Efficient Proactive Caching for Supporting Seamless Mobility

Vasilios Siris, Xenofon Vasilakos, George Polyzos
PhD student, AUEB, MMLAB
mm.aueb.gr/~xvas
xvas@aueb.gr
Problem

- Reduce propagation delay
  - $f(#\text{network hops})$
Approach (1/2)

- **Proactively** fetch data-objects to attachment points
- Is this a *typical* proactive caching approach?
Approach (2/2)

• Handoff mobility probabilities $q_1, q_2$

• Exploit **Individual** mobility & requests
  – *Not* data-popularities
Efficient Proactive Caching (EPC)

• Individual requests imply higher demand for cache space

• Congestion pricing for cache storage
  – Efficient cache utilization

  ➢ EPC trades cache space (price) for reduced delay (delay cost)
Outline

1. EPC in a flat cache structure
2. EPC in a two-level cache hierarchy
3. Evaluation
EPC IN A FLAT CACHE STRUCTURE
Flat cache structure

- Decision Rule:
  \[
  \begin{cases}
    1 & \text{if } q (D_R - D_L) \geq p_l \\
    0 & \text{if } q (D_R - D_L) < p_l
  \end{cases}
  \]

- Autonomous prefetching/caching

17/06/2014
Flat cache structure

• Step-wise decision procedure
  – Optimal selection of cached objects?
Flat cache structure

• Optimal selection of cached objects?

  1. Objects with different sizes
     • Optimization is identical to 0/1 Knapsack Problem
     • NP-hard problem

  2. Optimal for equal-size objects
     • For each cache and each request, order by
       \[ q \left( D_R - D_L \right) \]
EPC IN A TWO-LEVEL CACHE HIERARCHY
Hierarchical cache structure
Hierarchical cache structure

- Leaf nodes solve 2 flat cache problems:
  1. Delay $D_R$
  2. Delay $D_M$

- Requires cooperation

\[ D_{\text{mid}}^R - D_{\text{mid}}^M \geq p_{\text{mid}} \]
Finding an optimal solution?

- Data Placement Problem
  - Different object sizes => **NP-complete**
  - Equal size objects => high polynomial degree time
EVALUATION
Evaluation

Comparison with a naive, an optimal, and an oracle scheme
Evaluation

Comparison with a naive, an optimal, and an oracle scheme

![Bar chart showing performance comparison between naive, optimal, and oracle schemes]
Evaluation

Comparison with a naive, an optimal, and an oracle scheme
Evaluation

(a) $D_M/D_L = 5$

(b) $D_M/D_L = 2$
Evaluation

Comparison with a naive, an optimal, and an oracle scheme
A distributed mobility support solution tailored to individual user mobility/requests that exploits user mobility and uses congestion pricing.
Bibliography

