Efficient Online Collaborative Caching in Cellular Networks with Multiple Base Stations

Pouya Ostovari, Jie Wu, and Abdallah Khreishah

Temple University

New Jersey Institute of Technology

Center for Networked Computing

http://www.cnc.temple.edu

Agenda

- Introduction
  - Motivation
- Collaborative caching
  - Offline
  - Online
    - Algorithm
    - Complexity and competitive ratio
  - Optimal solution
- Evaluations
- Conclusions

Motivation

- Mobile data volume will grow 11 times larger from 2013-2018
- Video traffic will dominate (> 90%)

  [Cisco Mobile Data Traffic Forecast 2014-2019]
- Wireless spectrum is limited

Motivation

- Data storage price is decreasing very fast
- Popular contents are requested frequently

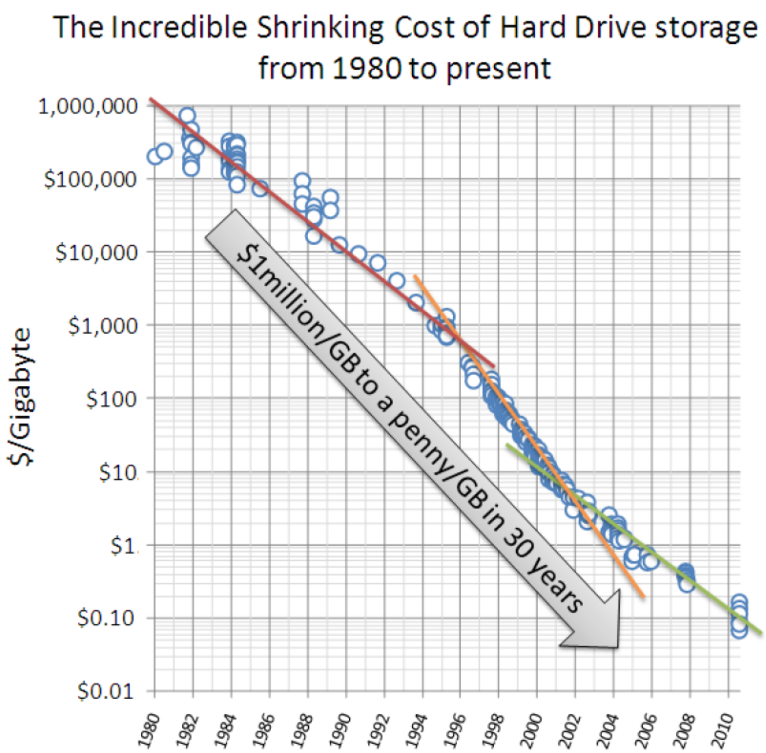
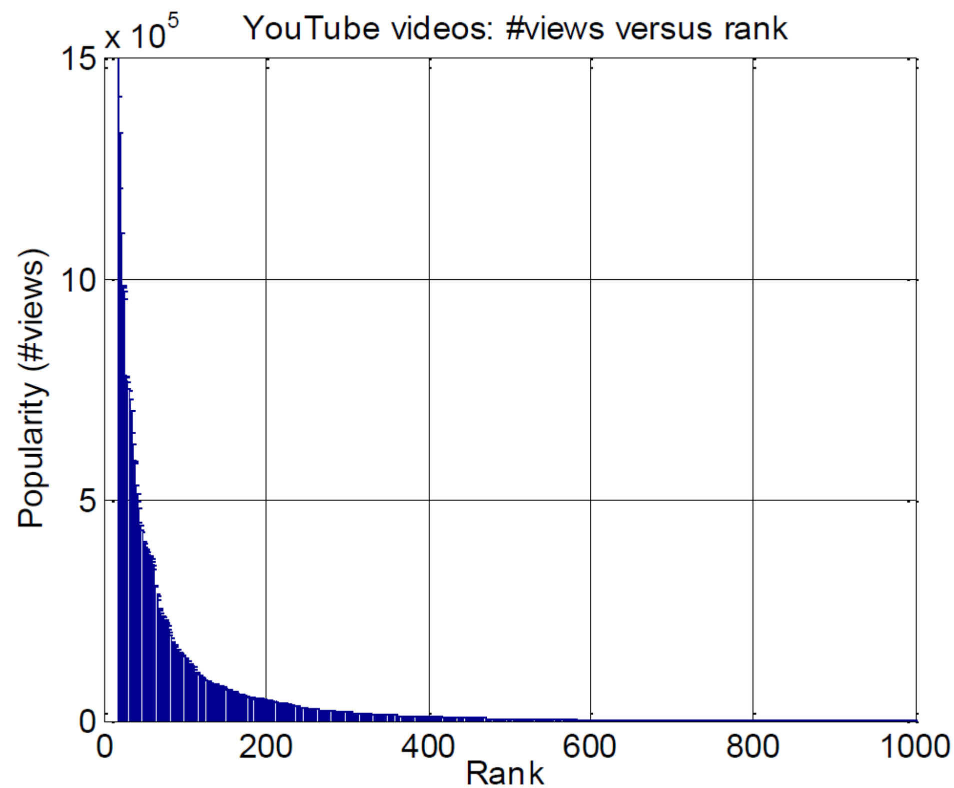


Chart: honesthypocrite.blogspot.com
Data: http://ns1758.ca/winch/winchest.html



Motivation

- Cellular carriers use LRU
- Hierarchical caching in several works

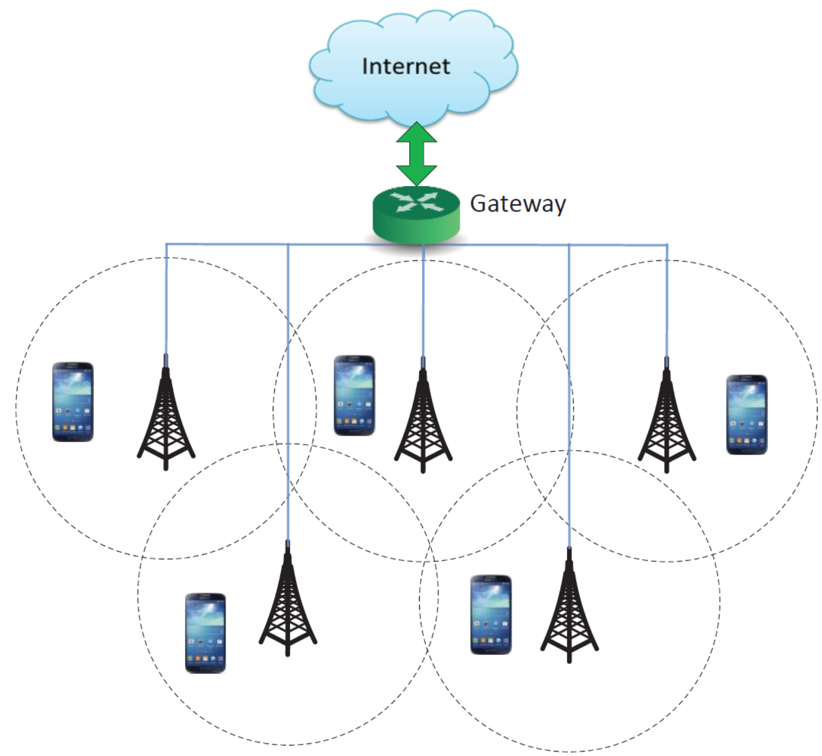
  [Ermalet. al. ‘11, Ahlehaghet. al. ’12]

  [Karamchandaniet. al. ‘14]
- Using helpers in other studies (No use of backhaul links)

  [Golrezaeiet. al. ‘12, Poularakiset. al. ‘14]
- In this work, we utilize the backhaul links to introduce collaborative caching

Setting

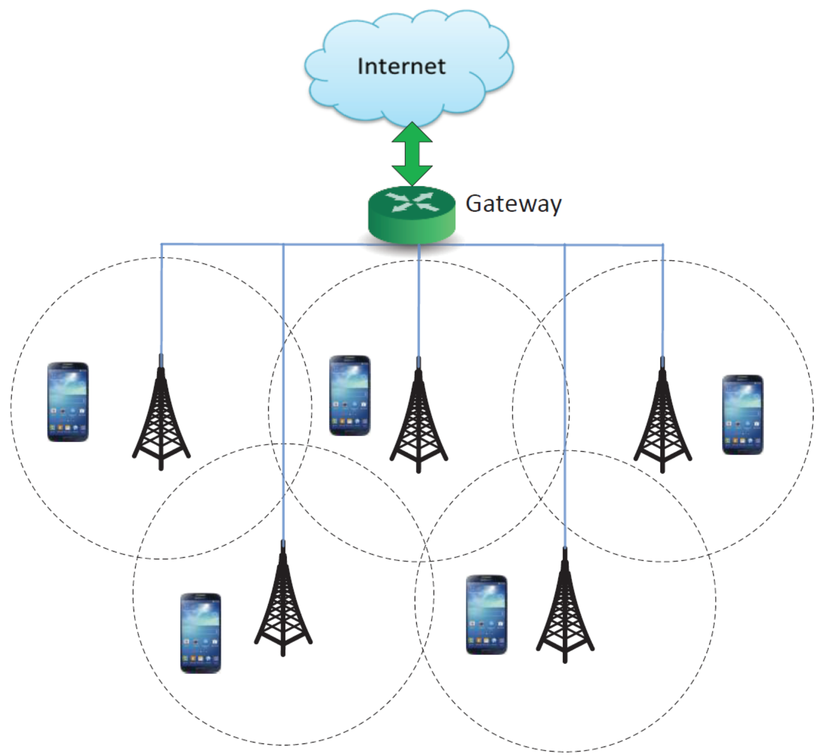
- Cellular networks
- Local cache storages
- Backhaul links



- Two models
  - Unlimited cache size
  - Limited cache size

Setting

- Two sources of cost
  - Caching cost
    - Paid to the cellular network or cloudlet service providers
  - Traffic on the backhaul links



- Objective
  - To minimize the aggregated caching and download cost of the cellular network

Problem Formulation

- Offline caching
- The popularity of each video is known

$$\min \sum_{i=1}^{n} [D_i + C_i]$$

$$s.t \quad y_{ik} \in \{0, 1\} \; \forall \; i, k$$

$$D_i = \sum_{k=1}^{m} \sum_{j=1}^{n} d_{i(j)_i} v_k \gamma_k r_i \left[\prod_{h=1}^{j-1} (1 - y_{(h)_i k}) \right] y_{(j)_i k}$$

$$+ \sum_{k=1}^{m} d_{ib} v_k \gamma_k r_i \left[\prod_{h=1}^{n} (1 - y_{(h)_i, k}) \right] y_{bk}$$

$$C_i = \sum_{k=1}^{m} g_i y_{ik} v_k t$$

Complexity

- The above problem is NP-complete
  - Reduction from the set-cover problem
- The optimization is a submodular function
- A greedy algorithm can achieve an approximation within a factor 2 of the optimum solution for the optimization

Online Caching

- The future requests are unknown
- Determines the caching based on incoming requests
- The first few requests will be served through the Internet
- The history of the requests show whether or not a content is popular
  - Popular contents will be cached

Online Caching

- We need to decide whether to serve a request through
  1. The Internet
  2. By caching the content at a base station
  3. Or by retrieving the content from a base station that has already cached the content

Online Caching

- Define a potential function for each base station and content

- Potential function denotes how much a base station (cache) can be useful in reducing the download cost of a content

- (*potential - caching cost*)>0 means that caching a content is beneficial

Online Caching

- Content *k* that is not requested for a while should be removed from the cache to reduce the caching cost
  - Base stations calculate the amount of help provided from caching content *k*
  - In the case that the total caching cost becomes greater than a fraction of this benefit (β), content *k* will be removed from the cache

Online Caching- Algorithm

- On upcoming request for content k to base station i
  1. Update potential functions u_{jk} for all base stations j
  2. Find base station $h = \arg\max_{j \in N}(u_{jk} - g_j)$
  3. If $u_{hk} - g_h > 0$
  4. Cache k at base station j
  5. Serve the request from the base station with the minimum download cost
  6. Update b_{hk} # amount of total cost reduction by caching # content k at base station h

Online Caching- Algorithm

- Iterative cost calculation and cache release
  1. for all base stations and contents
  2. $c_{jk} = c_{jk} + g_j v_k$ # updating caching cost
  3. if $c_{jk} > b_{jk}/\beta$
  4. remove content k from base station j

If the caching cost is greater than a given fraction of the gained benefit of caching content k at base station j remove content k

Online Caching- Limited Caches

- The same idea as the model with unlimited caches
- In the case that a cache becomes full
  - A new cached content *k* will replace the cached content(s) with smaller potential functions

Optimal Solution

- To evaluate the performance of the proposed online algorithms
- We assume that we know the exact time of all of the requests
- Formulating the problem as a linear programming optimization
- We use network coding

Network Coding

- Random linear network coding
  - Linear combinations of the packets
  - Gaussian elimination

$$\begin{cases} q_1 = \alpha_{1,1} p_1 + \alpha_{1,2} p_2 + \alpha_{1,3} p_3 \\ q_2 = \alpha_{2,1} p_1 + \alpha_{2,2} p_2 + \alpha_{2,3} p_3 \\ \vdots \\ q_\mathrm{n} = \alpha_{n,1} p_1 + \alpha_{n,2} p_2 + \alpha_{n,3} p_3 \end{cases}$$

- Applications of network coding
  - Reliable transmissions
  - Throughput/capacity enhancement
  - Content distribution

Optimal Solution- Unlimited Cache

$$\min D + C$$

$$s.t \quad D \geq \sum_{i,j=1}^{n} \sum_{k=1}^{m} \sum_{\tau=1}^{\tau} d_{ij} x_{ij}^{k\tau} v_k + \sum_{i=1}^{n} \sum_{k=1}^{m} \sum_{\tau=1}^{t} d_{ib} x_{ib}^{k\tau} v_k$$

$$C \geq \sum_{i=1}^{n} \sum_{k=1}^{m} \sum_{\tau=1}^{t} g_i y_{ik}^{\tau} v_k$$

$$x_{ij}^{k\tau} \leq y_{jk}^{\tau}, \ \forall \ i, j, k, \tau$$

$$x_{ib}^{k\tau} + \sum_{j=1}^{n} x_{ij}^{k\tau} \geq r_{ik}^{\tau}, \ \forall \ i, k, \tau$$

$$x_{ib}^{k\tau} \geq y_{ik}^{\tau} - y_{ik}^{\tau-1}, \quad \forall i, k, \tau$$

$$y_{ik}^{\tau}, x_{ij}^{k\tau} \in [\, 0, 1] \,, \ \forall \ i, j, k, \tau$$

Optimal Solution- Limited Cache

$$\min D + C$$

$$s.t \quad D \geq \sum_{i,j=1}^{n} \sum_{k=1}^{m} \sum_{\tau=1}^{\tau} d_{ij} x_{ij}^{k\tau} v_k + \sum_{i=1}^{n} \sum_{k=1}^{m} \sum_{\tau=1}^{t} d_{ib} x_{ib}^{k\tau} v_k$$

$$C \geq \sum_{i=1}^{n} \sum_{k=1}^{m} \sum_{\tau=1}^{t} g_i y_{ik}^{\tau} v_k$$

$$x_{ij}^{k\tau} \leq y_{jk}^{\tau}, \ \forall\ i, j, k, \tau$$

$$x_{ib}^{k\tau} + \sum_{j=1}^{n} x_{ij}^{k\tau} \geq r_{ik}^{\tau}, \ \forall\ i, k, \tau$$

$$\sum_{k=1}^{m} y_{ik}^{\tau} v_k \leq s_i, \ \forall i$$

$$x_{ib}^{k\tau} \geq y_{ik}^{\tau} - y_{ik}^{\tau-1}, \quad \forall i, k, \tau$$

$$y_{ik}^{\tau}, x_{ij}^{k\tau} \in [0, 1], \ \forall\ i, j, k, \tau$$

Evaluations

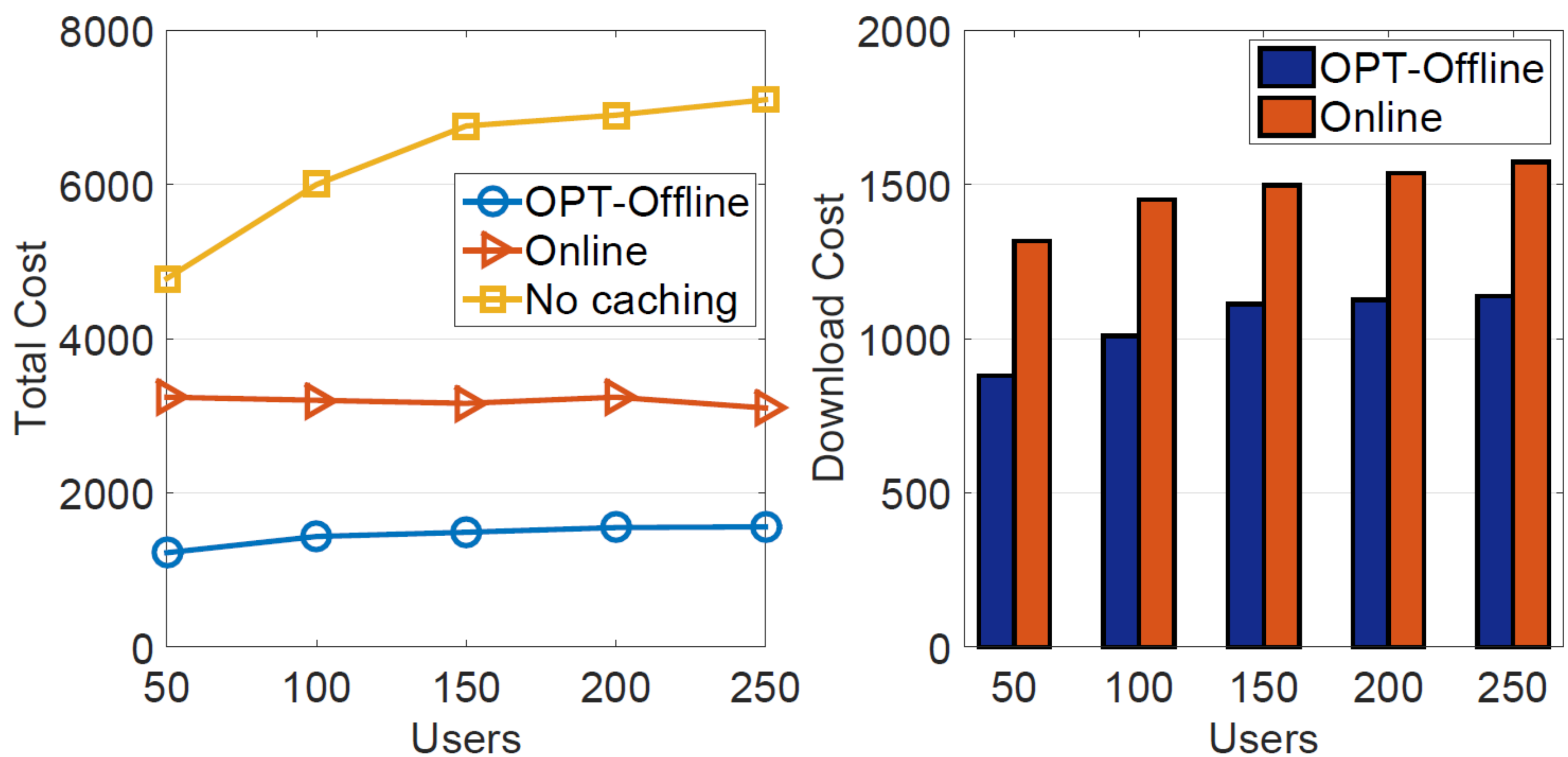
- Simulator in Matlab environment

- 100 random runs
  - Content requests by the users are randomly selected
  - Download and caching costs are randomly chosen

- Comparing the proposed online algorithm against that of the optimal solution
  - Total cost
  - Download cost

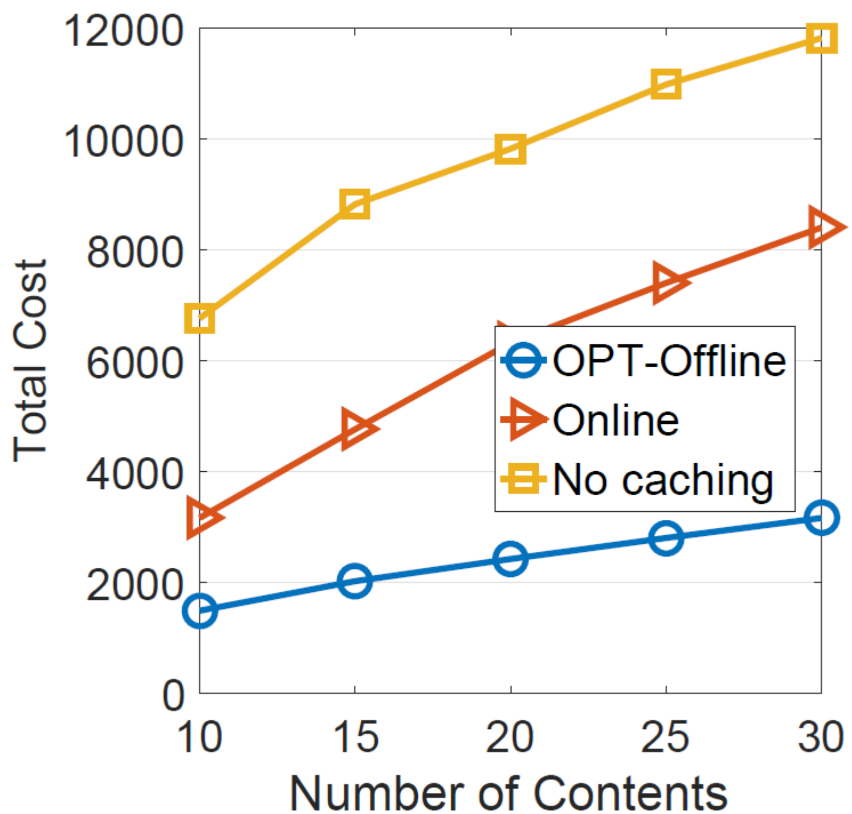
Evaluations

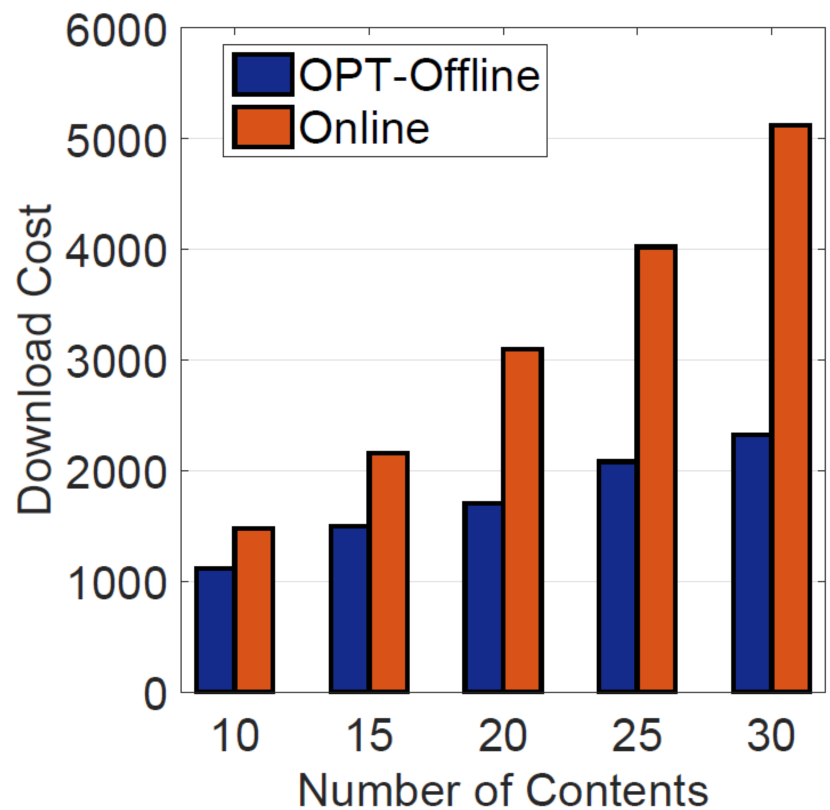
- 5 base stations
- Unlimited cache



Evaluations

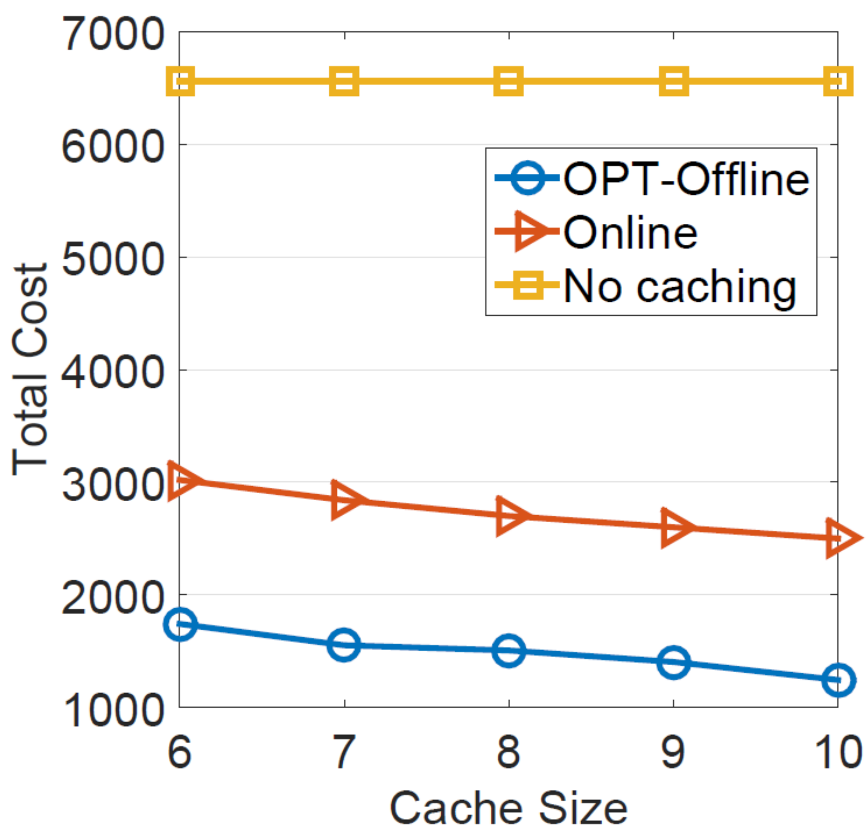
- 5 base stations
- Unlimited cache



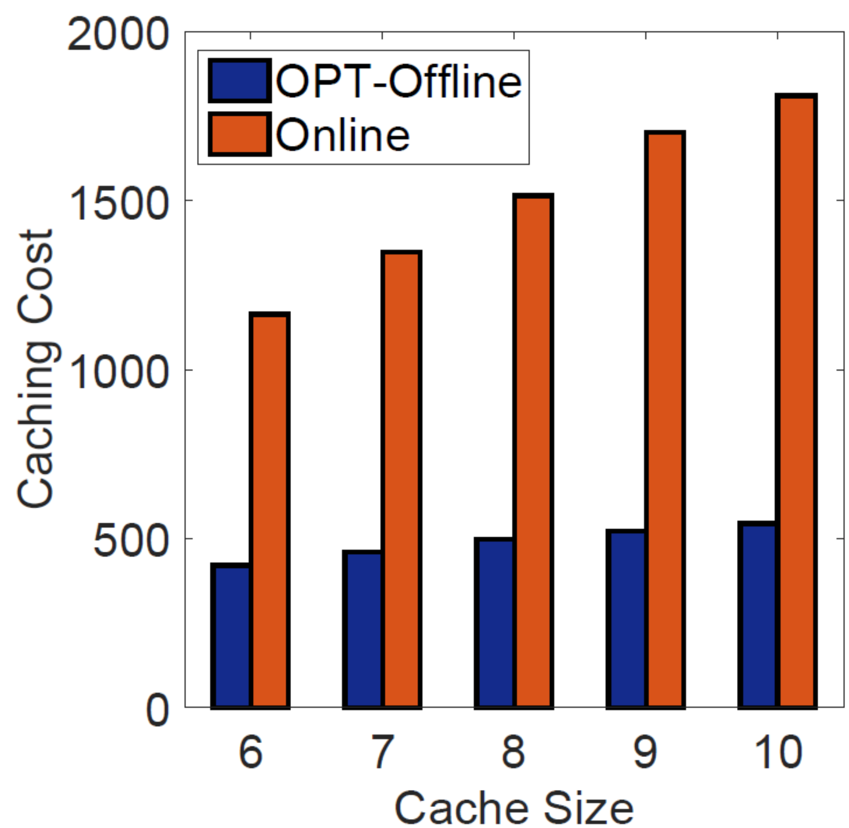


Evaluations

- 5 base stations
- Limited cache



- 10 base stations
- Limited cache



Conclusion

- Collaborative caching in cellular networks
- Offline caching
  - Integer programming
  - NP-complete problem
- Online caching algorithm
  - Using potential function
- Optimal scheme assuming complete knowledge about the future requests
  - Using network coding
  - Linear programming

Thank you