View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by Explore Bristol Research

Singh, R., Armour, S. M. D., Khan , A., Sooriyabandara, M., & Oikonomou, G. (2020). *Identification of the key parameters for computational offloading in Multi-access Edge computing*. Paper presented at IEEE Cloud Summit 2020, . https://doi.org/10.1109/IEEECloudSummit48914.2020.00026

Publisher's PDF, also known as Version of record

Link to published version (if available):
10.1109/IEEECloudSummit48914.2020.00026

Link to publication record in Explore Bristol Research
PDF-document

University of Bristol - Explore Bristol Research

General rights

This document is made available in accordance with publisher policies. Please cite only the published version using the reference above. Full terms of use are available: http://www.bristol.ac.uk/red/research-policy/pure/user-guides/ebr-terms/

Identification of the Key Parameters for Computational Offloading in Multi-Access Edge Computing

Raghubir Singh[†], Student Member, IEEE, Simon Armour[†],
Aftab Khan[‡], Mahesh Sooriyabandara[‡], and George Oikonomou[†]
Communication Systems & Networks Research Group, University of Bristol, Bristol, UK[†]
Department of Electrical and Electronic Engineering, University of Bristol, UK[†]
Telecommunications Research Laboratory, Toshiba Research Europe Limited, Bristol, UK[‡]
E-mail: raghubir.singh@bristol.ac.uk

Abstract—**Computational offloading is a strategy by which mobile device (MD) users can access the superior processing power of a Multi-Access Edge Computing (MEC) server network. This paper investigates the impact of CPU workloads (on both the user and server-side) on overall processing times and energy consumption as well as We provide a comprehensive mathematical model using two applications of varying complexity are tested on a range of cases. Our findings show that the relationship between the CPU workloads on the MD and MEC server and the link speed between them are the crucial parameters that determine the success of offloading in the MEC network. We demonstrate that a certain threshold of link speed is required for shorter completion times by offloading, and the MD CPU workload determines it. Furthermore, MD energy usage can be reduced considerably by offloading for varying complexity applications provided a sufficiently link speed is available to the MEC network.**

Keywords—Computation Offloading, Multi-Access Edge Computing, CPU Workloads, Energy Usage.

Nomenclature

γ^{UL}	Uplink speed between an MD and a MEC
Π	Proportion of data size reduction after the data processed
T^{DL}	Receiving time to send the processed data from a MEC
T_c^{Total}	Total offloading time on MEC
α	Processor speed of an MD
β	Processor speed of a MEC server
γ^{DL}	Downlink speed between a MEC and an MD
λ^c	Complexity of an application on a MEC
λ^m	Complexity of an application on an MD
C^{MD}	Total number of instructions to compute given computational data
E^{DL}	Total uplink energy consumption
E^{Idle}	Total Idling energy of an MD
E^{MD}	Total energy consumption to process a job on an MD
E_m^{Total}	Total energy consumption of a MD
E^{UL}	Transmit energy consumption of an MD to send the data to a MEC
L^{MD}	Given CPU workload on an MD
L^{MEC}	Given CPU workload on a MEC
P^{idle}	Idling power rating of an MD
P^{MD}	Power rating of an embedded processor on an MD
P^{rec}	Power rating of receiving computational data from a MEC
P^{send}	Power rating of an MD to send the computational data
T^{MD}	Total time to compute the data on an MD
T^{MEC}	Processing time of computational data on a MEC
T^{UL}	Transmission time to transfer the computational data to a MEC
X_j	Computational data on an MD

I. Introduction

Computational offloading seeks to leverage the superior processing power offered by server-based networks. This concept gained popularity with the advances made in the cloud computing paradigm [1]. While the cloud computing paradigm has demonstrated the potential of computational offloading, a key limitation in this approach is the remote location of the cloud-based services that results in latency and low bandwidth issues [2]. State of the art in computational offloading is focusing on bringing the computational capability closer to a user, and this paradigm is commonly known as Multi-Access Edge Computing (MEC). The primary motivation of MEC is that it significantly cuts the computational times of transmission and receiving of data from a MEC server network [3], [4].

With the growing number of mobile device users, the MEC facilities are expected to ration for requests that they receive for offloading jobs. The decision to entertain a job for offloading depends on several parameters, including mobile devices' operating conditions (i.e., state of charge, workload) and the operating conditions of the MEC servers. Several studies in recent years have claimed significant savings in both task completion time and energy usage by the mobile device (MD) offloading to both Mobile Cloud Computing and Multi-Access Edge Computing (MEC) environments [5]–[9]. However, in such studies, little attention has been paid to the importance of various parameters involved in decision-making in computational offloading problems. Previous work from the authors investigated the impact of mobile devices' computational power (MDs) and MEC servers on job completion times [10]. The work did not take into account CPU workloads and the link speed that connect MD to MEC servers. Such parameters are of importance while making informed decisions regarding computational offloading and are a subject of this paper. This paper investigates key parameters that are important to take into consideration for making offloading decisions.

This paper presents a novel mathematical model which incorporates CPU workloads (as percentages, i.e. where 100% is where the maximum number of instructions can that be executed per second at constant code efficiency). No previous study has considered this parameter in both on-board and server-side processors. The main contributions from this paper are twofold:

- investigation of the impact of varying the CPU workloads of an MD and a MEC server on the computational processing time and energy consumption;
- definition of minimal link speeds required for successful offloading with varying MD and MEC server CPU workloads.

The remainder of the paper is organised as follows: Section II presents a mathematical model to calculate total completion time and energy consumption for a given number of jobs. Section III demonstrates the model on two different arrangements of mobile devices and the MEC network. Section IV provides discussion and insights that are learned from our modeling work. The paper concludes in Section V where some challenges and future research directions are discussed.

II. Mathematical Model

Let $u_{j,c}$ be the binary variable that models the offloading of a job j on a MEC c , respectively. The binary variable is defined as follows:

$$u_{j,c} = \begin{cases} 1 & \text{if job } j \text{ is offloaded to } c, \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

The offloading decision-making strategy deals with determining, for a given computation task j, whether to compute it locally on an MD or leverage the computing facilities offered by a MEC server c. In other words, the computational offloading decision is to determine the set: $\{u_{j,c} : j \in J, c \in C\}$, that models the decision on each job that need to be processed. In the following subsections, mathematical relations are derived that model the dynamics of the computational offloading.

A. Computational processing time on a mobile device

Considering a mobile device 'MD', let X_j denote the computational data (in bits) to be processed. Let λ^m denote the complexity of the application that processed the data (in bits per instruction). The total numbers of instructions, C^{MD}, are calculated using the following Equation:

$$\mathrm{C}^{\mathrm{MD}} = \frac{\mathrm{X}_j}{\lambda^m} \tag{2}$$

Let α be the on-board processor speed of MD (in instructions per second). The time to compute the job on MD is given as follows:

$$T^{\mathrm{MD}} = \frac{\sum_{j \in J} X_j(u_{j,c})}{(1 - \frac{L^{\mathrm{MD}}}{100}) \times \alpha} \tag{3}$$

Equation (3) provides a relationship between the completion time, computational data, MD load and the processing speed of the device. From this Equation, we note that the completion time is directly proportional to the computational data and mobile device loading. In contrast, completion time is inversely proportional to the processing speed of the device.

B. Local Energy Consumption

Processing a computational task on an MD will require a certain amount of energy. Let P^{MD} be the power rating of the embedded processor. That energy consumption can be quantified as follows:

$$E^{\mathrm{MD}} = P^{\mathrm{MD}} \times T^{\mathrm{MD}} \tag{4}$$

where T^{MD} is obtained from the solution of Equation (3).

1) Computational processing time on a MEC: Let X_j^c denote the computational data (in bits) as the size of input data that needs to be processed from an application that is running on a MEC at λ^c (in bits per instruction). Let β be the on-board processor speed of MEC c (in instructions per second). The computational time to process a job on a MEC server is given as follows:

$$T^{\text{MEC}} = \frac{\sum_{j\in J} X_j u_{j,c}}{(1-\frac{L^{\text{MEC}}}{100})\beta\lambda^c} \quad (5)$$

Let γ^{UL} be the up-link speed (in bits/second). The following Equation gives the time to send the job over the link.

$$T^{\text{UL}} = \frac{\sum_{j\in J_i} u_{j,c} X_j}{\gamma^{\text{UL}}} \quad (6)$$

Let γ^{DL} be the downlink speed (in bits/second). The links between the MD and MEC are symmetric, which means MD can send and receive data to and from MEC at the same rate. The receiving time of the processed data can be calculated as follows:

$$T^{\text{DL}} = \Pi \frac{\sum_{j\in J_i} u_{j,c} X_j}{\gamma^{\text{DL}}} \quad (7)$$

where Π ($0 \leq \Pi \leq 1$) is defined as the proportion of data size reduction after a job is processed. Furthermore, without the loss of generality we assume that the uplink and downlink speed are equal i.e. $\gamma^{\text{UL}} = \gamma^{\text{DL}} = \Gamma$.

Equations (5), (6) and (7) can be represented as:

$$T_c^{\text{Total}} = \underbrace{\frac{\sum_{j\in J} X_j u_{j,c}}{(1-\frac{L^{\text{MEC}}}{100})\beta\lambda^c}}_{\text{MEC Processing Time}} + \underbrace{\frac{\sum_{j\in J_i} u_{j,c} X_j}{\gamma^{\text{UL}}}}_{\text{Transmission Time}} + \underbrace{\frac{\sum_{j\in J_i} \Pi X_j}{\gamma^{\text{DL}}}}_{\text{Receiving Time}} \quad (8)$$

C. Offload Energy Consumption

Here we are only concerned with the energy consumption of the mobile device. Let P^{send} and P^{rec} denote the power rating of the MD to send and receive the request for offloading the job (in W) respectively. The total energy consumption for this step is given as:

$$\left(E^{\text{UL}}, E^{\text{DL}}\right) = \left(P^{\text{send}} \times T^{\text{UL}}, P^{\text{rec}} \times T^{\text{DL}}\right) \quad (9)$$

Let P^{idle} denote the power rating of the MD (in W) when it is in the idle state and is waiting to receive the solution of the computational task back from the MEC. The energy consumption of the idle state is given as follows:

$$E^{\text{idle}} = P^{\text{idle}} \times T^{\text{MEC}} \quad (10)$$

Equations 9 and 10 can be represented as:

TABLE I: Parameters used in simulations to demonstrate the mathematical model

Entity	Parameter	Value	Unit
Jobs Size	X_{j}	1-20	MB
MD	α	3.60×10^9	IPS
MEC	β	1.40×10^{11}	IPS
Application 1	C^{MD}	3.7×10^9	Ins/MB
Application 2	C^{MD}	3.7×10^8	Ins/MB
Network	Γ	20	Mbps

$$E_m^{\text{Total}} = \underbrace{P^{\text{send}} \times T^{\text{UL}}}_{\text{Transmit energy consumption}} + \underbrace{P^{\text{idle}} \times T^{\text{MEC}}}_{\text{Idling energy consumption}} + \underbrace{P^{\text{rec}} \times T^{\text{DL}}}_{\text{Receiving energy consumption}} \quad (11)$$

III. Numerical Results

In this section, we demonstrate the use of the mathematical on two applications. The parameters for this simulation are provided in Table I. The processor speeds were taken from [11].

The values used for Application 1 are taken from [11]. Application 2 had a 10-fold reduction in the numbers of instructions generated, following the ratio proposed in [12] for MCC to distinguish between the high- and low-complexity applications considered for offloading.

The authors of [13] quote three values for power ratings (energy usage) by a mobile device: P^{MD} is the energy consumption of a mobile device while computing (0.9 W), P^{idle} is the energy consumption of the device while idling (0.3 W) and P^{send} and P^{rec} are the energy consumption of the device while transmitting and receiving information (1.3 W). These values have been used for calculating the energy used by an MD computing locally or offloading to a MEC server. Furthermore, the value of $\Pi = 0.4$ is assumed in Equation (7), which means that the data returned from a MEC server is 60% less (only 40% of what was transmitted).

To model variable MD power ratings with increasing CPU workload, data from [14] were used to derive a linear relationship from data plotted for a Xeon processor: relative power rating = 0.0096 × MD CPU workload + 0.8967. This relationship was used to adjust all MD power ratings P^{MD}, P^{idle} and P^{send} and P^{rec} were assumed to be approximately equal [15].

A. The impact of increasing job data size on the completion time

Job size and completion time have a linear relation as shown in equations (3) and (5). Fig. 1 shows the effect of

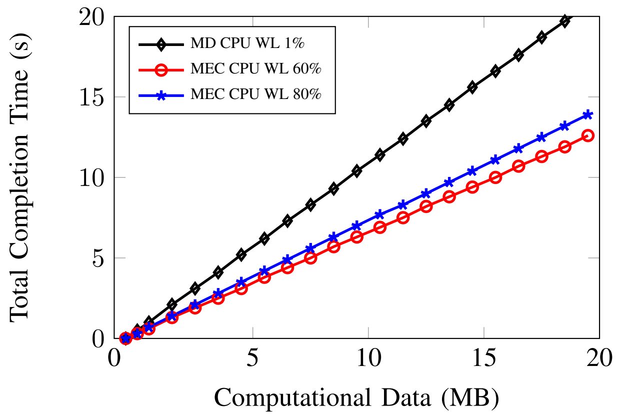


Fig. 1: Effect of computational data size on task completion time with the higher complexity application 1 at a 20 Mbps communication link speed to/from a MEC server.

increasing job (from 1 MB to 20 MB) size on the total completion when computing locally or offloading files for Application 1. Three cases are plotted: 1% MD CPU loading, 60% MEC CPU loading and 80% MEC CPU loading. Each case showed a linear increase in the total completion time. Even with the very low (1%) MD loading, the local job completion time was longer than offloading the task to the MEC server at any job size and at either server CPU workload.

B. The impact of MEC workload on the completion time

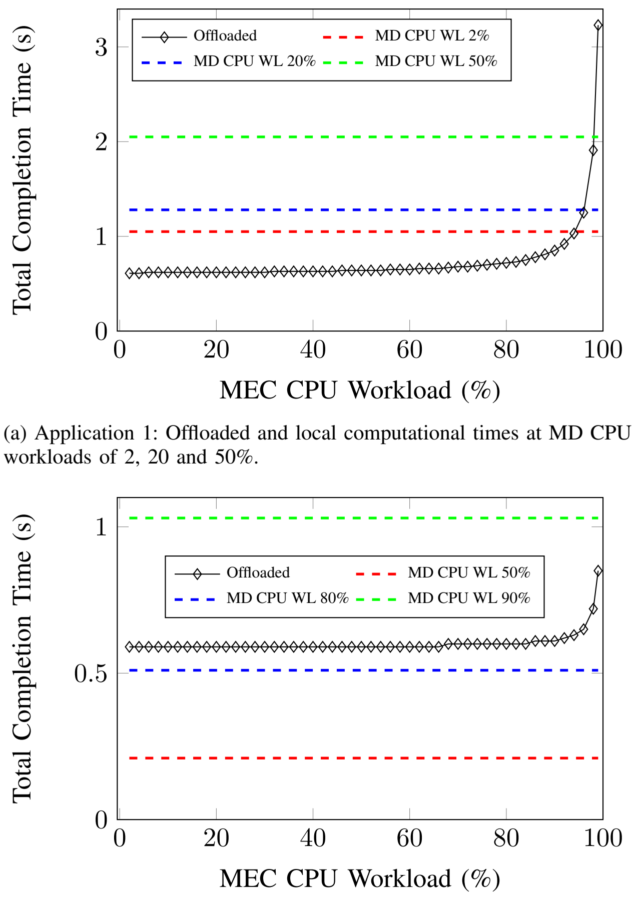
Fig. 2 presents the results with the two applications when the MEC CPU workload was increased from 1% to 99%. Fig. 2(a) shows that the total task completion time with the higher complexity Application 1 increased greatly as the MEC server CPU workload approached 100%. Nevertheless, offloading could result in a shorter task completion time even at a >90% MEC CPU workload if the CPU workload on the MD processor was >50%. Fig. 2(b) shows that, whatever the MEC server CPU workload, local computation was faster with the lower complexity Application 2 until the MD CPU workload became high.

C. The impact of MD workload on the completion time

Fig. 3 presents the results with the two applications with the MD CPU varying up to 99%. Fig. 3(a) shows that local computation was faster at low MD CPU workloads (<20%) but at higher MD CPU workloads offloading was beneficial for reducing task completion time at a MEC server CPU workload of 96%; even at 99% server CPU workload, offloading was beneficial if the MD CPU workload exceeded 70%. Fig. 3(b) shows that local computation was faster with the lower complexity Application 2 than offloading to a very high CPU workload server until the MD CPU workload approached 90%

D. The impact of link speed on offloading decision

The higher the MD processor CPU workload, the lower was the minimum communication link speed required for shorter total task completion time by offloading; this is shown in Fig. 4(a). With the much smaller local computation demands required for Application 2, minimum communication link speeds required for shorter total task completion time by offloading were much higher than for Application 1; this is shown in Fig. 4(b). At high MD CPU workloads, link speeds were compatible with 4G wireless networks but 5G range speeds were required if local computation was performed at low MD CPU workloads.



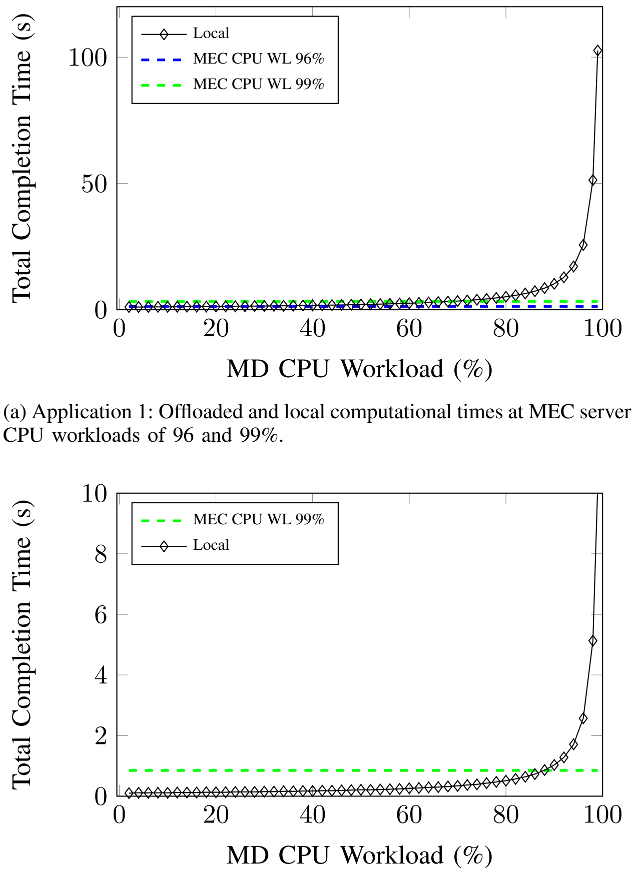
(a) Application 1: Offloaded and local computational times at MD CPU workloads of 2, 20 and 50%.

(b) Application 2: Offloaded and local computational times at MD CPU workloads of 50, 80 and 90%.

Fig. 2: Effect of varying MEC server CPU workload on task completion time for 1 MB data file offloaded at 20 Mbps connection link speed or processed locally at selected MD CPU workloads.

E. Energy Usage by an MD

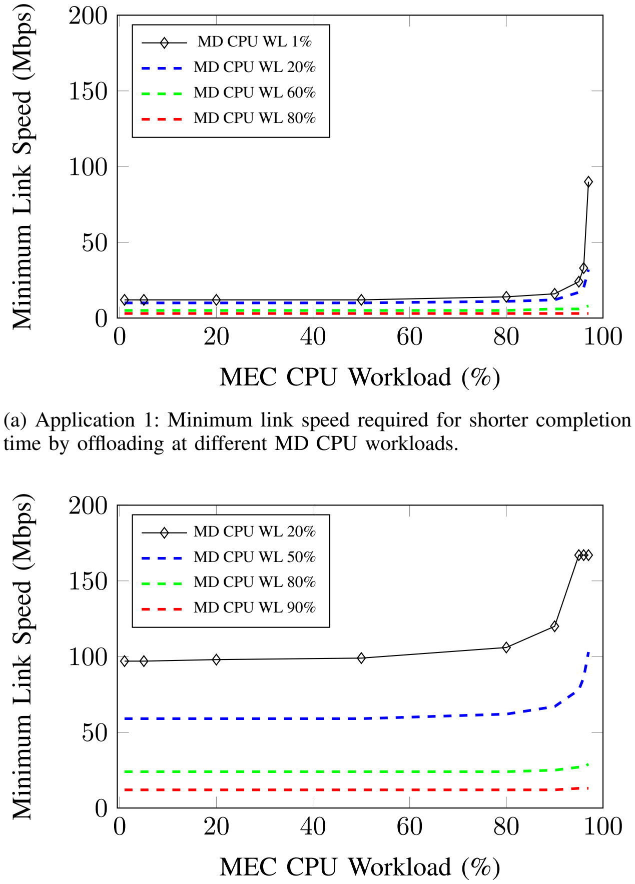
Fig. 5(a) shows that energy usage for local processing by the MD increased as the MD CPU workload increased with the higher complexity Application 1. In contrast, MD energy usage while offloading increased only very little when very high MD CPU workloads were reached. The energy saving for the MD occurred at all MD CPU workloads but increased greatly as MD CPU workloads increased and reached nearly 90% of local energy use when the CPU workload reached 90%. With the lower complexity Application 2, however, no energy



(a) Application 1: Offloaded and local computational times at MEC server CPU workloads of 96 and 99%.

(b) Application 2: Offloaded and local computational times at MEC server CPU workload of 99%.

Fig. 3: Effect of varying of MD server CPU workload on task completion time for a 1 MB data file offloaded at 20 Mbps connection link speed or processed locally at selected MEC server CPU workloads.



(a) Application 1: Minimum link speed required for shorter completion time by offloading at different MD CPU workloads.

(b) Application 2: Minimum link speed required for shorter completion time by offloading at different MD CPU workloads.

Fig. 4: Effect of varying MEC server CPU workload on the minimum link speed required for shorter task completion time with a 1 MB data at selected MD CPU workloads.

savings were possible for the MD by offloading until the MD CPU workload approached 90%, as shown in Fig. 5(b). This was because the low complexity of the application resulted in very short completion times when computed locally and communication times with the MEC network caused total task completion times by offloading to be longer than local processing.

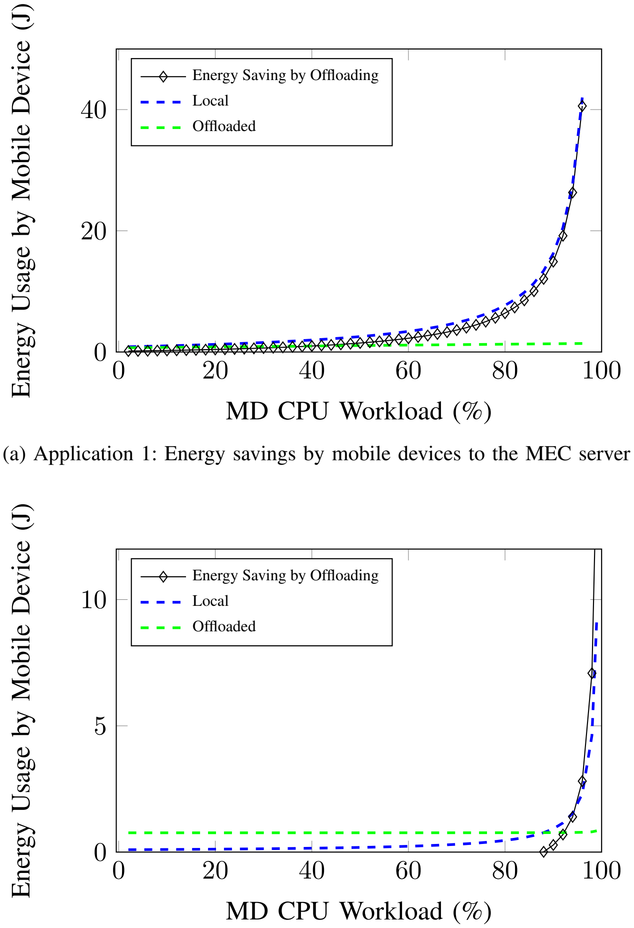
IV. Discussion

Our work has shown that the computation offloading decisions are critically dependent on the following four independent variables: onboard processor CPU workload, server-side CPU workload, communication link speed, and task complexity. Modern MDs are equipped with high processing power, and therefore, it is not always beneficial to offload tasks to a MEC network. We demonstrate that the offloading is only beneficial after a certain threshold of link speed is achieved. Such a threshold depends on the mobile device processor and the computing capabilities offered on the MEC side.

With a ten-fold lower computational task complexity, data transmission time is the dominant factor in rejecting offloading but higher link speeds eliminate any advantage of a high-power MD processor. Energy usage by the MD is not reduced by offloading until the MD CPU workload is very high. For a MEC network, therefore, ease of use and the Quality of Experience for mobile users and subscribers can only be established if the network functions smoothly and efficiently. For this, high link speeds for data transmission and reception, the highest possible ratio of server-side to onboard processor speeds and the avoidance of overuse of the server, CPU is essential. A congested MEC the network will disappoint users of MDs with high onboard processor speeds searching for faster task completion times, although energy use reduction will be paramount for some users with a low battery charge.

V. Conclusions and Future Research

We have demonstrated that any decision-making process for offloading must be able to compute advantages of task completion time and energy savings in a dynamic environment where widely fluctuating user numbers are expected to make use of such facilities, for example, city centers. Similarly, the



(a) Application 1: Energy savings by mobile devices to the MEC server

(b) Application 2: Energy savings by mobile devices to the MEC server.

Fig. 5: Effect of varying CPU workloads on energy consumption by a MD for a 1 MB data processed locally or offloaded at 20 Mbps.

Edge Computing facilities' provider must build in sufficient flexibility to cope with peak demand without overloading the network or linking servers to back-up servers in more massive and responsive network architectures. Our results demonstrate that link speed is a critical parameter that determines the offloading decisions. This is an important observation, especially in the introduction of the 5G network, which can significantly increase the connection speed between mobile users and the MEC servers.

Our future work will focus, firstly, on analysing offloading decisions for multiple jobs from a single MD or (in a MEC network) offloading from multiple MDs, and secondly on developing multi-objective optimization to minimize MD energy consumption, total task completion times and the cost to the MD user of using an offloading service to replace or augment local computation.

VI. Acknowledgment

This work is supported by Engineering and Physical Sciences Research Council (EPSRC) and Telecommunications Research Laboratory - Toshiba Europe (Bristol).

References

[1] L. Lin, X. Liao, H. Jin, and P. Li, "Computation offloading toward edge computing," *Proceedings of the IEEE*, vol. 107, no. 8, pp. 1584–1607, Aug 2019.

[2] M. Carroll, A. van Der Merwe, and P. Kotze, "Secure cloud computing: Benefits, risks and controls," in *Information Security for South Africa (ISSA), 2011*. IEEE, 2011, pp. 1–9.

[3] M. Satyanarayanan, "The emergence of edge computing," *Computer*, vol. 50, no. 1, pp. 30–39, Jan 2017.

[4] T. Taleb, K. Samdanis, B. Mada, H. Flinck, S. Dutta, and D. Sabella, "On multi-access edge computing: A survey of the emerging 5G network edge cloud architecture and orchestration," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 3, pp. 1657–1681, 2017.

[5] H. Guo, J. Liu, and J. Zhang, "Computation offloading for multi-access mobile edge computing in ultra-dense networks," *IEEE Communications Magazine*, vol. 56, no. 8, pp. 14–19, 2018.

[6] H. Guo, J. Liu, and J. Zhang, "Efficient computation offloading for multi-access edge computing in 5G hetnets," in *2018 IEEE International Conference on Communications (ICC)*, 2018, pp. 1–6.

[7] N. I. M. Enzai and M. Tang, "A heuristic algorithm for multi-site computation offloading in mobile cloud computing," *Procedia Computer Science*, vol. 80, pp. 1232–1241, 2016.

[8] B. Yang, X. Cao, J. Bassey, X. Li, T. Kroecker, and L. Qian, "Computation offloading in multi-access edge computing networks: A multi-task learning approach," in *IEEE International Conference on Communications (ICC)*. IEEE, 2019, pp. 1–6.

[9] S. Chouhan, "Energy optimal partial computation offloading framework for mobile devices in multi-access edge computing," in *2019 International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*. IEEE, 2019, pp. 1–6.

[10] R. Singh, S. Armour, A. Khan, M. Sooriyabandara, and G. Oikonomou, "The advantage of computation offloading in multi-access edge computing," in *2019 Fourth International Conference on Fog and Mobile Edge Computing (FMEC)*, June 2019, pp. 289–294.

[11] S. Melendez and M. P. McGarry, "Computation offloading decisions for reducing completion time," in *14th Annual Consumer Communications & Networking Conference (CCNC), 2017*. IEEE, 2017, pp. 160–164.

[12] A. R. Khan, M. Othman, A. N. Khan, J. Shuja, and S. Mustafa, "Computation offloading cost estimation in mobile cloud application models," *Wireless Personal Communications*, vol. 97, no. 3, pp. 4897–4920, 2017.

[13] K. Kumar and Y.-H. Lu, "Cloud computing for mobile users: Can offloading computation save energy?" *Computer*, vol. 43, no. 4, pp. 51–56, 2010.

[14] J. V. Kistowski, H. Block, J. Beckett, K.-D. Lange, J. A. Arnold, and S. Kounev, "Analysis of the influences on server power consumption and energy efficiency for CPU-intensive workloads," in *Proceedings of the 6th ACM/SPEC International Conference on Performance Engineering*, 2015, pp. 223–234.

[15] G. P. Perrucci, F. H. Fitzek, and J. Widmer, "Survey on energy consumption entities on the smartphone platform," in *2011 IEEE 73rd Vehicular Technology Conference (VTC Spring), Yokohama*. IEEE, 2011, pp. 1–6.