QAVA: Quota Aware Video Adaptation

Jiasi Chen, Amitabha Ghosh, Josphat Magutt, Mung Chiang
Princeton University

Dec. 12, 2012
Rise of Usage-Based Pricing

$10/GB charged by AT&T Wireless for 3G/4G data usage above 2GB
Rise of Video Traffic

70

Percentage of mobile data from video in 2016

Source: Cisco Visual Networking Index 2012
The Conflict Between Two Trends

Two emerging trends of Internet application:

- **Video traffic** becoming dominant
  
  High-resolution devices (e.g., iPhone, iPad, Android tablets)

- **Usage-based pricing** becoming prevalent

Can the consumer consume content without worrying about her wallet?
Current Video Adaptation Solutions

Two main approaches:

- Consumers may be warned by service providers or applications
  Android 4.0 provides data usage monitoring app; other iOS / Android apps

- “One size fits all” cutting back bit rates across all videos, for all users, at all times
  Youtube: channel-based quality adaptation depending on connection type
  Netflix: static quality adaptation to address wireline ISP quota constraints

Onavo: mobile app that compresses images and text to use less data

Adaptive HTTP streaming for bandwidth constraints

- Adobe Dynamic Streaming for Flash
- Microsoft Smooth Streaming for Silverlight and Windows Phone
- Apple HTTP Live Streaming for iOS
Video Consumption Tradeoff

A 3-way tradeoff

Within budget:
- Cost
  - MB of video

Minimize:
- Distortion
  - Video compressibility

Supply:
- # Videos watched
  - Usage profile
Quota-Aware Video Adaptation (QAVA)

Is every bit needed for every user at every time?

**Key idea**: All bytes are *charged* the same on cellular data plans, but not all bytes are equally *valuable* to mobile video experience.

Toy example: [http://www.youtube.com/watch?v=0sUBDpS9e2U](http://www.youtube.com/watch?v=0sUBDpS9e2U)

Stream Selector
Choose the right bitrate to maximize video quality

Video Profiler
Estimate compressibility of video

User Profiler
Predict user’s behavioral patterns from past history
QAVA System Architecture

Stream selector: located on user device / network / content provider
User profiler: requires access to user request logs
Video profiler: requires access to videos
**Video Profiler**
Estimate video compressibility

![Diagram of video delivery process]

- **Video request**
  - User Profiler (online)
  - Stream Selector (online)
  - Video Profiler (offline)

- **User device**
- **Video delivery**
- **Content provider’s server**
Leveraging Video Compressibility

Utility-cost tradeoff: diminishing returns for increasing cost

Different types of videos have different tradeoff curves – leverage this!

H.264/AVC video
Encoded at 100-900 kbps
720×480 pixels
Duration 6 mins

H.264/AVC videos
Encoded at 100,150,200, 300 kbps
640x480 pixels
Takeaway. Users have different perception of low- and high-motion videos. Low-motion videos are more compressible without perceptually noticeable distortion.
User Profiler
Predict user’s future data consumption patterns

User device

Video request

User Profiler (online)

Stream Selector (online)

Video Profiler (offline)

Content provider’s server

Video delivery
Seasonality and Trend in Time Series

**Seasonality**
Regularly spaced peaks and troughs with a consistent direction and approximately the same magnitude

Customer arrival in Starbucks who use Wi-Fi, NYC March 2010

**Trend**
Long term movement with an underlying upward or downward direction

Electric power consumption between 1975 and 1990

Our approach: estimate probability of request arrival in each time period
estimate video type preferences of each user
Stream Selection
How to choose the delivered video bitrate while staying under quota?

Video request

User device
User Profiler (online)

Stream Selector (online)

Content provider’s server
Video Profiler (offline)

Video delivery
Offline Stream Selection

If all video requests are known, we have the offline problem:

\[
\begin{align*}
\text{maximize} & \quad \sum_{t=1}^{T} \sum_{j=1}^{M_t} u_{t,j} x_{t,j} \\
\text{subject to} & \quad \sum_{t=1}^{N} \sum_{j=1}^{M_t} c_{t,j} x_{t,j} \leq B \\
& \quad \sum_{j=1}^{M} x_{t,j} \leq 1, \quad \forall \ t \\
& \quad x_{t,j} \in \{0, 1\}, \quad \forall \ t, j
\end{align*}
\]

maximize the total / average utility
spend less than budget
choose at most one bitrate per video

This is the multiple-choice knapsack problem

Online vs. Offline Stream Selection

Budget: 3
Goal: Maximize total utility
Items: (utility, cost)

Video 1

v11 (1,1)

v12 (2,2)
Online vs. Offline Stream Selection

Budget: 3
Goal: Maximize total utility
Items: (utility, cost)

Video 1

v11 (1,1)
v12 (2,2)
Online vs. Offline Stream Selection

Budget: 3
Goal: Maximize total utility
Items: (utility, cost)

Video 1
- v11 (1,1)
- v12 (2,2)

Video 2
- v21 (2,1)
- v22 (4,2)
Online vs. Offline Stream Selection

Budget: 3
Goal: Maximize total utility
Items: (utility, cost)

Video 1:
- v11: (1,1)
- v12: (2,2)

Video 2:
- v21: (2,1)
- v22: (4,2)
Online vs. Offline Stream Selection

Budget: 3
Goal: Maximize total utility
Items: (utility, cost)

Video 1
- v11 (1,1)
- v12 (2,2)

Video 2
- v21 (2,1)
- v22 (4,2)
Online vs. Offline Stream Selection

Budget: 3
Goal: Maximize total utility
Items: (utility, cost)

Video 1
- v11 (1,1)
- v12 (2,2)

Video 2
- v21 (2,1)
- v22 (4,2)
Online vs. Offline Stream Selection

Budget: 3
Goal: Maximize total utility
Items: (utility, cost)

Video 1
- v11 (1,1)
- v12 (2,2)

Video 2
- v21 (2,1)
- v22 (4,2)

Offline optimal: v11, v22
Total utility: 1+4 = 5
Total cost: 1+2 = 3

Online greedy: v12, v21
Total utility: 2+2 = 4
Total cost: 2+1 = 3
Modeling using Markov Decision Process

Possible videos $V = \{ (u,c), (u,c), (u,c) \}$; videos arrive randomly
Which bitrate to choose?

*Markov property.* Future bitrate decisions depend only on remaining budget, independent of past bitrate decisions

\[ t=1 \]

- $b, (u, c)$
  - *choose bitrate 1*
    - $b-c_1, (u, c)$
  - *choose bitrate 2*
    - $b-c_2, (u, c)$

\[ t=2 \]

- $b-c_1, (u, c)$
- $b-c_1, (u, c)$
- $b-c_2, (u, c)$
- $b-c_2, (u, c)$

- $b', (u, c)$
  - *choose bitrate 1*
    - $b'-c_1, (u, c)$
  - *choose bitrate 2*
    - $b'-c_2, (u, c)$

- $b'-c_1, (u, c)$
- $b'-c_1, (u, c)$
- $b'-c_2, (u, c)$
- $b'-c_2, (u, c)$
Simulation using Video Request Traces

YouTube request traces from wireless campus network
- 14 days, 16,337 users, 611,968 requests

4 bitrate selection algorithms:
- MDP: Our proposed approach
- MCKP: State-of-the-art literature
- Netflix: Solution in practice
- Offline: Hindsight offline optimal

Caveat: assumes perfect knowledge of number of video requests

Stream Selection Algorithm Comparison

How do algorithms perform for different user request traces, sweeping across quotas?

Conclusion: MDP achieves greater utility than other algorithms, without exceeding the quota.
Effects of Prediction Error

How robust is MDP algorithm to wrong user profiler or video profiler information?

Conclusion: Incorrect information only slightly decreases solution optimality
Implementation

Goals

- Test our architecture and system design
- Understand consumption behavior of real people
- Understand user perception of video quality
- Evaluate the algorithm
- Fun to run a trial involving real people
Silverlight Web Browser

Proof-of-concept implementation in web browser using Microsoft Silverlight
Android App Volunteer Trial

Developed QAVA as an Android application

Content provider: QAVA server

~500 videos encoded at 25 Kbps granularity (100 Kbps – 500 Kbps)

Participants: ~15 volunteers from Princeton community

Evaluation: Video quality feedback from users

Database logs:
- Video request
- Time stamp
- Android ID
- Video MB delivered
- Video quality feedback
Android App Screenshots

Category selection
Tailored to user preferences

Video selection
Regularly updated with new content

Video feedback
Primary means of evaluating user satisfaction
Conclusions & Future Work

Discussed conflicting trends of:
- Usage based pricing
- Increasing video consumption

Developed system design for quota-aware video adaptation
- Key idea: Not every bit needed for every user at every time
- Compared state-of-the-art literature and practical algorithms for video rate adaptation

Next: evaluate system performance with real user trial
explore client-based implementation architectures
Thank you!

QUESTIONS?