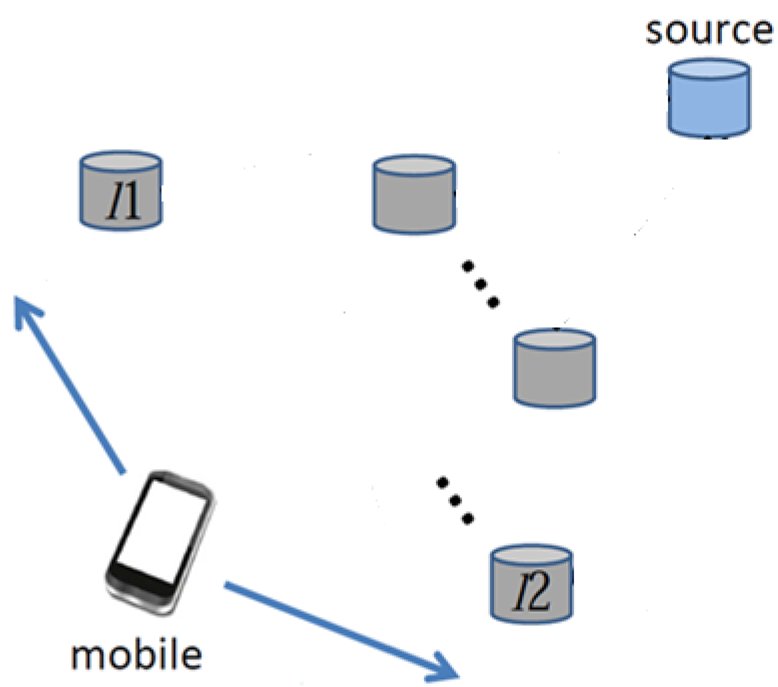
Efficient Proactive Caching for Supporting Seamless Mobility

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

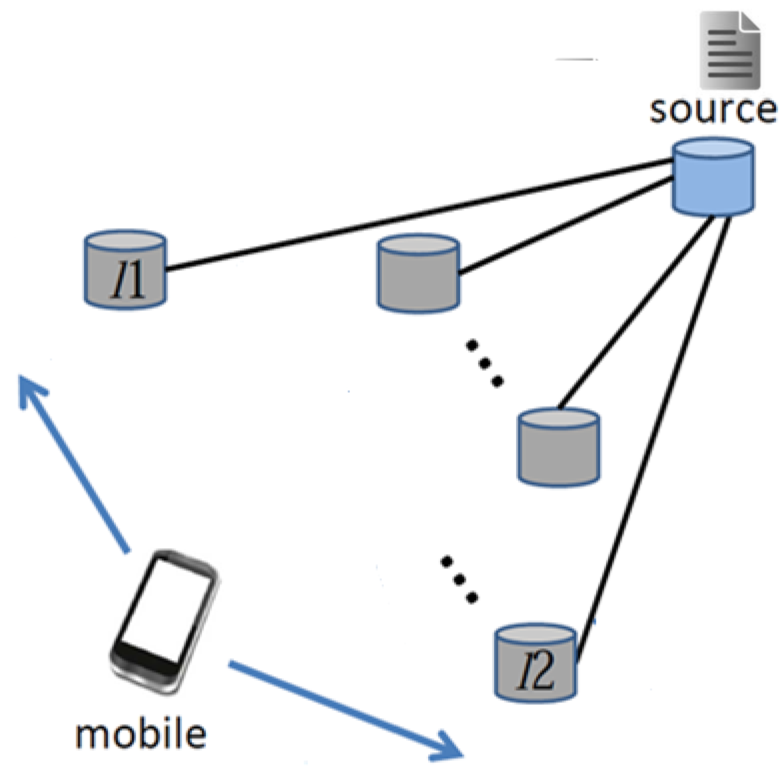
Problem



- Reduce **propagation delay**
  - f(#network hops)

Approach (1/2)

- **Proactively** fetch data-objects to attachment points
- Is this a *typical* proactive caching approach?

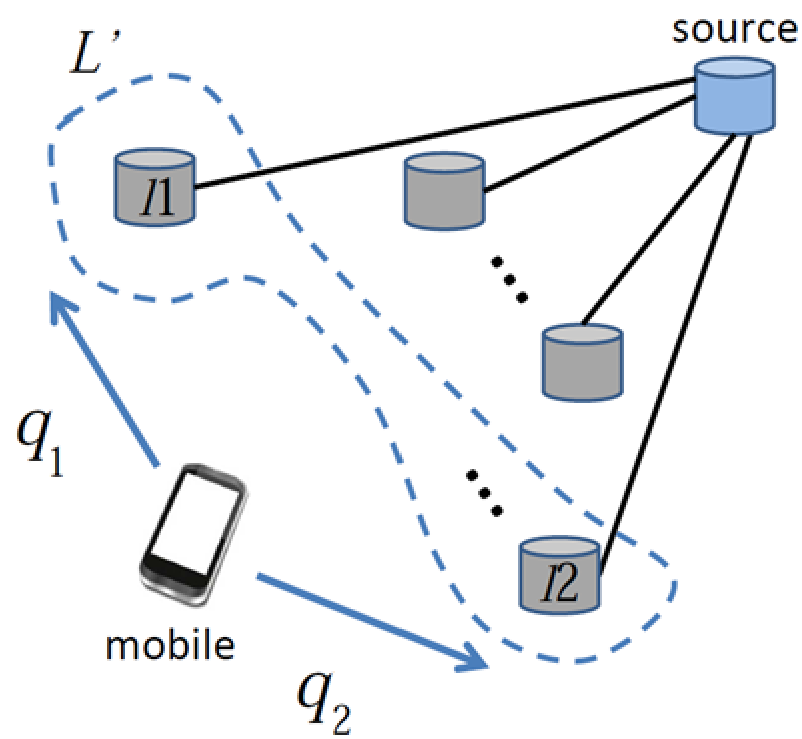


Approach (2/2)

- 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

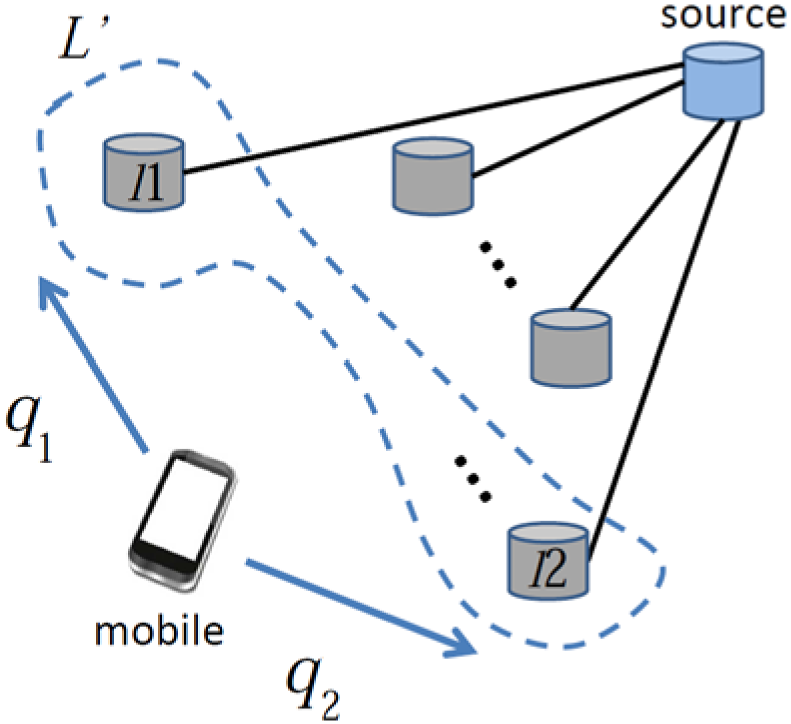
- **Congestion pricing** for cache storage
  - *Efficient* cache utilization

➢ **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

Flat cache structure



- **Decision Rule:**

$$\begin{cases} 1 & \text{if} \quad q\,(D_R - D_L) \geq p_l \\ 0 & \text{if} \quad q\,(D_R - D_L) < p_l \end{cases}$$

➢ **Autonomous** prefetching/ caching

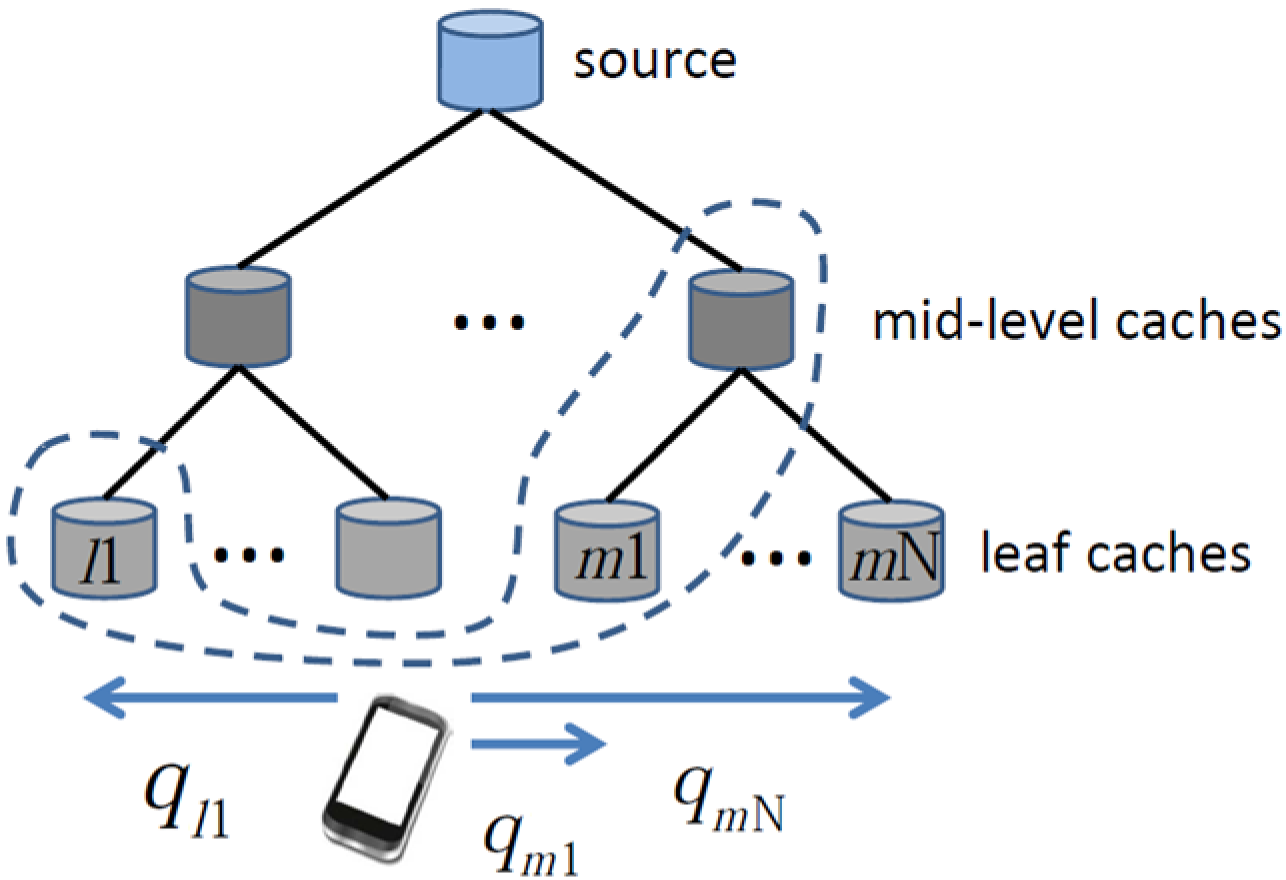
Flat cache structure

- Step-wise decision procedure
  - Optimal selection of cached objects?

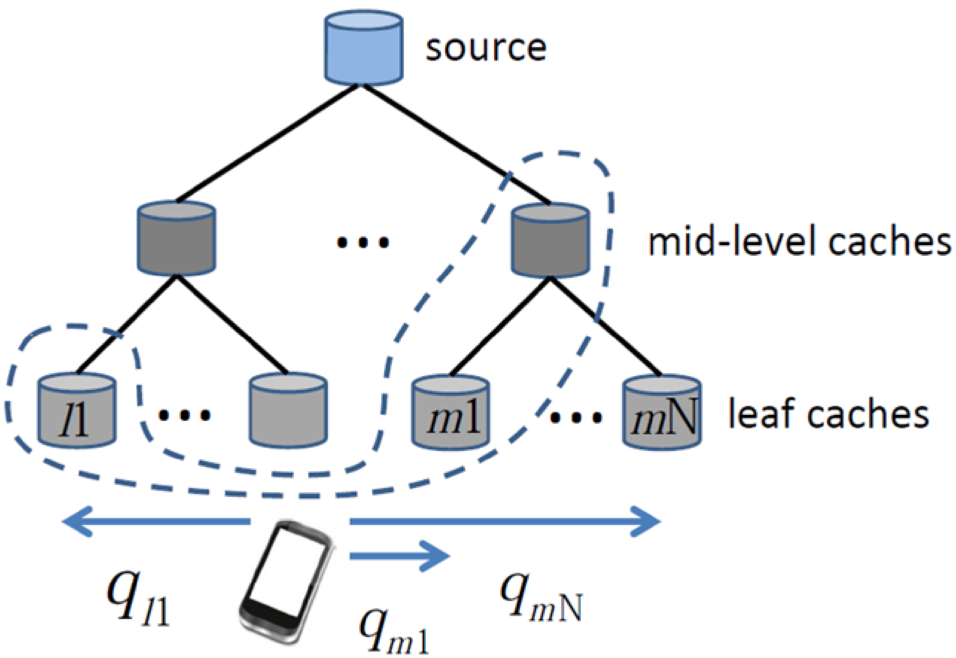
Flat cache structure

- 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 $q_{.}\,(D_{R} - D_{L})$

Hierarchical cache structure



Hierarchical cache structure



- Leafs solve 2 flat cache problems :
  1. Delay D_R
  2. Delay D_M

- $\mathcal{D}_{\mathrm{R}}^{\mathrm{mid}} - \mathcal{D}_{\mathrm{M}}^{\mathrm{mid}} \geq p_{\mathrm{mid}}$

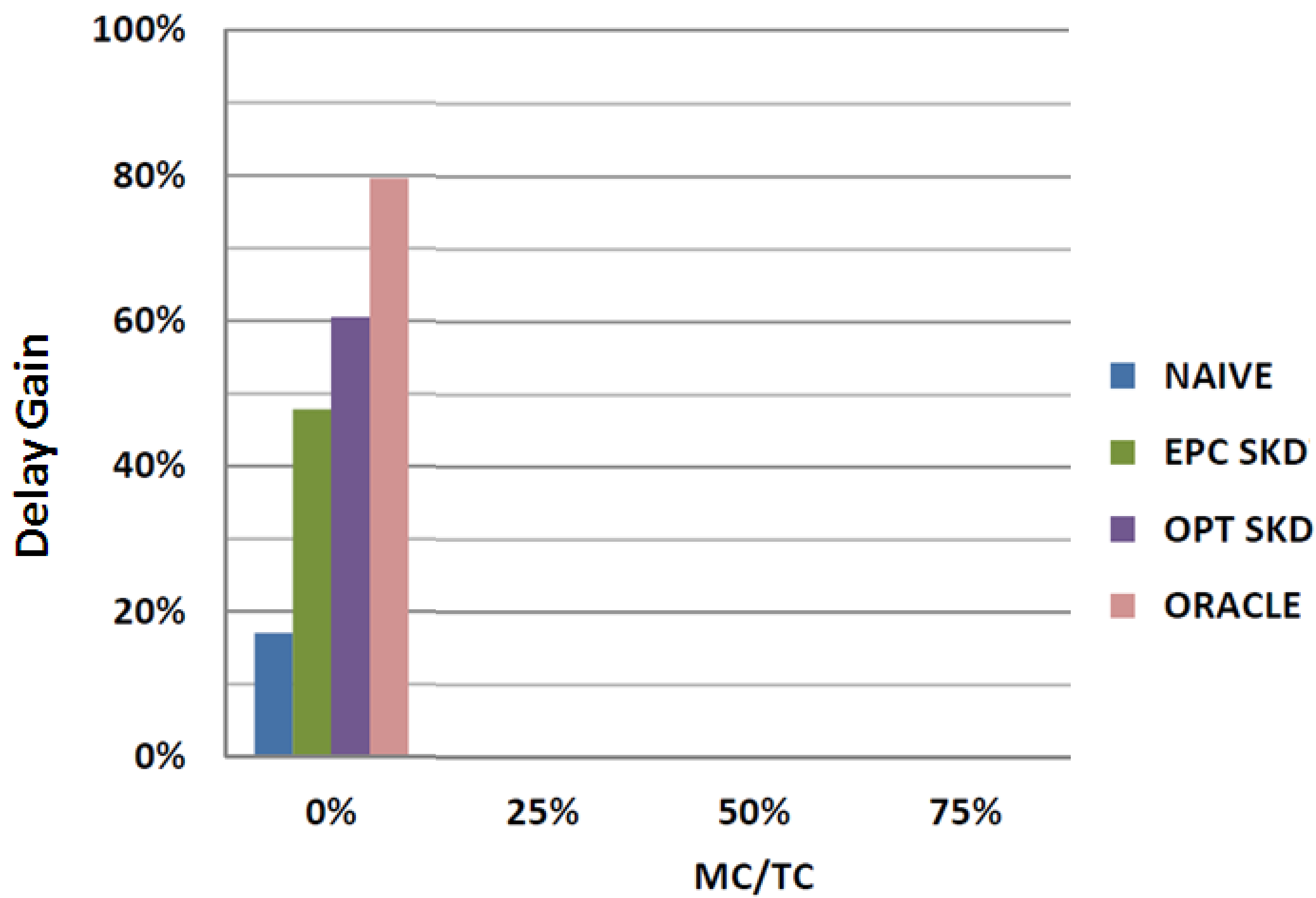
➢ Requires ***cooperation***

Finding an optimal solution ?

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

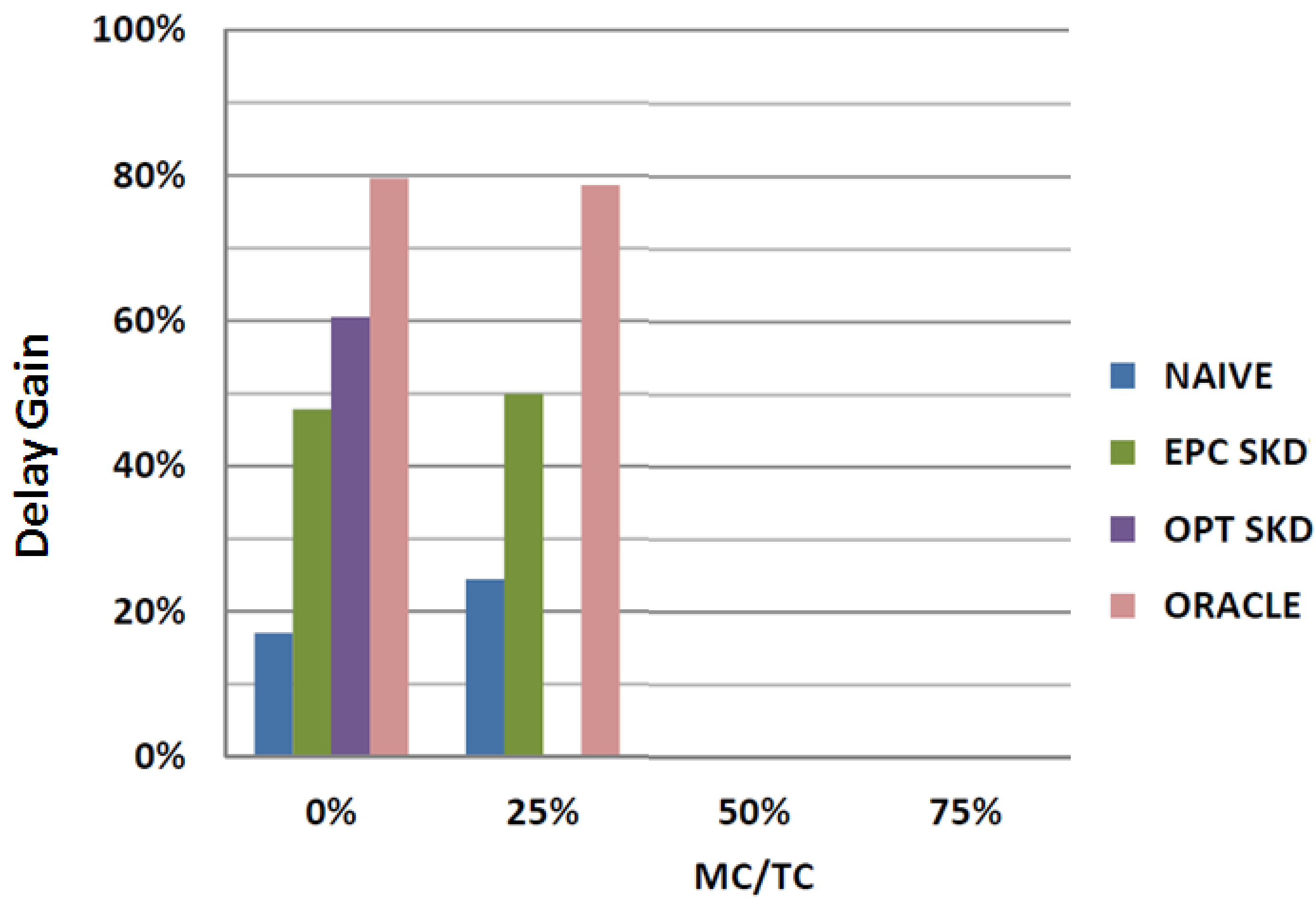
Evaluation

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



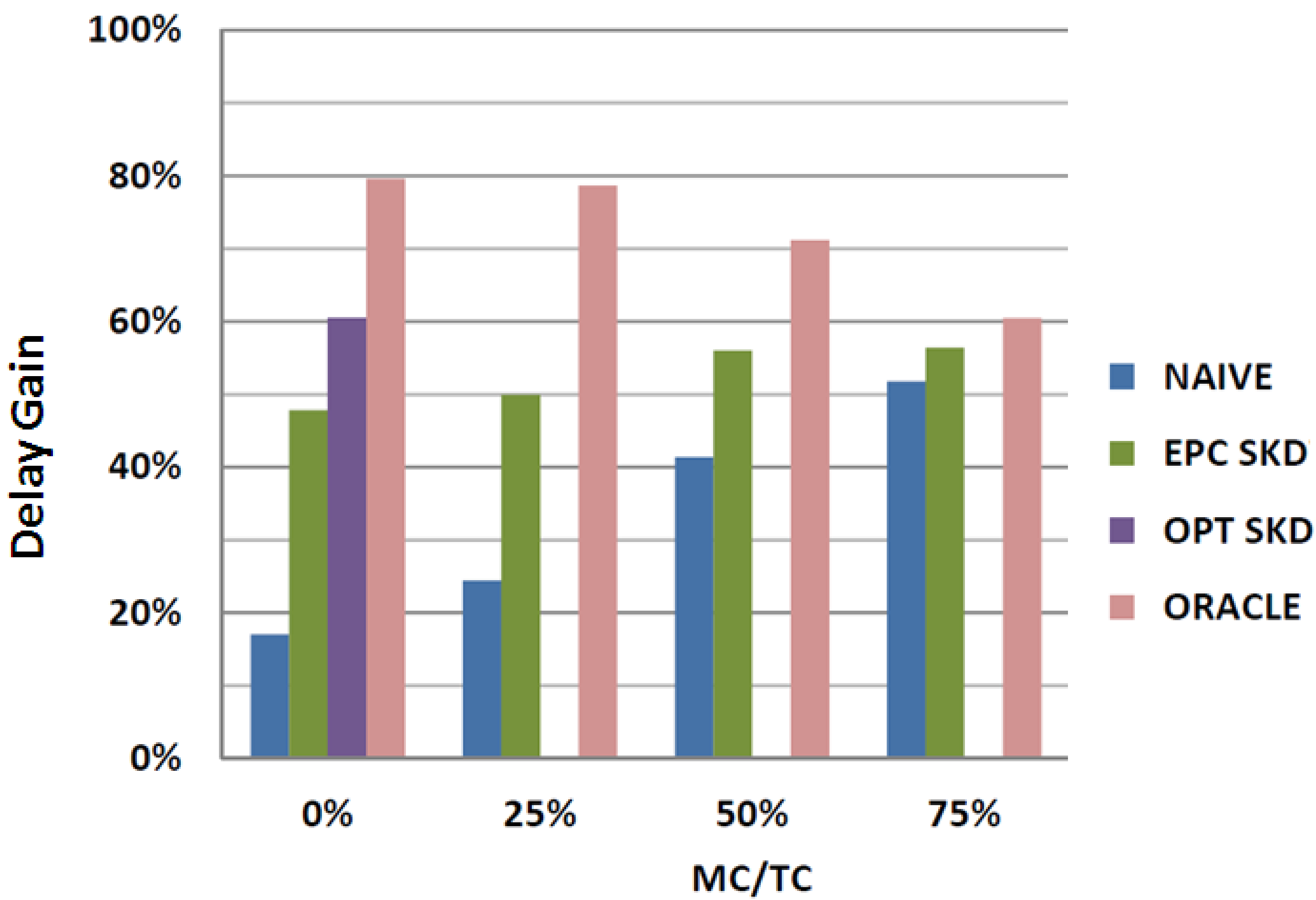
Evaluation

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

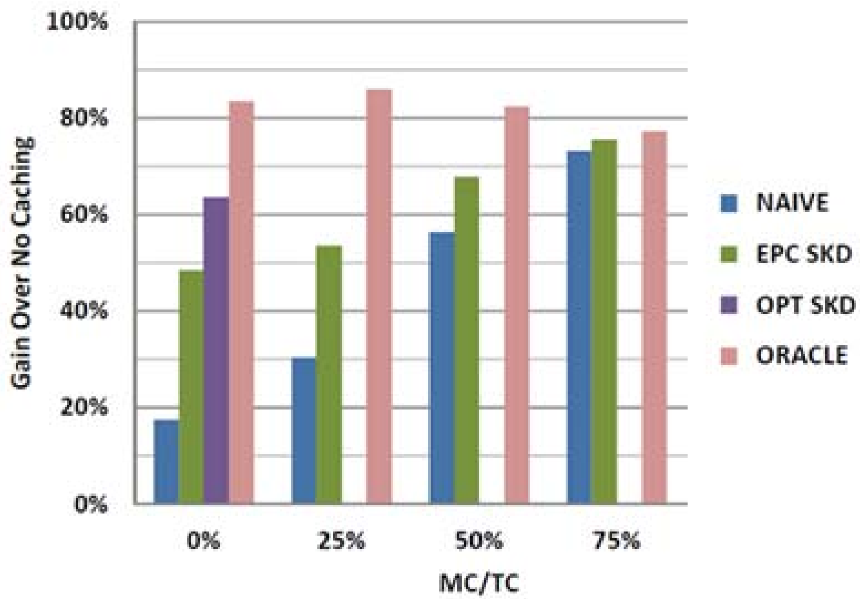


Evaluation

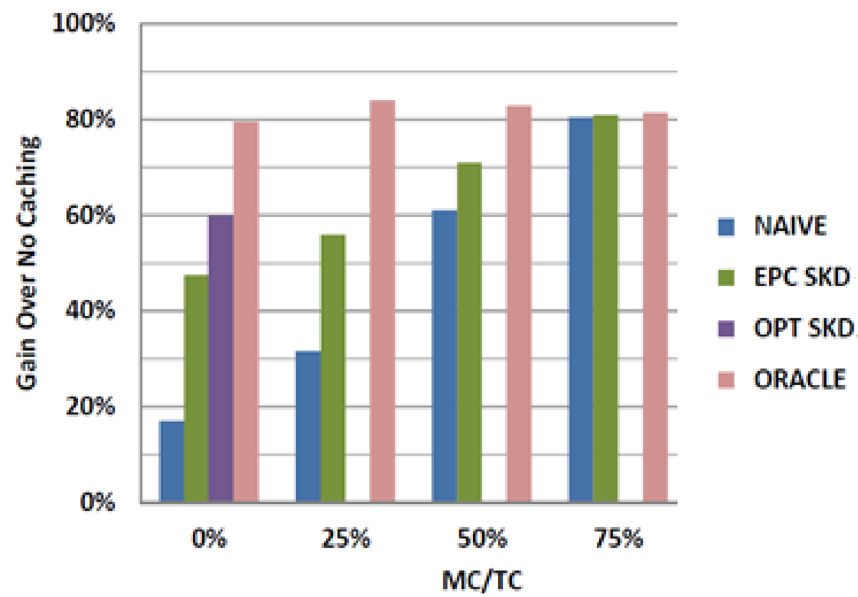
Comparison with a naive, an optimal, and an oracle scheme



Evaluation



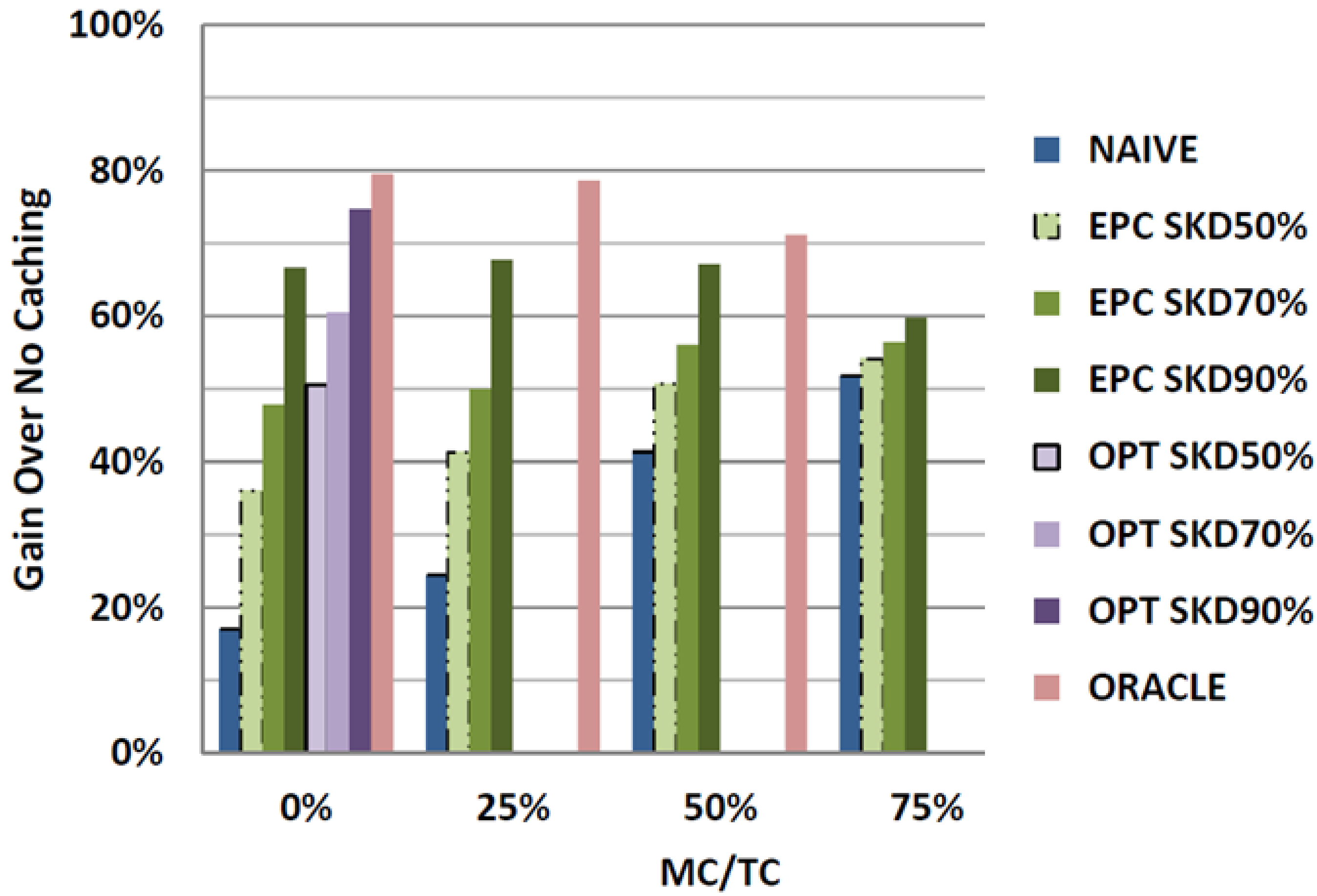
(a) $D_M/D_L = 5$



(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**