Dynamic Resource Allocation Exploiting Mobility Prediction in Mobile Edge Computing

Jan Plachy, Zdenek Becvar
Department of Telecommunication Engineering
Faculty of Electrical Engineering
Czech Technical University in Prague
Technicka 2, 166 27 Prague, Czech Republic
jan.plachy@fel.cvut.cz, zdenek.becvar@fel.cvut.cz

Emilio Calvanese Strinati
CEA-Leti
17 rue des Martyrs, 38000 Grenoble, France
Email: emilio.calvanese-strinati@cea.fr

***Abstract*—In 5G mobile networks, computing and communication converge into a single concept. This convergence leads to introduction of Mobile Edge Computing where computing resources are distributed at the edge of mobile network, i.e., in base stations. This approach significantly reduces delay for computing tasks offloaded from users' devices to cloud and reduces load of backhaul. However, due to users' mobility, optimal allocation of the computational resources at the base stations might change over time. The computational resources are allocated in a form of Virtual Machines (VM), which emulate a given computer system. User's mobility can be solved by VM migration, i.e., transfer of VM from one base station to another. Another option is to find a new communication path for exchange of data between the VM and the user. In this paper, we propose an algorithm enabling flexible selection of communication path together with VM placement. To handle dynamicity of the system, we exploit prediction of users' movement. The prediction is used for dynamic VM placement and finding of the most suitable communication path according to expected user's movement. Comparing to state of the art approaches, the proposal leads to reduction of the task offloading delay between 10% and 66% while energy consumed by user's equipment is kept at similar level. The proposed algorithm also enables higher arrival rate of the offloading requirements.**

***Keywords*—mobile network, 5G, Mobile Edge Computing, Virtual Machine, Offloading.**

I. Introduction

Demands of users on computation power of user equipment (UE) are rising due to computation demanding tasks in form of applications such as facial/object recognition, video/speech processing, etc. However, as the UEs are powered by the battery with a limited capacity, these applications can be used for a limited time only because of high energy consumption. Moreover, if the UE cannot provide sufficient computation power (e.g., due to outdated processor) the applications cannot be run at all. In order to enable demanding applications on low-class UEs or to extend battery life time of the UEs, a possibility to offload computing tasks to the cloud has been introduced [1].

As the newly introduced applications push their requirements towards real time, overall delay of communication (including communication over radio, backhaul, and to the cloud) in the conventional mobile cloud computing can be a limiting factor. To reduce communication delay and energy consumption of the UE, computational power can be allocated closer to the users, i.e., at the base stations (eNB). This concept is known as Small Cell Cloud (introduced in [2]) or Mobile Edge Computing (MEC), which is an extension of the Small Cell Cloud towards general exploitation of cloud computing also for control and management of mobile networks. The MEC aims to overcome limitations of an offloading delay and backhaul congestion.

To compute a task in the cloud, a Virtual Machine (VM) is to be created over allocated computational resources in the cloud. The VM emulates a computer system, in which the same application as the one, which computing is being offloaded, is being run to process the offloaded task. The overall delay of offloading (denoted as offloading delay) perceived by users, consists of: i) transmission of the offloaded task to eNB where the VM is allocated, ii) processing of the task by the VM and iii) transmission of the results back to the UE. With focus on offloading of real-time applications, the VM assigned to the UE should be ready when the computing task is being offloaded [3]. Otherwise, the delay due to creating and starting VM would make such a service unusable.

To meet demands of users on quality of service for offloading of heavy computation demanding applications, communication and computing resources have to be allocated jointly [4]. The motivation for this is the overall offloading delay consisting of communication and computation. However, the problem solved in [4] aims at static users only. With moving users, the problem becomes significantly more complicated and more complex.

Two general approaches are suitable for handling offloading for mobile users. The first one assumes to keep the VM at the eNB at which resources were allocated at the time of the task offloading [5]. In this case, the serving eNB is selected in a common way according to radio signal quality. To keep the UE connected to the eNB with the highest quality of signal, common handover procedure [6] (consisting of handing off connection from one eNB to another one) is used. However, due to the backhaul constraints and potential load of radio channels, using only the serving base station, can lead to degraded performance as demonstrated in [7]. Therefore, in

[7], the authors propose an algorithm extending the common handover procedure. The extension consists in modification of handover decision by consideration of communication with the VM assigned to the UE. Therefore, the communication path (radio as well as backhaul) for offloading of the task or receiving the results is selected among all eNBs in a communication range of the UE. This algorithm, denoted as Path Selection with Handover (PSwH), mitigates a problem of high delay in case of mobile users.

The second approach for handling users' mobility in MEC is focused on migration of the VM. The VM is migrated to a new serving eNB when the UE changes its serving eNB. In this case, the VM is migrated closer to the UE, leading to reduction in communication delay. An algorithm for deciding whether and where to migrate VM is proposed in [8]. The authors consider distance as the sole metric for decision on VM migration. This work is further enhanced by mobility prediction [9]. Also, the paper [9] considers the number of UEs utilizing VM's resources at a given eNB as a metric for decision on VM placement. As shown in [10], a long-term prediction can achieve accuracy of 93%. Moreover, if also a short-term prediction is taken into account, the accuracy can be improved to 95% [11]. Nevertheless the algorithm for VM placement proposed in [9] still delivers offloaded task via serving eNB selected according to radio channels.

In this paper, we extend our previous work [7], by use of mobility prediction and we design a complementary algorithm which decides where to place the VM. Therefore we exploit both selection of communication path and VM placement to solve the problem with mobility of users. The proposed solution consist of two cooperative algorithms. The first algorithm, dynamic VM placement, decides whether there is a more suitable placement for VM based on predicted mobility of users and load of eNBs' communication and computation resources. The second designed algorithm, the PSwH enhanced by mobility prediction, is used when the UE starts offloading its task to select a suitable communication path. Note that we do not target design of a new prediction algorithm, but we exploit existing approaches for our algorithms as these reach sufficient accuracy. Then, we compare the proposed algorithm with related work to demonstrate superiority of our proposal.

The rest of this paper is organized as follows. In the next section, we define model of the investigated MEC system. In Section III., the proposed algorithm is described. Simulation environment and results are presented in Section IV. Last, Section V concludes the paper and outlines future work plans.

II. System model

In this section, the system model is presented. The system is assumed to be composed of set S of base stations $s\{\in S\}$. To generalize the system model, the base station can be represented either by macro cell (eNB), small cell (SCeNB), or femto cell (HeNB). Unless otherwise stated, the label eNB covers all types of base stations. For each UE, a serving eNB $s' \in S$ is selected as the one with the highest received signal strength (RSS). As the UE moves, the serving eNB is updated following a conventional hard handover procedure (see [6]). The handover introduces an interruption in communication with a duration of D_{HO} (known as handover delay or handover interruption). This delay consists of time required to break the connection with current serving eNB and to establish a connection with a new eNB. Note that the UE can neither transmit nor receive data during hard handover in mobile networks.

To facilitate MEC, a VM for the UE is created at a base station, denoted as $s_{VM} \in S$. Number of the UEs with VM allocated at the s-th eNB is denoted as $n_s^{VM}(t)$, whereas number of UEs utilizing communication resources of the s-th eNB is denoted as $n_s^c(t)$. The VM can be placed at i) eNB selected based on offloading delay or energy and kept there [5], ii) the serving eNB and migrated to a new serving eNB after handover [12], iii) dynamically placed by considering UE's movement [9].

The offloaded task is defined by its size L (in bits), the number of instructions to be processed B, and the size of computed results R (in bits). As a possibility to migrate the VM is considered, the size of the migrated VM is defined as a number of bits G.

In Figure 1, an example of UE's movement for two adjacent time instances t and $t+\Delta t$ is depicted. The UE communicates with the serving eNB via radio channel with $SINR_s$ and capacity c_s^R. Each eNB is connected to mobile operator's core network (network connection through which the eNB is connected to the Internet) via backhaul with capacity c_s^B.

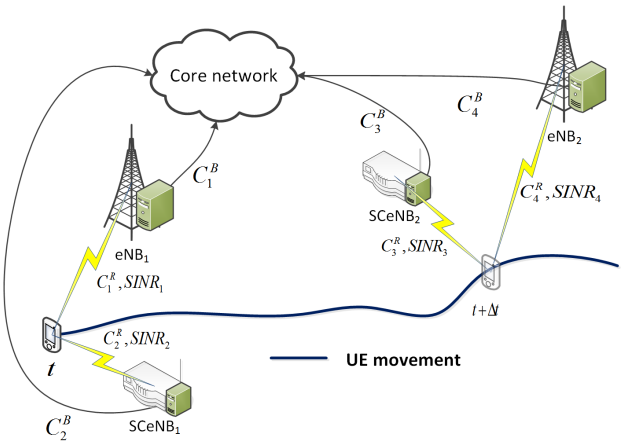

Figure 1. System model.

A set of eNBs with which the UE can communicate over radio channel is denoted as I; $I \subset S$. The set I includes the eNBs for which the $SINR$ observed by the UE is above $SINR_{min}$. An example of $SINR_{min}$ for LTE-A network and BLER of 10% is a value of -6.9 dB [13]. If the UE needs to deliver an offloaded task to s_{VM}, the transmission can be done directly via radio if $s' \neq s_{VM}$ or the offloaded task is transmitted via radio of s' and then via backhaul connection between s' and s_{VM}. The capacity available for delivery of

the offloaded task from the UE to the eNB with allocated VM is calculated as:

$$c_{UE,s_{VM}} = \begin{cases} c^R_{s'} & s' = s_{VM} \\ min\{c^R_{s'}, c^B_{s',s_{VM}}\} & otherwise \end{cases}$$

where capacity between two eNBs s' and s_{VM} is calculated as $c^B_{s',s_{VM}} = min\{c^B_{s'}, c^B_{s_{VM}}\}$.

As prediction is considered in our model, predicted communication capacity and available computing capacity (in instructions per second) are denoted as $\tilde{c}(t)$ and $\tilde{k}(t)$, respectively, where t is the time instance at which capacity and computational resources are being predicted. The predicted radio capacity is derived from predicted position of the UE mapped to SINR maps as introduced in [14]. From information about SINR and $n^c_s(t)$, available capacity of radio is computed as:

$$\tilde{c}^R_s(t) = thr\{MCS\{SINR_s(t)\}, \frac{n^{RB}_s}{n^c_s(t)}\}$$

where $MCS\{SINR_s(t)\}$ maps SINR to Modulation and Coding scheme (MCS) (e.g. by [13]), n^{RB}_s specifies the number of all Resource Blocks (RB) of the $s-$th eNB and function $thr()$, maps MCS and the number of RBs to the number of bits for transmission as described in [15]. For backhaul, the predicted capacity is calculated as $\tilde{c}^B_s(t) = \frac{\bar{c}^B_s}{n^c_s(t)}$, where $\bar{c}^B_s$ denotes backhaul capacity of the s-th eNB. Apart from capacities, the predicted delay of offloading consists of: i) delay due to uploading the offloaded task to the VM, $\tilde{D}^{UL}_s = \frac{L}{\tilde{c}_{UE,s_{VM}}(t)} + \sum D_{HO}$, ii) computation delay, i.e., time required to process the offloaded task by the VM, $\tilde{D}^W_s = \frac{B}{\tilde{k}(t)}$, iii) delay due to collecting results by the UE from the VM, i.e., downloading computed results from the eNB where the VM is allocated, $\tilde{D}^{DL}_s = \frac{R}{\tilde{c}_{s_{VM},UE}(t)} + \sum D_{HO}$, iv) delay of the VM migration represented by time required to copy and start the VM from current serving eNB to a new serving eNB, $\tilde{D}^{VM}_s = \frac{G}{\tilde{c}_{s,s_{VM}}(t)}$, v) delay of starting VM instead of migrating VM, D_{APP}. As we target offloading of real-time applications, we assume that the VM is pre-allocated [16]. Thus, delay due to starting the VM is equal only to delay of starting the offloaded application on the side of the VM. Total delay of one offloaded task is then defined as $\tilde{D}^T_s = \tilde{D}^{UL}_s + \tilde{D}^W_s + \tilde{D}^{DL}_s + \sum D^{VM}_s + \sum D_{APP}$ and it is a sum of communication, computation, VM migrations and starts of VMs.

III. Dynamic Resource Allocation

The proposed dynamic resource allocation in this paper is based on our previous PSwH algorithm described in [7], which exploits reward function from Markov Decision Process (MDP) to select the communication path q. The PSwH forces the UE to perform handover to new eNB if it is profitable for the UE from the offloading point of view. In this paper, we enhance the PSwH algorithm by mobility prediction and we design cooperative algorithm for dynamic VM placement based on calculating reward in terms of communication capacity and incorporating load balancing. Both algorithms are based on reward function from MDP and utilize prediction window denoted by τ.

The idea of cooperation, between algorithms in the proposal, is to dynamically place VM before the UE starts offloading as migration (or start) of VM when offloading has already started would increase offloading delay by VM migration (start of VM). Therefore, when the UE starts offloading its task, VM will be ready at the suitable eNB. Also the UE will utilize the PSwH enhanced with mobility prediction to select a suitable communication path (i.e. serving eNB) in order to further reduce offloading delay. Cooperation is achieved by starting algorithm for dynamic VM placement in-between offloading of two consecutive tasks, when certain radio conditions are met (SINR is below a given threshold). As both, the PSwH enhanced by mobility prediction and dynamic VM placement algorithms are based on MDP, we describe its reward function. The reward function for selection of communication path q (or dynamic VM placement by replacing q by s_{VM}) is defined as:

$$V^k_\pi = Pred[\sum_k R^t|\pi, q] = R(q) + \sum_k T(q, \pi(q,k), q')V^{k-1}_\pi(q')]$$

where k denotes total offloading delay in terms of discrete time steps, $R(q)$ denotes immediate reward for communication over path q instead of path q', $\sum_k T(q, \pi(q,k), q')V^{k-1}_\pi(q')$ represents expected future payoff as a sum of rewards over k steps, $Pred$ represents the fact that the reward is predicted. With respect to PsWH in [7], the reward function is based on prediction.

As the VM have to be ready to process the offloaded task when offloading starts [3], decision on VM placement have to be made before offloading starts. Therefore, algorithm for dynamic VM placement is started if in communication proximity of the UE are eNBs with $(SINR_s > SINR_{s_{VM}} | s \in S, s \neq s_{VM})$.

To find the best placement of VM, $SINR$ to set S have to be predicted as in [9], authors consider every eNB as a possible VM placement. However the set S could be quite large and thus in our proposal, we define a reduced set $Z\{z \in Z | (SINR_z > SINR_{min}) \cap (n^{VM}_z(t) < n_{limit})\}$. In this set each eNB z has $SINR$ above $SINR_{min}$. Also, the set is further reduced by avoiding over utilized eNBs to distribute computational load more equally.

The proposed algorithm for dynamic VM placement is described in Algorithm 1. For each eNB in Z (step 1) and each eNB from set I (step 2), $SINR$ is predicted by applying SINR map [14] on predicted UE's mobility (step 3). Communication capacity is predicted from predicted $SINR$ and $n^c_s(t)$(step 4). In order to prefer eNBs with good channel quality in the future (next time steps), a slope of SINR is calculated as shown in step 5 and eNBs with negative slope are discarded from set

I (steps 6 and 7). To suppress an impact of shadowing and fast fading, the slope is calculated over a whole period of prediction interval τ. For each VM placement, we select the eNB with the highest available capacity (step 10) and then eNB with the highest predicted gain in capacity is selected for VM placement (step 13). Following selection of eNB for VM placement, the VM migration delay is predicted (step 14) and the option with lower delay between start of VM and VM migration is selected (step 15).

Algorithm 1 VM dynamic placement

1: **for** $z \in Z$ **do**
2: **for** $i \in I$ **do**
3: predict $SINR_i(t, t+\Delta t, ..., t+\tau)$
4: predict $\tilde{c}_{z,i}(t, t+\Delta t, ..., t+\tau)$
5: $\alpha = \frac{dSINR_i}{dt}$
6: **if** $\alpha \leq 0$ **then**
7: $I = I \setminus i$
8: **end if**
9: **end for**
10: $\tilde{c}_z = \max_i\{\tilde{c}_{z,i}\}$
11: **end for**
12: $\hat{s}_{VM} = s_{VM}$
13: $s_{VM} = \arg\max_z(\tilde{c}_z - \tilde{c}_{current})$
14: $\tilde{D}^{VM} = \frac{G}{\tilde{c}_{\hat{s}_{VM}, s_{VM}}}$"
15: $option = min(D_{APP}, \tilde{D}^{VM})$

The enhancement of the PSwH by mobility prediction is described in Algorithm 2. First, available capacities of eNBs in set I are predicted (step 2) and handover vector $\rho = \{\rho_1, \rho_2, ..., \rho_{|I|}\}$ is initiated by setting its elements ρ_i to be 1 if eNB i is also the serving eNB (step 4) or to $1 - D_{HO}$ otherwise (step 6). The handover vector is used for modification of communication capacity to each eNB as no data can be transferred between the UE and the eNB during D_{HO}. Until all required data L are transmitted (step 9), eNB with the highest communication capacity is selected as $q(t)$. Also, vector ρ is modified to be in line with $q(t)$ (step 12).

As the optimal solution for selecting VM placement and path selection is a combinatorial problem, it is required to go through every combination of serving eNB, VM placement, and every step during offloading. This would have a large computation complexity and therefore, in our proposal, we reduce candidates for VM placement and path selection, Therefore, algorithm for VM placement has time-complexity of $O(|Z||I|\tau)$ and path selection $O(|I|\tau)$. Both time-complexities are lower than time-complexity of algorithm proposed in [9].

Algorithm 2 PSwH with prediciton

1: **for** $i \in I$ **do**
2: predict $\tilde{c}_i(t, t+\Delta t, ..., t+\tau)$
3: **if** $s' = i$ **then**
4: $\rho_i = 1$
5: **else**
6: $\rho_i = 1 - D_{HO}$
7: **end if**
8: **end for**
9: **while** $L > 0$ **do**
10: $q(t) = \arg\max_i(\tilde{c}_i(t) \times (\Delta t \cdot \rho))$
11: $L = L - max(\tilde{c}_i(t) \times (\Delta t \cdot \rho)$
12: **if** $i = q(t)$ **then**
13: $\rho_i = 1$
14: **else**
15: $\rho_i = 1 - D_{HO}$
16: **end if**
17: $t = t + \Delta t$
18: **end while**

IV. Performance Evaluation

In this section, models and scenario for performance evaluation are defined. The evaluation is carried out by means of simulations in MATLAB.

A. Simulation scenario and models

Major parameters of the simulation, presented in Table I, are in line with recommendations for networks with small cells as defined by 3GPP in [17]. We also follow parameters of the physical layer and frame structure for LTE-A mobile networks defined in the same document.

Signal propagation is modeled according to 3GPP [17] with path loss model $PL = 128.1 + 37.6log10(d)$, where d is a distance between the UE and the eNB. A mapping function between SINR and MCS [13] with BLER=10%. The backhaul of eNBs is modeled as optical fiber with capacities (in Mbit/s) generated from normal distribution with $\mu = 100$ and $\sigma^2 = 2$.

Since we target to real-time applications, offloaded task has a size of 200 kB (as in [18] authors consider task to be in tenths of kB) and its arrival rate is specified by λ. The size of data transferred during VM migration (data in RAM of offloaded task) is 20 MB. Time before VM is prepared to process an offloaded task (start time) is 500 ms, which consist only of starting an offloaded application at the VM.

Radio and backhaul resource allocation is done by round-robin scheduling.

We assume the hexagonal grid of 19 eNB like in [9] and we further drop 57 HeNBs into the simulation area. There are 200 UEs moving within the area of all eNBs.

B. Performance evaluation

In our simulations, the proposed algorithm is compared with three competitive algorithms:

- SO [12] - The VM is kept at the serving eNB, so the VM is migrated each time handover is performed.
- Wang's algorithm [9] - VM placement is based on predicted future costs of its placement.
- PSwH without prediction [7] Communication path (serving eNB) is selected so that communication delay is minimized.

In Figure 2 we show the average offloading delay (consisting of uploading offloaded task, computing, and collecting results) of the task in dependency on task inter-arrival rate

TABLE I. Simulation parameters

Parameter	Value
Simulation area	800 x 800 m
Carrier frequency	2 GHz
Bandwidth for downlink/uplink	10/10 MHz
Tx power of eNB/SCeNB/UE	27/15/10 dB
Number of eNB/SCeNB	19/57
VM size/start time	20 MB/500 ms
Offloaded task/results size	200/200 kB
Offloaded task number of instructions	$1e6$ instructions
eNB/SCeNB CPU	3300 MIPS
Prediction window τ/Wang's algorithm	20s/60s
Prediction accuracy	90%
Shadowing factor	6 dB
Handover interruption duration	30 ms
Number of UEs	200
Speed of users	1 m/s
Backhaul capacity-Normal distribution	$\mu = 100, \rho^2 = 2$ Mbit/s
Simulation time T	2 000 s
Number of simulation drops	10 drops

(λ). From this figure, we can see that with decreasing λ, the average offloading delay increases as the load of communication and computation resources increases. The proposed algorithm reduces the average offloading delay significantly comparing to all competitive algorithms. For lightly loaded network ($\lambda = 40s$), the average offloading delay is reduced by the proposed algorithm by 27.3%, 15.6%, and 9.7% with respect to the SO, Wang's algorithm, and PsWH, respectively. For heavily loaded network ($\lambda = 10s$) the gain is 30.5%, 29.2%, and 26.6% with respect to the SO, Wang's algorithm, and PsWH, respectively. The gain is caused by cooperation between VM placement and path selection according to predicted situation in the network.

Note that results for the SO and Wang's algorithm for $\lambda < 10s$ are not depicted as these algorithms cannot handle such load of network as delay of tasks can lead to tasks being buffered at the UE and thus leading to congestion of communication and computation resources. The proposal, by combining both VM placement and path selection avoids over utilized eNBs a thus works even for $\lambda = 1s$. The proposal outperforms all compared algorithms as compared to the PsWH, which has the second lowest delay, reduces offloading delay by up to 66%.

In Figure 3, we compare CDF of the average offloading delay for $\lambda = 10s$. We show CDF for $\lambda = 10s$ as it corresponds to heavily loaded network, which is more challenging than lightly loaded network.

The offloading delays reached by UEs in case of the SO, Wang's algorithm, and PsWH are spread significantly from relatively low values (115 ms) to extremely high delays not acceptable for real time services (even more than 2s). Contrary, the proposed algorithm offers stable delay around 200 ms for almost all UEs. For example, the delay experienced by 95% of UEs is below 250ms for the proposal while competitive SO, Wang's algorithm, and PsWH requires 610ms (144%more), 610ms (144% more), 500ms (100% more), respectively. Consequently, almost all UEs exploiting the proposed algorithm can exploit real-time services with high quality.

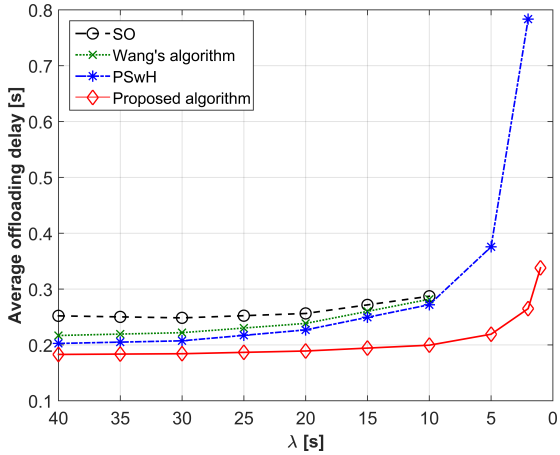

Figure 2. Average times required to offload, compute and collect results of the offloaded task.

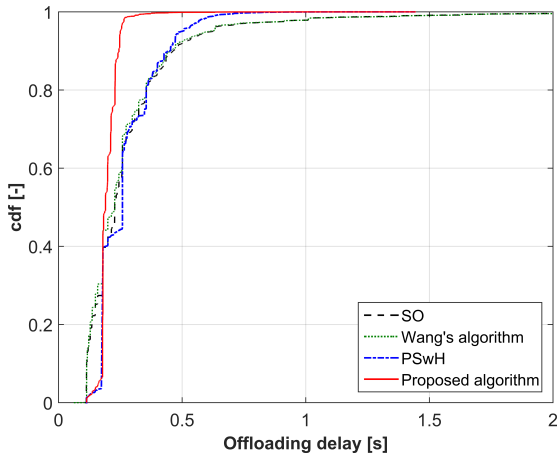

Figure 3. CDF of the task offloading delay for $\lambda = 10s$.

In Figure 4, comparison of average energy consumed by the UE for communication of a single task is shown. With decreasing λ, consumed energy increases as offloading delay is higher due to increased network load and relation between UE's energy consumption and delay [19]. From Figure 4, we can see, that the PsWH is the most energy hungry and it consumes between 10.7% and 188% more energy than the proposed algorithm. The SO and Wang's algorithm require less energy (up 9%) per offloaded task than the proposed algorithm if the network is lightly loaded ($\lambda > 15s$). Contrary, for heavily loaded network ($\lambda < 15s$), the proposed algorithm becomes more energy efficient (saving of 9%). The reason for increase in energy consumption by the proposal at light network load is the fact that the proposed algorithm targets solely on offloading delay and disregard energy consumption. Extension towards consideration of the energy consumption is considered as a future work. Note also that the SO and Wang's algorithm cannot serve tasks with λ lower then 10s.

In Figure 5, we show CDF of the energy spent by the UE for communication for $\lambda = 10s$. The energy consumption reached by the SO and Wang's algorithm is spread more wide so energy consumption of some UEs is reduced comparing to the

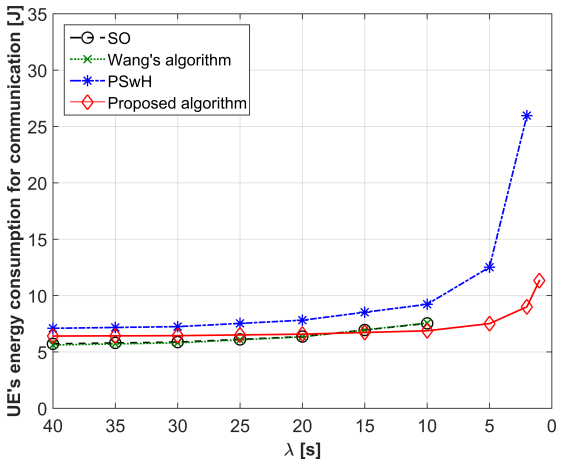

Figure 4. Average energy consumption of UE communication.

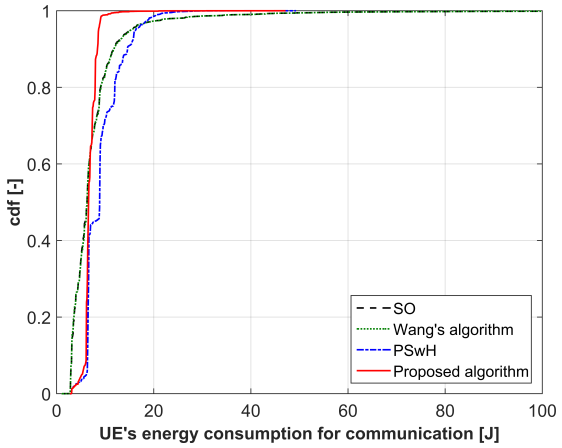

Figure 5. CDF of energy consumed for communication for $\lambda = 10s$.

proposed algorithm while some UEs consumes significantly more energy. This shows fairness of the proposed algorithm among users and there are no users significantly punished for unfair allocation of resources for computation. The energy consumed by 95% UEs is below 8.61J for the proposal while competitive SO, Wang's algorithm, and PsWH consumes 15.3J (77.7%more), 15.3J (77.7% more), 16.3J (89.3% more), respectively.

V. Conclusion

In this paper, we have proposed an algorithm for dynamic allocation of computing and communication resources for Mobile Edge Computing. The algorithm dynamically places VMs considering load of eNBs and selects communication path between the UE and the eNB with allocated VM. The algorithm is based on MDP and exploits mobility prediction.

Comparing to state of the art approaches, the proposed algorithm reduces the offloading by 10-66%. The superiority of the proposed algorithm is more notable for high arrival rate of the offloading requests, i.e., for heavily loaded network. At the same time, the energy consumed by the UEs for offloading is kept at similar level as for the state of the art algorithms. The proposed algorithm also balances fairness among users in terms of experienced delay and energy consumption so that all UEs can exploit real-time services even for very high arrival rates of the offloading requests.

In the future, we will focus on extension of the algorithm towards energy consumption awareness.

References

[1] S. S. Qureshi, T. Ahmad, K. Rafique, *et al.*, "Mobile cloud computing as future for mobile applications-implementation methods and challenging issues," in *IEEE CCIS*, pp. 467–471, 2011.

[2] F. Lobillo, Z. Becvar, M. A. Puente, P. Mach, F. Lo Presti, F. Gambetti, M. Goldhamer, J. Vidal, A. K. Widiawan, and E. Calvanesse, "An architecture for mobile computation offloading on cloud-enabled lte small cells," in *IEEE Wireless Communications and Networking Conference Workshops (WCNCW) 2014, Istanbul*, pp. 1–6, 2014.

[3] M. Barbera, S. Kosta, A. Mei, V. Perta, and J. Stefa, "Mobile offloading in the wild: Findings and lessons learned through a real-life experiment with a new cloud-aware system," in *IEEE INFOCOM 2014*, pp. 2355–2363, 2014.

[4] S. Sardellitti, G. Scutari, and S. Barbarossa, "Joint optimization of radio and computational resources for multicell mobile-edge computing," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 1, no. 2, pp. 89–103, 2015.

[5] V. Di Valerio and F. Lo Presti, "Optimal virtual machines allocation in mobile femto-cloud computing: An mdp approach," in *IEEE Wireless Communications and Networking Conference Workshops (WCNCW)*, pp. 7–11, 2014.

[6] 3GPP, "Handover procedures," TS 23.009, 3rd Generation Partnership Project (3GPP), 9 2014.

[7] Z. Becvar, J. Plachy, and P. Mach, "Path selection using handover in mobile networks with cloud-enabled small cells," in *IEEE Personal, Indoor, and Mobile Radio Communication (PIMRC) 2014*, pp. 1480–1485, 2014.

[8] S. Wang, R. Urgaonkar, M. Zafer, T. He, K. Chan, and K. K. Leung, "Dynamic service migration in mobile edge-clouds," in *IEEE IFIP Networking Conference 2015*, pp. 1–9, 2015.

[9] S. Wang, R. Urgaonkar, K. Chan, T. He, M. Zafer, and K. K. Leung, "Dynamic service placement for mobile micro-clouds with predicted future costs," in *IEEE Communications (ICC) 2015*, pp. 5504–5510, 2015.

[10] A. Sadilek and J. Krumm, "Far out: Predicting long-term human mobility.," in *AAAI*, 2012.

[11] A. Hadachi, O. Batrashev, A. Lind, G. Singer, and E. Vainikko, "Cell phone subscribers mobility prediction using enhanced markov chain algorithm," in *IEEE Intelligent Vehicles Symposium 2014*, pp. 1049–1054, 2014.

[12] K. Wang, M. Shen, J. Cho, A. Banerjee, J. Van der Merwe, and K. Webb, "Mobiscud: A fast moving personal cloud in the mobile network," in *ACM Workshop on All Things Cellular: Operations, Applications and Challenges*, pp. 19–24, 2015.

[13] C. Mehlführer, J. C. Ikuno, M. Simko, S. Schwarz, M. Wrulich, and M. Rupp, "The vienna lte simulators-enabling reproducibility in wireless communications research.," *EURASIP EURASIP Journal on Advances in Signal Processing*, vol. 2011, p. 29, 2011.

[14] H. Li, S. Habibi, and G. Ascheid, "Handover prediction for long-term window scheduling based on sinr maps," in *IEEE Personal Indoor and Mobile Radio Communications (PIMRC) 2013*, pp. 917–921, 2013.

[15] 3GPP, "Technical Specification Group Technical Specification Group Radio Access Network; Evolved Universal Terrestrial Radio Access (E-UTRA);Physical layer procedures," TS 36.213, 3rd Generation Partnership Project (3GPP), 2016.

[16] I. Eyal, F. Junqueira, and I. Keidar, "Thinner clouds with preallocation.," in *HotCloud*, Citeseer, 2013.

[17] 3GPP, " Further advancements for E-UTRA physical layer aspects," TS 36.814, 3rd Generation Partnership Project (3GPP), 3 2010.

[18] K. Ha, Z. Chen, W. Hu, W. Richter, P. Pillai, and M. Satyanarayanan, "Towards wearable cognitive assistance," in *ACM Mobile systems, applications, and services*, pp. 68–81, 2014.

[19] M. Lauridsen, L. Noël, T. B. Sørensen, and P. Mogensen, "An empirical lte smartphone power model with a view to energy efficiency evolution," *Intel Technology Journal*, vol. 18, no. 1, pp. 172–193, 2014.