Video

Lecture 24, Computer Networks (198:552)
Videos over the Internet

- Downloaded full-length videos
- Streaming videos
- Real-time conference
- Video gaming
Internet video growth

Source:
Internet trends report 2018, Mary Meekers
Internet video growth

Source: Internet trends report 2018, Mary Meekers
How should net support video streaming?

• **Improved underlying transport protocols**
  • Better UDP? Do we need all of TCP’s heavy hammers?

• **Co-designing transports with applications**
  • Better app design based on transport info? App-directed network probing? Rewrite data written to a socket?

• **Coordinating clients based on network information**
  • Example: same ISP ➔ similar performance
Streaming video basics

• **Average bit-rate**
  - *Average* bits per second of the video
    - Faster “movement” in video ➔ more bits needed
    - Correlates with perceived video quality
    - Discrete bit-rates

• **Video buffer**
  - Client-local buffer
  - Measured in *seconds*
  - If buffer runs out (ex: if no data received), *rebuffering* results
Streaming video basics

- **Video chunks**
  - Fixed time segments of the video -- granularity of control

- Clients can request chunks of a given (avg) bit-rate
  - **Control loop** involves the bit-rate and buffer size
Streaming video QoE

• Quality (ex: average bit-rate)

• Rebuffering

• Smoothness

• Join time

Understanding the Impact of Video Quality on User Engagement, Dobrian et al., 2011
A Buffer-Based Approach to Rate Adaptation

Te-Yuan Huang et al., SIGCOMM‘14
Context: Highly-variable net throughput
System model
Current practice (in 2013)

\[ R(t) = F(B(t))\hat{C}(t) \]

- **Initial video rate**
- **Download & measure**
- **Buffer Occupancy**
- **Capacity estimation** $\hat{C}(t)$
- **Pick a rate**
- **Adjustment function** $F(B(t))$

Video rate for the next video segment.
Rate estimation for single buffered chunk

\[ R(t) < \left( \frac{B(t)}{V} \right) C(t) \]

\[ F(B(t)) < \left( \frac{B(t)}{V} \right) \left( \frac{C(t)}{\hat{C}(t)} \right) \text{ for all } t. \]
A purely buffer-based approach?

• **Goals**
  • (1) Avoid unnecessary rebufferings
  • (2) Maximize average video rate

• **Simplifications**
  • (1) Infinitesimal chunk size
  • (2) Any bit-rate available between two limits $R_{\text{min}}$ and $R_{\text{max}}$
  • (3) Videos encoded at constant bit-rate
  • (4) Infinitely long video
Rate map

(1) Continuous

(2) Strictly increasing between $R_{\text{min}}$ and $R_{\text{max}}$

(3) Pinned at both ends, $B=0$ ($R_{\text{min}}$) and $B=B_{\text{max}}$ ($R_{\text{max}}$)
But reality is more complex…

- (1) Infinitesimal chunk size — Chunks force discrete decisions
- (2) Any bit rate available — Only discrete bit-rates available
- (3) Videos encoded at constant bit rate — Variable bit-rate videos
- (4) Infinitely long video — Startup phase (building initial buffer) matters
BBA-0 handles (1) and (2)
Improvements in rebuffering rate

(a) Number of rebuffers per playhour during the day.
Average bit-rate
(3) Chunk sizes keep changing
Solution: Chunk map! (BBA-1)
BBA-1 bit-rate improvement

Figure 15: The BBA-1 algorithm improved video rate by 40–70 kb/s compare to BBA-0, but still 50–120 kb/s away from the Control.
But chunk map can cause bit-rate flips!
(4) Startup bit-rate estimation

- **Key idea:** Use previously achieved bit-rate/buffer occupancy when buffer alone doesn’t provide enough info.

- Increase bit-rate from $R_i$ to $R_{i+1}$ when

\[
\Delta B \geq V - \left( \frac{\text{ChunkSize}}{R_{i+1}} \right)
\]
BBA-2 bit-rate improvement
Conclusions

• Bit-rate and buffer control matters for streaming video

• Interesting problems & solutions in video control loop design
  • Lots of follow-up work innovating on basic ideas
  • … including mechanisms to “guess” bit-rates out of band

• However, very different ideas needed for real-time streaming