SmartTransfer

Transferring Your Mobile Multimedia Contents at the "Right" Time

Yichuan Wang, Xin Liu
University of California, Davis

Angela Nicoara
Deutsche Telekom Innovation Laboratories

Ting-An Lin, Cheng-Hsin Hsu
National Tsing Hua University
Mobile Multimedia Content

Why should we care?
30 million+ registered users
1 billion+ photos uploaded
5 million+ photos per day
575 likes per second
81 comments per second
Mobile Video: The Next Big Thing

• 16 Million Users (including Mark Zuckerberg)

• Just passed 20 million mark (12 million last week)
New kind of Content

Created on Mobile

Consumed on Mobile
To and from the Cloud

• Web album
  • Update user photos/videos automatically

• Email/Facebook/RSS feed/Tweets update
  • Fixed sync vs. dynamic sync

• Software update delivery
  • OS and App update over the air
  • Carriers decide the deadlines for delivery
Big Players are Trying It

• Apple iOS/Mac PhotoStream to iCloud

• Google Android InstantUpload to Picasa/Google+

• Google Android SyncManager
What are the Problems?

Network overload

Sporadic connectivity

Battery constraints
What are the Opportunities?

- Mobility
  - Network condition varies over time
  - We are all different

- Delay Elasticity
  - 55% of uplink data > 1 day old [Trestian Infocom’11]
  - Not all traffic is created equal

- App developers need a unified, easy to use, and effective API/Framework
SmartTransfer

• A framework that *intelligently* schedules mobile content transfer

• Predict network condition based on *individual* user profile

• Leverage network dynamics at a *larger* time scale

• Decide the optimal transmission time to maximize resource efficiency subject to deadline

• User can request instantaneous data transmission anytime
User Profile

- You are unique (and predictable)
  - 85% of the time, humans are visiting top 5 places [Gonzalez Nature’08]

- Network condition is tightly coupled with mobility
  - Signal variations are small on same path [Schulman Mobicom ‘10]
SmartTransfer Components

- **API**
  - Submit jobs through API
  - Specify deadlines

- **Scheduler**
  - Generate lookup table using scheduling algorithms
  - Schedule the actual transfer

- **Profiler**
  - Network condition trace
  - Usage pattern trace
Scheduling Model

\[
\begin{align*}
\text{Time} & \quad X_t \quad \text{Wait} \quad X_{t+1} \\
\text{User request} & \quad \text{Transfer} \quad \text{User request} \\
\text{End} & \\
\end{align*}
\]
Scheduling Algorithm

- Optimal Stopping Scheduling (OSS)
  - Markovian model for transfer cost
  - Transfer cost depends on last transfer cost and time of the day, etc.

- Lightweight OSS (OSSₗ)
  - Transfer cost independent over time
  - Depends on time of the day, day of the week

- Instant transfer (INS)
  - The current state-of-the-art

- Optimal schedule (OPT)
  - Offline optimal schedule, for benchmark
Optimal Stopping in the Literature

- Classic Secretary problem

- Secretary problem with random freeze equivalent to apply discount to payoff.
  [Samuel-Cahn Journal of the American Statistical Association ‘96]

- To the best of our knowledge, optimal stopping with arbitrary payoff function and random freeze haven’t been studied.
OSS and OSS\(_L\)

- Principle of Optimality
- Transfer when the *current cost* is smaller the *future expected cost*.

- Expected cost at time \(i\): \(C_i^*\)
- User request transfer at \(M\)

\[
\text{Action}_t = \begin{cases} 
\text{Transfer,} & M = i \\
\text{Transfer,} & M > i, X_i \leq C_{i+1}^* \\
\text{Wait,} & M > i, X_i > C_{i+1}^* 
\end{cases}
\]
Simulation with Real User Trace

- 12 users, 3 month long real traces
- OSS and OSSL algorithms always outperform the INS (up to 17 times)
Simulation with Real User Trace

- OSS and \( \text{OSS}_L \) introduce less delay than OPT
- \( \text{OSS}_L \) introduces less delay than OSS
Implication of Deadline

- OSS outperforms OSS\textsubscript{L} when the deadline is shorter than 12.5 hrs
- OSS\textsubscript{L} is more suitable for job with larger delay tolerance
Running Time

- OSS algorithm does not scale well with the time horizon
Android Implementation

• Android Open Source Project (AOSP)

• Andriod SyncManager API
  • AddPeriodicalSync

• Android SmartSync
  https://github.com/fmaker/SmartSync
Android SmartSync Library

- SmartTransfer Library
- Facebook PhotoSync App
  - Will be out in App Market soon.
- Video time!
Thanks!

- Questions?