

Computation and Caching for Data of Things in Mobile Edge Computing

Anselme Ndikumana, and Choong Seon Hong*

Department of Computer Science and Engineering, Kyung Hee University
{anselme, cshong}@khu.ac.kr

Abstract

In Internet of Things (IoT), things or IoT devices are continuing to increase rapidly, and this increase is expected to continue in the coming years, where things will be in many services such as smart grid, smart agriculture, smart retails, smart home, smart hearth, smart factory, and many more. Furthermore, with things, more Data will be collected by enterprises in order to improve their daily operations. Therefore, for getting Data insights, it requires Data analytics, such as customer spending habits analysis, sales analysis, demography analysis, etc. In other words, getting business Data insights involves time-critical Data analytics. However, things have limited resources for Data storage, processing, and analyzing. On the other hand, moving Data of Things (DoT) to the remote cloud involves high end-to-end delay. Therefore, to overcome these challenges, we proposed computation and caching for DoT in Mobile Edge Computing (MEC). In order to reduce end-to-end delay, for services that require real-time processing, we formulate computation and caching for DoT as an optimization problem that consider MEC resources, where DoT will be cached, processed, analyzed at edge of mobile the network (at the base station) near where Data is created. The simulation results show that our proposal is easy to be implemented.

1. Introduction

Background: Things or IoT devices are continuing to increase rapidly, and in future “anything that can be connected will be connected” [1]. According to CISCO prediction, in year 2020, 50 billion of things will be connected to the internet, which is equivalent to 6 devices per person on planet [2]. Furthermore, in order to make intelligent decisions, things require Data collection, exchange, and processing. Consequently, in the future, we will have Big-Data of things, where Data of Things (DoT) will bring many benefits (through the use of Data insights) that may help in enhancing the quality of our daily services.

Challenges in IoT for dealing with DoT: Business Data insights come from Data analytics, which are time-critical and require enough computation resources. However, things have limited resources (e.g., power, CPU cycles, memory, I/O Data rate, etc.). Therefore, things will lead to increased use of cloud computing. On the other hand, computation and Data exchange between end-users’ devices and remote clouds are also increasing astronomically [3]. Consequently, combining DoT and Data generated by end-users’ devices will create a tsunami of computation and Data exchange between edge-devices (things and end-users’ devices) and remote cloud. Furthermore, accessing computing resources from a remote cloud will involve high end-to-end delay.

Our contributions: To overcome the above highlighted challenges, for services that require real-time processing, we propose computing and caching DoT in Mobile Edge Computing (MEC) network. We formulate joint computation and caching for DoT as an optimization problem that consider MEC resources (where MEC server is deployed at base station

[4]). DoT caching, processing, and analyzing at MEC server can significantly reduce end-to-end delay, and huge Data exchange between things and remote clouds. Moreover, it will help in future Big-Data and edge analytics through solving the problem of moving DoT to remote compute and returning computation outputs to the things.

The rest of the paper is structured as follows, Section 2 describes in details our system model, while Section 3 presents our computation and caching for DoT. Section 4 provides performance evaluation of our proposal. We conclude our paper with conclusion in Section 5.

2. System Model

In this section, we discuss in details our system model described in Fig. 1.

Smart things: We consider smart things as IoT devices which include but not limited to thermostats, cars, door locks, light bulbs, fridges and many more that can make intelligent decisions, and be connected to the Internet via a smart IoT gateway. We denote I as a set of things, where each thing $i \in I$ has limited resources. Things are connected to the smart IoT gateways, where smart IoT gateways are intelligent devices which can perform edges DoT computation and caching. We consider K as a set of smart IoT gateways, where each smart IoT gateway can be connected to various connections (wired or wireless). Here, we consider only wireless connection (mobile network), where each smart IoT gateway is connected to its nearest base station (BS) based on signal strength. Each smart IoT gateway $k \in K$ has both caching and computational resources that are sliced for being allocated to many things, where C_k and P_k are cache and computation capacities of smart IoT gateway $k \in K$,

respectively. However, when smart IoT gateway's resources are not enough to perform DoT computation and caching. Smart IoT gateway forwards requests to MEC network.

MEC network: As shown in Fig. 1, we consider an MEC network composed of a set of M MEC servers, where each MEC server $m \in M$ is attached to one BS. Each MEC server collaborates with other MEC servers through sharing resources at h_{mn} hop count distance from BS $m \in M$ to any other BS $n \in M$. Each MEC server m has both caching and computational resources that are sliced for being allocated to various smart IoT gateways, where C_m and P_m are cache and computation capacities of MEC server $m \in M$, respectively.

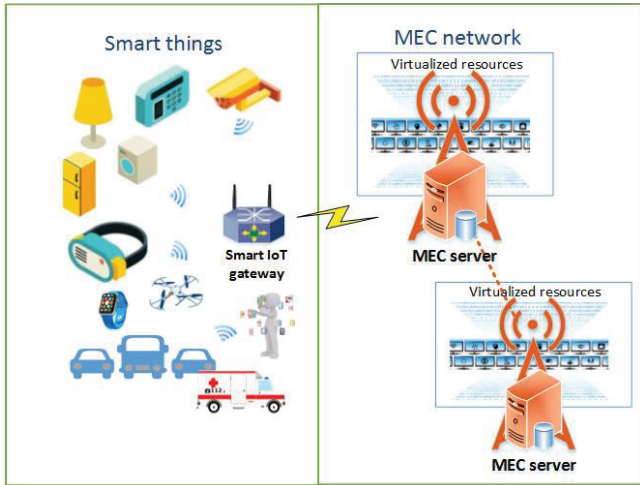


Fig. 1. System model

Task: We consider binary task in which a task is a single entity that cannot be partitioned. In the other word, each task from thing is either computed at smart IoT gateway or to the MEC server. We denote task $T_{ik} = (d_{ik}, \omega_{ik}, z_{ik})$ from thing $i \in I$ at smart IoT gateway $k \in K$. d_{ik} is the size of input Data in terms of bits from the thing i to the IoT gateway k , τ_{ik} is the task computation deadline, and z_{ik} is the computation workload or intensity in terms of CPU cycles.

3. Computation and caching for DoT

To reduce end-to-end delay and huge Data exchange between things and remote clouds, we proposed DoT computation and caching in MEC network. Here, we focus on task offloading from smart IoT gateways to MEC server.

3.1. Communication model

Task offloading from smart IoT gateway to MEC server involves the communication cost, where smart IoT gateway offloads tasks through the use of the wireless channel. For this reason, we define y_{km} as binary offloading decision variable (where $y_{km} = 1$ if smart IoT gateway offloads computation demand, 0 otherwise), which indicates whether or not smart IoT gateway k offloads the task to MEC server m via the wireless channel. Therefore, the spectrum efficiency for IoT gateway k is expressed as:

$$Y_{km} = \log_2 \left(1 + \frac{\rho_k |G_{km}|^2}{\sigma_k} \right), \quad (1)$$

where ρ_k is the transmission power of smart IoT gateway k , and $|G_{km}|^2$ is the channel gain between smart IoT gateway k and BS m , while σ_k is the power of Gaussian noise at smart IoT gateway k . The instantaneous Data rate for of smart IoT gateway k is given by:

$$r_{km} = y_{km} a_{km} B_{km} Y_{km}, \quad (2)$$

where a_{km} indicates the percentage of radio spectrum allocated to smart IoT gateway k , and B_{km} is the bandwidth of BS m . Based on instantaneous Data rate, the transmission delay for offloading the task to the MEC server m is expressed as:

$$\tau_{km} = \frac{y_{km} d_{ik}}{r_{km}}. \quad (3)$$

However, when the MEC server m does not have enough resources to satisfy the demands, BS m forwards request to any other BS n . Therefore, based on link capacity r_{mn} between BS m and BS n , the transmission delay is given by:

$$\tau_{mn} = \frac{y_{mn} d_{ik}}{r_{mn}}, \quad (4)$$

where y_{mn} is offloading decision variable from BS m to BS n at h_{mn} hop count distance.

3.2. Computation model

Computation at smart IoT gateway: We consider P_k as the total computation resource of smart IoT gateway k . The computation allocation for thing i at smart IoT gateway k is given by:

$$p_{ki} = P_k \frac{z_{ik}}{\sum_{j=1}^I z_{jk}}, \quad (5)$$

where $\sum_{j=1}^I z_{jk}$ is the computation demands of other things than i . The execution latency τ_{ki} of task $T_{ik} = (d_{ik}, \omega_{ik}, z_{ik})$ at IoT gateway k is given by:

$$\tau_{ki} = \frac{z_{ik} d_{ik}}{p_{ki}}. \quad (6)$$

Computation at MEC server: We consider that IoT gateway k can satisfy demands. However, when $p_{ki} \leq z_{ik}$ or $\omega_{ik} \leq \tau_{ki}$, IoT gateway k offloads demands to MEC server. The computation allocation for smart IoT gateway k at MEC server m is given by:

$$p_{km} = P_m \frac{y_{km} z_{ik}}{\sum_{g=1}^K z_{ig}}, \quad (7)$$

where $\sum_{g=1}^{K \setminus k} z_{ig}$ is the computation demands of other IoT gateways than k .

The execution latency τ_m is computed based on (6). Therefore, the total execution time τ_{em} for offloaded task by IoT gateway k at MEC server m is given by:

$$\tau_{em} = \tau_{km} + \tau_m. \quad (8)$$

However, when MEC server m does not have enough computation resource to meet the computation deadline, MEC server m offloads the task to another MEC server n . Therefore, the total execution time τ_{en} for offloaded task by

IoT gateway k at MEC server n is given by:

$$\tau_{en} = \tau_{km} + \tau_{mn} + \tau_n. \quad (9)$$

3.3. Caching model

In caching, DoT can be input Data d_{ik} and computed Data o_{ik} (computation output).

Caching at smart IoT gateway: We define x_{ki} as binary caching decision variable ($x_{ki} = 1$ if smart IoT gateway caches Data, 0 otherwise), which indicates whether or not smart IoT gateway k has to cache Data. Therefore, the cache allocation at smart IoT gateway is given by:

$$c_{ki} = x_{ki} C_k \frac{d_{ik} + o_{ik}}{\sum_{j=1}^I d_{jk} + o_{jk}}, \quad (10)$$

where $\sum_{j=1}^I d_{jk} + o_{jk}$ is the cache capacity needed by other things than i at smart IoT gateways K .

Caching at MEC server: We define x_{km} as binary caching decision variable which indicates whether or not MEC server m has to cache Data offloaded by smart IoT gateway k . The cache allocation at MEC server is given by:

$$c_{km} = x_{km} C_m \frac{d_{ik} + o_{ik}}{\sum_{g=1}^K d_{jg} + o_{jg}}, \quad (11)$$

where $\sum_{g=1}^{K \setminus k} d_{jg} + o_{jg}$ is the computation demands of other IoT gateways than k . However, when MEC server m does not have enough caching resource, MEC server m offloads the request to another MEC server n .

3.4. Problem formulation

For DoT computation and caching in MEC, we formulate the following optimization problem that aims at minimizing total delay $\tau_{ki} + y_{km} (\tau_{em} + y_{mn} \tau_{en})$, subject to the computation (P_m, P_n) and caching (C_m, C_n) resources constraints:

$$\begin{aligned} \min_{\{y \in [0, 1]\}} & (1 - y_{km})\tau_{ki} + y_{km} (\tau_{em} + y_{mn} \tau_{en}) \quad (12) \\ \text{Subject to:} & \sum_{k=0}^K y_{km} P_{km} \leq P_m \\ & \sum_{k=0}^K y_{km} C_{km} = C_m \\ & \sum_{k=0}^K y_{km} y_{mn} P_{kn} = P_n \\ & \sum_{k=0}^K y_{km} y_{mn} C_{kn} = C_n \\ & \sum_{k=0}^K y_{km} a_{km} \leq 1, \forall CP \quad i \in P \end{aligned}$$

4. Performance evaluation

In this section, we evaluate the performance of our proposal through numerical analysis, where gurobi solver [5] is used for solving our optimization problem (12).

In our experimental setup, we consider the number of smart IoT gateways $K=12$, while the total number of MEC server $M=5$. Furthermore, we consider caching demands in the range from 2 to 15 terabytes (TB), while computation demands in the range from 1 to 12 million instructions per second (MIPS). We consider that when smart IoT gateways do not have enough resources, they forward demand to MEC servers. In order to reduce end-to-end delay and satisfying the demands from smart IoT gateways, MEC servers collaborate among

themselves. The Fig.2 shows the total resources at MEC servers, while Fig.3 shows the total cache and computation resources allocated to all 12 smart IoT gateways in all MEC servers.

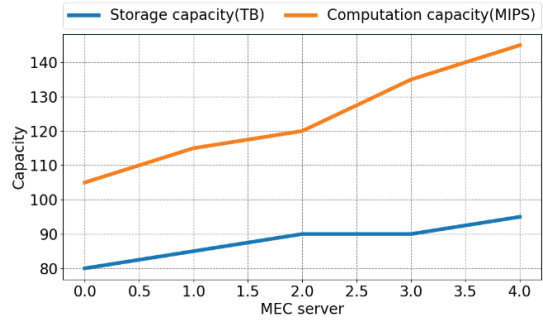


Fig. 2. Total resources at MEC servers

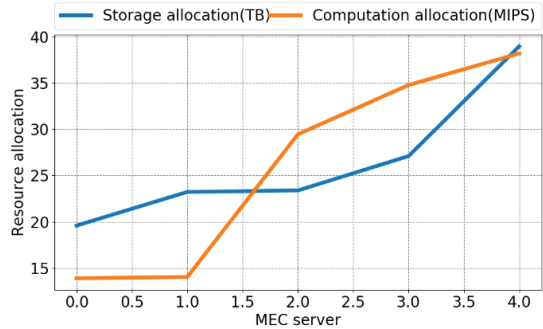


Fig. 3. Resource allocation at MEC servers

5. Conclusion

In this paper, we proposed computation and caching for DoT in MEC. We formulate computing and caching DoT as an optimization problem that aims at minimizing total delay. The simulation results show that our proposal is easy to be implemented. In the future, we aim to extend our proposal with more analysis.

6. Acknowledgement

This work was supported by Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIT) (No.2015-0-00557, Resilient/Fault-Tolerant Autonomic Networking Based on Physicality, Relationship and Service Semantic of IoT Devices). *Dr. CS Hong is the corresponding author.

7. References

- [1]. <https://www.forbes.com/sites/jacobmorgan/2014/05/13/simple-explanation-internet-things-that-anyone-can-understand/#7e0e244c1d09>
- [2]. <https://blogs.cisco.com/cle/10-predictions-for-the-future-of-the-internet-of-things>
- [3]. Ndikumana A, Ullah S, LeAnh T, Tran NH, Hong CS. Collaborative cache allocation and computation offloading in mobile edge computing. In Network Operations and Management Symposium (APNOMS), 2017 19th Asia-Pacific 2017 Sep 27 (pp. 366-369). IEEE.
- [4]. Hu YC, Patel M, Sabella D, Sprecher N, Young V. Mobile edge computing—A key technology towards 5G. ETSI White Paper. 2015 Sep 5;11(11):1-6.
- [5]. <http://www.gurobi.com>