A Fuzzy Logic-based Mechanism for An Efficient Cloud Resource Planning

Abdelquoddouss Laghrissi[1], Tarik Taleb[1,2], Miloud Bagaa[1] and Jonathan Prados-Garzon[3]
[1] Communications and Networking department, Aalto University, Espoo, Finland
[2] Centre for Wireless Communications, University of Oulu, Oulu, Finland
[3] Research Centre for Information and Communications Technologies of the University of Granada,
Granada, Spain
Emails: abdelquoddouss.laghrissi@aalto.fi, tarik.taleb@aalto.fi, miloud.bagaa@aalto.fi, jpg@ugr.es

Abstract—**The key concept beneath Multi-Access Edge Computing (MECs) is to place cloud resources in closer proximity to end-users, through the installation of small-scale cloud infrastructures at the network edge. In MEC environments, we identify two issues: 1) data about users' activities are not always available, and 2) the available virtual resource planning mechanisms (i.e., algorithms for the placement of Virtual Network Functions - VNFs) are not efficient enough to fulfill the QoS requirements and deployment costs. In this vein, we design a layered framework to define the presence of Mobile BroadBand User Equipments (UEs) and automate the underlying virtual resource placement and management based on the Fuzzy Logic Controller paradigm (FLC). Experimentation results show that our framework, compared to baseline solutions, achieves good performance results; the end-to-end delay is enhanced by 25%, the resource consumption is reduced by 30%, and the environmental impact, reflected by the carbon footprint that depends on the amount of deployed Virtual Machines (VMs), is reduced by 50%.**

I. Introduction

It is observed that a tremendous amount of data is generated at the edge of the network. This includes videos streamed from smart-phones and Internet of Thing services (IoT) [1]. End-users would benefit from bandwidth and short latency if the resources/services needed are available/moved either partially or completely to the network edge [2]. This is of high importance particularly that Cloud Service Providers (CSPs) are facing an interesting phenomenon known as "data tsunami" or "data deluge" whereby millions of connected users exchange a massive number of text messages, as well as audio and video contents. The evolving technologies must process this amount of data and manage it with the insurance of ultra-short latency, Quality of Service (QoS) and Quality of Experience (QoE).

Unfortunately, the classical approaches are unable to assess the required computational resources and provide them in real-time. Additionally, in the granularity of Edge Clouds (ECs) which streamline the traffic flow from mobile User Equipments (UEs), a "smart" solution is demanded to provide local data analysis of the traffic in real-time, and then predict and adjust the needed virtual resources (i.e., VNFs) mainly in scenarios with similar behavioral patterns (e.g., peak working hours and tech-trends). In this vein, we propose a framework dubbed "FL-QUEME" to quantify the end-users' presence in EC environments using fuzzy logic.

Unlike the conventional logic, based on the choice between true or false for any problem statement, fuzzy logic adds shades of truth and falseness in the same statement [3]. It is defined as the mathematical evaluation of a given problem based on the degree of truth, and adds the notion of perspective.

One of the most important phases of the fuzzy logic design is to define the Fuzzy Logic Controllers (FLCs). The primary concept in developing a FLC [4] relies on the information gathered from various sources of experience or experts. In our case, when considering the design of a FLC for VNF placement, we have to keep in-mind the following information:

- Type of service (e.g., streaming services, social networks, chat services, and web browsing).
- Type of data exchanged (e.g., text, image, video, and content-mixed web page).
- Duration of the session and subsequently the frequency of service requests' generation.
- Estimated number of packets and packet size.
- Needed computation resources (i.e., bandwidth, storage, and CPU).

As a basic design of our FLC, we consider a VNF placement core agent, which takes as inputs the aforementioned information for each given area and gives as an output the needed virtual resources (i.e., number of required CPU cores). In a traditional cloud infrastructure, this would be a viable option, as it is meant to fulfill general-purpose computing and increase resource utilization. In EC environments, an optimized design should consist of at least two types of controllers, an eNB-level controller and an EC-level controller in order to serve latency-sensitive services in large-scale environment setups. In this vein, our work proposes a fuzzy logic-based framework, the objective of which is twofold: i) Quantify the service usage of UEs in a spatio-temporal fashion (i.e., variability of UE behavior in terms of service usage defined for a given period of time and considering the mobility); ii) Based on the first objective, provide an efficient mechanism for placing the needed virtual resources.

The remainder of this paper is organized as follows. Section II briefly reviews the related literature. Section III describes each step of our framework design (i.e., membership functions and rule base). Section IV evaluates the performance of our solution against those of the literature. Finally, Section V concludes the paper with insights and future avenues.

II. Related Work

The optimal placement of virtual mobile core network functions is known to be NP-hard [7], and several strategies tackling this issue have been proposed in the literature. In this section, we discuss the work related to the management and evaluation of service requests, and the placement of virtual resources.

Moens et al. consider in [8] the management of service and VM requests separately, for two types of service chains. The proposed management algorithm was tested through a scenario of a small service provider. Based on Integer Linear Programming (ILP), the proposed algorithm finishes in few seconds (i.e., 16 seconds), which makes it quick to cope with sudden changes in demand for resources due to NFV burstiness. In this solution, the virtualized services handle the spillover and the hardware handles the base load [9], [10], but without considering the restrictions on the link usage, nodes capacity, and cost. Those restrictions are discussed in the following.

With the objectives of minimizing the usage cost of link and node resources, Baumgartner et al. [11] addressed the placement of different VNFs, such as Serving Gateway (S-GW), Packet Data Network Gateway (P-GW), Home Subscriber System (HSS) and Mobility Management Entity (MME), excluding VNFs of the RAN. They also considered the VNF requirements (i.e., processing, storage, and bandwidth) excluding latency on the end-to-end network and that on VNF nodes. The RAN domain was taken into account by Riggio et al. in [12]. They aimed to satisfy VNF requirements (i.e., memory, CPU, radio, storage and bandwidth), while minimizing the cost of mapping VNFs, but without taking into account the end-to-end delay.

Indeed, this poses challenges related to the management of communication among data and control plane elements and to the consequent delay budgets among cellular core components. In alignment, many approaches such as in [13], [15]–[18] aimed to define efficient placement algorithms for the typical Evolved Packet Core (EPC) functions, instantiated in decentralized and load-aware physical machines, and improve load-balance, energy (i.e., reduce number of active machines), bandwidth, and link utilization.

Although in several other deployments, UE-related parameters (e.g., delay and mobility) were neglected, they were considered in the recent work of the authors [14], [19], [23]–[25], [28], [29]. In [14], a new simulator dubbed Network Slice Planner (NSP) was introduced. NSP defines a tool to simulate mobile service usage over a particular geographical area and in real time[1]. It allows to generate data about UEs service usage (e.g., video streaming, social networks, and instant messaging) and mobility (e.g., driving users and walking users). The simulation data can be exported as JSON files and exploited in testing VNF placement algorithms. More importantly, it offers a spatio-temporal viewpoint of the generated data. Several reactive placement algorithms were tested using the simulator. As a continuation of this work, we propose in this paper a new approach to place VNFs in closer proximity to where the data was generated, in order to reduce latency and optimize virtual resources usage (i.e., CPU) using a fuzzy logic-based learning process. The approach will be detailed in the following section.

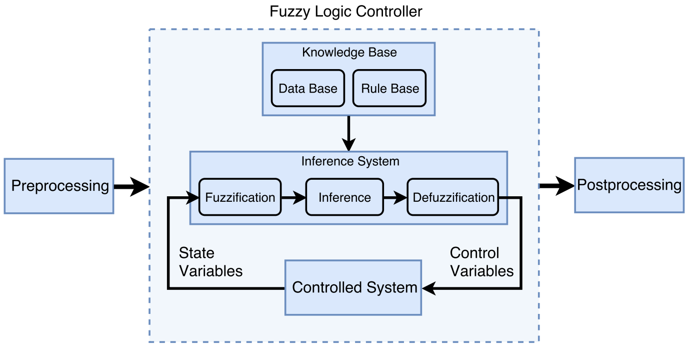


Fig. 1: Generic design of a fuzzy logic controller and its requirements.

III. FL-QUEME Framework description

FLCs were founded and refined by Zadeh, Mamdani and Assilian in [4]–[6] as a generalization of crisp sets using a membership function μ. The main decision phases for our FLC are depicted in Fig. 1. These steps will be detailed in the following subsections. Initially, we introduce the main design components, namely, the preprocessing, membership functions, rule base, and fuzzification and defuzzification methods. Then, the mathematical formulation used to tune the membership functions throughout the simulation is detailed. Finally, we define two algorithms used for virtual resources dimensioning, implemented in eNB and EC controllers, respectively.

A. *Preprocessing*

Prior to this phase, the inputs are in a crisp form, resulting from equipment measurements. The preprocessing consists of converting the crisp inputs into linguistic values. This is handled by the set of rules (i.e, rule base). This conversion should reflect the variations of the crisp values into the best-level discrete universe. In our case, we consider experts' guidelines for the preprocessing, based on the following studies [26], [27], [30] and translated through the rule base that will be detailed in the following subsection. The considered UEs are tablets, smart-phones, and laptops.

B. *Membership functions*

A membership function, usually denoted as $\mu(x)$, takes values from within the interval $[0, 1]$ and reflects the degree

[1] Network Slice Planner: http://mosaic-lab.org/implementations.aspx

of membership of x to given fuzzy sets. We chose the small-medium-large classification for data rates and service duration, both for the up-link and down-link. The initial membership graphs are generated using JFuzzyLogic Java library [31] based on the experts' values. Some of them are depicted in Fig. 2. The input values are manipulated by the rule base in order to provide, in our case, the output CPU needed for up-link and down-link.

Algorithm 1 Some of the rules of our rule base.

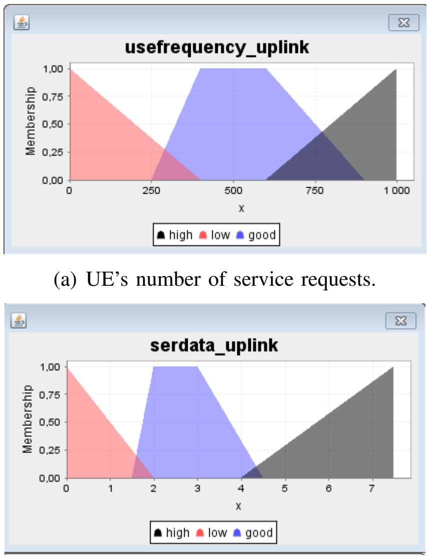
```
RULEBLOCK
// Use 'min' for 'and' (also implicit use 'max'
// for 'or' to fulfill DeMorgan's Law)
$AND : MIN$;
// Use 'min' activation method
$ACT : MIN$;
// Use 'max' accumulation method
$ACCU : MAX$;
RULE 1 : IF $serduration\_uplink$ IS $low$ OR $serdata\_uplink$ IS $low$
THEN $cpu\_u$ IS $low$;
RULE 2 : IF $serdata\_uplink$ IS $good$
THEN $cpu\_u$ IS $medium$;
RULE 3 : IF $serdata\_uplink$ IS $high$ AND $serduration\_uplink$ IS $high$
THEN $cpu\_u$ IS $high$;
RULE 4 : IF $serduration\_downlink$ IS $low$ OR $serdata\_downlink$ IS $low$ THEN $cpu\_d$ IS $low$;
RULE 5 : IF $serdata\_downlink$ IS $good$
THEN $cpu\_d$ IS $medium$;
RULE 6 : IF $serdata\_downlink$ IS $high$ AND $serduration\_downlink$ IS $high$
THEN $cpu\_d$ IS $high$;
$END\_RULEBLOCK$
```

C. Rule base

While the fuzzy logic design can be simply defined as "the control with sentence rather than equations", the empirical rules that help into making decisions based on the inputs manipulated as linguistic variables are known as the rule base. They are usually written as sets of IF (i.e., premise) and Else (i.e., conclusion) pairs. This set of rules is the abstraction of the control strategy which can be the equivalent to an equation-based description.

We depict in Algorithm 1 some of the main entries of our rule base. The inputs are the data size, service usage duration, and number of service requests generated for the up-link and down-link. The output is the CPU resources needed.

Initially, based on experts' values, we estimate in the fuzzification phase the number values for fuzzy sets as: low, medium, and high for each input (see sub-section III.D.1). Inspired by [20] and given the obtained trapezoidal fuzzy partitions (see Fig. 2), we use the Center of Gravity as the defuzzification method. Finally, we define the set of rules to compute the resources needed in terms of CPU (e.g., if the service duration and the data size are high, high CPU is needed).



(a) UE's number of service requests.

(b) Service Data usage (Mb/min).

Fig. 2: Some of the initial membership functions graphs generated using *JFuzzyLogic*.

D. Core functions of FL-QUEME

1) Temporal Membership Functions: In this sub-section, we give details on the mathematical formulation of temporal membership functions for fuzzy sets low, medium and high.

Let us consider a universal set $A_1 : \overline{H} \in A_1, \overline{M} \in A_1 and\ \overline{L} \in A_1$, whereby $\overline{H}$ is the fuzzy set for heavy usage, composed of two fuzzy sets for the up-link and down-link $\overline{H}_u$ and $\overline{H}_d$, respectively. Formally: $\overline{H} = \overline{H}_u o\ \overline{H}_d$.

$\overline{M}$ is the fuzzy set for regular usage, composed of two fuzzy sets for the up-link and down-link $\overline{M}_u$ and $\overline{M}_d$, respectively. Formally: $\overline{M} = \overline{M}_u o\ \overline{M}_d$.

$\overline{L}$ is the fuzzy set for low usage, composed of two fuzzy sets for the up-link and down-link $\overline{L}_u$ and $\overline{L}_d$, respectively. Formally: $\overline{L} = \overline{L}_u\ o\ \overline{L}_d$

$\overline{R_d}$ is the relation matrix for the down-link and $\overline{R_u}$ is the one for the up-link. They are populated by the values computed for the previous fuzzy sets.

To decide if a user is considered to belong to fuzzy set $\overline{H}$, $\overline{M}$, or $\overline{L}$, we consider the following conditions. The membership of a user depends on the probability $P(x)$; the probability that the user's generated data x exceeds a given threshold T (i.e., high usage), comprised between $\frac{T}{2}$ and T (i.e., medium usage), or under $\frac{T}{2}$ (i.e., low usage).

Formally, in case of high usage:

$$\mu_{\overline{H_d}}(\mathrm{x}) = P(x > T_d) \tag{1}$$

$$\overline{R_d} = (\overline{A_1} \times \overline{H_d}) \cup (\overline{A_1} \times \overline{H}\,) \tag{2}$$

Applying the same rule for each fuzzy set, we can obtain $\overline{R_u}$ the relation matrix characterizing the service usage for the up-link as follows:

$$\overline{R_u} = \left\{((x,y,z)\,,\ \mu_{\overline{R}}(x,y,z)) \;\middle|\; (x,y,z) \in \overline{H_u} \times \overline{M_u} \times \overline{L_d}\right\} \tag{3}$$

$$\begin{aligned}\mu_{\overline{R_u}}(x,y,z) &= \mu_{\overline{H_u} \times \overline{M_u} \times \overline{L_u}}(x,y,z) \\ &= \min\{\mu_{\overline{H_u}}(x)\,, \mu_{\overline{M_u}}(y)\,, \mu_{\overline{L_u}}(z)\}\end{aligned} \tag{4}$$

And $\overline{R_d}$ the relation matrix characterizing the service usage for the down-link as follows:

$$\overline{R_d} = \left\{((x,y,z)\,,\ \mu_{\overline{R}}(x,y,z)) \;\middle|\; (x,y,z) \in \overline{H_d} \times \overline{M_d} \times \overline{L_d}\right\} \tag{5}$$

$$\begin{aligned}\mu_{\overline{R_d}}(x,y,z) &= \mu_{\overline{\mathrm{H}_d} \times \overline{M_d} \times \overline{L_d}}(x,y,z) \\ &= \min\{\mu_{\overline{H_d}}(x)\,, \mu_{\overline{M_d}}(y)\,, \mu_{\overline{L_d}}(z)\}\end{aligned} \tag{6}$$

It would have been a straight forward process to decide where to place the needed resources if we could obtain the probabilities needed for relation matrices (4) and (6) which reflect each user's activity (i.e., service requests, data usage, etc.) during the overall time of their activity. But since we cannot obtain such values, we opted for a workaround using $\alpha - cuts$. Basically, $\alpha - cuts$ are used to map each number α from the interval $[0,1]$ into an interval as follows: $\dot{x}(\alpha) = \{\dot{x} : \mu(\alpha) \geq \alpha\}$. We define our $\alpha - cuts$ in the following.

Definition 1.1

A trapezoidal fuzzy number Y can be expressed as [a, b, c, d] and its membership function is defined as:

$$\mu_Y(x) = \begin{cases} \frac{x-a}{b-a},\ a \leq x < b \\ 1,\ b \leq x < c \\ \frac{d-x}{d-c},\ c \leq x \leq d \end{cases} \tag{7}$$

Y reflects fuzzy numbers for a user activity (i.e., service requests, data usage for up-link, etc.) and [a, b, c, d] the threshold values we defined using experts values (e.g., low, good/medium, and high number of service requests). However, since such values are not computed for only one snapshot of time but as a set of snapshots (i.e., whence the use of spatio-temporal), we should be able to obtain a sum of memberships for the user activity using $\alpha - cuts$ as follows.

Definition 1.2

Let Y= [a, b, c, d] and Z= [p, q, r, s] be two fuzzy numbers whose membership functions are:

$$\mu_Y(x) = \begin{cases} \frac{x-a}{b-a},\ a \leq x < b \\ 1,\ b \leq x < c \\ \frac{d-x}{d-c},\ c \leq x \leq d \end{cases} \tag{8}$$

$$\mu_Z(x) = \begin{cases} \frac{x-p}{q-p},\ p \leq x < q \\ 1,\ q \leq x < r \\ \frac{s-x}{s-r},\ r \leq x \leq s \end{cases} \tag{9}$$

Then $Y = [(b-a)\alpha + a, d - (d-b)\alpha]$ and $Y = [(q-p)\alpha + p, s - (s-q)\alpha]$ are the $\alpha - cuts$ of fuzzy numbers Y and Z respectively, and will permit to calculate the addition of fuzzy numbers Y and Z. To do so, we first add the $\alpha - cuts$ of Y and Z using interval arithmetic. If Y reflects the activity of a given user in time T_1, and Z reflects the activity of the same user but in time T_2, the sum will reflect his total activity in T_1 and T_2. The membership function of sum of "Y + "Z is given as follows:

$$\begin{aligned}``Y\ +\ ``Z\ &= [(b-a)\alpha + a,\ d - (d-b)\alpha] \\ &\quad +\ [(q-p)\alpha + p,\ s - (s-q)\alpha\ ]\end{aligned} \tag{10}$$

$$\begin{aligned}``Y\ +\ ``Z\ &= [a + p + (b - a + q - p)\alpha,\ d + s \\ &\qquad - (d - b + s - q)\alpha]\end{aligned} \tag{11}$$

By equating left and right sides to x, we obtain:

$$x = \ a + p + (b - a + q - p)\alpha \tag{12}$$

$$x = \ d + s - (d - b + s - q)\alpha \tag{13}$$

The expression of α in terms of x, by setting α to its border limits 0 and 1, becomes:

$$\alpha = \frac{x - (a+p)}{(b+q) - (a+p)},\ (a+p) \leq x \leq (b+q) \tag{14}$$

and

$$\alpha = \frac{(d+s) - x}{(d+s) - (b+q)},\ (b+q) \leq x \leq (d+s) \tag{15}$$

Thus,

$$\mu_{Y+Z}(x) = \begin{cases} \frac{x-(a+p)}{(b+a)-(a+p)},\ (a+p) \leq x \leq (b+q) \\ \frac{(d+s)-x}{(d+s)-(b+q)},\ (b+q) \leq x \leq (d+s) \end{cases} \tag{16}$$

We apply Equation (16) to be able to compute the temporal membership values for each user. This will be used by the virtual resources placement agent, detailed in the following sub-section.

E. Virtual resources placement agent

In this sub-section, we define the main function of FL-QUEME, which is the service requests dispatcher and virtual resources placement agent. The algorithm for this function (see Algorithm 2) can be split into two parts:

- The first part consists of fetching data of previous simulations, and generate the linguistic values for each UE, in eNB level (e.g., low number of service requests of UE_1, high number of service requests of UE_{55}, etc.), and the needed resources for each UE, in EC level (high data usage of UE_{33}, low CPU needed by UE_1, etc.). These outputs, calculated using the previously detailed membership functions and rule base, can be seen as a service oriented cartography of the previous simulations and will be used by the second part.
- As depicted in Algorithm 2, the second part bases the placement decisions with respect to the linguistic values generated in the first phase. After every placement decision, these linguistic values are updated, accordingly.

Algorithm 2 Algorithm of the main function of FLQUEME

```
Require:
    $E$: Set of events.
  Each event contains entries about the type of
  requested service, UE ID, UE position, the concerned
  ENodeB ID, ENodeB position, the concerned EC ID,
  EC position, Data size, and service duration.
    $M$: The set of linguistic values.
Ensure:
    $ALs$: The set of allocated resources.
for all $e_i \in E$ do
    $M_{tmp} < -$Fetch_Lunguistic_Values();
    $EC_{tmp} < -$Get (Best_Choice($M_{tmp}$));
    if $EC_{tmp}$ contains_enough_resources()) then
        Allocate_resources($EC_{tmp}$);
        Update_lunguistic_values($M_{tmp}$);
    else
        Create_new_resources($EC_{tmp}$);
        Update_lunguistic_values($M_{tmp}$);
    end if
    Update($\mathcal{AL}s$);
end for
return  $\mathcal{AL}s$;
```

IV. Experimentation and results

In this section, we evaluate the performance of our proposed solution against that of existing base-line approaches, namely the BestFit [21], [22] and conformal mapping [19]. NSP is used to simulate the network and users' behavior, interchangeably UEs, in terms of service consumption and mobility. The simulation parameters used in NSP are given in Table I. The solutions are evaluated in terms of the following metrics:

TABLE I: Simulation parameters

Parameters	Value
Simulation duration (hours)	25
Number of UEs	500
Percentage of walking users (%)	50
Percentage of driving users (%)	35
Percentage of biking users (%)	15
Number of ECs	15
Number of Tracking Areas	5
Number of eNodeBs	45
Range of eNodeBs (m)	5000
Instant Messaging usage probability	0.2
Video streaming usage probability	0.45
Social networks usage probability	0.3

- End-to-End delay: This metric is defined as the end-to-end delay of signaling messages between variant UEs and the chosen host server. Formally, the end-to-end delay is computed in terms of the haversine distance between two longitudes and latitudes positions (i.e., UE and host server) by the speed of signal propagation.
- Number of instantiated VMs: This metric is defined as a sum of instantiated VMs.
- Carbon footprint: This metric is defined as the carbon footprint of a VM calculated as follows: $C_f = X \times P_a$ with X is the number of instantiated VMs, and P_a is the carbon emission of one VM (see [32]). P_a becomes higher when the resource is farther from the service request generation.
- Percentage of used virtual resources: This metric is defined as the percentage of CPU used in a given time-span.
- Execution time: This metric is defined as the time needed to execute each algorithm.

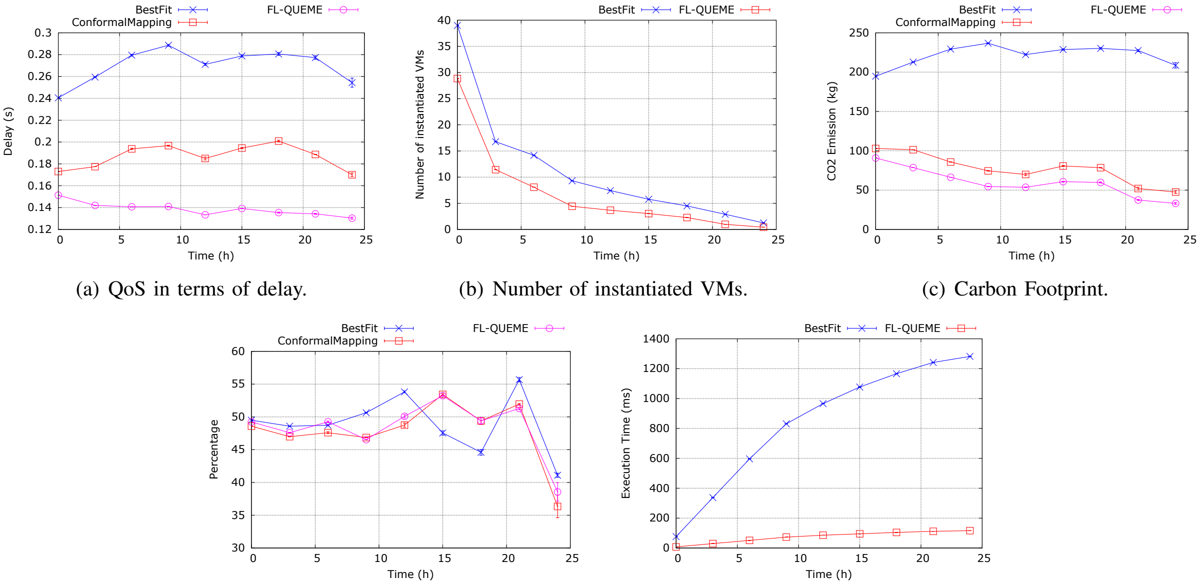
Fig. 3 illustrate the results computed in each time-span of the simulation, which in our case is hourly-based. For a clear analysis of the results, we make a comparison of the three algorithms in the following sub-sections.

A. End-to-End delay:

As depicted in Fig 3.a, FL-QUEME outperforms the baseline solutions in terms of delay, with the conformal mapping being second. This can be explained by the fact that FL-QUEME bases its placing decisions with regards to the inputs given of previous simulation data, recorded on the eNB local controller level. This results in placing virtual resources in closer proximity to end users. The gap in total delay becomes more important as the simulation goes by reaching $0.13s$ for FL-QUEME, $0.17s$ for the ConformalMapping, and $0.26s$ for the BestFit.

B. Number of instantiated VMs:

As illustrated in Fig. 3.b, the number of instantiated VMs reaches its peak value in the first 5 hours of the simulation, with 40 VMs created to meet UEs service usage for the BestFit, and 29 VMs for FL-QUEME. The ConformalMapping



(a) QoS in terms of delay.

(b) Number of instantiated VMs.

(c) Carbon Footprint.

(d) Percentage of used resources.

(e) Execution time (s).

Fig. 3: Spatio-temporal results.

obtained very close values to those of Fl-QUEME with a difference of $4-6$ VMs. As the simulation goes by, the number of instantiated VMs decreases which is explained by the fact that the virtual resources in place are meeting users' requirements. Since FL-QUEUME permits to dispose resources in closer proximity to end-users by making placement decisions, with regards to the inputs given of previous simulation data recorded on the EC local controller level, its overall number of created VMs outperforms BestFit and ConformalMapping by approximately 30%.

C. Carbon footprint:

Due to the fact that this metric depends on the number of instantiated VMs (i.e., the higher is the number of created VMs, the higher CO_2 emission values are reached) as well as the distance between a UE position where the service request was generated and where the resource is made available, it is justified that FL-QUEME outperforms the baseline approaches, with the ConformalMapping being in the second position, in terms of CO_2 emission with more than 50% (see Fig. $3.c$). The gap becomes more important as the simulation goes by, mainly after 15 hours of simulation.

D. Percentage of used virtual resources:

As depicted in Fig. $3.d$, the percentage of used CPU in the created VMs of the three algorithms are overall good, which reflects how much the created VMs are enough to answer to user requirements and justifies the results obtained in terms of instantiated VMs. Nevertheless, Fl-QUEME achieves better resource utilization with a lower percentage obtained mainly between hours 5 and 8 and after hour 23. As future work, it would be interesting to investigate new ways to select given VM flavors rather than others to fit the requirements of end-users and get the most of their performance and utilization.

E. Execution time:

In addition to the very good results obtained for each metric as aforementioned, the execution time of FL-QUEME is very low in comparison to the baseline approaches (see Fig. $3.e$), with the BestFit being in the second position, due to the fact that Fl-QUEME relies on the computation at the levels of eNB and EC controllers, while the other approaches rely on a central controller which creates the need for more computation time.

V. Conclusion

To cope with the dynamic nature of users' service usage and mobility, we proposed in this work a new approach for the evaluation of service requests generated in a given area and for the placement of the needed virtual resources (i.e., CPU) accordingly. The approach is based on two fuzzy logic controllers, one at the level of eNBs and the other at the level of ECs. This separation enables a robust and efficient management of virtual resources, different workloads, and mobility patterns.

The experimentation results demonstrated the efficiency of our approach in comparison to the baseline solutions, in terms of QoS, created virtual resources, energy consumption, and execution time. However, this framework is applied in the case of general network functions placement. We intend to extend this solution in order to satisfy the requirements of specific network functions, such as the relocation cost for Serving Gateways, and latency to Packet Data Network Gateways, considering the EPC case.

Acknowledgement

This work was partially supported by the European Unions Horizon 2020 research and innovation programme under the 5G!Pagoda project with grant agreement No. 723172, the European Union's Horizon 2020 research and innovation programme under the MATILDA project with grant agreement No. 761898, and the Academy of Finland's Flagship programme 6Genesis under grant agreement No. 318927.

References

[1] H. Chang, A. Hari, S. Mukherjee, and T. V. Lakshman, "Bringing the cloud to the edge," in *Proc. IEEE Conference on Computer Communications Workshops*, Toronto, ON, 2014, pp. 346-351.

[2] A. Chandra, J. Weissman, and B. Heintz, "Decentralized Edge Clouds", *IEEE Internet Computing*, vol. 17, no. 5, pp. 70-73, Sep. 2013.

[3] T. T. P. Guanrong Chen, "Introduction to Fuzzy Sets, Fuzzy Logic, and Fuzzy Control Systems," in CRC Press, ISBN. 9780750676052, 2000.

[4] A. Ibrahim, "Fuzzy Logic for Embedded Systems Applications," in Newnes, ISBN. 9780750676052, 2003.

[5] E. H. Mamdani and S. Assilian, "An experiment in linguistic synthesis with a fuzzy logic controller," in *International Journal of Man-Machine Studies*, vol. 7, no. 1, pp. 1-13, 1975.

[6] L. A. Zadeh, "Fuzzy sets," Information and Control, vol. 8, no. 3, pp. 338-353, 1965.

[7] J. Gil-Herrera and J. F. Botero, "Resource Allocation in NFV: A Comprehensive Survey," *IEEE Trans. on Net. and Service Manag.*, vol. 13, no. 3, pp. 518–532, Sep. 2016.

[8] H. Moens and F. D. Turck, "VNF-P: A model for efficient placement of virtualized network functions," in Proc. 10th International Conference on Network and Service Management (CNSM) and Workshop, Rio de Janeiro, Brazil, Nov. 2014, pp. 418-423.

[9] B. Ahmad, T. Taleb, A. Vajda, and M. Bagaa, "Dynamic Cloud Resource Scheduling in Virtualized 5G Mobile Systems," in *Proc. 2016 IEEE Global Communications Conference (GLOBECOM)*, Washington, DC, USA, Dec. 2016, pp. 1–6.

[10] B. Ahmad, A. Vajda, and T. Taleb, "Impact of Network Function Virtualization: A Study based on Real-Life Mobile Network Data," in Proc. 2016 IEEE Int. Wireless Communications and Mobile Computing Conf. (IWCMC), Paphos, Cyprus, Sep. 2016, pp. 541–546.

[11] A. Baumgartner, V. S. Reddy, and T. Bauschert, "Mobile core network virtualization: A model for combined virtual core network function placement and topology optimization," in *Proc. 1st 2015 IEEE Conf. on Net. Softwarization (NETSOFT)*, London, UK, April 2015, pp. 1-9.

[12] R. Riggio, A. Bradai, T. Rasheed, J. Schulz-Zander, S. Kuklinski, and T. Ahmed, "Virtual Network Functions Orchestration in Wireless Networks," in *Proc. 11th Int. Conf. on Network and Service Management (CNSM)*. IEEE, Barcelona, Spain, Nov. 2015, pp. 108-116.

[13] F. Ben Jemaa, G. Pujolle and M. Pariente, "QoS-Aware VNF Placement Optimization in Edge-Central Carrier Cloud Architecture," in *Proc. 2016 IEEE Global Communications Conference (GLOBECOM)*, Washington, DC, USA, Dec. 2016, pp. 1-7.

[14] A. Laghrissi, T. Taleb, M. Bagaa, and H. Flinck, "Towards Edge Slicing: VNF Placement Algorithms for a Dynamic & Realistic Edge Cloud Environment, in Proc. 2017 IEEE Global Communications Conference (GLOBECOM), Singapore, Singapore, Dec. 2017, pp. 1–6.

[15] F. Z. Yousaf, J. Lessmann, P. Loureiro and S. Schmid, "SoftEPC Dynamic instantiation of mobile core network entities for efficient resource utilization," in *Proc. 2013 IEEE International Conference on Communications (ICC)*, Budapest, Hungary, June 2013, pp. 3602-3606.

[16] S. Oechsner and A. Ripke, "Flexible Support of VNF Placement Functions in OpenStack," in *Proc. of the 2015 1st IEEE Conference on Network Softwarization (NetSoft)*, London, UK, April 2015, pp. 1–6.

[17] X. Meng, V. Pappas, and L. Zhang, "Improving the scalability of data center networks with traffic-aware virtual machine placement," in *Proc. of the 29th Conf. on Information Communications (INFOCOM'10)*, San Diego, CA, USA, Mar. 2010, pp. 1154–1162.

[18] F. Machida, M. Kawato, and Y. Maeno, "Redundant virtual machine placement for fault-tolerant consolidated server clusters," in *2010 IEEE Network Operations and Management Symposium (NOMS)*, Osaka, Japan, Apr. 2010, pp. 32-39.

[19] A. Laghrissi, T. Taleb and M. Bagaa, "Conformal Mapping for Optimal Network Slice Planning based on Canonical Domains," *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 3, pp. 519–528, Mar. 2018.

[20] E. V. Broekhoven, B. D. Baets,"Fast and accurate center of gravity defuzzification of fuzzy system outputs defined on trapezoidal fuzzy partitions," *Fuzzy Sets and Systems*, vol. 157, no. 7, pp. 904-918, Apr. 2006.

[21] M. Guazzone, C. Anglano, and M. Canonico, "Exploiting VM migration for the automated power and performance management of green cloud computing systems," in *Proc. of the 1st International Workshop on Energy Efficient Data Centers*, Springer, Madrid, Spain, 2012, pp. 81-92.

[22] L. Tomas and J. Tordsson, "An autonomic approach to risk-aware data center overbooking," *IEEE Transactions on Cloud Computing*, vol. 2, no. 3, pp. 292-305, Jul. 2014.

[23] A. Laghrissi, T. Taleb, and M. Bagaa, "Canonical domains for Optimal Network Slice Planning," in *Proc. of the 2018 IEEE Wireless Communications and Networking Conference (WCNC)*, Barcelona, Spain, Apr. 2018, pp. 1–6.

[24] A. Laghrissi and T. Taleb, "A Survey on the Placement of Virtual Resources and Virtual Network Functions," in IEEE Communications Surveys Tutorials. (to appear)

[25] J. Prados-Garzon, A. Laghrissi, M. Bagaa, T. Taleb, and J. M. Lopez-Soler, "A Complete LTE Mathematical Framework for the Network Slice Planning of the EPC," in IEEE Transactions on Mobile Computing.

[26] Ericsson AB., Ericsson Mobility Report On the Pulse of the Networked Society, Stockholm, Sweden, Nov. 2013 http://www.ericsson.com/res/docs/2013/ericsson-mobility-report-november-2013.pdf.

[27] Cisco, Global Mobile Data Traffic Forecast Update, 2012 - 2017, from Visual Network Index (VNI) White Paper, Cisco Systems, California, USA, Feb. 2013 http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns827/white_paper_c11-520862.pdf.

[28] J. Prados, A. Laghrissi, M. Bagaa, and T. Taleb, "A Queuing based Dynamic Auto Scaling Algorithm for the LTE EPC Control Plane," in IEEE Globecom18, Abu Dhabi, UAE, Dec. 2018.

[29] M. Bagaa, T. Taleb, A. Laghrissi, A. Ksentini, and H. Flinck, "Coalitional game for the creation of efficient virtual core network slices in 5g mobile systems," IEEE Journal on Selected Areas in Communications, vol. 36, no. 3, pp. 469484, Mar. 2018.

[30] Cisco, The Zettabyte Era - Trends and Analysis,from Visual Network Index (VNI) White Paper, Cisco Systems, California, USA, May 2013 http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns827/VNI_Hyperconnectivity_WP.pdf.

[31] P. Cingolani and J. Alcalá-Fdez, "jFuzzyLogic: a Java Library to Design Fuzzy Logic Controllers According to the Standard for Fuzzy Control Programming", *International Journal of Computational Intelligence Systems*, vol. 6, no. sup1, pp. 61-75, Jun. 2013.

[32] A. Nadkarni, E. Sheppard, B. Casemore, "Data Center Energy and Carbon Emission Reductions Through Compute, Storage, and Networking Virtualization," IDC, Sep. 2017.