URSI GASS 2021, Rome, Italy, 28 August - 4 September 2021

A Partial Computation Offloading Strategy for Microcell-femtolet based Future Generation Edge-Cloud Network

Anwesha Mukherjee*[1]
(1) Department of Computer Science, Mahishadal Raj College,
Mahishadal, Purba Medinipur, West Bengal, PIN - 721628, India

Abstract

Fast and low power offloading is an emerging area of interest for future generation mobile network. This paper proposes a partial computation offloading strategy for microcell-femtolet based edge-cloud network. In the microcell-femtolet network microcells contain femtolets inside their coverage area to provide good communication and computation facilities to the user devices at indoor region. An indoor user is connected with the femtolet and an outdoor user is connected with the microcell base station in the microcell-femtolet network scenario. When a computation has to be partially offloaded, the task is partitioned into two segments, one is locally executed and the other one is offloaded to the femtolet or edge server according to the user's presence at indoor or outdoor region. The theoretical and experimental results illustrate that the proposed partial offloading scheme reduces the latency and power consumption of the mobile device than the cloud based approach.

1 Introduction

The smart phone users desire computation and communication services simultaneously at low power and low latency. For faster service provisioning edge computing comes into the scenario for future generation cellular network and Internet of Things (IoT), by bringing the storage and computational resources at the edge of the network [1, 2]. Usually the edge servers are placed along with the base stations [1]. In a fifth generation (5G) small cell network, due to poor signal strength at indoor regions, small cell base stations such as femtocell base stions (FBSs) and picocell base stations (PBSs) are used inside the coverage area of the large cell base stations i.e. macrocell base stations (MBSs) or microcell base stations (MiBSs) [3]. For joint computational and communication services, small cell with computation and storage abilities such as Small cell cloud enhanced eNodeB (SCceNB) and femtolet, has been discussed in [4]. The remote cloud based offloading [5, 6] increases the latency [7, 8], which the use of edge/fog computing [1, 9] can resolve. In this paper, a partial computation offloading strategy is proposed for microcell-femtolet based network, where femtolets [4] are used inside the microcell to provide good coverage at indoor region. The objectives of the proposed work are summarized as:

- A task containing a number of jobs will be offloaded. Therefore, in partial offloading the decision making regarding which job will be offloaded and which one will be locally executed, is a major concern.
- As the user device is a mobile device, the user will be at indoor or outdoor region. Therefore, it is another challenge to provide the user partial computation offloading at minimal latency and power consumption despite its presence at indoor or outdoor region.

To attain the objectives, the contributions of this paper are:

- A partial computation offloading strategy is proposed for microcell-femtolet based edge-cloud network. When a computation has to be partially offloaded, the task is decomposed into two segments: one containing the jobs to be locally executed and another segment containing the jobs to be offloaded. Based on the deadline, amount of computation to be performed, and inter-dependency among the jobs, it is decided whether to locally execute or offload a job.
- If the requesting device is registered under a femtolet at indoor region, the computation is partially offloaded to the femtolet. Otherwise, if the user is under the MiBS, the computation is partially offloaded to the edge server. If the user is disconnected before delivering the result, the edge server/femtolet sends the result to the cloud along with the device ID. The cloud sends the result to the mobile device via a push notification message. Theoretical and experimental results show that the proposed method provides partial computation offloading at low latency and low power consumption of the mobile device than the existing schemes.

Rest of this paper is organized as: Section 2 discusses the proposed partial computation offloading method, in Section 3 the calculation of offloading latency and power consumption of the user device are illustrated, Section 4 discusses the theoretical and experimental results, and finally Section 5 concludes the paper.

2 Partial Computation Offloading for Microcell-femtolet Network

The microcell-femtolet network contains microcells, where each microcell contains femtolets to provide good coverage at indoor region. The users present at the indoor regions use femtolet to get communication services as well as they can offload their computations inside the femtolet under which the respective mobile device is registered. For the users at outdoor regions, the MiBS provides the communication service, and the users can offload their computation inside the edge servers attached with the MiBS. In this work, the case of partial computation offloading is only considered. Firstly, Algorithm 1 is proposed for making decision regarding segmentation of the task containing a number of jobs. Then Algorithm 2 is proposed for offloading device selection for indoor and outdoor users.

Algorithm 1 Segmentation of a task into jobs to be locally executed and offloaded

```
Input: Computational Task $T$
Output: Segment $S_l$ containing jobs to be locally executed, Segment $S_o$
    containing jobs to be offloaded
decompose $T$ into a set of jobs $\{J_1, J_2., , , , J_n\}$ where $n$ is the number
    of jobs in task $T$;
partition the jobs into $p$ segments $\{S_1, S_2, ..., S_p\}$ based on their inter-
    dependency;
for each segment $S_i$ where $1 \leq i \leq p$ do
    refer $S_i$ as a task $T_i$
    if $T_i$ has soft deadline then
        if exhaustive computation is required then
            put all jobs of $S_i$ into segment $S_o$
        else
            put all jobs of $S_i$ into segment $S_l$
        end if
    else
        if exhaustive computation is required then
            if device is able to execute $T_i$ then
                put all jobs of $S_i$ into $S_l$
            else
                ask user to extend the deadline
            end if
        else
            put all jobs of $S_i$ into $S_l$
        end if
    end if
end for
```

Algorithm 2 Offloading device selection based on user's presence at indoor and outdoor regions

```
Input: Segment $S_o$ to be offloaded
Output: Result after execution
if the user is at indoor region then
    offload $S_o$ to the femtolet under which the user device is registered;
else
    offload $S_o$ to the edge server attached with the microcell base sta-
    tion under which the user device is registered;
end if
if the user is disconnected before delivering the result then
    edge server/femtolet delivers the result to the cloud along with the
    device ID;
    the cloud sends the result to the device via a push notification mes-
    sage;
end if
```

3 Latency and Power Consumption of Mobile Device during Offloading

The offloading latency is calculated as the sum of the data transmission latency (L_{tr}) and computation execution latency (L_{ex}), given as,

$$L_{off} = L_{tr} + L_{ex} \tag{1}$$

The power consumption of the user device during the period of offloading is given as,

$$P_{off} = P_{tr} + P_{ex} \tag{2}$$

where P_{tr} and P_{ex} denotes the power consumption of the user device during the period of data transmission and computation execution respectively. If the user is at indoor region, then the data transmission takes place from the mobile device to the femtolet and the result is sent from the femtolet to the mobile device. In that case, the data transmission latency is $((1+u_f) \cdot (D_t/R_{mf}) + (1+d_f) \cdot (D_r/R_{fm}))$, where u_f and d_f are the link failure rate, D_t and D_r are the amount of data transmitted, and R_{mf} and R_{fm} are the rate of data transmission from mobile device to femtolet, and from femtolet to mobile device respectively. The power consumption of the user device during this period is given as $((P_t \cdot L_{tf}) + (P_r \cdot L_{rf}))$, where $L_{tf} = ((1+u_f) \cdot (D_t/R_{mf}))$ and $L_{rf} = ((1+d_f) \cdot (D_r/R_{fm}))$, and P_t and P_r are the power consumption of the user device per unit time in data transmission and reception modes. The computation execution latency for the device registered under the femtolet is $((A_c/S_f) + L_{qf})$, where A_c is the amount of computation executed, S_f is the speed of the femtolet, and L_{qf} is the queuing latency for the femtolet respectively. The power consumption of the user device during this period is given as $(P_i \cdot L_{cf})$, where $L_{cf} = ((A_c/S_f) + L_{qf})$ and P_i is the power consumption of the user device per unit time in idle mode. If the user is at outdoor region, then the data transmission takes place from the mobile device to the edge server and the result is sent from the edge server to the mobile device. In that case, the data transmission latency is $((1+u_e) \cdot (D_t/R_{me}) + (1+d_e) \cdot (D_r/R_{em}))$, where u_e and d_e are the link failure rate, D_t and D_r are the amount of data transmitted, and R_{me} and R_{em} are the rate of data transmission from mobile device to edge server, and from edge server to mobile device respectively. The power consumption of the user device during this period is given as $((P_t \cdot L_{te}) + (P_r \cdot L_{re}))$, $L_{te} = ((1+u_e) \cdot (D_t/R_{me}))$ and $L_{re} = ((1+d_e) \cdot (D_r/R_{em}))$. The computation execution latency for the user registered under the microcell is $((A_c/S_e) + L_{qe})$, where A_c is the amount of computation executed, S_e is the speed of the edge server, and L_{qe} is the queuing latency for the edge server respectively. The power consumption of the user device during this period is given as $(P_i \cdot L_{ce})$, where $L_{ce} = ((A_c/S_e) + L_{qe})$. Let assume R_1 number of requests arrive for partial computation offloading to the femtolet and R_2 number of requests arrive for partial computation offloading to the edge server respectively. Let the offloading latency to the femtolet for a request r is L_{fem_r}, and if the offloading takes place to the edge server, the latency is L_{edge_r}. The

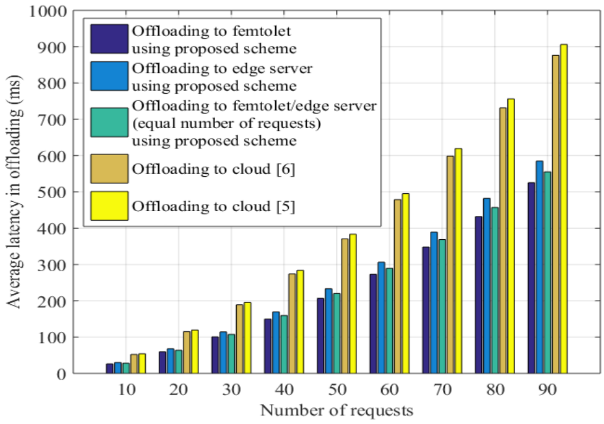

Figure 1. Latency in partial computation offloading using proposed and existing frameworks

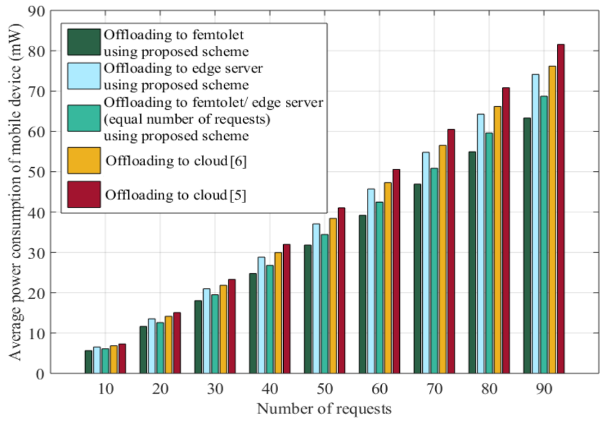

Figure 2. Power consumption of the mobile device in partial computation offloading using proposed and existing frameworks

Table 1. Total latency including local execution and offloading

Code	Segment	Locally executed/ Offloaded	Latency while partial offloading is done to edge device	Latency while partial offloading is done to cloud	Reduction in latency using edge device
Binary search on a list of items	Sorting	Offloaded	5.582 sec (3.154 sec (offloading)+ 2.428 sec (local))	8.082 sec (5.654 sec (offloading) + 2.428 sec (local))	30.93%
	Finding an item	Locally executed			
Adjoint of a matrix	Co-factor calculation	Offloaded	11.954 sec (6.196 sec (offloading)+ 5.758 sec (local))	16.016 sec (10.258 sec (offloading)+ 5.758 sec (local))	25.36%
	Transpose	Locally executed			
Copy the content of a file and encrypt	Copying a file	Locally executed	9.514 sec (3.884 sec (local)+ 5.63 sec (offloading))	13.521 sec (3.884 sec (local)+ 9.637 sec (offloading))	29.64%
	Encryption	Offloaded			

Table 2. Power consumption of the mobile device in the total period including local execution and offloading

Code	Segment	Locally executed/ Offloaded	Power consumption of the mobile device during total period if partial offloading is done to edge device	Power consumption of the mobile device during total period if partial offloading is done to cloud	Reduction using edge device
Binary search on a list of items	Sorting	Offloaded	0.614 W	0.889 W	30.93%
	Finding an item	Locally executed			
Adjoint of a matrix	Co-factor calculation	Offloaded	1.315 W	1.762 W	25.37%
	Transpose	Locally executed			
Copy the content of a file and encrypt	Copying a file	Locally executed	1.047 W	1.487 W	29.59%
	Encryption	Offloaded			

average offloading latency is therefore given as,

$$L_{offp} = \frac{\sum_{r\in R_1} L_{fem_r} + \sum_{r\in R_2} L_{edge_r}}{R_1 + R_2} \tag{3}$$

Let the power consumption of the mobile device while offloading computation for a request r to the femtolet is P_{fem_r} and while offloading computation for a request r to the edge server is P_{edge_r}, then the average power consumption of the user device during offloading considering both the cases is given as,

$$P_{offp} = \frac{\sum_{r\in R_1} P_{fem_r} + \sum_{r\in R_2} P_{edge_r}}{R_1 + R_2} \tag{4}$$

If the user gets disconnected and the cloud delivers the result via a push notification message, then the data transmission latency from the femtolet/edge server to the cloud has to be considered and in that case the result will be sent to mobile device from the cloud. Therefore, the data transmission latency from the femtolet/edge server to the mobile device will not be considered. Instead of that the data transmission latency from the cloud to the mobile device will be considered. If the femtolet sends the result to the cloud, the data transmission latency is $((1+u_{fc})\cdot(D_{rd}/R_{fc}))$, where u_{fc} is the link failure rate from femtolet to the cloud, D_{rd} is the amount of data containing result and device ID, and R_{fc} is the rate of data transmission from femtolet to the cloud. The power consumption of the user device during this period is given as $(P_i \cdot L_{fc})$, where $L_{fc} = ((1+u_{fc})\cdot(D_{rd}/R_{fc}))$. If the edge server sends the result to the cloud, the data transmission latency is $((1+u_{ec})\cdot(D_{rd}/R_{ec}))$, where u_{ec} is the link failure rate from edge server to the cloud, D_{rd} is the amount of data containing result and device ID, and R_{ec} is the rate of data transmission from edge server to the cloud. The power consumption of the user device during this period is given as $(P_i \cdot L_{ec})$, where $L_{ec} = ((1+u_{ec})\cdot(D_{rd}/R_{ec}))$. The data transmission latency from the cloud to the mobile device is $((1+d_{cm})\cdot(D_r/R_{cm}))$, where d_{cm} is the link failure rate from cloud to the mobile device, D_r is the amount of data containing result, and R_{cm} is the rate of data transmission from cloud to the mobile device. The power consumption of the mobile device during this period is given as $(P_r \cdot L_{cm})$, where $L_{cm} = ((1+d_{cm})\cdot(D_r/R_{cm}))$.

4 Results and Discussion

In this section, the performance of the proposed strategy is evaluated using theoretical analysis and experimental analysis.

Theoretical analysis: For theoretical analysis MATLAB2015 is used. In Fig.1 the average latency in partial computation offloading using the proposed and existing strategies are presented. In Fig.2 the average power consumption of the mobile device during offloading using the proposed and existing strategies are presented. In the proposed scheme partial computation offloading is performed either to the femtolet or edge server. It is considered that for half of the requests, offloading takes place to the femtolet, and for the rest half of the requests, offloading takes place to the edge server. The average offloading latency for the proposed scheme is calculated using equation (3) and the average power consumption of the mobile device during offloading is calculated using equation (4). To compare with the proposed approach, the latency in case of partial offloading to the cloud [5, 6] is also calculated. This is observed that the offloading to the femtolet using the proposed scheme reduces the latency by approximately 40-50% than offloading to the cloud. This is also observed that the offloading to the edge server using the proposed scheme reduces the latency by approximately 30-40% than offloading to the cloud. This is also observed from the offloading to the femtolet/edge server (considering equal number of requests) using the proposed scheme reduces the latency by approximately 35-45% than offloading to the cloud respectively. This is observed that the offloading to the femtolet using the proposed scheme reduces the power consumption of the mobile device by approximately 20% than offloading to the cloud. This is also observed that the offloading to the edge server using the proposed scheme reduces the power consumption of the mobile device by approximately 2-10% than offloading to the cloud. This is also observed that the offloading to the femtolet/edge server (considering equal number of requests) using the proposed scheme reduces the power consumption of the mobile device by approximately 10-15% than offloading to the cloud.

Experimental analysis: In experimental analysis a mobile phone with 2GB RAM is used as the user device and the device is connected to an edge device having 4 GB RAM and 250 GB HDD. To compare the proposed scheme with the cloud based scheme a VM instance of 3.75 GB RAM and 250 GB HDD has been taken in Google Cloud Platform. For experimental analysis, three codes are partially offloaded. Each of the three codes are divided into two segments, which execute separately. However, to get the final result the output of one segment has to be provided as input to another segment. Here, the codes of binary search, finding adjoint of a matrix, and copy and encrypt the content of a file, are considered. The latency in local execution and offloading while using the proposed scheme and cloud based scheme are presented in Table 1. The power consumption of the mobile device during the total period is presented in Table 2. From the experimental results this is observed that partial offloading to the edge device reduces the latency and power consumption of the user device by approximately 25-31% than the cloud based partial offloading scheme.

The theoretical and experimental results show that partial offloading to the edge device using the proposed scheme reduces the latency and power consumption of the user device. Thus, the proposed approach can be referred as a fast and green offloading scheme.

5 Conclusion

In this paper a partial computation offloading strategy is proposed for microcell-femtolet based future generation edge-cloud network. This is assumed that a mobile device is either connected with a femtolet if the user is at indoor region or the MiBS if the user is at outdoor region. When a computation has to be partially offloaded, the task is partitioned into two segments, each containing jobs. Based on the deadline, computation intensity and inter-dependency among the jobs it is decided whether to locally execute or offload a segment. For offloading a segment either the femtolet or the edge server attached with the MiBS is used, based on the user's presence at indoor or outdoor region. This is observed from the theoretical and experimental results that partial offloading to edge device using the proposed scheme reduces the latency and power consumption of the user device with respect to the cloud based partial offloading scheme.

References

[1] K. Peng, V. Leung, X. Xu, L. Zheng, J. Wang, and Q. Huang, "A survey on mobile edge computing: focusing on service adoption and provision," *Wireless Communications and Mobile Computing*, vol. 2018, 2018.

[2] Y. Liu, C. Yang, L. Jiang, S. Xie, and Y. Zhang, "Intelligent edge computing for iot-based energy management in smart cities," *IEEE Network*, vol. 33, no. 2, pp. 111–117, 2019.

[3] P. Deb, A. Mukherjee, and D. De, "A study of densification management using energy efficient femto-cloud based 5g mobile network," *Wireless Personal Communications*, vol. 101, no. 4, pp. 2173–2191, 2018.

[4] A. Mukherjee and D. De, "Femtolet: A novel fifth generation network device for green mobile cloud computing," *Simulation Modelling Practice and Theory*, vol. 62, pp. 68–87, 2016.

[5] L. Jiao, R. Friedman, X. Fu, S. Secci, Z. Smoreda, and H. Tschofenig, "Cloud-based computation offloading for mobile devices: State of the art, challenges and opportunities," in *2013 Future Network & Mobile Summit*. IEEE, 2013, pp. 1–11.

[6] A. Mukherjee and D. De, "Low power offloading strategy for femto-cloud mobile network," *Engineering Science and Technology, an International Journal*, vol. 19, no. 1, pp. 260–270, 2016.

[7] A. Mukherjee, D. De, and S. K. Ghosh, "Power-efficient and latency-aware offloading in energy-harvested cloud-enabled small cell network," in *2020 XXXIIIrd General Assembly and Scientific Symposium of the International Union of Radio Science*. IEEE, pp. 1–4.

[8] D. De, A. Mukherjee, and D. G. Roy, "Power and delay efficient multilevel offloading strategies for mobile cloud computing," *Wireless Personal Communications*, pp. 1–28, 2020.

[9] J. Das, A. Mukherjee, S. K. Ghosh, and R. Buyya, "Spatio-fog: A green and timeliness-oriented fog computing model for geospatial query resolution," *Simulation Modelling Practice and Theory*, vol. 100, p. 102043, 2020.