Time-Optimized Task Offloading Decision Making in Mobile Edge Computing

Ibrahim Alghamdi, Christos Anagnostopoulos and Dimitrios P. Pezaros

School of Computing Science
University of Glasgow, UK

Second UK Mobile, Wearable and Ubiquitous Systems Research Symposium
1st-2nd July 2019, Dept of Computer Science, University of Oxford, UK

Outline

1. Background
2. Computation Task Offloading
3. Previous Work
4. System Model & Problem Statement
5. Optimal Stopping Theory
6. Performance Evaluation
    1. Data Sets
    2. Performance Metrics & Assessment
7. Conclusions

Background[1]

- The recent advances in mobile devices
  - Example
  - Limitation
- Mobile Cloud Computing (MCC)
  - What is MCC?
  - Limitation
- Mobile Edge Computing (MEC)
  - What is MEC?
  - Names
    - iCloud
    - Fog Computing
    - **Mobile Edge Computing**
  - Use cases

[1] Pavel Mach and Zdenek Becvar. "Mobile edge computing: A survey on architecture and computation offloading". In: *IEEE Communications Surveys & Tutorials* 19.3 (2017), pp. 1628–1656.

Computation Offloading

- Dispatching intensive tasks to an external server, i.e., Cloud or an Edge server.[2]
  - Face/speech recognition;
  - Augmented, assisted or virtual reality;
  - Low latency applications, such as online gaming or remote desktop;
  - Big data analytic.
- The authors[3] demonstrated on a real MEC testbed that the reduction of **latency** up to 88% and **energy consumption** of the mobile device up to 93% can be accomplished by the computation/task offloading in MEC.

---

[2] Pavel Mach and Zdenek Becvar. "Mobile edge computing: A survey on architecture and computation offloading". In: *IEEE Communications Surveys & Tutorials* 19.3 (2017), pp. 1628–1656.

[3] Jakub Dolezal, Zdenek Becvar, and Tomas Zeman. "Performance evaluation of computation offloading from mobile device to the edge of mobile network". In: *2016 IEEE Conference on Standards for Communications and Networking (CSCN)*. IEEE. 2016, pp. 1–7.

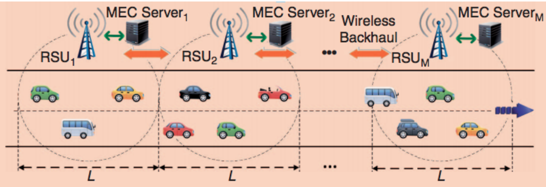
Challenges[4]

- Offloading Sequential Decision Making
  - Doing the tasks **either** locally **or** offloading them
  - Locally, Cloud, or at the Edge
  - **Which Edge server to offload?**

[4] Pavel Mach and Zdenek Becvar. "Mobile edge computing: A survey on architecture and computation offloading". In: *IEEE Communications Surveys & Tutorials* 19.3 (2017), pp. 1628–1656.

Previous Work

- Different from previous work, we focus on the **decision of when** to offload to an edge server, i.e., the selection of MEC servers/time.
- ST-CODA[5]: A spatial and temporal computation offloading decision algorithm.
- A predictive off-loading framework in vehicular networks.[6]

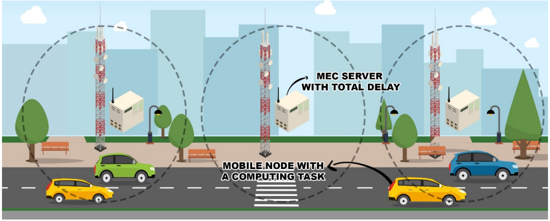


[5] Haneul Ko, Jaewook Lee, and Sangheon Pack. "Spatial and Temporal Computation Offloading Decision Algorithm in Edge Cloud-Enabled Heterogeneous Networks". In: *IEEE Access* 6 (2018), pp. 18920–18932.

[6] Ke Zhang et al. "Mobile-edge computing for vehicular networks: A promising network paradigm with predictive off-loading". In: *IEEE Vehicular Technology Magazine* 12.2 (2017), pp. 36–44.

System Model

- Mobile device
- MEC server
- Computing task with total delay D such that:
  - $D_{offload} < D_{local}$
- $D_{offload}$ delay includes:
  1. Transmission Time;
  2. Processing Time;
  3. Time spent to receive the processed data from MEC server to mobile device.



Problem Statement

The mobile node should find the best time instance t^* such that the expected total delay $\mathbb{E}[D]$ for the offloading is minimized, i.e., the optimal stopping time t^* achieves the essential infimum:

$$\operatorname*{ess\,inf}_t \mathbb{E}[D_t] \tag{1}$$

This problem is a sequential decision making solved based on the principles of the Optimal Stopping Theory (OST).

Optimal Stopping Theory

- Concerned with the problem of choosing the **best time instance** to take a given action based on sequentially observed random variables in order to minimize an expected cost.
- In our problem, we have two actions: **offload** or **continue observing**.
- We have two states: the user has offloaded the data, or still looking for a MEC server.
- **Abstraction**: we cast our offloading problem as a finite horizon OST problem, in which we know the upper bound n, i.e., the number of stages at which one may stop[7].

[7] Ke Zhang et al. "Mobile-edge computing for vehicular networks: A promising network paradigm with predictive off-loading". In: *IEEE Vehicular Technology Magazine* 12.2 (2017), pp. 36–44.

Optimal Offloading Rule

- We provide an **estimate** of the optimal offloading time.
- The optimal offloading time is determined by the scalar values $a_1, a_2, \ldots, a_n$ through which the mobile node decides either to offload or not:

Optimal Task Offloading Rule

Offload the data at the k-th MEC server if $D_k \leq a_k$; otherwise continue the observation if $D_k > a_k$.

Cont'd

Problem: Estimation of the scalar variable $\{a_k\}$.
The scalar variable a values are calculated once through *backward induction*.

$$a_k = \frac{1}{1+r}\left(a_{k+1}(1 - F_D(a_{k+1})) + \int_0^{a_{k+1}} udF_D(u)\right) \quad (2)$$

$$a_n = \frac{1}{1+r}\int_0^1 udF_D(u) = \frac{1}{1+r}\mathbb{E}[D], \quad (3)$$

$F_D(u) = P(D \leq u)$ is the cumulative distribution function of the total delay D.

From Model to Algorithm

Input: Decision scalar values $a_1, a_2, ..., a_n$
Output: Decision of which MEC server to offload

```
Offload ← FALSE
for k = 1 : n do
    if current total delay D_k ≤ a_k then
        MEC-Server ← k ;
        Offload ← TRUE; break;
    end if
end for
if Offload == FALSE then
    MEC-Server ← n;
end if
Offload tasks/data to the MEC-Server;
```

Experiment: Data Set

- We used the real dataset of taxi cabs' movements in Rome[8].
- The dataset contains GPS coordinates of 320 taxis collected over 30 days.
- For each movement, the mobile node is observing a server to check the the expected delay D.

Car id	Time	lat	long	Delay	Server
156	"2014-02-0100:00:00.73"	41.88	12.48	80.61	4
156	"2014-02-0100:00:16.47"	41.88	12.48	62.97	4
156	"2014-02-0100:00:30.70"	41.88	12.48	4.53	4
156	"2014-02-0100:00:45.30"	41.88	12.49	4.37	4
187	"2014-02-0100:00:01.14"	41.92	12.46	70.17	1
187	"2014-02-0100:00:16.15"	41.92	12.46	66.59	1
187	"2014-02-0100:00:30.81"	41.92	12.47	31.65	4

[8] Lorenzo Bracciale et al. *CRAWDAD dataset roma/taxi (v. 2014-07-17)*. Downloaded from https://crawdad.org/roma/taxi/20140717. July 2014. DOI: 10.15783/C7QC7M.

Evaluation

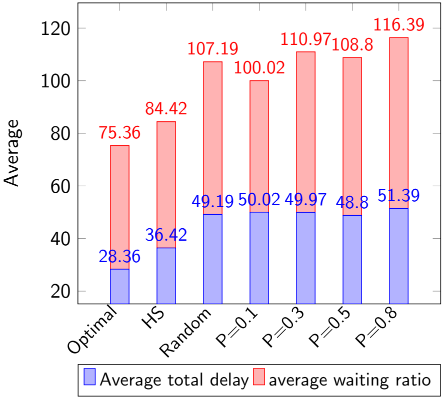


Figure: Average total delay and average waiting ratio of all models

Future Work and Conclusions

- We aim to consider the case where the number of the servers (times) is **unknown and not provided** to the mobile nodes.

The End

Thank you!

Contact: Ibrahim Alghamdi

Email address: i.alghamdi.1@research.gla.ac.uk