ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

ARCHIVIO ISTITUZIONALE
DELLA RICERCA

Alma Mater Studiorum Università di Bologna
Archivio istituzionale della ricerca

An Energy-Aware Offloading Clustering Approach (EAOCA) in Fog Computing

This is the final peer-reviewed author's accepted manuscript (postprint) of the following publication:

Published Version:
Arash, B., Daniele, T., Giovanni Emanuele Corazza, (2017). An Energy-Aware Offloading Clustering Approach (EAOCA) in Fog Computing. IEEE [10.1109/ISWCS.2017.8108146].

Availability:
This version is available at: https://hdl.handle.net/11585/609301 since: 2020-10-09

Published:
DOI: http://doi.org/10.1109/ISWCS.2017.8108146

Terms of use:

Some rights reserved. The terms and conditions for the reuse of this version of the manuscript are specified in the publishing policy. For all terms of use and more information see the publisher's website.

This item was downloaded from IRIS Università di Bologna (https://cris.unibo.it/).
When citing, please refer to the published version.

(Article begins on next page)

12 October 2024

This is the final peer-reviewed accepted manuscript of:

A. Bozorgchenani, D. Tarchi and G. E. Corazza, "An Energy-Aware Offloading Clustering Approach (EAOCA) in fog computing," ***2017 International Symposium on Wireless Communication Systems (ISWCS)*****, Bologna, 2017, pp. 390-395.**

The final published version is available online at: 10.1109/ISWCS.2017.8108146

Rights / License:

The terms and conditions for the reuse of this version of the manuscript are specified in the publishing policy. For all terms of use and more information see the publisher's website.

This item was downloaded from IRIS Università di Bologna (https://cris.unibo.it/)

When citing, please refer to the published version.

An Energy-Aware Offloading Clustering Approach (EAOCA) in Fog Computing

Arash Bozorgchenani, Daniele Tarchi, Giovanni Emanuele Corazza
Department of Electrical, Electronic and Information Engineering
University of Bologna, Italy
Email: {arash.bozorgchenani2,daniele.tarchi,giovanni.corazza}@unibo.it

***Abstract*—Fog computing is an interesting paradigm which has drawn attention recently, based on the presence of several Fog Nodes (FNs) able to interact among each other for sharing their tasks. To boost the interactions among FNs, an Energy-Aware Offloading Clustering Approach (EAOCA) is proposed for raising the network fairness in terms of FNs' energy level. In each cluster, there is a Fog Cluster Head (FCH) that aggregates the traffic from its Fog Cluster Members (FCMs) to be computed. EAOCA considers different policies for evaluating the impact of FCH and FCM selection and how the remained energy of the FNs influences the performance of the network in terms of fairness, delay and energy consumption. The simulation results later demonstrate how cluster updating frequency has a profound impact on the network lifetime.**

I. Introduction

Fog computing brings the computing capability of the cloud to the edge of the network aiming at minimizing the time required for responding to a task and reducing the traffic at the fronthaul, by extending the mobile cloud computing approach [1]. Fog Radio Access Networks (F-RANs) have been proposed as a 5G communication architecture by taking advantage of fog computing and Heterogeneous Cloud Radio Access Network (H-CRAN) [2]. There are various use cases of edge computing that have been demonstrated: active device location tracking, augmented reality content delivery, video analytics, radio access network aware content optimization, distributed content and DNS caching and application-aware performance optimization [3].

In this architecture there are mainly three forms of communications. First, Device-to-Device (D2D) communication in which Fog Nodes (FNs) are able to share their resources or outsource their tasks to the adjacent FNs. Second, FN to Fog-Access Points (F-APs) in which a request from an FN is sent to an F-AP for performing the computation. This form of communication is desirable when there is no other FNs near the requested FN for performing the task. Third, FN to centralized cloud, to be considered in case the previous cases cannot be used. However, in some cases, e.g. for real time applications, this delay might not be acceptable. For this reason, in this work, a new form of communication among FNs is introduced by organizing them in clusters and increasing the cooperation among the FNs which are close to each other. In this idea, we take into account the remained energy of the FNs for their selection for giving service to the adjacent FNs. By updating the clusters, the FNs having consumed more energy for performing the computations will then send their own tasks to an FN with higher remained energy.

There have been plenty of investigations on F-RAN recently. In the system model in [2], for the FN layer, some of the nodes are able to act as a relay to help the interaction among nodes which are not in their coverage. In [4] the clustering was performed among the access points considering channel condition and caching status. To target a fog server from the user point of view, [5] considers both communication and computing delays in their scenario. However, an assumption made in [5] was that all the users can access all the fog servers. One of the closest works in the literature can be [6] that considered some cloudlets for mobile cloud computing. In this architecture mobile devices send their tasks to a gateway and the gateways in the same cloudlet send the tasks to a master device. A clustering algorithm was also proposed in [7] for the radio access points dealing with joint computation and communication resource allocation inside the cluster.

Most of these works have discussed clustering the F-APs. However, none of them have tried to propose an idea for increasing the D2D communication among edge nodes and increasing the fairness in terms of FNs energy level. By the proposed Energy-Aware Offloading Clustering Approach (EAOCA), we will be able to raise the D2D interactions by providing fast content access and giving the responsibility of computation to proper FNs based on their energy level. As a result, the remained energy of the FNs will be balanced in the network. This scenario is applicable in remote areas where FNs are isolated due to a natural disaster and are disconnected from F-APs. With this idea, the lifetime of the network will also be extended and the FNs which are battery powered will stay alive for a longer time.

II. System Model

Our focus in this work is only on the communication among FNs, where $\mathcal{U} = \{u_1, \ldots, u_i, \ldots, u_n\}$ represents the set of FNs. All FNs have computational and storage capabilities and are battery-powered. FNs are considered to be fixed with the

This work has been partially supported by the project "GAUChO - A Green Adaptive Fog Computing and Networking Architecture" funded by the MIUR Progetti di Ricerca di Rilevante Interesse Nazionale (PRIN) Bando 2015 - grant 2015YPXH4W_004.

possibility of offloading their tasks to the neighboring FNs. It is of vital importance to exploit these capabilities in a proper way. FNs can communicate with each other within a specific range depending on the deployed wireless technology. A task in our work is defined as a request sent by an FN to other FNs for computing a specific application. For performing the computation, FNs consume some of the available resources.

The focus in our scenario is on FN to FN communications with the objective of increasing the FN to FN interactions in an efficient way and allowing to build a network able to efficiently share the computational load even if not connected to the core network. To this aim, we propose here a clustering approach for organizing the interactions among the FNs. We have classified the FNs in two types: Fog Cluster Heads (FCH) and Fog Cluster Members (FCM). Each cluster is composed of one FCH and several FCMs. FCHs are capable of performing the computations of the tasks requested by the FCMs within their cluster. In this way, by trying to find some FNs which are capable of performing the tasks for more than one FN, the interaction between FNs will increase without requesting the F-AP or centralized cloud. The FNs which are not able to be in a cluster due to the coverage limitation or capacity constraint will perform the task locally.

Among several parameters, we consider the energy for the selection of FCHs and FCMs. Each FN can be in any of the two possible states of $S = \{1, 0\}$ in which 1 represents that FN is an FCH and 0 shows that FN is an FCM. Depending on the state of the FN, the energy consumption is different and it can be defined as:

$$E_{FCM} = P_{tx} \cdot T_{tx} + P_{id} \cdot T_{com} + P_{rx} \cdot T_{rx} \tag{1}$$

$$E_{FCH} = P_{rx} \cdot T_{rx} + P_{com} \cdot T_{com} + P_{tx} \cdot T_{tx} \tag{2}$$

where P_{tx}, P_{id}, P_{com} and P_{rx} respectively represent the power required for transmission, the state of being idle, computation and reception. While T_{tx}, T_{rx} and T_{com} respectively show the time required for transmission, reception and computation of a task. Due to the fact that FCMs are offloading their tasks, the energy consumption for an FCM is defined as sum of the energy consumed for transmitting a task, the energy consumed in its idle time, while the task is being computed by the FCH, and the reception time of the task result. While the FCH consumes energy for receiving the task from an FCM, processing it and sending the result back to the FCM.

The set of all clusters is shown as $\mathcal{W} = \{w_1, \ldots, w_m, \ldots, w_M\}$. Let us define the set of FCHs as the set $\mathcal{U}_c$ having the cardinality of M. If $i \in (\mathcal{U}\text{-}\ \mathcal{U}_c)$ and $j \in \mathcal{U}_c$, r_{ij} represents the data rate of the link between nodes i and j.

On the other hand, the transmission time for the lth task when the ith node is in its s_i state, where $s_i \in S$, can be written as:

$$T^l_{tx,s_i} = L_{ls_i}/r_{ij} \tag{3}$$

where L_{ls_i} is the length of the lth task sent from ith node in its s_i state. Length of the task depends on the state of the FN [1]. On the other hand, the reception time of the FCM is the same as transmission time of the FCH and the reverse. They are defined as:

$$T^l_{rx,s_i} = T^l_{tx,s_j} \tag{4}$$

where $s_i \neq s_j$, meaning that one FN is an FCM and the other is FCH. On the other hand, the computational time for the lth task is defined as:

$$T^l_{com} = O_l/Flop \tag{5}$$

where O_l represents the number of operations required for computing the lth task and $Flop$ is the Floating-point Operation Per Second (FLOPS) which depends on the CPU capability of the device.

The ith device is supposed to have an initial energy $E^i_r(0)$. All FNs consume a certain amount of energy when they transmit, receive or compute tasks or when they are idle. Therefore, by a certain time t in the network each FN has consumed $E^i_c(t)$ Joule of energy. Thus, the remained energy of the FNs in a certain time of the network can be calculated as:

$$E^i_r(t) = E^i_r(0) - E^i_c(t) \tag{6}$$

where,

$$E^i_c(t) = \sum_{\tau=0}^{t} s_i(\tau) \cdot E^i_{FCM}(\tau) + \sum_{\tau=0}^{t} (1 - s_i(\tau)) \cdot E^i_{FCH}(\tau) \tag{7}$$

where the state of the ith FN might change at each time instant τ, meaning that it can be an FCH or FCM at different times.

Moreover, the delay for processing the lth task can be defined as:

$$D^l = T^l_{tx,s_i} + T^l_w + T^l_{com} + T^l_{rx,s_i} \tag{8}$$

where T^l_w is the waiting time for the lth task to be computed. The delay is sum of the time required for sending a task, waiting for the task to be computed, computing the task and having the result back from the FCH. Each FCH is supposed to have a buffer which keeps the incoming tasks from the FCMs and when a task is processed it is removed from the buffer and the result is sent back to the requested FCM. $\mathcal{B}_{FCH_m} = \{s^m_1, \ldots, s^m_l, \ldots, s^m_{\bar{l}}\}$ is the buffer of the mth FCH where s^m_l is the lth task in the mth FCH and $\bar{l}$ is the maximum number of tasks the FCH can keep in its buffer. Now, we define waiting time for the lth task as:

$$T^l_w = \sum_{\lambda=1}^{l-1} T^\lambda_{com} \tag{9}$$

which is the sum of the computation time for all tasks in the buffer of a generic FCH that have not yet been processed.

To have a clear understanding of the offloading connection among the FCMs and FCHs, an n by n matrix C is defined. The element $c_{i,j}$ of the matrix corresponds to an active connection between the ith and the jth FN. If $c_{i,i}$ is equal

[1]The length of a packet sent by an FCM is usually larger than the packet sent back from FCH as an output of the processing

to 1, it means that the ith node is acting as an FCH and as a result $s_i = 1$ and having $c_{i,j}$ equal to 1 corresponds to having s_i equal to 0.

Having the remained energy of the FNs and their state using C, our goal is to optimize the system by maximizing the Jain's fairness index of the remained energy of the FNs resulting in:

$$\max\left\{F(\overline{E_r})\right\} = \max\left\{\frac{\left(\sum_{i=1}^{n} E_r^i\right)^2}{\left(n\sum_{i=1}^{n} {E_r^i}^2\right)}\right\} \tag{10}$$

where $\overline{E_r}$ is the set of remained energy of n FNs and $0 \leq F(\overline{E_r}) \leq 1$.

In this work, the goal is to find a proper connection matrix C whose elements define the connections between the ith and the jth FNs and maximize the fairness in terms of remained energy of the FNs, as defined in (10). Such matrix is defined in order to keep as minimum as possible the number of clusters while maximizing the number of FCMs within each cluster. At the same time, network lifetime will also be extended by finding the best connections in C where every FCM has an FCH to offload its task to. This aim is subject to the constraints:

$$\sum_i c_{i,j} \leq \overline{k} \qquad \forall j \tag{11}$$

$$\sum_j c_{i,j} = 1 \qquad \forall i \tag{12}$$

$$c_{i,j} = 0 \qquad \forall i, \qquad \text{if } c_{i,i} = 0 \tag{13}$$

$$c_{i,j} = 0 \qquad \text{if } d_{i,j} > R \tag{14}$$

Constraint (11) ensures that number of FCMs connected to an FCH can not exceed a threshold due to the FCH's capacity limit. Constraint (12) shows that an FCM can not connect to more than one FCH. An FN which is not an FCH can not receive tasks from other FNs and this is shown in Constraint (13). Constraint (14) ensures that the distance between an FCH and its FCM should not exceed threshold R, which is the coverage area of the device.

Since there is no closed solution to solve the problem defined in (10), in the following section we propose a suboptimal solution for FNC resulting in decomposing the problem into sub problems.

III. Energy-Aware Offloading Clustering Approach

We have proposed three phases to solve the problem which are FN classification, FCH selection and FCM assignment to the selected FCHs. Our hypothesis is that the FNs having a higher remained energy in the network are seen as better candidates to be selected as FCHs, due to their capability to perform the computation of the incoming tasks. On the other hand, FNs not having much energy left require an external FN to offload their tasks. By this, the FNs selected as FCHs in the clusters change based on their energy level and the frequency of updating the clusters. The proposed EAOCA with different policies is shown in Algorithm 1.

Algorithm 1 EAOCA

Input: $\mathcal{U}$
Output: $\mathcal{W}$
Quantile ($\mathcal{U}$) **giving** I_1 **and** I_2
for each $u_i \in \mathcal{U}$ **do**
  if $E_r^i \leq I_1$ **then**
    $\mathcal{LPFN} \leftarrow u_i$
  else if $I_1 \leq E_r^i \leq I_2$ **then**
    $\mathcal{MPFN} \leftarrow u_i$
  else if $I_2 \leq E_r^i$ **then**
    $\mathcal{HPFN} \leftarrow u_i$
  end if
end for
FCH_m**= max from** $\mathcal{HPFN}$ **or** $\mathcal{MPFN}$ **(depending on the rule)**
$c_{m,m}$=1
for each $u_j \in \mathcal{U}$ **do**
  while $\sum_j c_{j,m} \leq \overline{k}$ **do**
    FCM_j**=An FN from** $\mathcal{LPFN}$ **or** $\mathcal{MPFN}$ **(depending on the rule)**
    if $d_{j,m} \leq R$ **then**
      $FCH_m \leftarrow FCM_j$
      $c_{j,m}$=1
    end if
  end while
end for
$\mathcal{W} \leftarrow w_m$
Repeat the FCH and FCM selection until all FNs have been evaluated
The remained FNs will do the computation locally

In the first phase of the algorithm, all FNs are classified into three groups, High Power FNs (HPFN), Medium Power FNs (MPFN) and Low Power FNs (LPFN), using a quantile function, which does this classification considering the remained energy of the FNs, and put in their lists which are respectively $\mathcal{HPFN}$, $\mathcal{MPFN}$ and $\mathcal{LPFN}$. The FNs whose energy is higher than the upper quantile of the energy level distribution of all FNs, I_2, are seen as HPFNs, and the ones lower than the lower quantile, I_1, are the LPFNs. Also, the FNs whose energy level is between the upper and the lower quantile of the energy level distribution of all FNs are the MPFNs. After the classification of the FNs, in the FCH selection phase, an FCH which has the highest remaining energy shown as max in Algorithm 1 is selected according to one of the following possible rules:

1) FCHs are selected at first from the HPFN and then MPFN list.
2) FCHs are selected only from the HPFN list.

Then, in the last phase, the FCH is assigned some FCMs, as long as the FCMs are within its coverage range and the maximum number of FNs in each cluster does not allow to

TABLE I
DEFINITION OF POLICIES

policy	FCH Selection rule	FCM Selection rule
policy 0	Random	Random
policy 1	2	2
policy 2	2	1
policy 3	1	2
policy 4	1	1

exceed its capacity threshold. The FCM selection is performed by one of the following rules:

1) FCMs are selected at first from MPFN and then LPFN list.
2) FCMs are selected only from the LPFN list.

After the selection of the FCH according to any of the above rules, if there is no FN remained to be assigned to an FCH meeting the requirement of the capacity of the cluster and the coverage constraint, a cluster with a single node is shaped and the FN performs the computation of the task locally. The rules affect how the algorithm works. For instance, if rule number 2 is considered for FCH selection phase and also among the two possible rules, the second rule is selected in the FCM selection phase, MPFNs perform the tasks locally. Then the low power FNs, which are the only FCMs, are connected to their FCHs, which are the HPFNs. As a result, the LPFNs will not consume its remained energy for computation, and this responsibility is given to an HPFN.

Moreover, it is of great importance to update the clusters after a certain time. The FCHs performing the computation for low power FCMs will spend energy and by updating, an FCH can be selected as the next FCM due to the reduction in its remained energy. In this case, most of the FNs can be both FCH and FCM in the network based on their remained energy which sharply increases the fairness of the network in terms of remained energy.

IV. NUMERICAL RESULTS

In this section, a simulation-based analysis of our proposed method, EAOCA, is performed. We have considered 5 policies with different perspectives for the selection of FCHs and FCMs and updating frequency of the cluster to have a comprehensive evaluation of all the possible conditions and see what the behavior of the policies are. The policies are shown in Table I and the values in each column represent the rules that were previously discussed.

Policy 0 is the basic policy where energy is not considered for FCH and FCM selection phases and the clusters are shaped randomly only based on the coverage constraint and maximum number of FNs in a cluster which is set to 5. Moreover, for each policy we have considered 3 updating frequency values for clusters which are:

(a) once at the beginning of simulation
(b) every 5 seconds
(c) every 1 minute

TABLE II
SIMULATION PARAMETERS

Parameter	Value
Dimension	200m x 200m
Communication Protocol	IEEE 802.11
Task size (L_{s_i}), $s_i = 0$	5 MB
Task size (L_{s_i}), $s_i = 1$	1 MB
Transmission range (R)	25 m
FN Maximum energy $E_r^i(0)$	5000 J
Data Rate (r_{ij})	20 Mbit/s
Task Operation (O)	50G
Flops	12G FLOPS
Computation power (P_{com})	0.9 W
Idle power	0.3 W
Transmission power (P_{tx})	1.3 W
Reception power (P_{rx})	1.1 W
Maximum Number of FNs	2000

The simulation is performed in Matlab; the detailed information regarding the parameters are shown in Table II. The task generation rate is considered 0.02 task per FN in our work. The simulation is carried out once for 7500 seconds in terms of average task delay, average node energy consumption and fairness and once for 12500 seconds for FNs' lifetime. The considered performance parameters are defined as:

- Average Task Delay: The average time spent for a task for transmitting, waiting, computing and receiving back the result.
- Average Node Energy Consumption: The average energy that all FNs have consumed.
- Energy Consumption Fairness: An evaluation on how fair the energy consumption of the FNs in the network is [8].
- Network Lifetime 1 (NL1): The earliest time instant at which any of the FNs in the network depletes its battery, as defined in [9].
- Network Lifetime 2 (NL2): The time instant beyond which 20 percent of the FNs deplete their battery, as defined in [9].

In the following figures, we report the comparison of the selected policies, by considering the three possible updating frequencies of the cluster formation; for each line in the legend the number corresponds to the policy, while the letter to the cluster updating frequency, as previously listed.

Figure 1 depicts the fairness in terms of energy consumption; it is possible to note that updating the cluster formation is beneficial from the fairness point of view, while updating every 5 seconds or every 1 minute seems to lead to similar results. Similar to the previous cases the policies, when the FCMs are selected among all the remaining nodes, seem to have the best performance.

In order to observe the behavior of the policies in term of network lifetime we carried out the simulation for a longer time period (i.e., 12500 s). Figure 2 shows that when the cluster updating is done only once at the beginning, all the policies have similar performance. However, when it is done every minute or every 5 seconds, the first FN depletes its battery nearly 5000 seconds after the basic policy and 3500

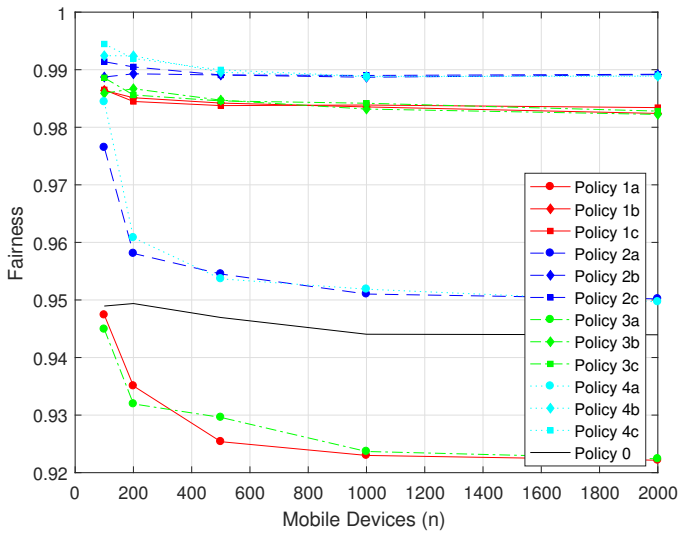

Fig. 1. Fairness

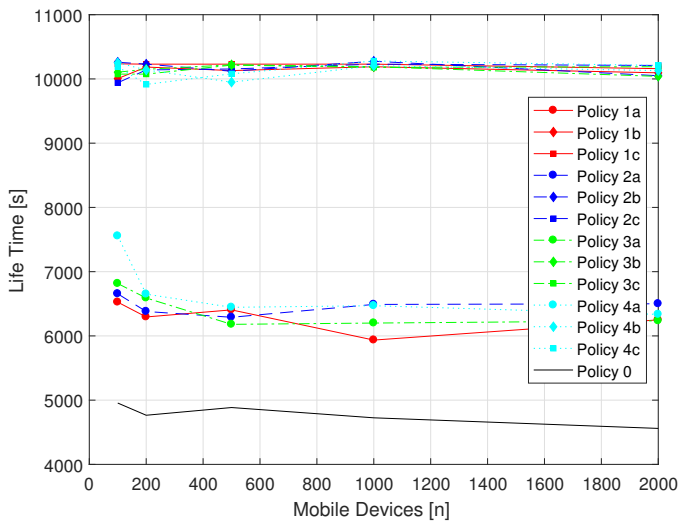

Fig. 2. Network Lifetime 1 (NL1)

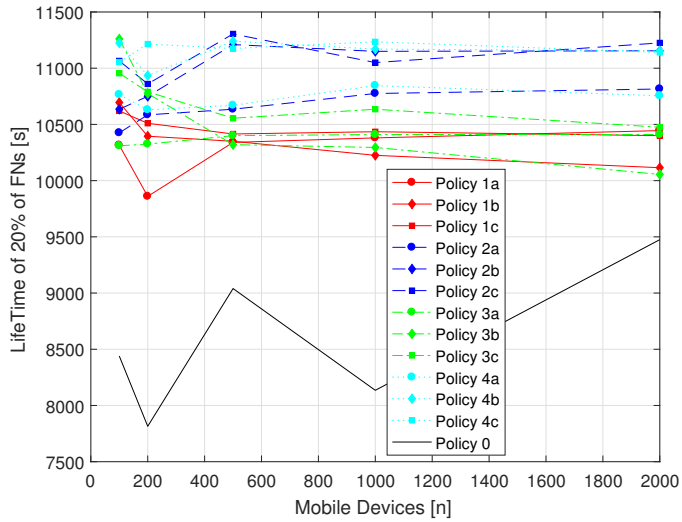

Fig. 3. Network Lifetime 2 (NL2)

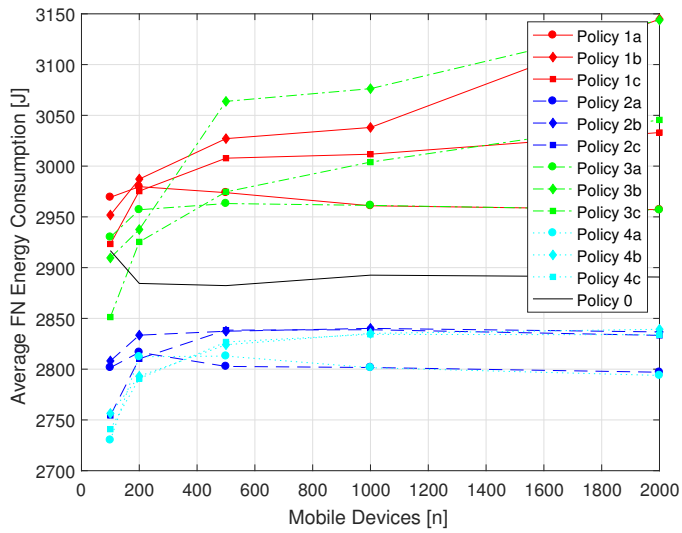

Fig. 4. Average FN Energy Consumption

seconds after updating only once and this shows the significant impact of clustering in the lifetime of the network.

Interestingly in Figure 3 it can be seen that no matter how many FNs are in the network, when it comes to clustering, lifetime of the network when considering 80% of them are still on, is longer than the basic policy, denoting however again that policies 2 and 4 seem to perform better.

Energy consumption performance is depicted in Figure 4, where it is possible to note that, similarly to the delay performance, the policies 2 and 4 have the best performance results. The reason is that the majority of the FNs are selected as FCM, meaning that the only energy LPFNs consume is for transmitting or receiving the tasks, or being idle, which is less than computing locally.

In Figure 5, the performance in terms of average task delay is depicted. Since in this work, we are dealing with offloading and computation of the whole task, we have considered the scale of the delay to be in second It is possible to notice that all policies allow to complete the tasks in less time comparing to the basic policy when the number of FNs are fewer than 1000; for FNs higher than 1000, the time becomes higher if the FCM selection is limited to the LPFNs, and when the updating is done at every run. In these cases, indeed, for a very high density of nodes we may have that some nodes remain isolated, demonstrating the effectiveness of the clustering approach. On the other side, the best performance is obtained when the FCMs are selected among MPFNs and LPFNs; moreover, the number of FNs does not impact largely on the delay performance.

The simulation results underscore that an energy-aware clustering approach in the first layer of fog network can reduce the energy consumption dramatically and make the network

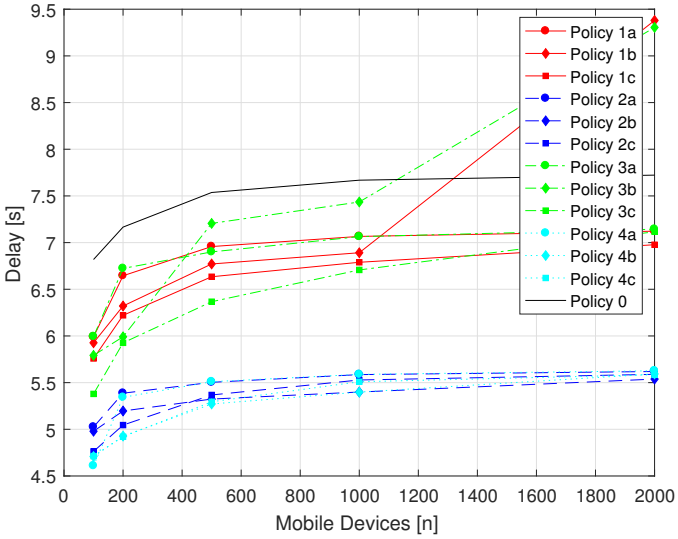

Fig. 5. Average Task Delay

much fairer. Furthermore, as the number of updates are more, the consumption of energy among FNs is considered more and as a result the FCHs are elected based on their capabilities which increase the overall lifetime of the network.

V. Conclusions

In this paper, we describe a novel EAOCA for having a broader interactions among FNs. We have defined various policies considering energy with different perspectives in FCH and FCM selection and frequency of cluster updates. With EAOCA FNs are more balanced in terms of energy consumption which shows the fairness of the proposed method and this has been proven in the simulation results. Assigning the LPFNs to HPFNs for task offloading leads to prolonging the lifetime of FNs and lowering average FN energy consumption. Furthermore, with the suggested method, life time of the network when the first FN goes off and when 80 % of FNs are alive is much longer when the clusters are updated constantly.

References

[1] D. Mazza, D. Tarchi, and G. E. Corazza, "A unified urban mobile cloud computing offloading mechanism for smart cities," *IEEE Commun. Mag.*, Mar. 2017, in print.

[2] M. Peng, S. Yan, K. Zhang, and C. Wang, "Fog-computing-based radio acees networks: issues and challenges," *IEEE Netw.*, vol. 30, pp. 46–53, 2016.

[3] Y. Chao Hu, M. Patel, D. Sabella, N. Sprecher, and V. Young, "Mobile edge computing - a key technology towards 5G," ETSI, White Paper 11, Sep. 2015. [Online]. Available: http://www.etsi.org/images/files/ETSIWhitePapers/etsi_wp11_mec_a_key_technology_towards_5g.pdf

[4] M. Tao, E. Chen, H. Zhou, and W. Yu, "Content-centric sparse multicast beamforming for cache-enabled cloud RAN," *IEEE Trans. Wireless Commun.*, vol. 15, no. 9, pp. 6118–6131, Sep. 2016.

[5] K. Intharawijitr, K. Lida, and H. Koga, "Analysis of fog model considering computing and communication latency in 5G cellular networks," in *2016 IEEE International Conference on Pervasive Computing and Communication Workshops (PerCom Workshops)*, Sydney, Australia, Mar. 2016.

[6] M. Jo, T. Maksymyuk, B. Strykhalyuk, and C.-H. Cho, "Device to device based heterogeneous radio access network architecture for mobile cloud computing," *IEEE Trans. Wireless Commun.*, vol. 22, no. 3, pp. 50–58, Jun. 2015.

[7] J. Oueis, E. Calvanese Strinati, and S. Barbarossa, "Distributed mobile cloud computing: A multi-user clustering solution," in *2016 IEEE International Conference on Communications (ICC)*, Kuala Lumpur, Malaysia, May 2016.

[8] H. Shi, R. V. Prasad, E. Onur, and I. Niemegeers, "Fairness in wireless networks: Issues, measures and challenges," *IEEE Commun. Surveys Tuts.*, vol. 16, pp. 5–24, First 2014.

[9] H. Yetgin, K. T. K. Cheung, M. El-Hajjar, and L. Hanzo, "A survey of network lifetime maximization techniques," *IEEE Commun. Surveys Tuts.*, 2017.