Caching Policies for Delay Minimization in Small Cell Networks with Joint Transmissions

Guilherme I. Ricardo $^{1,2}$
Giovanni Neglia $^2$  Thrasyvoulos Spyropoulos $^1$

$^1$EURECOM, Communication Systems Department

$^2$Inria, Université Côte d’Azur
Agenda

1. Motivation
2. Single Server Caching
3. FemtoCaching Problem
4. Cooperative MultiPoint Systems
5. CoMP Caching Policies
6. Conclusion
**Motivation**

**Content Distribution Networks**

- **Scenario**: Increasing mobile and cellular data usage.
- **Question**: How to provide better QoS under such scenario?
- **Solution**: Content replication closer to final user - Caching!

*Figure* – CDN Multiserver Caching Strategy – Source

Guilherme Ricardo
Single Server Caching

Introduction

- Problem: What to cache?
- Performance metric: Hit Ratio
- Popularity is known: Store the most popular contents
- Popularity is unknown/dynamic: Caching algorithms (policies)
Single Server Caching

Policies Examples - Least Frequently Used (LFU)

**Figure** – LFU Caching Policy – Source
Single Server Caching
Policies Examples - Least Recently Used (LRU)

Variations:
- $q$LRU – probabilistic insertion, $0 \leq q \leq 1$
- $k$LRU – multilevel cache, $k = 1, 2, ...$
FemtoCaching Problem

5G Heterogeneous Networks Topology

Figure – Heterogeneous Network Topology – Source
FemtoCaching Problem

The Optimization Formulation

Let $X$ be the allocation matrix such that $x_{hf} = 1$ if helper $h$ caches content $f$ and $x_{hf} = 0$ otherwise. The problem is:

$$\begin{align*}
\text{maximize} & \quad F(X) = \frac{1}{U} \sum_{f=1}^{F} p_f \sum_{u=1}^{U} \mathbb{1}\{k(u,f)>0\} \\
\text{subject to} & \quad \sum_{f=1}^{F} x_{hf} = C, \ h = 1, \ldots, H,
\end{align*}$$

where $F$ is the catalog size, $C$ is the cache capacity, $U$ is the number of users, $k(u,f) \triangleq \sum_{h \in \mathcal{H}(u)} x_{hf}$, and $\mathcal{H}(u)$ is the set of helpers covering user $u$. 
FemtoCaching Problem
The Offline Solution – Femto (2015)

- NP-Hard Problem (Combinatorial Nature)
- Greedy Algorithm:
  - $F(X)$ is monotone and submodular
  - Constraints form a matroid partition
  - $1/2$-Approximation ratio
- Drawbacks: Strong assumptions, e.g.,
  - Centralized intelligence
  - Network topology and popularities are static and known
FemtoCaching Problem

The Online Solutions – Caching Policies

- LRU-One and LRU-All – Giovanidis (2016)
- qLRU-Lazy – Neglia (2018)

qLRU-Lazy Policy Description

1. Only the helper that served the file can update its cache; and
2. It only does so if it is the only one able to actually serve it
Definition

The delay $d(u, f, X)$ for user $u$ to download content $f$ under allocation $X$ is

$$
d(u, f, X) = \begin{cases} 
  d_B + \frac{M}{W \log_2 (1 + \max_h g_{hu})}, & \text{if cache miss} \\
  \frac{M}{W \log_2 (1 + \sum_h g_{hu} x_{hf})}, & \text{if cache hit,}
\end{cases}
$$

where $d_B$ is the backhaul delay, $g_{hu}$ is the SNR from $h$ to $u$, $M$ is the file size, and $W$ is the channel bandwidth.
Delay Minimization Problem

\[
\text{minimize } \quad F(X) = \frac{1}{U} \sum_{f=1}^{F} \sum_{u=1}^{U} p_f d(u, f, X) \\
\text{subject to } \quad \sum_{f=1}^{F} x_{hf} = C, \ h = 1, \ldots, H
\]

Remark

Submodularity Proof and Greedy Algorithm
Cooperative Multipoint Systems

Hit Ratio → Avg. Delay

**Figure** – Static allocation for different overlapping levels (full rep.)
Cooperative Multipoint Systems

Optimal Allocation: \( d_B \times \text{SNR Bounds} \)

Assumptions:
- Completely overlap
- Homogeneous SNR (\( \gamma \))

For a given \( \gamma \), if \( d_B \geq d_{B,\max} \) such that

\[
d_{B,\max}(C, H, \alpha, \gamma) \triangleq (HC)^\alpha \frac{M}{W} \left( \frac{1}{\log_2(1 + \gamma)} - \frac{1}{\log_2(1 + 2\gamma)} \right)
\]

then the optimal allocation is full diversity.

For a given \( \gamma \), if \( d_B \leq d_{B,\min} \) such that

\[
d_{B,\min}(C, H, \alpha, \gamma) \triangleq \left( \frac{C + 1}{C} \right)^\alpha \frac{M}{W} \left( \frac{1}{\log_2(1 + (H - 1)\gamma)} - \frac{1}{\log_2(1 + H\gamma)} \right)
\]

then the optimal allocation is full replication.
Cooperative Multipoint Systems

Optimal Allocation: $d_B \times$ SNR Bounds, Example

Tradeoff: Backhaul Delay $x$ SNR

$F=1000000; C=100; T=10; \alpha=1.5$
CoMP Caching Algorithms

\textit{qLRU-\(\Delta d\) Policy Notation}

- Let \(I_u\) be the set of helpers covering user \(u\) and \(J_{u,f} \subseteq I_u\) be the subset of those helpers caching \(f\).
- The marginal gain for adding a copy of file \(f\) at helper \(h\) is defined as:
  \[
  \Delta d^{(h)}(u, f, X) \triangleq d(u, f, X \ominus e^{(h)}) - d(u, f, X)
  \]
- Normalizers:
  \[
  \beta \triangleq 1/(\max_{f,h,u,X} \Delta d^{(h)}(u, f, X))
  \]
  \[
  \gamma \triangleq 1/(\max_{f,h,u,X} \Delta d^{(h)}(u, f, X \oplus e^{(h)})).
  \]
CoMP Caching Algorithms

$qLRU-\Delta d$ Policy Introduction

$qLRU-\Delta d$ Policy General Description

At every request \((u, f)\), each \(h \in I_u\) updates its cache as follows:

- If \(h \in J_{u,f}\), reset \(f\)'s cache position with probability:
  \[
  \rho^{(h)}(u, f, X) = \beta \cdot \Delta d^{(h)}(u, f, X)
  \]

- If \(h \in I_u \setminus J_{u,f}\), store \(f\) to \(h\)'s cache with probability:
  \[
  q \cdot \sigma^{(h)}(u, f, X), \text{ where } q \in (0, 1] \text{ is fixed and}
  \]
  \[
  \sigma^{(h)}(u, f, X) = \gamma \cdot \Delta d^{(h)}(u, f, X \oplus e^{(h)})
  \]
CoMP Caching Algorithms

$qLRU-\Delta d$ Policy Introduction

$qLRU-\Delta d$ Policy Algorithmic Description

**Input:** $I_u$, $J_{u,f}$, and $g_{h',u}, \forall h' \in I_u$

**for** $h \in I_u$ **do**

  **if** $h \in J_{u,f}$ **then**

  Move $f$ to the front with prob. $\rho^{(h)}$

  **else**

  Evict file in the cache’s last position;

  Insert $f$ with prob. $q \cdot \sigma^{(h)}(u, X_f)$.

**end**

**end**
Remark – Ricardo (2020)

Under IRM, Che’s, and Exponentialization approximations, a network of $q_{LRU-\Delta d}$ caches converges to a locally-optimal caching configuration when $q \to 0$. 

![Graph](image-url)
CoMP Caching Algorithms

2LRU-Δd Policy Notation

- IRM ≠ Real request process (Temporal locality)
- Each helper deploys a 2-levels cache: the physical cache storing the actual file and the virtual cache storing files’ metadata (i.e., ID)
- Let $I_u$ be the set of helpers covering user $u$ and let $J_{u,f}, \hat{J}_{u,f} \subseteq I_u$ be the subsets of those helpers storing $f$ at the physical cache and at the virtual cache, respectively.
CoMP Caching Algorithms

2LRU-\(\Delta d\) Policy Introduction

2LRU-\(\Delta d\) Policy General Description

At every request \((u, f)\), each \(h \in I_u\) updates its cache as follows:

- If \(h \in \hat{J}_{u,f}\), move \(f\)'s ID to the front of \(h\)'s virtual cache and,
  - if \(h \in J_{u,f}\), move \(f\) to the front of \(h\)'s physical cache with prob. \(\rho^{(h)}(u, f, X)\);
  - else, evict the file in the physical cache’s last position and insert \(f\).

- If \(h \notin \hat{J}_{u,f}\), with prob. \(q \cdot \sigma^{(h)}(u, f, X)\), evict the ID in \(h\)'s virtual cache’s last position and insert \(f\)'s ID.
2LRU-Δd Policy Introduction

**Input:** \( I_u, J_{u,f}, \hat{J}_{u,f}, \) and \( g_{h',u}, \forall h' \in I_u \)

for \( h \in I_u \) do

if \( h \in \hat{J}_{u,f} \) then

Move \( f \)'s ID to the front of the virtual cache;

if \( h \in J_{u,f} \) then

Move \( f \) to the front of the physical cache with prob. \( \rho(h) \)

else

Evict file in physical cache’s last position;

Insert \( f \).

end

else

Evict file’s ID in virtual cache’s last position;

Insert \( f \)'s ID with prob. \( \sigma(h) \).

end

end
CoMP Caching Algorithms
Numerical Results – IRM, Homogeneous SNR
CoMP Caching Algorithms

Numerical Results – Real, Homogeneous SNR
CoMP Caching Algorithms
Numerical Results – Real, Heterogeneous SNR
Conclusion and Future Works

- Conclusions
  - Delay cost function under CoMP provides different allocation with potentially better download rates
  - $q$LRU-$\Delta d$ Policy outperforms other Hit Ratio dynamic policies for synthetic requests

- Future Work
  - Finish Real Traces Experiments
  - Greedy Algorithm with pair of files
  - Finish Algorithm
Thank You!