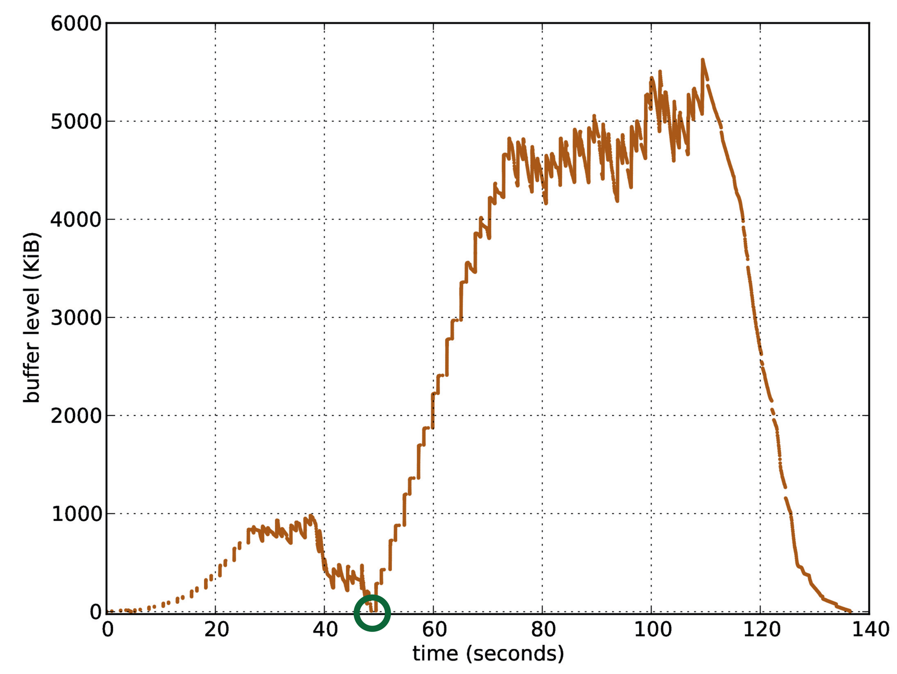
Demonstrate the range of possible stalling trade-offs, but are impractical in real situations.

Minimal buffering / Playback stalling

- Start immediately, stall immediately
- Starts playback as soon as there is at least one complete frame in the buffer
- Shortest initial playback delay
- Optimal in controlled situations with sufficient transmission rates at any time

Playback without interruption / Initial playback delay

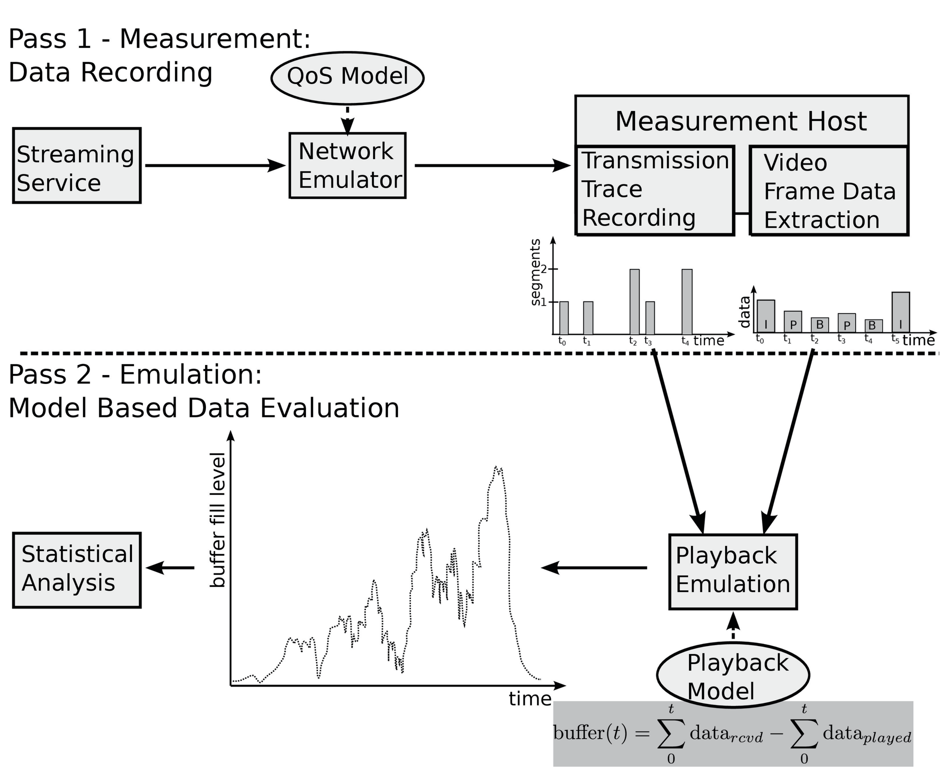
- Download exactly as much as needed to play back without any stalls
- Lower limit of total stalling time and number of stalls
- Impossible to implement − requires perfect knowledge
- Could be approximated by guessing transmission and bitrate



HTML5 video strategy in Firefox 4

- Factor in moving averages of transmission and playback bitrates
- If $MA_{transmission} > MA_{bitrate}$ then wait until 20s of video is in the buffer or for 20s in total, else 30s
- Limits stalling to few but long events, requires large buffer

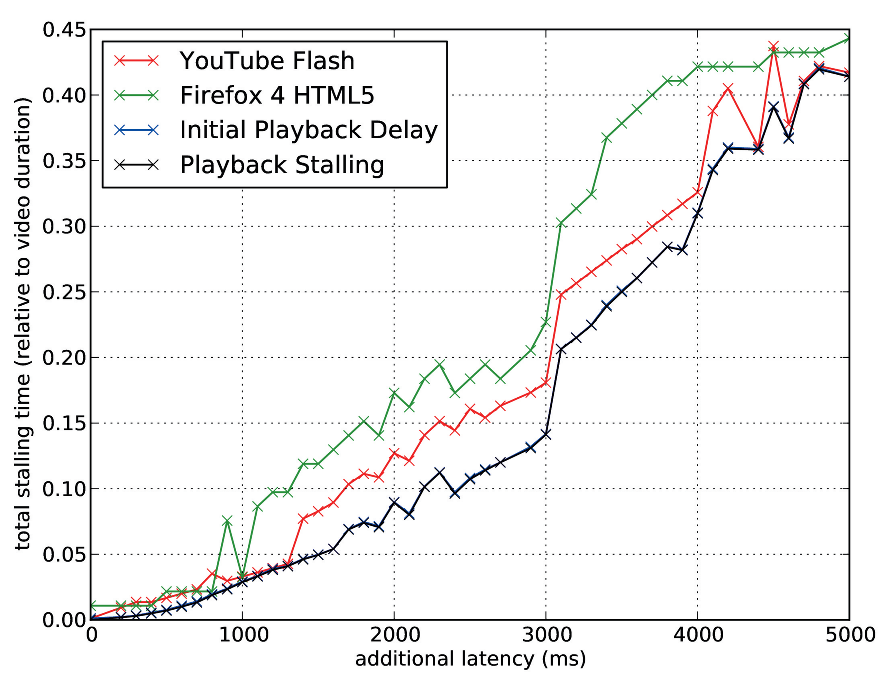
Measurement Framework



Measurement pass
- Request content from streaming server through QoS model in network emulator
- Record network and decoded video playback trace

Emulation pass
- Use trace files to do buffer fill level calculations according to playback startegies
- Can apply multiple emulated strategies to the same network trace
- Evaluate stalling statistics: stalling duration and frequency as potential input for QoE estimations

Evaluations



Impact of latency/loss on stalling characteristics

- Frequency: YT/FF suffer on average from one additional stall at a latency larger than 1000ms
- Lower total stalling limit given by theoretical strategies
- Impact of packet loss greater than 1% more noticeable, possibly due to TCP timeouts and retransmissions
- Practical implementations must make trade-offs between frequency and durations

Outlook

Adaptive Streaming Emulation

- Currently used strategies may not be the optimal choice for high latency networks (e.g. mobile networks)
- Adaptive streaming protocols as a possible solution
- Already many variations available, e.g. DASH, Smooth Streaming, HTTP Live Streaming
- Has a larger parameter space to observe and model
  - Segment retrieval times (i.e. client side throttling)
  - Quality adaptation through alternate segment encodings
- Transfer emulation testbed approach to adaptive mechanisms
- Explore strategies, trade-offs, and evaluation metrics feasible

Mobile Core Network Dataset

- Investigation of a one week mobile operator core network dataset
- Includes user traffic flow data, HTTP specifics and GTP signaling traffic between SGSN and GGSN
- Attempt to correlate mobile device types to GTP signaling patterns
- Determine PDP context life cycle and overhead
- Has streaming traffic a noticeable impact on the core network? How can it be modeled?