Service migration versus service replication in Multi-access Edge Computing

Pantelis A. Frangoudis and Adlen Ksentini
EURECOM, Communication Systems Dept., Sophia Antipolis, France
{firstname.lastname}@eurecom.fr

***Abstract*—Envisioned low-latency services in 5G, like automated driving, will rely mainly on Multi-access Edge Computing (MEC) to reduce the distance, and hence latency, between users and the remote applications. MEC hosts will be deployed close to mobile base stations, constituting a highly distributed computing platform. However, user mobility may raise the need to migrate a MEC application among MEC hosts to ensure always connecting users to the optimal server, in terms of geographical proximity, Quality of Service (QoS), etc. However, service migration may introduce: (i) latency for users due to the downtime duration; (ii) cost for the network operator as it consumes bandwidth to migrate services. One solution could be the use of service replication, which pro-actively replicates the service to avoid service migration and ensure low latency access. Service replication induces cost in terms of storage, though, requiring a careful study on the number of service to replicate and distribute in MEC. In this paper, we propose to compare service migration and service replication via an analytical model. The proposed model captures the relation between user mobility and service duration on service replication as well as service migration costs. The obtained results allow to propose recommendations between using service migration or service replication according to user mobility and the number of replicates to use for two types of service.**

***Index Terms*—Mobile and Multi-access Edge Computing, 5G, Markov chains**

I. Introduction

Mobile or Multi-access Edge Computing (MEC) [1] is seen as one of the key enablers of 5G services, particularly for those requiring ultra Reliable Low Latency Communication (uRLLC), such as automotive and industry 4.0 services. MEC allows reducing access latency to remote services (computation and storage), by hosting them at the network edge. Therefore, rather than being connected to remote servers in the cloud, where the average Round Time Trip (RTT) is around 200 ms, using MEC allows to reduce this latency by a factor of 4 [2]. Such reduction of latency enables services like automated driving to quickly react to events reported by sensors enrolled in the vehicle.

MEC is being standardized by ETSI [3], where a first architecture has been released. Several functional blocks and interfaces have been defined. However, user mobility raises the challenge of migrating the service among the MEC servers (or *Follow Me Edge*) in order to allow users to be always connected to the closed MEC server, so as to keep low-latency access to the service. Service migration has been explored in the context of Follow Me Cloud [4], [5], where cloud services are migrated among Data Centers (DC) according to user mobility. While in the case of cloud services the downtime of the service due to service migration could be tolerated, this represents a challenge in the case of MEC, as it should be avoided as much as possible for uRLLC services. One solution introduced in [6] is to employ proactive service replication in order to reduce the service migration events. Similarly to the principle of proactive caching, the authors proposed to replicate the service on neighbor MEC servers. They solve a multi-criteria optimization problem, deciding at each user handover event where to duplicate the service by minimizing both the probability of service migration and the number of duplicate instances of a service. Their solution is interesting but at the same time challenging to deploy in practice, since the optimization problem should be solved at each handover. In addition, their model has specific limitations: it does not integrate the user service type (e.g., duration) nor the number of active users in the system.

In this paper, we address such limitations and present an analytical model based on Markov chains to capture the relation between the cost of service replication and the cost of service migration regarding (i) the service duration, (ii) the size of area where the service is replicated, (iii) the number of users, and (iv) the user mobility.

This paper is organized as follows: Section II gives an overview on the MEC architecture and the state of the art. Section III introduces our models. Our results are presented in Section IV. Finally, the paper is concluded in Section V.

II. Related work

MEC is deemed as a critical technology to enable the transition toward 5G. It enables the deployment of two types of service: (i) services requiring low-latency access and real-time reaction to events, such as automated driving applications; (ii) services that require information on user context (e.g., radio channel quality) and adapt the service accordingly, such as a video transcoding service which may adapt the video bitrate according to users' radio quality. So far, there is no clear indication where the MEC servers (hosted in MEC hosts) will be located in the mobile network. Some implementations, such as in [2] consider placing the MEC host (i.e., the physical machine hosting the services) at the end of the tunnel established between the eNodeB and the S/P-GW (gateway to the Internet), whereas some proposals consider to locate MEC servers close to the eNodeB. In all the cases, Software-Defined Networking (SDN) will be used

to dynamically offload user traffic to the ME host as per the operator's or the application's request, using appropriate APIs exposed to the MEC application. Moreover, the number of MEC hosts to deploy in the mobile network is still open, and may depend drastically on the operator policy, the number of users to cover, etc. Nevertheless, it is expected to have one ME host covering a small area (a certain number of eNodeBs) in order to: (i) be close to users, hence reducing access latency; (ii) limit the overhead caused by the gathered radio information to treat and expose to MEC applications. One major concern for MEC is the management of user mobility. Indeed, since users are mobile, they may move away from the serving MEC host, leading to an increased end to end latency and hence user quality of experience (QoE) [7] degradation when using uRLLC services. With a good dimensioning mechanism, the moving UE may approach an area covered by another MEC host, where it deems appropriate to migrate its MEC application from the distant MEC host to the new one. The concept of migrating a service according to user mobility is not new. It has been introduced in [3], under the name of Follow Me Cloud (FMC). The goal is that services follow the mobile users. Each time the UE has a new data anchor gateway (could be a P-GW or mobile IP router), which gives access to a better data center (in terms of geographical proximity or QoS), the service is migrated between the two DCs. Although FMC could be very relevant in the context of MEC, as MEC hosts are expected to be more present at local DCs, migration may be costly for the operator in terms of network overhead, and it may introduce service disruption due to the downtime of the service during the migration. The latter may disturb considerably the low latency services. One solution would be to use a service replication (or duplication) mechanism, which may prove very efficient particularly if the application is highly popular and shared by a large number of users. The service replication concept has been studied for VM placement to ensure high availability [8], and to reduce the duration of service migration in [9]. Furthermore, the concept of service duplication could be considered similar to CDN caching [10], where the service is deployed in a proactive manner, contrariwise to service migration, which is more reactive. However, the cost of service duplication could be high as it requires more storage than the case of service migration. Therefore, a mechanism to find a trade-off is highly necessary to optimize MEC resource utilization, while ensuring low latency access to the service. In [6] the authors proposed a multi-criteria optimization problem formulation to decide at each handover where to duplicate the service, minimizing the probability of service migration and the number of duplicate instances of the service. While this approach seems to be promising, it is difficult to deploy in practice, as the optimization problem should be solved at each user handover.

III. Model

A. Service replication and migration model

The proposed model aims to capture the impact of user mobility and service duration on proactive service deployment, as well as on service migration. For the mobility model, we consider a 2D hexagonal deployment of a cellular wireless network as depicted in Figure 1. We assume that a cluster includes one or several cells, where each cell has its own MEC host. The number of rings belonging to a cluster indicates the number of MEC hosts where the service is pro-actively deployed. For instance, if a cluster includes ring 0 and ring 1, it means that the service has been deployed in 7 MEC hosts. We note by k the number of rings belonging to a cluster. Therefore, the number of duplicate services per user in the cluster is $1 + \sum_{i=1}^{k-1} 6i$.

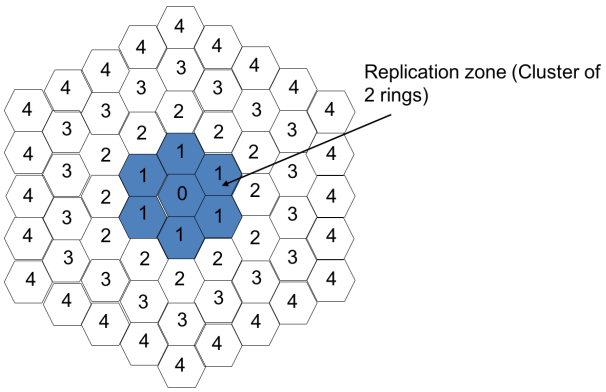


Fig. 1: The clustering model.

We assume that n users have started a session with an application duplicated on the MEC hosts belonging to a cluster. Our objective is to capture the dynamics of users and their impact on service migration and duplication, that is, to derive the number of users that have moved to another cluster without ending the service (hence they require a service migration from one cluster to another), and the number of users that have finished the service while being in the same initial cluster. Accordingly, we can calculate the cost of service replication and service migration based on user mobility as well as service duration. Let us assume that each user consumes a MEC application for a duration that follows an exponential distribution with rate μ_s. We assume that a user resides in cluster 0 for a duration following an exponential distribution with rate μ_m. The residence time of a user in a cluster will be described in the next section. The above assumptions conduct us to model the system using a Markov chain $\{X_t, t \geq 0\}$ on the state space $S = \{(i,j) | i = 0, \ldots, n \text{ and } j = 0, \ldots, n-i\}$, for every $n \geq 1$. $X_t = (i,j)$ means that, at time t, there are i UEs using a MEC-hosted service which remain in cluster 0, and j users that have left the initial cluster. The latter case implies that a service migration is needed. Fig. 2 illustrates the transition graph of the system. We notice that the chain contains absorbing states, which indicate that the initial N users have either left the cluster or finished the service within the same cluster. For example, the state $(0,k)$ means that k users have left the cluster before the end of the service, while $n-k$ users have finished the service while being in the same initial cluster. This design allows us to know the expected number of active users that have left the initial cluster, hence

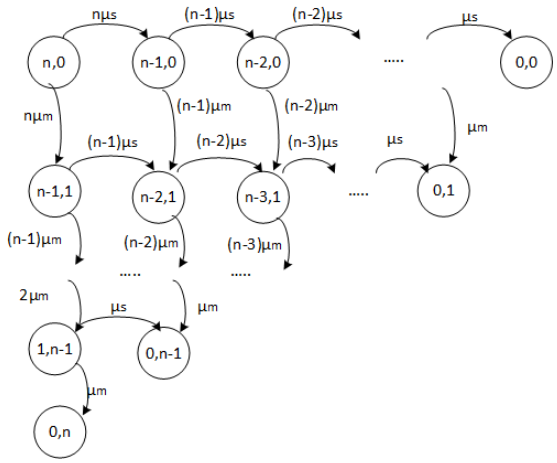


Fig. 2: Markov chain corresponding to our clustering model.

generating migration cost. The different transitions are as follows:

- If a user has finished his service in cluster 0, there is a transition from $(i+1,j)$ to (i,j) with rate $(i+1)\mu_s$
- If a user hands off to another cluster, there is a transition from (i,j) to $(i,j+1)$ with rate $(n-i)\mu_m$

We denote by Q_B the transition matrix between the non-absorbing states. It is worth noting that this matrix does not represent the infinitesimal generator of the chain. Based on Fig. 2, we decompose the chain in sub-chains, according to the level where states belong to. Level 1 corresponds to all the states $(i,0)$, level 2 is composed of states $(i,1)$, etc. Q_B is obtained as follows:

$$Q_B = \begin{bmatrix} A & AB & \dots & 0 & 0 \\ 0 & B & BC & \dots & 0 \\ 0 & 0 & C & CD & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & 0 & Z \end{bmatrix}$$

where $A(n,n)$, $B(n-1,n-1)$, $C(n-2,n-2)$, $\dots$, $Z(1,1)$ are the matrices containing the transitions between non absorbing states of level 1 (respectively, level 2, level 3, $\dots$, level n+1), and $AB(n,n-1)$, $BC(n-1,n-2)$, $CD(n-2,n-3)$, etc. are the transition matrices between non-absorbing states of level 1 and level 2 (respectively, level 2 and level 3, level 3 and 4, etc.). A, B, C, and in general all intra-level matrices, have the same structure; their main difference is about the size of the matrix and the rates of the transitions. Z contains only one element, which is equal to $-(\mu_m+\mu_s)$. For example, A and B are defined as follows:

$$A = \begin{bmatrix} -n(\mu_s+\mu_m) & n\mu_s & 0 & \dots & 0 \\ 0 & -(n-1)(\mu_s+\mu_m) & (n-1)\mu_m & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & -(\mu_m+\mu_s) \end{bmatrix}$$

$$B = \begin{bmatrix} -(n-1)(\mu_s+\mu_m) & (n-1)\mu_s & 0 & \dots & 0 \\ 0 & -(n-2)(\mu_s+\mu_m) & (n-2)\mu_m & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & -(\mu_m+\mu_s) \end{bmatrix}$$

On the other hand, AB, BC, CD, and in general each inter-level matrix, have the same structure. As in the case of intra-level matrices, the differences are in the size of the matrices and the transition rates; see, for example, AB and BC:

$$AB = \begin{bmatrix} n\mu_m & 0 & 0 & 0 \\ 0 & (n-1)\mu_m & 0 & 0 \\ \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & 2\mu_m \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$BC = \begin{bmatrix} (n-1)\mu_m & 0 & 0 & 0 \\ 0 & (n-2)\mu_m & 0 & 0 \\ \dots & \dots & \dots & 2\mu_m \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

The probability $\pi_{0,k}$ to be in an absorbing state $(0,k)$, i.e., the probability that k users have moved to another cluster and hence the migration of services is required, can be obtained as follows:

$$\pi_{0,k} = -\sigma_B (Q_B)^{-1} Q_{B,k}^T \,. \tag{1}$$

Let σ_B be the initial probability distribution vector of the chain states. This is equal to $(1,0,\dots,0)$, as the system shall start from the state $(n,0)$. $Q_{B,k}$ represents the vector containing the transition rates from the $\frac{n(n+1)}{2}$ non-absorbing states to the absorbing state $(0,k)$. $Q_{B,k}$ has the same structure for any k, except for $Q_{B,0}$ and $Q_{B,n}$, which contain only one non-zero element. The non-zero elements of these vectors are as follows (the position of the element in the vector is in parentheses): $Q_{B,0}(n) = \mu_s$, $Q_{B,n}(\frac{(n+1)n}{2}) = \mu_m$, $Q_{B,k}(\sum_{i=1}^{k}(n-i+1)) = \mu_m$ and $Q_{B,k}(\sum_{i=1}^{k+1}(n-i+1)) = \mu_s$, for $k = 1,\dots,n-1$.

Having described how we model the number of users moving outside the replication zone, we can derive the expected number of users served by the duplicated services inside the cluster and the expected number of users that require service migration. The number of service replications in cluster 0 is obtained as follows:

$$N_{rep} = n(1 + \sum_{i=1}^{k-1} 6i) \,. \tag{2}$$

The expected number of users served by the duplicated service is given by

$$E[n_{dupl}] = \sum_{i=1}^{n-1} (n-i)\pi_{0,i} \,. \tag{3}$$

The expected number of users requiring service migration is derived as follows:

$$E[n_{migr}] = \sum_{i=1}^{n} i\pi_{0,i} \,. \tag{4}$$

We define the efficiency of the service duplication as the proportion of users served with the duplicated service without service migration. It is given by the following expression:

$$Eff = \sum_{i=1}^{n-1} (\frac{n-i}{n})\pi_{0,i} \,. \tag{5}$$

We define the service migration rate as the proportion of users that required service migration. It is obtained as follows:

$$Rate = \sum_{i=1}^{n} (\frac{i}{n}) \pi_{0,i} \,. \tag{6}$$

Now, we can compare the cost of service replication (storage cost) and service migration (storage cost and network cost - bandwidth usage). S_c denotes the cost of storage of a single service in one MEC host, and N_c the cost of the network. We define the cost incurred by service migration and service duplication as follows:

$$Cost_m = (S_c + N_c) E[n_{migr}] \tag{7}$$

$$Cost_r = S_c N_{rep} \,. \tag{8}$$

To compare these costs we introduce Pr, i.e., the proportion of migration cost to service replication cost. By setting $m = \frac{N_c}{S_c}$, we obtain Pr as follows:

$$Pr = \frac{(m+1) E[n_{migr}]}{N_{rep}} \,. \tag{9}$$

The proportion Pr will allow us to identify if the cost of migration is higher than the cost of replication. If Pr is higher than 1, then the cost of migration is higher. In contrast, if Pr is lower than 1, then the cost of replication is higher.

B. Mobility model

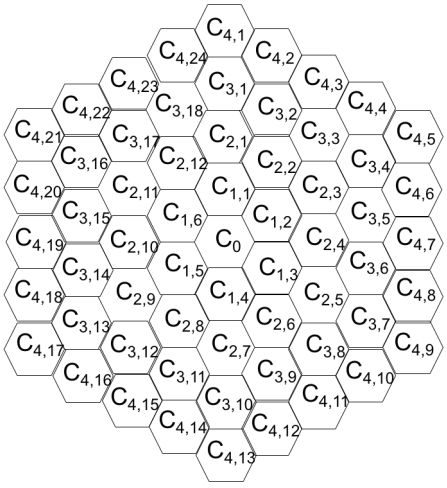


Fig. 3: Our network model, where each cell is identified by the ring it belongs to and its position in the ring.

Having described the model to obtain the cost of both service replication and migration, we show here how to derive the mobility duration (μ_m), which depends on the cluster size (number of rings) and the residence duration in each cell composing the cluster noted by α. Again, let us consider the hexagonal design of a cellular network represented in Fig. 3, wherein a cell is represented by the ring to which it belongs and its identity inside the ring. For instance, cells in ring k are denoted as $C_{k,j}$, with $1 \leq j \leq 6k$. In this model, we are interested in the average time a UE spends in the cluster of size k, where the service is replicated. A UE is assumed to move from one cell to any of the neighbor cells with the same probability $p = 1/6$. As in [11], each cell is represented by its ring label and its position in this ring. Consequently, we consider a Markov chain $\{X_t, t \geq 0\}$ on the state space $S = \{(0,0) \cup (i,j) | 1 \leq i \leq k-1, 1 \leq j \leq 6i\}$, where state $(0,0)$ corresponds to C_0. $X_t = (i,j)$ means that, at time t, the UE is in cell j of ring i.

As in [11], we propose to reduce the state space by aggregating states that show the same behavior. We obtain a new chain, noted A_t, with a lower number of states. To do so, we take advantage of the symmetry of the 2-D model. Indeed, we observe that UEs in the first ring have the same behavior and can move to each neighbor cell with the same probability. That is, UEs come back to C_0 with probability p, stay in the same ring with probability $2p$, and move to ring 2 with probability $3p$. Thereby, all states of ring 1 can be aggregated into one state. Regarding the second ring, we differentiate between two cases: (i) cells with three neighbors in ring 3, two neighbors in ring 2 and one neighbor in ring 1 (e.g., $C_{2,1}$), and (ii) cells with two neighbors in each of the three rings (e.g., $C_{2,2}$). In the first case, the UE leaves the ring towards a more distant one with probability $3p$, while in the second case this happens with probability $2p$. Therefore, we obtain two aggregated states: state $C^*_{2,0}$ aggregates states $\{C_{2,1}, C_{2,3}, C_{2,5}, C_{2,7}, C_{2,9}, C_{2,11}\}$ and state $C^*_{2,1}$ aggregates states $\{C_{2,2}, C_{2,4}, C_{2,6}, C_{2,8}, C_{2,10}, C_{2,12}\}$. We continue in the same manner, based on the algorithm presented in [4], to obtain the remaining aggregated states.

Fig. 4 shows the transition graph for two cases; $k = 4$ and $k = 5$. The absorbing state A, represents the case that a UE has left the cluster. Here, our objective is to capture the residence time of UE in a cluster. We define M_t as the average time before the absorbing state is reached, that is, the average time the UE spends in the cluster before handing off to another one. M_t represents μ_m, which has been used in the precedent section. Let us denote by L_B the transition matrix between the non-absorbing states and σ_B the initial probability distribution (including only non-absorbing states). Then μ_m is obtained as follows:

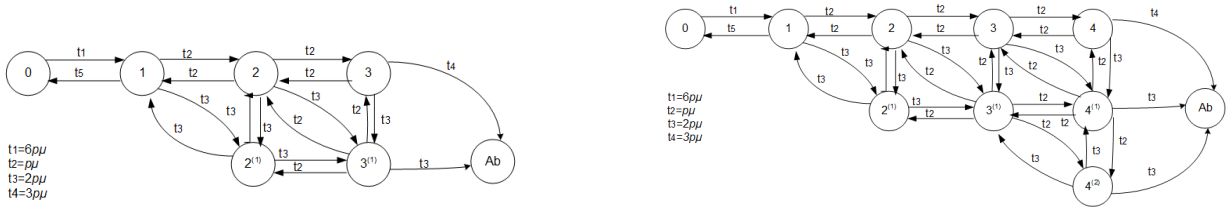
$$\mu_m = \sum_{i \in N} L_i(\infty) \,, \tag{10}$$

where $L_i(\infty)$ is defined as $L_i(\infty) Q_B = -\sigma_B$.

IV. Results

After solving both Markov chains using Matlab, we present in this section numerical results considering two scenarios:

- Scenario 1: The service duration (μ_s) is five times higher than the residence duration in a cell (α). In this scenario we consider the case of an application like video streaming or caching, or an office application.
- Scenario 2: In this scenario we consider the case of an application with short interactions with users, such as



(a) k=4 (b) k=5

Fig. 4: Transition states of the aggregate chain.

automated driving. The service duration (μ_s) is five times lower than the residence duration.

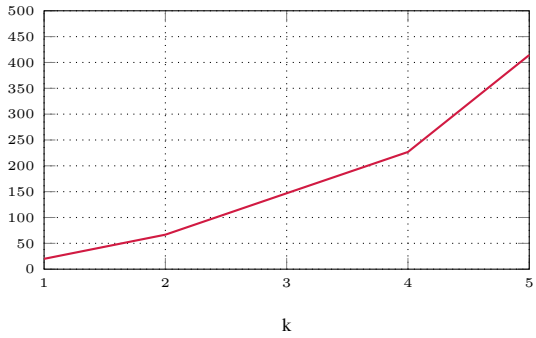


Fig. 5: μ_m versus k

In Figure 5 we draw the residence time in a cluster according to the number of rings. It is clearly observed that increasing the size of the cluster leads to increase the residence time of UE in the cluster. This means lower probability to have service migration. However, this comes with an increased number of service instance duplicates, which leads to increase the cost of replication, as Fig. 6. These two figures clearly show the trade-off between reducing the service migration probability and increasing the cost due to replication. Fig. 7 illustrates

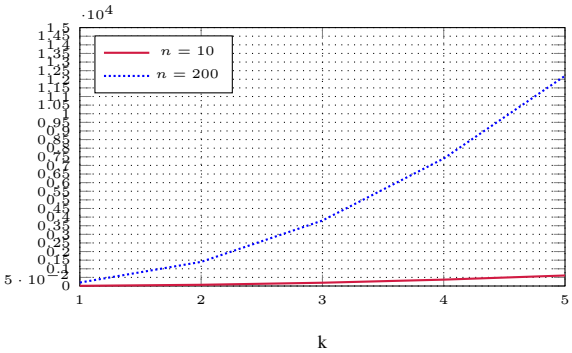
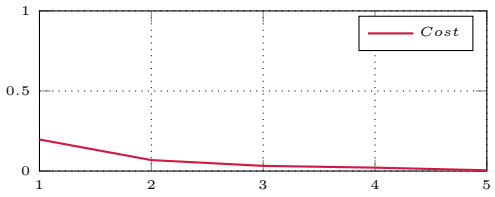


Fig. 6: Number of replication versus k

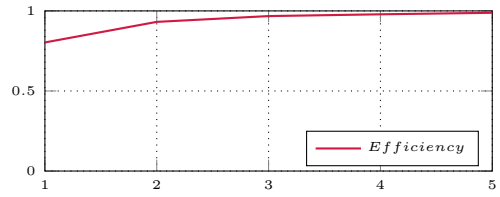
the cost of service migration and the efficiency of the service replication for the two scenarios. The values are obtained by varying the size of the cluster. Fig. 7(a) and Fig. 7(b) represent the rate of service migration and the efficiency versus the cluster size k for the first scenario. We clearly observe that the rate of migration is decreasing with the increase of k. This is indicating that increasing the size of the cluster has a positive impact on reducing service migrations in this case. We argue that this is due to the fact that users are continuously connected to the MEC server and their service ends inside the cluster, particularly when increasing the size of the cluster. Moreover, the efficiency is significantly higher than 50%, whatever the value of k. The efficiency is close to 100% when the size of the cluster is higher than 4 rings; that is, UEs are mostly served by the duplicated services avoiding service migration. We conclude that in case of short-duration applications, service duplication may increase highly the system performance by avoiding service migrations. A value for k around 3 could be a good trade-off between the cost of service replication (number of replicates) and service migration.

Fig. 7(c) and Fig. 7(d) illustrate the performance in case of scenario 2. We observe that unlike the precedent case, the cost of migration is very high. It reaches practically 100% when $k = 1$ (i.e. no service replication). This is explained by the fact that since the duration of the service is very long, users end by leaving the cluster whatever its size. We note that when $k = 5$, the efficiency of the system exceeds 50%. However, this gain is lost with the high number of needed replicates of the service. We can conclude that for services of long duration, it is better to use less replicates, as users end by leaving the cluster and require service migration.

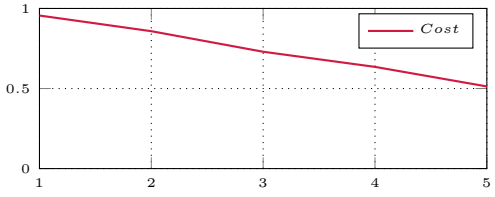
Figures 8(a) and 8(b) indicate the proportion between migration cost and replication cost as defined in (9) for scenario 1 and scenario 2, respectively. Besides the size of the cluster, for each scenario, we varied the value of m, which corresponds to the proportion of the network cost with respect to storage cost. As expected, the case of no replication shows the highest cost for service migration in both scenarios and for all values of m. As shown in Fig. 7(c), the cost of migration is higher for scenario 2. However, from $k = 3$ the cost of migration is lower than the cost of service replication. Therefore, for scenario 2 (i.e., service with short duration), it is more beneficial to select $k = 3$, which demonstrates the best trade-off between service replication and service migration costs. Although in this case Pr is close to 1 (around 0.9), the cost of service replication is mitigated by avoiding downtime (and hence reducing latency). For all other cases, it is better to allow service migration as the cost of replication is very high.



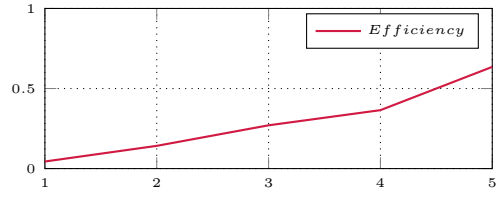
(a) Scenario1



(b) Scenario1

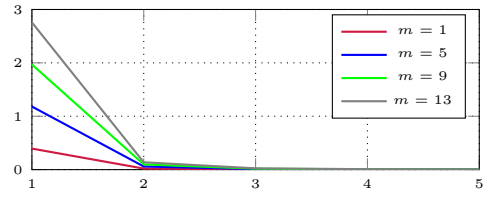


(c) Scenario2

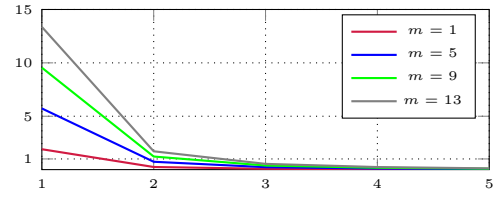


(d) Scenario2

Fig. 7: Cost and Efficiency versus k



(a) Scenario1



(b) Scenario2

Fig. 8: The proportion of migration cost over replication cost

V. Conclusion

In this paper we compared the cost of service migration and service replication in the context of MEC service deployment. The proposed model, based on Markov chains, captures the impact of user mobility and service duration on the cost of service replication and service migration. Our results allowed us to provide recommendations regarding the use either of service replication or service migration. The optimal solution depends on different parameters, and particularly on the number of replicates to use and the type of service. For a short-duration service, it is better to use service replication, while for long duration services it is better to use service migration, as this type of service could be tolerant to the downtime duration and the cost of service replication is high compared to its gains.

VI. Acknowledgement

This work was partially funded by the European Union's Horizon 2020 research and innovation program under the 5G-Transformer project (grant no. 761536).

References

[1] T. Taleb, S. Dutta, A. Ksentini, M. Iqbal, and H. Flinck, "Mobile Edge Computing potential in making cities smarter," *IEEE Commun. Mag.*, vol. 55(3), pp. 38–43, 2017.

[2] A. Huang, N. Nikaein, T. Stenbock, A. Ksentini, and C. Bonnet, "Low Latency MEC Framework for SDN-based LTE/LTE-A Networks," in *Proc. IEEE ICC*, 2017.

[3] ETSI ISG MEC, "Mobile Edge Computing; Framework and Reference Architecture", v1.1.1, 2016-03.

[4] T. Taleb and A. Ksentini, "An analytical Model for Follow Me Cloud," in *Proc. IEEE Globecom*, 2013.

[5] A. Ksentini, T. Taleb, and M. Chen, "A Markov Decision Process-based Service Migration Procedure for Follow Me Cloud," in *Proc. IEEE ICC*, 2014.

[6] I. Farris, T. Taleb, M. Bagaa, and H. Flinck, "Optimizing Service Replication for Mobile Delay-sensitive Applications in 5G Edge Network," in *Proc. IEEE ICC*, 2017.

[7] K. Piamrat, K. Singh, A. Ksentini, C. Viho, J. Bonnin "QoE-aware scheduling for video-streaming in High Speed Downlink Packet Access," in *Proc. IEEE WCNC*, 2010.

[8] C. Colman-Meixner, C. Develder, M. Tornatore, B. Mukherejee, "A Survey on Resiliency Techniques in Cloud Computing Infrastructures and Applications," *IEEE Commun. Surveys Tuts.*, vol. 18(3), pp. 2244–2281, 2016.

[9] S.K. Bose, S. Brock, R. Skeoch, and S. Rao, "Cloudspider: Combining replication with scheduling for optimizing live migration of virtual machines across wide area networks," in *Proc. IEEE/ACM CCGrid*, 2011.

[10] X. Wang, M. Chen, T. Taleb, A. Ksentini, and V.C.M. Leung, "Cache in the air: exploiting content caching and delivery techniques for 5G systems," in *IEEE Comun. Mag.*, vol. 52(2), pp.131–139, Feb. 2014.

[11] T. Taleb, K. Samdanis, and A. Ksentini, "Supporting Highly Mobile Users in Cost-Effective Decentralized Mobile Operator Networks," in *IEEE Trans. Veh. Technol.*, vol. 63(7), pp. 3381–3396, 2014.