DYNAMIC JOINT RESOURCE ALLOCATION AND USER ASSIGNMENT IN MULTI-ACCESS EDGE COMPUTING

Mattia Merluzzi, Paolo Di Lorenzo, Sergio Barbarossa

DIET Department, Sapienza University of Rome, Via Eudossiana 18, 00184, Rome, Italy
E-mail: {mattia.merluzzi,paolo.dilorenzo,sergio.barbarossa}@uniroma1.it

ABSTRACT

Multi-Access Edge Computing (MEC) is one of the key technology enablers of the 5G ecosystem, in combination with the high speed access provided by mmWave communications. In this paper, among all services enabled by MEC, we focus on computation offloading, devising an algorithm to optimize computation and communication resources jointly with the assignment of mobile users to Access Points and Mobile Edge Hosts, in a dynamic scenario where computation tasks are continuously generated according to (unknown) random arrival processes at each user. To formulate and solve the dynamic allocation/assignment problem, we merge tools from stochastic optimization and matching theory, thus developing a low complexity algorithmic solution that works in an online fashion. Numerical results illustrate the potential advantages of the proposed approach.

***Index Terms*—** Multi-access edge computing, computation offloading, matching theory, stochastic optimization.

1. INTRODUCTION

Nowadays, we are in the middle of the so-called *fourth industrial revolution*, a drastic change of paradigm in which technology plays a key role, being at the center of this transformation. Some of the key points of this change are Internet of Things (IoT), automated vehicles, remote surgery, etc. The management of such complex system, with billions of interconnected devices, requires a new rethinking of the network, and finds its first solution in the fifth generation of mobile systems (5G), which is now in its second phase [1]. Differently from the previous evolution of mobile communication systems, the aim of 5G is to go beyond the simple enhancement of the physical layer, thus enabling new services from different application domains as, e.g., those aforementioned. To enable services with such diverse requirements, the new network has to be designed with high flexibility, in such a way that resources can be allocated when and where needed. This is possible thanks to network function virtualization and network slicing [2]. Moreover, to enable low latency services, Multi-Access Edge Computing is seen as a promising technology [3], thanks to the deployment of computation and storage resources at the edge of the network, in edge servers called Mobile Edge Hosts (MEH) using the ETSI terminology. The combination of MEC and mmWave communications is the main idea of the H2020 EU/JP project 5G-Miedge [4]. Among all possible services enabled by MEC, in this paper we focus on computation offloading, through which computationally heavy applications can be transferred from mobile terminals to a MEH, in order to reduce power consumption and/or to enable resource-contrained devices (e.g. sensors) to run sophisticated applications within strict latency constraints.

This work was funded by the H2020 EU-Japan Project 5G-MiEdge nr. 723171 and by Sapienza University.

Related works. The problem of resource allocation for computation offloading with MEC has received a lot of attention in the research community in the last few years [5–13]. In [5], the concept of joint optimization of radio and computation resources is presented and shown to be the best solution in terms of energy consumption. In [6], [8], computation offloading is investigated in a mmWave scenario, considering the effect of blocking events on the data rate and power consumption. A comprehensive survey on computation offloading with MEC can be found in [14]. Recently, computation offloading was also extended to a dynamic case [10–12], which is useful for those applications in which new tasks arrive at each time slot, according to a certain random process. In particular, in [10] the problem is formulated as the minimization of the long-term average power consumption under the constraint of mean-rate stability of the computation queues, in a single MEH scenario, i.e., without considering the assignment problem. In [11], the authors address the problem of mutual user association in a fog-enabled D2D network, with the aim of minimizing the long-term average energy consumption under computation queue stability constraints. In [12], a user assignment problem is cast as an energy-constrained delay minimization in a multiple AP's and MEH's scenario, but not considering allocation of computation/communication resources.

Contributions. In this paper, we focus on a dynamic computation offloading problem, considering continuously demanding applications, and performing the optimization in a per-slot fashion. In particular, differently from [10–12], we consider a scenario with *multiple* AP's and MEH's, addressing *jointly* the problem of user assignment and allocation of radio/computation resources. We introduce a novel algorithmic framework that hinges on stochastic optimization [10] to deal with the dynamic nature of offloading requests, channel states, computation queues, etc. User mobility is also taken into account in the problem by reducing frequent handovers through a suitable penalty function. In principle, the method would require the solution of a mixed integer nonlinear program (MINLP) per time slot. To find simple solutions amenable for online implementation, the user assignment is handled using tools from matching theory [15]. Once the assignment is chosen at each time slot, the optimal allocation of computation/communication resources is then given in closed form. The proposed method is able to efficiently handle users' mobility, and naturally implements a balance of communication/computation load among the multiple AP's/MEH's. Finally, numerical results illustrate the advantages of the proposed strategy for dynamic computation offloading with MEC.

2. PROBLEM FORMULATION

Let us consider a scenario with K UE's and N AP's, each one associated with its corresponding MEH. Each UE is allowed to access a single AP and can offload some computation tasks to its associated MEH. Time is divided in slots of equal duration τ. Then, at time

slot t, the association of user k to the pair AP-MEH n is described by the binary variable $a_{kn}(t)$. In particular, denoting by $\mathcal{S}_n(t)$ the set of users assigned to AP n during time slot t (also called *coalition* later on), we have $a_{kn}(t) = 1$ if $k \in \mathcal{S}_n(t)$, whereas $a_{kn}(t) = 0$ otherwise. Considering the radio access part, $h_{kn}(t)$ is the (random) channel coefficient between UE k and AP n, $p_k(t)$ denotes the transmit power for uplink transmission, and $\beta_{kn}(t)$ represents the fraction of bandwidth allocated to user k during time slot t. Then, the maximum transmission rate during time slot t is given by:

$$R_{kn}(t) = \beta_{kn}(t) B \log_2 \left(1 + \frac{h_{kn}(t) p_k(t)}{\beta_{kn}(t) N_0 B}\right), \tag{1}$$

where N_0 is the noise power spectral density, and B is the available bandwidth. Concerning the computation part, let us denote with $f_k^l(t)$ the local CPU cycle frequency at UE k. The dynamic amount of bits associated with computation tasks of UE k evolves according to the following queue backlog rule:

$$Q_k^l(t+1) = \max\Bigg(Q_k^l(t) - \tau f_k^l(t) J_k - \tau \sum_{n=1}^{N} a_{kn}(t) R_{kn}(t), 0\Bigg) + A_k(t), \tag{2}$$

where $Q_k^l(t)$ is the local queue backlog at time t; $A_k(t)$ is the random arrival process of bits associated with computation tasks; J_k represents the number of bits processed within one CPU cycle (a parameter that depends on the specific application); thus, the term $\tau f_k^l(t) J_k$ denotes the amount of bits processed locally at user k during the time slot; finally, the term $\tau \sum_{n=1}^{N} a_{kn}(t) R_{kn}(t)$ represents the overall number of bits transmitted by UE k to the assigned AP n (i.e., the one such that $a_{kn}(t) = 1$) to enable computation offloading. We assume that computation tasks can be arbitrarily split between local and remote processing. Note that our approach subsumes also the cases where applications are entirely run either locally or remotely. Since part of the bits in the local queue $Q_k^l(t)$ are transferred for computation offloading [cf. (2)], the assigned MEH maintains a remote queue backlog, say, $Q_k^r(t)$, which quantifies the number of bits associated with offloaded computation tasks of UE k. Letting $f_{kn}(t)$ be the CPU cycle frequency allocated to UE k by MEH n, the remote queue $Q_k^r(t)$ evolves as:

$$Q_k^r(t+1) = \max\left(Q_k^r(t) - \tau \sum_{n=1}^{N} a_{kn}(t) f_{kn}(t) J_k, 0\right) + \min\left(\max\left(Q_k^l(t) - f_k^l(t) J_k \tau, 0\right), \tau \sum_{n=1}^{N} a_{kn}(t) R_{kn}(t)\right) \tag{3}$$

where the term $\tau \sum_{n=1}^{N} a_{kn}(t) f_{kn}(t) J_k$ quantifies the number of bits of UE k processed by the assigned MEH n during the time slot; the second term in the RHS of (3) represents the arrival rate.

In this paper, our aim is to find the joint assignment strategy and dynamic allocation of computation/communication resources in order to minimize the long-term average power consumption of all UE's under queue stability constraints. The power consumption of UE k is given by the sum of the transmit power $p_{kn}(t)$ and the power spent for local computation $p_k^l(t) = \gamma_k f_k^l(t)^3$, where γ_k is the effective switched capacitance of the CPU [16]. Then, we can define the total power consumption of UE k as:

$$p_k^{\text{tot}}(t) = p_k(t) + \gamma_k f_k^l(t)^3. \tag{4}$$

Thus, the dynamic joint assignment/allocation problem can be formulated mathematically as follows:

$$\begin{aligned}
\min_{\mathbf{\Psi}(t)} \ & \lim_{t\to\infty} \frac{1}{t} \sum_{n=1}^{N} \sum_{\tau=0}^{t-1} \sum_{k=1}^{K} \mathbb{E}\left\{p_k^{\text{tot}}(\tau) + \sigma \cdot \tilde{a}_{kn}(\tau; \tau-1)\right\} \\
\text{s.t} \quad & (a) \quad \lim_{T\to\infty} \frac{\mathbb{E}\big[Q_k^l(T)\big]}{T} = 0, \ \ \forall k; \\
& \qquad \lim_{T\to\infty} \frac{\mathbb{E}[Q_k^r(T)]}{T} = 0, \ \ \forall k; \\
& (b) \quad 0 \le p_k(t) \le P_k, \qquad \forall k, t; \\
& (c) \quad 0 \le \beta_{kn}(t) \le 1, \qquad \forall k, n, t; \\
& (d) \quad \sum_{k=1}^{K} a_{kn}(t) \beta_{kn}(t) \le 1, \quad \forall n, t; \\
& (e) \quad a_{kn}(t) \in \{0, 1\}, \qquad \forall k, n, t; \\
& (f) \quad \sum_{n=1}^{N} a_{kn}(t) = 1, \qquad \forall k, t; \\
& (g) \quad 0 \le f_{kn}(t) \le F_n, \qquad \forall k, n, t; \\
& (h) \quad \sum_{k=1}^{K} a_{kn}(t) f_{kn}(t) \le F_n, \quad \forall n, t; \\
& (i) \quad 0 \le f_k^l(t) \le F_k^l, \quad \forall k, t;
\end{aligned} \tag{5}$$

where $\mathbf{\Psi}(t) = \big[\{p_k(t)\}_k, \{f_{kn}(t)\}_{k,n}, \{f_k^l(t)\}_k, \{a_{kn}(t)\}_{k,n}, \{\beta_{kn}(t)\}_{k,n}\big]$; P_k and F_k^l are the maximum power budget and the maximum local CPU cycle frequency of user k, respectively; whereas F_n is the maximum CPU cycle frequency of MEH n. The quantity $\tilde{a}_{kn}(t; t-1)$ in (5) is a penalty function defined as:

$$\tilde{a}_{kn}(t; t-1) = (a_{kn}(t) - a_{kn}(t-1))^2, \tag{6}$$

whose aim is to reduce the number of handovers in the dynamic process, and $\sigma > 0$ is a penalty parameter. Indeed, note that choosing the association in each time slot can result in frequent changes of the assignment variables, i.e., handovers between pairs AP-MEH. This not only increases the complexity due to the necessary control signaling for AP switch, but it can also incur in an additional delay on the execution of the application due to the necessity of transferring the state of the application from the old MEH to the new one.

The constraints in (5) have the following meaning: (a) The computation queues are mean rate stable; (b) The transmission power is non negative and does not exceed the power budget of the device; (c) Each user is given a portion of the bandwidth; (d) The sum of the bandwidth portions given to all users does not exceed the unity; it is also assumed that different AP's operate in different frequency bands to avoid interference; (e) The assignment variables are binary; (f) Each user is associated to only one pair AP-MEH; (g) The CPU cycle frequency allocated by each MEH to users is non negative and does not exceed its maximum value; (h) The total number of CPU cycles assigned by the MEH does not exceed a maximum value; finally, (i) ensures that the local CPU cycles frequency is non negative and does not exceed its maximum value. Note that, by Little's law, the average queue length [cf. constraint (a) in (5)] and the average queueing delay are strictly linked through the arrival rate [17]. Thus, the solution of problem (5) aims at striking an optimal tradeoff between power consumption and average delay for processing tasks via computation offloading. In the next section, we will introduce a low-complexity online algorithm to solve (5), merging tools from stochastic optimization and matching theory with transfers.

3. ALGORITHM DEVELOPMENT

To solve the dynamic problem (5), we build on stochastic optimization as in [18]. Thus, given a system with K users, whose queue backlogs evolve as in (2)-(3), we define a Lyapunov function as:

$$L(\mathbf{\Theta}(t)) = \frac{1}{2}\sum_{k=1}^{K}\left[Q_k^l(t)^2 + Q_k^r(t)^2\right] \tag{7}$$

where $\mathbf{\Theta}(t) = \left[\{Q_k^l(t)\}_k, \{Q_k^r(t)\}_k\right]$. We can now define the *one-slot conditional Lyapunov drift* as:

$$\Delta(\mathbf{\Theta}(t)) \triangleq \mathbb{E}\{L(\mathbf{\Theta}(t+1)) - L(\mathbf{\Theta}(t))|\mathbf{\Theta}(t)\} \tag{8}$$

where the expectation depends on the control policy, and is taken with respect to the random channels and arrival rates. Minimizing (8) would stabilize the queues, but it can lead to an unnecessarily large power expenditure and number of handovers. For this reason, we introduce the so called *drift-plus-penalty* function as [18]:

$$\begin{aligned}\Delta_p(\mathbf{\Theta}(t)) &= \Delta(\mathbf{\Theta}(t)) \\ &+ V\cdot\mathbb{E}\left\{\sum_{n=1}^{N}\sum_{k=1}^{K}\left(p_k^{\text{tot}}(t) + \sigma\cdot\tilde{a}_{kn}(t;t-1)\right)\Big|\mathbf{\Theta}(t)\right\}\end{aligned} \tag{9}$$

where V is a control parameter used to trade-off power consumption and number of handovers with queues length. Following arguments as in [18], we first derive an upperbound of the drift-plus-penalty function in (9), and then we proceed by greedily minimize instantaneous values of such upper bound, thus obtaining the control policy dictated by the following dynamic optimization:

$$\begin{aligned}\min_{\mathbf{\Psi}(t)}\quad & V\cdot\sum_{n=1}^{N}\sum_{k=1}^{K}\left(p_k^{\text{tot}}(t) + \sigma\cdot\tilde{a}_{kn}(t;t-1)\right) \\ & -\tau\cdot\sum_{k=1}^{K}Q_k^l(t)\left[f_k^l(t)J_k + \sum_{n=1}^{N}a_{kn}(t)R_{kn}(t)\right] \\ & -\tau\cdot\sum_{k=1}^{K}Q_k^r(t)\sum_{n=1}^{N}a_{kn}(t)\left(f_{kn}(t)J_k - R_{kn}(t)\right) \\ \text{subject to}\quad & \mathbf{\Psi}(t)\in\mathcal{Z}\end{aligned} \tag{10}$$

where $\mathcal{Z}$ is the feasible set of control actions for problem (5), according to constraints $(b)-(i)$. The optimization in (10) requires the solution of a mixed integer nonlinear program (MINLP) at each time slot, whose optimal solution might be too complex to be computed even for a moderate number of UE's and AP's, especially in the dynamic context considered in this paper. As a consequence, to find a low-complexity solution that is amenable for online implementation, we propose a procedure that splits the solution of (10) in two parts. As we illustrate in the sequel, first the assignment problem is solved using tools from matching theory with transfers [15]; then, for a given assignment of UE's to AP-MEH pairs, the optimal resource allocation is provided in closed form. To build the utility functions used by the matching algorithm, it is useful to specify the resource allocation for a given assignment, as shown in the sequel.

3.1. Optimal Allocation of Radio and Computation Resources

Let us assume for the moment that the assignment of UE's to AP-MEH pairs has been selected for time slot t, i.e., all the coalitions $\mathcal{S}_n(t)$ have been chosen (cf. Sec. 3.2). The first task at each UE is the optimal allocation of the local CPU cycle frequencies. In particular, solving (10) with respect to $\{f_k^l(t)\}_k$, it is easy to obtain:

$$f_k^l(t) = \min\left\{F_k^l\ ,\ \sqrt{\frac{Q_k^l(t)J_k\tau}{3\gamma_k V}}\right\},\quad \text{for all } k. \tag{11}$$

The second subtask is the optimal bandwidth and power allocation for each UE/AP wireless link. Let us denote by $\widetilde{\mathcal{S}}_n(t)$ the set of users assigned to the pair AP-MEH n at time t, such that $Q_k^l(t) > Q_k^r(t), \forall k\in\mathcal{S}_n(t)$. To find a simple closed form solution useful in online implementations, we assume the bandwidth is equally shared by the users belonging to $\widetilde{\mathcal{S}}_n(t)$, i.e. $\beta_{kn}(t) = |\widetilde{\mathcal{S}}_n(t)|^{-1}$, if $k\in\widetilde{\mathcal{S}}_n(t)$, and 0 otherwise, where $|\mathcal{S}|$ denotes the cardinality of set $\mathcal{S}$. Then, from (10), the optimal power allocation writes as:

$$p_k(t) = \begin{cases}\min\left[\max\left(\zeta_{kn}(t),0\right),P_k\right], & \text{if } k\in\widetilde{\mathcal{S}}_n(t);\\ 0, & \text{otherwise;}\end{cases} \tag{12}$$

where

$$\zeta_{kn}(t) = \beta_{kn}(t)B\left[\frac{\left(Q_k^l(t) - Q_k^r(t)\right)\tau}{\log(2)\cdot V} - \frac{N_0}{h_{kn}(t)}\right]. \tag{13}$$

Finally, we consider the optimal scheduling at each MEH n. Since (10) is a linear problem with respect to the $\{f_{kn}(t)\}_{k,n}$, its optimal solution is always at the vertex of the polyhedral set [19]. In this case, we have:

$$f_{kn}(t) = \begin{cases}F_n, & \text{if } k = \underset{k\in\mathcal{S}_n(t)}{\text{argmax}}\{Q_k^r(t)J_k\};\\ 0, & \text{otherwise.}\end{cases} \tag{14}$$

This means that only the UE with the maximum value of $Q_k^r(t)J_k$ will be served by MEH n during time slot t.

3.2. UE's Assignment via Matching Theory

The user assignment strategy is based on matching theory for college admission [15], which implements a coalitional game with low complexity. In this case, we are dealing with a *many-to-one* matching problem, in which one agent (AP-MEH pair) can be associated to more than one agent from the other set (UE). The maximum number of UE's that can be assigned to an AP-MEH pair is called *quota*. Each UE builds a preference list with the aim of ranking all the AP-MEH pairs. The scope of matching theory is then to find a *stable* assignment between the two sets of agents [20], i.e., it must not exist a situation where two UE's α and β are assigned to AP-MEH pairs A and B, respectively, but β prefers A to B and A prefers β to α. The so called Deferred Acceptance (DA) algorithm solves this problem and converges to a stable matching with polynomial complexity [20]. However, when the preference functions of UE's are interdependent[1], the DA algorithm looses its efficiency. To cope with this issue, as recently proposed in [13, 21], our approach is based on two games, whose rationale is illustrated in the following.

3.2.1. Matching game with fixed utility functions

The first game is played only at the first time slot, and is based on the DA algorithm, with *expected* utility functions. To define the utilities

[1] In our case, this is true due to radio and computation resource sharing.

in the first slot, since each user is not aware of all other UE's utilities, these function are built assuming that every AP-MEH fills up its quota. In this way, for each UE, it is possible to evaluate the optimal resource allocation for each pair AP-MEH according to (11), (12), and (14), and thus evaluate the objective function in (10). Let us then define as G_{kn} the value of the objective in (10) obtained by UE k by accessing the pair AP-MEH n; and let $U_{kn} = -G_{kn}$ denote the corresponding utility associated with the pair k and n. Note that, in this first slot game, the penalty function in (6) is not considered in the user's utilities, since the assignment process is at its first stage. On the other side, the preference functions of the pairs AP-MEH are based on the Signal to Noise Ratio (SNR) of UE's, i.e., users will be ranked in descending order with respect to their SNR. Then, based on the aforementioned utility functions defined at both sets of agents (i.e., UE's and AP-MEH pairs), the network runs the DA algorithm. Since we have fixed the utility functions, this first phase converges to a stable matching, although this assignment can be very unbalanced, since users do not know a priori which coalition other users will belong to at the end of the procedure. This situation might lead to poor performance since many users will share radio and computation resources when assigned to the same AP-MEH pairs.

3.2.2. *Coalitional game for transfers among AP's*

To enhance the system performance, in each time slot after the first one, we perform a coalitional game, where UE's can request to be *transferred* to other AP's in order to improve their utilities. Denoting by Π_0 the initial assignment, i.e., the set of all coalitions $\mathcal{S}_n,\ \forall n$ after the first game, we define the welfare of a coalition as [21]:

$$W(\mathcal{S}_n) = \sum_{k \in \mathcal{S}_n} U_{kn}. \tag{15}$$

Given Π_0, each user builds the new utility functions based on this conditions in the same way as described for the first game. Once the preference functions are computed, a generic user k requests to be transferred from coalition $\mathcal{S}_n$ to coalition $\mathcal{S}_{n'}$ if its utility function is improved by the transfer, i.e., if $U_{kn'} > U_{kn}$. Once all the transfers are requested, the pair AP-MEH involved in each transfer can either accept or refuse it. In particular, a transfer is accepted if and only if the target pair (i.e., the one accepting the new user) does not exceed its quota and the social welfare of the two coalitions is improved. In formulas, denoting by k the requesting user, by n the source pair and by n' the target pair, these two conditions can be cast as:

1. $|\mathcal{S}_{n'} \cup \{k\}| \leq q_{n'}$;
2. $W(\mathcal{S}_n \setminus \{k\}) + W(\mathcal{S}_{n'} \cup \{k\}) > W(\mathcal{S}_n) + W(\mathcal{S}_{n'})$;

where $q_{n'}$ is the quota of n', and $\mathcal{S}_n \setminus \{k\}$ is the coalition obtained by removing user k from $\mathcal{S}_n$. This transfer phase stops if there are no more requests, or no more transfers can be accepted by the network. If two users request to be transferred to the same coalition, only the user with the highest SNR is taken into consideration. Interestingly, this second game can be proved to converge to a Nash stable partition [21]. This coalitional game is used in each time slot to perform transfers (i.e., handovers) whenever a user finds a better coalition according to the defined utilities. This procedure has a low complexity and is amenable for real-time implementations.

4. NUMERICAL RESULTS

In this section, we present some numerical results to assess the performance of the proposed algorithm when compared to a classical SNR-based association, i.e., where each user is assigned to the AP with the highest SNR. We use the channel pathloss model between UE k and AP n considered in [22]. UE's and AP's are equipped with arrays composed of $N_{tx} = 4$ antennas and $N_{rx} = 144$ antennas, respectively. The average arrival rate in (2) is 2×10^6 bits per slot, while the slot duration is set to 10 ms. The number J_k of bits per CPU cycle is 10^{-1}, for all k; the maximum CPU cycle frequencies of the UE's and the MEH's are 10^9 and 5×10^9 CPU cycles/s, respectively. We consider 10 users moving randomly in a squared area of size 150 m, and 6 AP's randomly deployed in the same area. Whenever a handover occurs, we assume that the application state is migrated over the destination AP during the subsequent N_{tr} time slots, so that during this time, only local computations are allowed. Figure 1 shows the tradeoff between the sum queue length [i.e., sum of local and remote queues in (2) and (3)] and the power consumption over all users, averaged over 100 simulations, considering two assignment strategies: the proposed one based on matching theory [cf. Sec. 3.2], and the classical SNR-based association. For any combination of V and N_{tr}, we have chosen the penalty parameter σ in (5) that gives the best empirical results. As expected, from Fig. 1, we notice that the performance decreases for higher values of N_{tr}, due to the larger delay required for migrating the computation after a handover occurs. However, we can appreciate the large gain achieved by our matching algorithm with respect to the SNR-based association. This is also due to the fact that our matching strategy keeps into account the handover cost thanks to the penalty function in (6), thus reducing the number of unnecessary handovers.

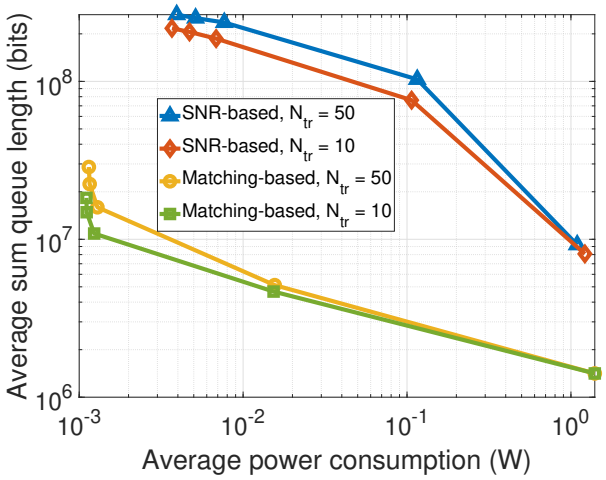

Fig. 1: Average behavior of Sum queue length versus sum power, for different assignment strategies and values of N_{tr}.

5. CONCLUSIONS

In this paper we have studied dynamic user assignment and resource allocation for computation offloading with MEC. We have introduced an online algorithmic framework that hinges on stochastic optimization to deal with the dynamic nature of the system, and performs user association using matching theory with transfers, in order to strike the best trade-off between power consumption and queue length (i.e., average delay). Numerical simulations assess the performance of the proposed approach, and illustrate the potential gain with respect to a classical SNR-based association in terms of long-term average power consumption and average queue length, considering a dynamic scenario with users' mobility.

6. REFERENCES

[1] "3GPP release 15," *http://www.3gpp.org/release-15*.

[2] B. Chatras, U. S. Tsang Kwong, and N. Bihannic, "NFV enabling network slicing for 5G," in *Proc. of Conference on Innovations in Clouds, Internet and Networks (ICIN)*, March 2017, pp. 219–225.

[3] "Etsi multi-access edge computing," *https://www.etsi.org/technologies-clusters/technologies/multi-access-edge-computing*.

[4] "5G-MiEdge millimeter-wave edge cloud as an enabler for 5G ecosystem," *Europe/Japan project co-funded by the European Commission's Horizon 2020 and Japanese Ministry of Internal Affairs and Communications; website: http://5g-miedge.eu*.

[5] S. Barbarossa, S. Sardellitti, and P. Di Lorenzo, "Communicating while computing: Distributed mobile cloud computing over 5G heterogeneous networks," *IEEE Signal Process. Mag.*, vol. 31, no. 6, pp. 45–55, Nov. 2014.

[6] S. Barbarossa, E. Ceci, M. Merluzzi, and E. Calvanese-Strinati, "Enabling effective mobile edge computing using millimeter-wave links," in *Proc. of IEEE Int. Conf. Commun. Work. (ICC Wkshps)*, May 2017, pp. 367–372.

[7] W. Labidi, M. Sarkiss, and M. Kamoun, "Energy-optimal resource scheduling and computation offloading in small cell networks," in *Proc. of 2015 22nd International Conference on Telecommunications (ICT)*, Sydney, NSW, Australia 2015, pp. 313–318.

[8] S. Barbarossa, E. Ceci, and M. Merluzzi, "Overbooking radio and computation resources in mmw-mobile edge computing to reduce vulnerability to channel intermittency," in *Proc. of 2017 Eur. Conf. Net. Commun. (EuCNC)*, Jun. 2017, pp. 1–5.

[9] Y. Mao, J. Zhang, and K. B. Letaief, "Dynamic computation offloading for mobile-edge computing with energy harvesting devices," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 12, pp. 3590–3605, Dec 2016.

[10] Y. Mao, J. Zhang, S. H. Song, and K. B. Letaief, "Stochastic joint radio and computational resource management for multi-user mobile-edge computing systems," *IEEE Trans. Wireless Commun.*, vol. 16, no. 9, pp. 5994–6009, Sept 2017.

[11] Y. Yang, S. Zhao, W. Zhang, Y. Chen, X. Luo, and J. Wang, "Debts: Delay energy balanced task scheduling in homogeneous fog networks," *IEEE Internet of Things Journal*, vol. 5, no. 3, pp. 2094–2106, June 2018.

[12] Y. Sun, S. Zhou, and J. Xu, "EMM: Energy-aware mobility management for mobile edge computing in ultra dense networks," *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 11, pp. 2637–2646, Nov 2017.

[13] S. Sardellitti, M. Merluzzi, and S. Barbarossa, "Optimal association of mobile users to multi-access edge computing resources," in *Proc. of 2018 IEEE International Conference on Communications Workshops (ICC Workshops)*, May 2018, pp. 1–6.

[14] P. Mach and Z. Becvar, "Mobile edge computing: A survey on architecture and computation offloading," *IEEE Communications Surveys Tutorials*, vol. 19, no. 3, pp. 1628–1656, 2017.

[15] Zhu Han, Yunan Gu, and Walid Saad, *Matching Theory for Wireless Networks*, Springer Publish. Comp., Inc., 1st edition, 2017.

[16] Thomas D. Burd and Robert W. Brodersen, "Processor design for portable systems," *J. VLSI Signal Process. Syst.*, vol. 13, no. 2-3, pp. 203–221, Aug. 1996.

[17] John DC Little, "Little's law as viewed on its 50th anniversary," *Operations research*, vol. 59, no. 3, pp. 536–549, 2011.

[18] Michael J. Neely, *Stochastic Network Optimization with Application to Communication and Queueing Systems*, Morgan and Claypool Publishers, 2010.

[19] S. Boyd and L. Vandenberghe, *Convex optimization*, Cambridge university press, 2004.

[20] D. Gale and L. S. Shapley, "College admissions and the stability of marriage," *The Amer. Math. Month.*, vol. 69, no. 1, pp. 9–15, 1962.

[21] W. Saad, Z. Han, R. Zheng, M. Debbah, and H. V. Poor, "A college admissions game for uplink user association in wireless small cell networks," in *Proc. of IEEE Conf. Comput. Commun. (INFOCOM 2014)*, Apr. 2014, pp. 1096–1104.

[22] K. Sakaguchi, G. K. Tran, H. Shimodaira, S. Nanba, T. Sakurai, K. Takinami, I. Siaud, E. Calvanese Strinati, A. Capone, I. Karls, R. Arefi, and T. Haustein, "Millimeter-wave evolution for 5G cellular networks," *CoRR*, vol. abs/1412.3212, 2014.