Received August 15, 2015, accepted September 5, 2015, date of publication October 12, 2015, date of current version November 9, 2015.

Digital Object Identifier 10.1109/ACCESS.2015.2489930

Toward Elastic Distributed SDN/NFV Controller for 5G Mobile Cloud Management Systems

ABDELKADER AISSIOUI[1], ADLEN KSENTINI[2], (Senior Member, IEEE), ABDELHAK (MOURAD) GUEROUI[1], AND TARIK TALEB[3], (Senior Member, IEEE)

[1]PriSM Laboratory, University of Versailles Saint-Quentin-en-Yvelines, Versailles 78035, France
[2]Institute for Research in IT and Random Systems, University of Rennes 1, Rennes 35065, France
[3]Communications and Networking Department, Aalto University, Espoo 02150, Finland

Corresponding author: T. Taleb (talebtarik@gmail.com)

This work was supported by the French Fond Unique Inter-ministeriel (FUI) DVD to Cloud (DVD2C) Project.

ABSTRACT Future 5G mobile network architecture is expected to offer capacities to accommodate the inexorable rise in mobile data traffic and to meet further stringent latency and reliability requirements to support diverse high data rate applications and services. Mobile cloud computing (MCC) in 5G has emerged as a key paradigm, promising to augment the capability of mobile devices through provisioning of computational resources on demand, and enabling resource-constrained mobile devices to offload their processing and storage requirements to the cloud infrastructure. Follow-me cloud (FMC), in turn, has emerged as a concept that allows seamless migration of services according to the corresponding users mobility. Meanwhile, software-defined networking (SDN) is a new paradigm that permits the decoupling of the control and data planes of traditional networks and provides programmability and flexibility, allowing the network to dynamically adapt to change traffic patterns and user demands. While the SDN implementations are gaining momentum, the control plane is still suffering from scalability and performance concerns for a very large network. In this paper, we address these scalability and performance issues in the context of 5G mobile networks by introducing a novel SDN/OpenFlow-based architecture and control plane framework tailored for MCC-based systems and more specifically for FMC-based systems where mobile nodes and network services are subject to constraints of movements and migrations. Contrary to a centralized approach with a single SDN controller, our approach permits the distribution of the SDN/OpenFlow control plane on a two-level hierarchical architecture: a first level with a Global FMC Controller (G-FMCC), and a second level with several Local FMC Controllers (L-FMCCs). Thanks to our control plane framework and Network Function Virtualization (NFV) concept, the L-FMCCs are deployed on-demand, where and when needed, depending on the global system load. Results obtained via analysis show that our solution ensures more efficient management of control plane, performance maintaining, and network resources preservation.

INDEX TERMS Follow me cloud, mobile cloud computing, scalability, network management, openflow, software defined networking, network function virtualization.

I. INTRODUCTION

Today, service provisioning finds in the emerging Cloud Computing paradigm a flexible and economically efficient solution, in particular for small and medium enterprises that do not want to invest huge capitals for creating and managing their own IT infrastructures. The basic tenet of cloud computing is that end users do not need to care about where a service is actually hosted, while service providers may dynamically acquire the resources they need for service provisioning in a pay-per-use model. While for most of elastic web applications the relative position of client and server end systems does not affect the perceived Quality of Experience, rich interactive applications are sensible to other communication metrics, such as delay and jitter [1]. In the absence of explicit QoS control mechanisms in the network, the only way to improve Quality of Experience is to locate servers as close as possible to user terminals. Such an approach, largely exploited by Content Delivery Networks, can be further advanced in the era of Cloud Computing [2]. Assuming that several cloud-enabled Data Centers are made available at the edges of the Internet (i.e. Federated Cloud), service providers may take advantage of them for optimally locating service instances as close as possible to their users. In such a context, mobility of user terminals makes such location decisions even more difficult. In this context, the Follow Me Cloud (FMC) principle was introduced in [3], wherein mobile users are always connected via the optimal data anchor gateway to access its data and/or service from the optimal DC, i.e. geographically/topogically nearest DC. To ensure an optimal end-to-end connection to the cloud for mobile users, users Virtual Machine (VM) (i.e. service) are migrated between DCs, when deemed appropriate [4], [5]. Accordingly, services are always provided from data center locations that are optimal for the current locations of the users. It is worth noting that VM migration is seamless and transparent to users. Thus, on-going sessions between users and services are not interrupted and connections do not need to be reestablished, even if users and/or servers (i.e., hosting services) change location. Besides improving users Quality of Service/Quality of Experience, FMC allows preserving operators network resources by offloading network traffic to data centers through the nearest points compared with users locations.

However, FMC control plane scaling still remains a serious concern in current FMC implementations. To the best of our knowledge, the only work that has investigated the FMC control plane scalability is the one by Bifulco *et al.* [6]. They studied the scalability of an FMC-based system from a static perspective, and proposed an architecture permitting to distribute the control plane on a number of FMC controllers that are statically located in the networks. Nevertheless, the static number and location of FMC controllers may not be suitable constantly because of the dynamic aspect of network load and traffic patterns over time.

To overcome these limitations, we propose in this paper a novel elastic approach based on a SDN/OpenFlow architecture and a control plane framework tailored for mobile cloud computing systems and more specifically for FMC-based systems where mobile nodes and network services are subject to constraints of movements and migrations. In contrary to centralized approach with single SDN controller, our approach permits to distribute the SDN/OpenFlow control plane on a two-level hierarchical architecture: (i) a first level with a global controller G-FMCC, (ii) and second level with several local controllers L-FMCC(s) deployed on-demand, where and when needed, depending on the network dynamics and traffic patterns.

The remainder of this paper is organized as follows. Section II discusses some related work. In Section III, we present the system description and functioning. Section IV studies the control plane scalability of the system. While the Section V addresses the distributed FMC controller operations, the Section VI provides an analytical evaluation of the solution with results' discussion. The paper concludes in Section VII.

II. RELATED WORK

A. THE FMC CONCEPT

The FMC concept was initially proposed in [3]. It was dedicated to the case where all mobility management procedures are handled at the 3GPP domain. In [4], an analytical model is presented to evaluate the performance of the FMC mechanism, while in [5] a Markov-Decision Process (MDP) was introduced for the service migration procedure. In [7], an OpenFlow-enabled implementation of FMC was proposed. The paper describes the components needed to enable FMC, in particular the detection of users movements, the decision logic for migrating services and the method for making migration seamless. The authors presented a proof-of-concept of FMC based on VMware (i.e., VMotion for live VM migration), a NOX-based FMC controller and OpenFlow switches. As the latter have to handle multiple per-flow rules, scalability of FMC rules became an issue. A distributed and hierarchical architecture of FMC controllers could be a remedy [6]. In [8], the authors use the concept of identifier/locator separation of edge networks to support service continuity in FMC. Effectively, in case of a VM migration, the old IP address serves as an identifier and the new IP address serves as a locator for the mobile node. Whilst this operation ensures somehow service continuity, it incurs an important overhead for manipulating the locator/identifier values on the edge networks. In [9], the authors proposed another implementation of FMC based on LISP (Local/Identifier Separation Protocol), whereby the main goal is to render FMC independent from the underlying radio access technology. Thanks to the features of LISP, both users, mobility and VM migration are jointly managed at the same control plane. Besides the LISP entities, all FMC entities were implemented as virtualized network functions running on VMs, facilitating further the concept of carrier cloud [10]. The results obtained from a real-life testbed of the proposed LISP-based FMC architecture showed that the architecture achieved its main design goals, transferring users, services in the order of milliseconds and with very minimal downtime.

B. SDN SCALABILITY

In the literature, several research efforts have been made to tackle the SDN scalability concerns, most of them can be classified in three categories: data plane, control plane, and hybrid.

In the first category, DevoFlow [11] is characterized by its capability to reduce the overhead by delegating some work to the forwarding devices. Thus, it permits to reduce the control plane invocation for most flow setups, and reduces statistics flows transfer. The Software-Defined Counters (SDC) [12] proposal aims to introduce general-purpose CPUs in forwarding devices (ASIC). The existence of such purpose-CPUs, and a fast connection to ASIC's data plane allow to replace traditional counters with a stream of rule-match records which is transmitted to and processed in the CPU. Software-defined counters permits to reduce the control plane overhead by allowing software based implementations of functions for data aggregation and compression.

The second category of efforts aims to improve the performance of the control plane. Maestro [13] is an OpenFlow controller which incorporates an abstraction layer that permits to keep the simple single-threaded programming model along with exploiting parallelism, techniques and designs, permitting to improve the performance of the OpenFlow control plane. HyperFlow [14] is another proposal aiming to increase the OpenFlow control plane performance. HyperFlow exploits the distribution of control plane to provide a physical local view and a logical global view of the system. A distributed file system (WheelFS) is used to maintain and synchronize HyperFlow global view state among distributed controllers. Kandoo [15] is a distributed control plane constructed of two-level hierarchical controllers. Local controllers with no interconnection, which take actions of local scope, and global controller that takes actions of global scope requiring global network view.

Among the hybrid category proposals, DIFANE [16] tries to split the control plane between controllers and specialized data plane switches, called authority switches. The latter are responsible for installing rules on the remaining switches, while the controller focuses on generating the needed rules by the applications. The utilisation of this approach ensures a better scale of the overall system.

Additionally, we can distinguish a specific category of proposals that addresses the notion of elasticity in SDN controllers. Elastic approaches aim to include dynamic adaptation of the controllers number and their locations in the design of scalable SDN solutions. In [17] the authors propose ElastiCon, an elastic distributed controller architecture which permits dynamically to expand or shrink the controllers pool according to the network traffic load. A novel protocol of switch migration is also presented permitting to shift traffic across controllers. Krishnamurthy *et al.* [18] presented Pratyaastha, an elastic distributed SDN control plane which permit to efficiently assign state partitions and switches to controller instances, while minimizing both inter-controller communication and resources consumption. The solution relies on assignment/reassignment algorithm to adapt the system to dynamic changes, which are modeled as an integer linear program (ILP) and solved via a heuristic approach. Bari *et al.* [19] proposed a management framework that allows to optimize the controllers' numbers and locations according to the network dynamics. The dynamic controller provisioning problem was addressed as an integer linear program (ILP) along with two heuristics-based solutions. The presented results are promising and lead to minimize both flow setup time and communication overhead.

III. FMC SYSTEM DESCRIPTION AND FUNCTIONING

The studied FMC system consists of several PMIPv6 domains (we denote this number by N), wherein each domain comprises two parts *(i)* the mobile operator part *(ii)* and the service provider part (Cloud). The SDN/OpenFlow architecture of the system is constructed of a Global Follow-Me Cloud Controller **(G-FMCC)**, multiple Local Follow-Me Cloud Controllers **(L-FMCCs)** and a set of OpenFlow-enabled devices which are **LMA(s)** and **DCG(s)**. First, the system starts with one Global Follow Me Cloud Controller, namely the G-FMCC which is responsible of generating, managing and installing OpenFlow Rules **(OFRs)** on all PMIPv6 domains' OpenFlow-enabled devices. It is worth noting that, the number of controllers is dynamically adjusted, while Local Follow-Me Cloud Controllers are dynamically deployed per PMIPv6 domains (one L-FMCC per PMIPv6 domain) according to network dynamics in order to offload the control plane overhead of the global controller G-FMCC.

Our design of Distributed Follow Me Cloud Controller called **(DFMCC)** is mainly focused on evaluating the scalability in term of managed OpenFlow Rules (OFRs). For this purpose, we introduce a new performance indicator called the OpenFlow Rule Management Rate **(OFRMRate)**, which represents the number of new OpenFlow rules managed per second by the OpenFlow controller associated to IPs addresses migrations experienced by a domain D_j. We introduced this indicator specifically for FMC-based systems and more generally for mobile cloud computing systems where mobile nodes and network services are subject to movements and migrations. The OFRMRate is a per-domain calculated parameter (denoted by L-OFRMRate$_i$ for domain D_i) and it can be associated to G-FMCC or L-FMCC (according to the assignment of the domain D_i to G-FMCC or L-FMCC$_i$ at time t). If we define a binary vector $Y = \langle y_1, y_2, \ldots, y_N \rangle$ indicating which domains are assigned to L-FMCCs (*i.e.*, $y_m = 1$) and which domains are assigned to G-FMCC (*i.e.*, $y_m = 0$) at any time. In this condition, the sum over the N domains of different L-OFRMRate$_i$ values to give the global indicator (denoted by G-OFRMRate) of the overall system is as follow:

$$G\text{-}OFRMRate = \sum_{i}^{N} L\text{-}OFRMRate_i(1 - y_i) \qquad (1)$$

The decision of deploying L-FMCC(s) is governed by a threshold-based system. We define two global threshold levels: High Global Threshold **(H-GThr)** and Low Global Threshold **(L-GThr)**. Our objective is to maintain the system performance in the predefined thresholds window, by deploying L-FMCC(s) when the G-OFRMRate goes over H-GThr (G-OFRMRate > H-GThr) and remove all deployed L-FMCC(s) to preserve system resources when the G-OFRMRate goes under L-GThr (G-OFRMRate < L-GThr). Depending on the value of G-OFRMRate, our DFMCC should:

$$\begin{cases} Deploy\ L\text{-}FMCC(s), & if \quad G\text{-}OFRMRate > H\text{-}GThr \\ Remove\ L\text{-}FMCC(s), & if \quad G\text{-}OFRMRate < L\text{-}GThr \end{cases}$$

To achieve this, every controller maintains a migration information table, namely Global Migration Information Table **(G-MITab)** for G-FMCC and Local Migration Information Table **(L-MITab)** for L-FMCC. The G-MITab is a global view, it contains inter-domain migrations information for all PMIPv6 domains which consists as shown in Table 1, of domain id (Domain), controller id (Controller), number of

OFRs (NbrOFR), list of OFRs (ListOFR), OpenFlow rule management rate (OFRMRate), and old number of OFRs (OldNbrOFR). For all newly inter-domain migration, the G-FMCC updates accordingly its G-MITab table. Whereas the L-MITab is a local view, it contains inter-domain migrations information as shown in Table 2 only for local PMIPv6 domain with the same attributes as G-MITab.

TABLE 1. Global Migration Information Table (G-MITab).

Domain	Controller	NbrOFR	ListOFR	OFRMRate	OldNbrOFR
D_1	$G\text{-}fmcc$	-	$list_1$	-	-
D_2	$L\text{-}fmcc_2$	-	$list_2$	-	-
D_3	$L\text{-}fmcc_3$	-	$list_3$	-	-
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
D_N	$G\text{-}fmcc$	-	$list_N$	-	-

TABLE 2. Local Migration Information Table (L-MITab) for domain D_i.

Domain	Controller	NbrOFR	ListOFR	OFRMRate	OldNbrOFR
D_i	$L\text{-}fmcc_i$	-	$list_i$	-	-

In order to cope with the problem of scalability and resiliency in centralized control plane architecture, we propose in this work an elastic control plane based on two-level architecture. *(i)* The first level is represented by the G-FMCC, it is a permanent active controller which is responsible of generating, managing and installing OpenFlow Rules (OFRs) in order to ensure a seamless migration of service on the cloud side, while following inter-domain mobility of MN in the mobile network side. *(ii)* The second level is represented by the L-FMCC(s) which are dynamically provisioned and deployed when and where needed according to the network dynamics in terms of MNs inter-domains mobilities, services migrations and traffic load. We envision deploying L-FMCC(s) controllers on-demand using the concept of Network Function Virtualization (NFV) [20] which aims at running network functions in virtualized environments on VMs on top of virtualized platforms, rather than on dedicated hardware. This is expected to help in rapid deployment of FMC solution, at least within the cloud side.

IV. CONTROL PLANE SCALABILITY

In order to study the performances of our system, we are aiming in this section at assessing the scalability of our distributed control plane architecture. We will mainly focus on evaluating the scalability from the perspective of managed OFRs. Lets $c_{jk,i}$ represents the number of correspondents nodes in the cloud side that are exchanging packets with the i-th IP address migration from the domain D_j to the domain D_k, and f_j, f_k represent the number of OpenFlow-enabled devices present in domain D_j and D_k respectively (LMA for mobile operator side, DCG for cloud provider side). As illustrated in Figure 1, the number of OFRs managed by the G-FMCC for the i-th migrated IP address from the domain D_j to the domain D_k is given by the following formula:

$$R_i^{D_jD_k} = (f_j + f_k)c_{jk,i} \tag{2}$$

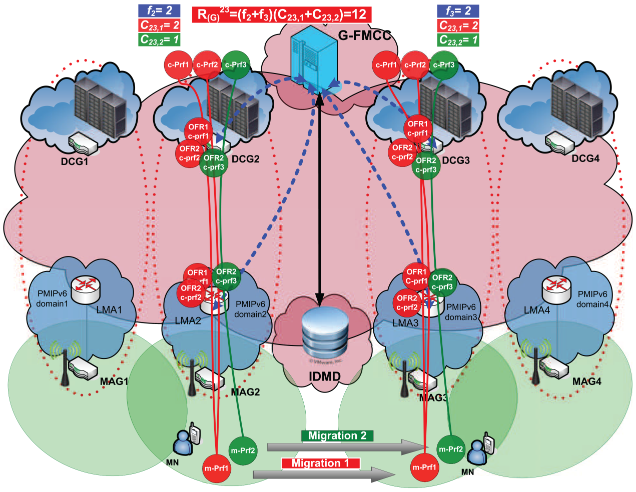


FIGURE 1. Number of managed OFRs in MN inter-domain migrations.

The total number of OFRs managed by the G-FMCC for all IPs address migrations from the domain D_j to the domain D_k is the sum over i of the rules as expressed in (2):

$$R^{D_jD_k} = \sum_i (f_j + f_k)c_{jk,i} \tag{3}$$

The total number of OFRs managed by the G-FMCC for all IPs address migrations originated from the domain D_j is:

$$R^{D_jD_*} = \sum_{\substack{k=1 \\ k\neq j}}^{N} \sum_i (f_j)c_{jk,i} \tag{4}$$

The total number of OFRs managed by the G-FMCC for all IPs address migrations toward the domain D_j is:

$$R^{D_*D_j} = \sum_{\substack{k=1 \\ k\neq j}}^{N} \sum_{i'} (f_j)c_{kj,i'} \tag{5}$$

The total number of OFRs managed by G-FMCC for the domain D_j is:

$$R^{D_j} = R^{D_jD_*} + R^{D_*D_j} \tag{6}$$

$$R^{D_j} = \sum_{\substack{k=1 \\ k\neq j}}^{N} \left(\sum_i (f_j)c_{jk,i} + \sum_{i'} (f_j)c_{kj,i'} \right) \tag{7}$$

The total number of OFRs managed by G-FMCC for all domains is the sum over j of the R^{D_j}:

$$R^G = \sum_{j=1}^{N} \sum_{\substack{k=1 \\ k\neq j}}^{N} \left(\sum_i (f_j)c_{jk,i} + \sum_{i'} (f_j)c_{kj,i'} \right) \tag{8}$$

From equations (2), (6) and (7) it is clear that the number of OpenFlow rules generated for a domain D_j is directly

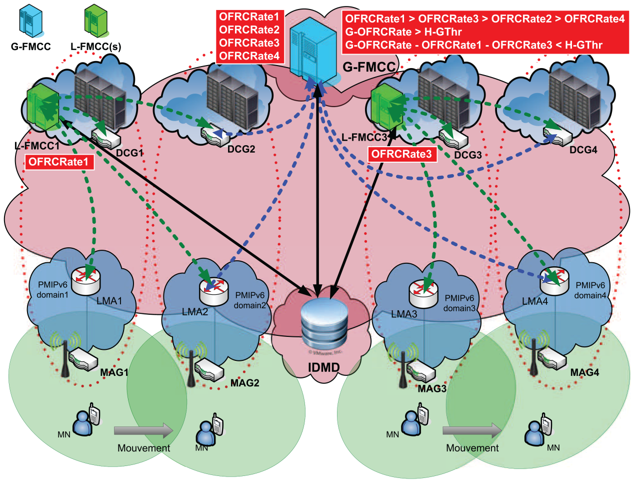


FIGURE 2. L-FMCC(s) deployment conditions.

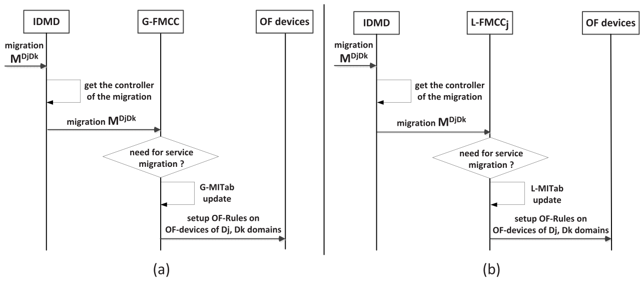

IDMD
G-FMCC
OF devices
migration M^DjDk
get the controller of the migration
migration M^DjDk
need for service migration ?
G-MITab update
setup OF-Rules on OF-devices of Dj, Dk domains
(a)
IDMD
L-FMCCj
OF devices
migration M^DjDk
get the controller of the migration
migration M^DjDk
need for service migration ?
L-MITab update
setup OF-Rules on OF-devices of Dj, Dk domains
(b)

FIGURE 3. (a) Workflow of new migration managed by G-FMCC (b) Workflow of new migration managed by L-FMCC.

TABLE 3. G-MITab update following inter-domain migration in case of D_j-FMCC = G-FMCC.

Domain	Controller	NbrOFR	ListOFR	OFRMRate	OldNbrOFR
.	.	.	.	.	.
D_j	$G\text{-}fmcc$	$NbrOFR_j{+}f_jc_{jk}$	$list_j,\, listR_{c_{jk}}^{D_jD_k}$	-	-
.	.	.	.	.	.
.	.	.	.	.	.

TABLE 4. L-MITab update following inter-domain migration in case of D_j-FMCC = L-FMCC$_j$.

Domain	Controller	NbrOFR	ListOFR	OFRMRate	OldNbrOFR
D_j	$L\text{-}fmcc$	$NbrOFR_j{+}f_jc_{jk}$	$list_j,\, listR_{c_{jk}}^{D_jD_k}$	-	-

proportional to: **(i)** the number of concurrent inter-domain migrations between the domain D_j and all the other domains (from domain D_j to all other domains or inversely from all other domains to domain D_j); **(ii)** the number of OpenFlow-enabled devices of each domain on which the OpenFlow rules are pushed (represented here by the f_i variables); **(iii)** the number of correspondent nodes on the cloud side that are exchanging packets with each migrated address (given here by the $c_{jk,i}$ variables) related to the i-th IP address migration from the domain D_j to the domain D_k.

Due to this characteristic, and in order to quantify the control plane performance in our architecture, we make use of our introduced parameter OFRMRate, which represents the number of new OpenFlow rules managed per second associated to IPs addresses migrations registered in a domain D_j. The OFRMRate parameter is a specific characteristic to each domain, it is directly related to the number of migrations experienced by the different domains. As illustrated in Figure 2 OFRMRate is calculated and registered globally on the G-FMCC for all domains. It is also calculated and registered locally if a L-FMCC is deployed for a specific domain. The key objective of our DFMCC solution is to maintain the G-OFRMRate value of the overall system within the prespecified threshold window (H-GThr, L-GThr). This is achieved by the dynamic adaptation of controllers number through the on-demand NFV deployment/removal of L-FMCC(s). The system should deploy one or several L-FMCCs when OFRMRate goes over the H-GThr in order to ensure the offload of the G-FMCC. Inversely, it should remove all deployed L-FMCCs when OFRMRate goes under the L-GThr in order to maintain the system performance and resources preservation.

V. DISTRIBUTED FMC CONTROLLER OPERATIONS

A. OPERATIONS RELATED TO INTER-DOMAIN MIGRATIONS

We point out here the existence of external elements which are the Inter-Domain Mobility Database (IDMD) ensuring the registration of mobility information of all PMIPv6 domains, and the Decision Making Application Module (DMAM) responsible for taking the decision on the relevance of service migration. For each inter-domain migration $M^{D_jD_k}$ from domain D_j to domain D_k, the IDMD acts as a trigger to advise the source domain controller D_j-FMCC about the MN inter-domain movement. The D_j-FMCC exploits these information and thanks to the DMAM module it takes a decision on the relevance of service migration and thus triggers the service migration on the cloud side according to user mobility. We note that the D_j-FMCC may be, as appropriate, G-FMCC or L-FMCC$_j$ depending on the current state of the DFMCC system. A new migration management workflow is illustrated in Figure 3. The details of control operations interactions between the D_j-FMCC and the different modules foregoing the service migration is out-of-scop for this paper the interested readers are invited to refer to our work on centralized FMCC architecture [21] for further details.

In addition, the IDMD maintains a local list of domain-to-controller mapping information indicating at all moment which domain is managed by which controller. This list is kept updated by the G-FMCC according to L-FMCC(s) deployment state. Upon reception of inter-domain migration message $Msg^{D_jD_k}$ from domain D_j to domain D_k, the IDMD extracts the source domain D_j of the migration, performs a lookup of its current deployed controller D_j-FMCC thanks to the domain-to-controller mapping list, and relays to it the migration message $Msg^{D_jD_k}$. In its turn, the D_j-FMCC activates the DMAM in order to take decision on the relevancy of service migration following the MN movement. If the service migration is deemed appropri-

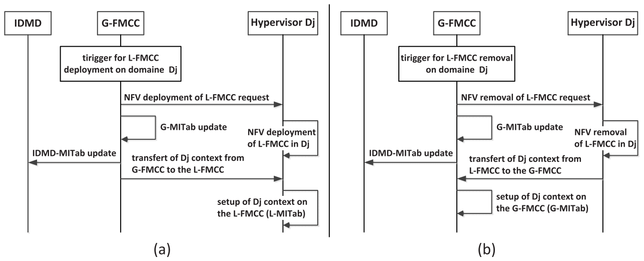

IDMD
G-FMCC
Hypervisor Dj
tirigger for L-FMCC deployment on domaine Dj
NFV deployment of L-FMCC request
G-MITab update
NFV deployment of L-FMCC in Dj
IDMD-MITab update
transfert of Dj context from G-FMCC to the L-FMCC
setup of Dj context on the L-FMCC (L-MITab)
(a)
IDMD
G-FMCC
Hypervisor Dj
tirigger for L-FMCC removal on domaine Dj
NFV removal of L-FMCC request
G-MITab update
NFV removal of L-FMCC in Dj
IDMD-MITab update
transfert of Dj context from L-FMCC to the G-FMCC
setup of Dj context on the G-FMCC (G-MITab)
(b)

FIGURE 4. **(a) Workflow for L-FMCC(s) deployment (b) Workflow for L-FMCC(s) removal.**

ate, the D_j-FMCC generates then the requisite OpenFlow rules in order to ensure a seamless service migration from domain D_j to domain D_k and updates its local table entry as shown in Tables 3 and 4 (G-MITab or L-MITab depending on the type of current D_j-FMCC controller: G-FMCC or L-FMCC$_j$) with information on NbrOFR, and ListOFR. This is achieved by installing the generated OpenFlow rules on all SDN-capable components of D_j and D_k domains. Figure 4 illustrates the workflow for L-FMCC(s) deployment/removal.

B. OPERATIONS RELATED TO LOCAL CONTROLLERS DEPLOYMENT

In this part we will develop the operations related to L-FMCC(s) deployment, we will present the parameters used to compute our OFRMRate indicator, then we will introduce the three building blocks algorithms of our elastic control plane framework: *(1)* OFRMRate Updating Algorithm **(OUA)**, *(2)* L-FMCC(s) Deployment Vector Generating Algorithm **(LDVGA)** and *(3)* L-FMCC(s) NFV Deployment Algorithm **(LNDA)**.

1) OFRMRate PERFORMANCE INDICATOR COMPUTATION

Lets $\tilde{V} = \langle \tilde{v}_1, \tilde{v}_2, \ldots, \tilde{v}_N \rangle$ and $V = \langle v_1, v_2, \ldots, v_N \rangle$ two vectors which represent respectively the previous OpenFlow rule management rate (OldOFRMRate) and the current OpenFlow rule management rate (OFRMRate) extracted from the G-MITab table of the G-FMCC. Hence, $\tilde{v}_m$ and v_m are respectively the previous (OldOFRMRate$_m$) and the current (OFRMRate$_m$) number of OpenFlow rules managed per second registered for the domain D_m. We assume that the OFRMRate attribute values are updated each T_{Rate} time interval for all domains in the G-MITab of the G-FMCC. The updated OFRMRate attribute value for the D_i domain is given as follows:

$$OFRMRate_i = \frac{NbrOFR_i - OldNbrOFR_i}{T_{Rate}} \tag{9}$$

In addition, we define a binary vector $Y = \langle y_1, y_2, \ldots, y_N \rangle$ indicating which domains have deployed their L-FMCCs (*i.e.*, $y_m = 1$) and which domains are not (*i.e.*, $y_m = 0$) at any time.

2) OFRMRate UPDATING ALGORITHM (OUA)

This algorithm is invoked by the G-FMCC every T_{Rate} time interval; it permits to compute the updated value of OFRMRate$_i$ indicator for each domain D_i on the G-MITab as shown in Table 5. Thereby preparing the next step for the execution of the L-FMCC(s) Deployment Vector Generating Algorithm; which performs the operations in Algorithm 1.

Algorithm 1 OFRMRate Updating Algorithm (OUA)

```
Input:  Current deployment vector of L-FMCCs, $Y$
        Current OpenFlow rule management rate vector, $V$
        Previous OpenFlow rule management rate vector, $\tilde{V}$
Output: Updated value of OFRMRate, $V$, $\tilde{V}$
for $i = 1$ to $N$ do
    if $y_i = 0$ then
        G-MITab[$D_i$][OFRMRate] ← $\frac{\text{G-MITab}[D_i][NbrOFR] - \text{G-MITab}[D_i][OldNbrOFR]}{T_{Rate}}$
        $v_i$ = G-MITab[$D_i$][OFRMRate]
        G-MITab[$D_i$][OldNbrOFR] ← G-MITab[$D_i$][$NbrOFR$]
        $\tilde{v}_i$ = G-MITab[$D_i$][OldNbrOFR]
    end if
repeat
```

TABLE 5. **G-MITab OFRMRate update every T_{Rate} time interval.**

Domain	Controller	NbrOFR	ListOFR	OFRMRate	OldNbrOFR
D_1	$G\text{-}fmcc$	-	-	$\frac{NbrOFR_1 - OldNbrOFR_1}{T_{Rate}}$	$NbrOFR_1$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
D_i	$G\text{-}fmcc$	-	-	$\frac{NbrOFR_i - OldNbrOFR_i}{T_{Rate}}$	$NbrOFR_i$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
D_N	$G\text{-}fmcc$	-	-	$\frac{NbrOFR_N - OldNbrOFR_N}{T_{Rate}}$	$NbrOFR_N$

3) L-FMCC(s) DEPLOYMENT VECTOR GENERATING ALGORITHM (LDVGA)

This algorithm is invoked by the G-FMCC every $T_{deployment}$ time interval (note that $T_{deployment} > T_{Rate}$), on the basis of the updated OFRMRate$_i$ indicator value for each domain D_i; it permits to generate a candidate deployment vector $\tilde{Y} = \langle \tilde{y}_1, \tilde{y}_2, \ldots, \tilde{y}_N \rangle$ of L-FMCC(s) destined to replace the current deployment vector $Y = \langle y_1, y_2, \ldots, y_N \rangle$, each one in its correspondent domain. This algorithm prepares the NFV deployment of L-FMCC(s) step accomplished by L-FMCC(s) NFV Deployment Algorithm, and performs the operations in Algorithm 2.

4) L-FMCC(s) NFV DEPLOYMENT ALGORITHM (LNDA)

This algorithm is triggered by the L-FMCC(s) Deployment Vector Generating Algorithm, marking the end of its execution. It is launched by the NFV module of the G-FMCC, based on the current deployment vector Y and the candidate deployment vector $\tilde{Y}$. It permits to deploy/remove L-FMCC(s) in order to adapt the overall system load, and performs the operations in Algorithm 3.

VI. EVALUATION

In this section, we present the evaluation of our Distributed Follow Me Cloud Controller DFMCC through a theoretical analysis. With regard to the scalability of the distributed architecture and to evaluate the total number of managed rules we mainly focus on the formulas given in (7) and (8). In order to simulate the inter-domain migration arrivals for a domain D_j, we will rely on Non-Homogeneous Poisson Process (NHPP) model with rate parameter function $\lambda_{jk}(t)$. Thus in the studied scenarios the inter-domain migrations

Algorithm 2 L-FMCC(s) Deployment Vector Generating Algorithm (LDVGA)

```
Input: Current OpenFlow rule management rate vector, $V$
       Current deployment vector of L-FMCCs, $Y$
Output: Candidate deployment vector of L-FMCCs, $\tilde{Y}$

$V^* \leftarrow V$, with $v_m$ sorted in descending order
$Y^* \leftarrow Y$, with $y_m$ sorted in the same index order as $v^*_m$
$\psi \leftarrow \sum_{i=1}^{N} v^*_i(1-y^*_i)$, the G-OFRMRate
if $\psi >$ H-GThr then
    $\psi^* \leftarrow \psi$
    for $i = 1$ to $N$ do
        if $y^*_i = 0$ then
            $\psi^* \leftarrow \psi^* - v^*_i$
            $y^*_i \leftarrow 1$
        end if
        if $\psi^* \leq$ H-GThr or $i = N$ then
            break
        end if
    repeat
end if
if $\psi <$ L-GThr then
    $\psi^* \leftarrow \psi$
    for $i = 1$ to $N$ do
        if $y^*_{N-i+1} = 1$ then
            $\psi^* \leftarrow \psi^* + v^*_{N-i+1}$
            $y^*_{N-i+1} \leftarrow 0$
        end if
        if $\psi^* \geq$ L-GThr or $i = N$ then
            break
        end if
    repeat
end if
$\tilde{Y} \leftarrow Y^*$, with $y^*_m$ sorted in the same index order as $y_m$
```

Algorithm 3 L-FMCC(s) NFV Deployment Algorithm (LNDA)

```
Input: Current deployment vector of L-FMCCs, $Y$
       Candidate deployment vector of L-FMCCs, $\tilde{Y}$
Output: New deployment vector of L-FMCCs, $Y$

for $i = 1$ to $N$ do
    if $y_i = 0$ and $\tilde{y}_i = 1$ then
        Deployment of L-FMCC in the domain $D_i$ with NFV
        G-MITab[$D_i$][Controller] $\leftarrow$ L-FMCC$_i$
        IDMD domain-to-controller list[$D_i$] $\leftarrow$ L-FMCC$_i$
        Transfer of context: L-MITab[$D_i$] $\leftarrow$ G-MITab[$D_i$]
    end if
    if $y_i = 1$ and $\tilde{y}_i = 0$ then
        Removal of L-FMCC in the domain $D_i$ with NFV
          L-MITab[$D_i$][Controller] $\leftarrow$ G-FMCC
          IDMD domain-to-controller list[$D_i$] $\leftarrow$ G-FMCC
          Transfer of context: G-MITab[$D_i$] $\leftarrow$ L-MITab[$D_i$]
     end if
  repeat
  $Y \leftarrow \tilde{Y}$, candidate vector becomes the current deployment vector
```

arrivals are assumed to follow a Non-Homogeneous Poisson Process with rate parameter function $\lambda_{jk}(t)$.

A. NON-HOMOGENEOUS POISSON PROCESS MODEL FOR INTER-DOMAIN MIGRATIONS ARRIVALS

In this scenario the inter-domain migrations arrivals for a given t from domain D_j to domain D_k (noted $N^{D_jD_k}(t)$) are assumed to follow Non-Homogeneous Poisson Process with rate parameter function $\lambda_{jk}(t)$; i.e.,

$$\begin{cases} P(N^{D_jD_k}(t) = R) = e^{-\Lambda_{jk}(t)}\dfrac{(\Lambda_{jk}(t))^R}{R!}, & t \geq 0 \\ \Lambda_{jk}(t) = \int_0^t \lambda_{jk}(s)\,\mathrm{ds} \end{cases} \tag{10}$$

In our evaluation we will consider three different scenarios, each scenario is run for 30 minutes with the L-FMCC(s) Deployment Vector Generating Algorithm (LDVGA) running every 4 minutes ($T_{deployment} = 4mn$). The choice of the latter depends on the current state of the DFMCC system, and can be further tuned through a more detailed analysis. Three scenarios were considered: *(1)* The first scenario is the growing phase in which the inter-domain migrations arrivals are assumed to increase; *(2)* The second scenario represents a constant phase with inter-domain migrations arrivals assumed to be constant; *(3)* The third scenario is the decaying phase in which the inter-domain migrations arrivals are assumed to decrease. In order to meet the conditions of this three scenarios: the growing phase, the constant phase and the decaying phase, the rate parameter function $\lambda_{jk}(t)$ from domain D_j to domain D_k is given by:

$$\lambda_{jk}(t) = \begin{cases} \dfrac{t}{5(j+1)}, & \text{if } 0 \leq t \leq 30mn \\ \dfrac{360}{(j+1)}, & \text{if } 30mn \leq t \leq 60mn \\ \dfrac{360}{(j+1)(1+\frac{t}{100})}, & \text{if } 60mn \leq t \leq 90mn \end{cases}$$

Where t is time and j is the index of the source domain D_j.

The expected number of inter-domain migration arrivals $M^{D_jD_k}(t)$ from domain D_j to domain D_k for the NHPP $N^{D_jD_k}(t)$ is given by:

$$M^{D_jD_k}(t) = E[N^{D_jD_k}(t)] = \Lambda_{jk}(t) \tag{11}$$

As a result,

$$M^{D_jD_k}(t) = \begin{cases} \dfrac{t^2}{10(j+1)}, & \text{if } 0 \leq t \leq 30mn \\ \dfrac{360t}{(j+1)}, & \text{if } 30mn \leq t \leq 60mn \\ \dfrac{36000}{(j+1)}\log(1+\dfrac{t}{100}), & \text{if } 60mn \leq t \leq 90mn \end{cases}$$

Throughout our evaluation, the number of domains N is fixed to 6 ($N = 6$) and the reference inter-domain migrations scheme is represented in Figure 5. We consider also that the $c_{jk,i}$ values are constant and equal to c value whatever the i-th inter-domain migration as well as the D_j and D_k domains ($\forall i, j, k,\ c_{jk,i} = c$). Moreover, the number of OpenFlow-enabled devices f_j are constant and equal to f value whatever the D_j domain ($\forall j,\ f_j = f$).

Regarding the preceding criteria and the formulas given in (7) and (8), we will obtain the equations summarized in Table 6.

B. SCENARIO 1: OFRMRate INCREASING PHASE

This scenario is characterized by an increasing number of inter-domain migrations arrivals according to the $M^{D_j}(t)$ function shape of the NHPP model of the domain D_j. Figure 6(a) plots the global G-OFRMRate(t) associated to the G-FMCC and the L-OFRMRate$_i$(t) associated to the L-FMCC$_i$ of domain D_i when this latter is activated. We can

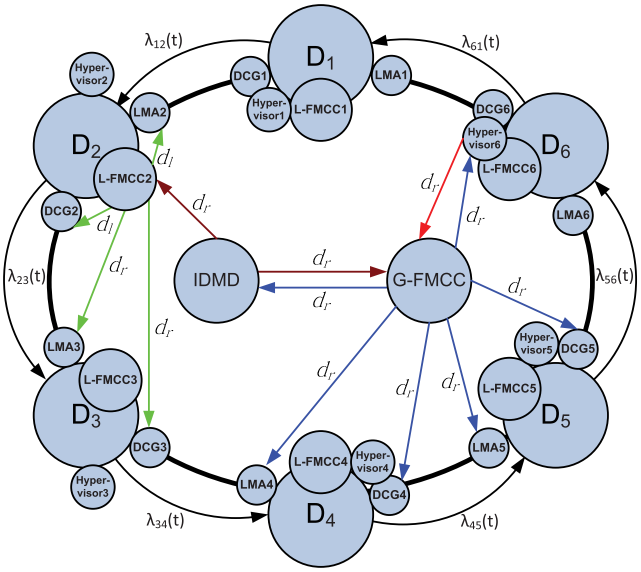


FIGURE 5. Network topology of DFMCC architecture.

TABLE 6. The equations to calculate OFRMRate, #Migrations and #Rules.

Parameter	Domain D_j	All domains
OFRMRate	$L\text{-}OFRMRate_j(t)=2fc\sum_{\substack{k=1\\k\neq j}}^{N}\lambda_{jk}(t)$	$G\text{-}OFRMRate(t)=2fc\sum_{j=1}^{N}\sum_{\substack{k=1\\k\neq j}}^{N}\lambda_{jk}(t)$
#Migrations	$M^{D_j}(t)=\sum_{\substack{k=1\\k\neq j}}^{N}\Lambda_{jk}(t)$	$M^{G}(t)=\sum_{j=1}^{N}\sum_{\substack{k=1\\k\neq j}}^{N}\Lambda_{jk}(t)$
#Rules	$R^{D_j}(t)=2fc\ M^{D_j}(t)$	$R^{G}(t)=2fc\ M^{G}(t)$

see that the application of our elastic control plane framework permits to offload the G-FMCC when the G-OFRMRate(t) goes over the G-HThr by the L-FMCC(s) NFV deployment for most loaded domain(s). The Figure 7(a) plots a comparison of the global number of rules managed by the G-FMCC under the application of our elastic control plane framework, and the case with a single centralized FMCC. We can see that the G-FMCC becomes less loaded then the centralized FMCC when the G-HThr is reached.

C. SCENARIO 2: OFRMRate STATIONARY PHASE

This phase is called stationary because the number of inter-domain migrations arrivals $M^{D_j}(t)$ grows according to a particular kind of NHPP with function parameter a constant (which is a Homogeneous Poisson Process HPP). In Figure 6(b) we can see that the application of our solution always ensures to have a G-OFRMRate(t) under the G-HThr, and as this phase is stationary (number of inter-domain migrations arrivals is constant) only the first invocation of our LDVGA and LNDA algorithms (the first $T_{deployment}$) is needed to deploy the sufficient number of L-FMCC(s) ensuring a G-OFRMRate(t) under the G-HThr for the entire duration of this phase . As we can see also in the Figure 7(b) there is only the first execution of our algorithms (the first $T_{deployment}$) that permits to reduce the number of OpenFlow rules managed by the G-FMCC. We can clearly distinguish differences in term of number of managed rules and the advantage provided by our solution.

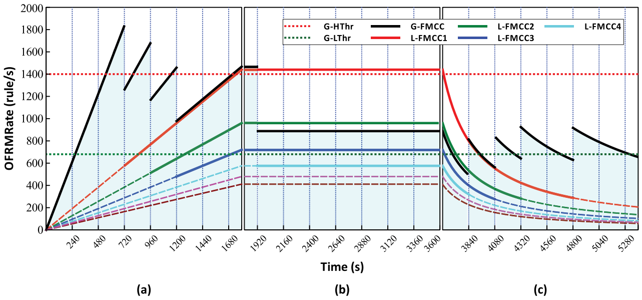


FIGURE 6. OFRMRate parameter adaptation between G-HThr and G-LThr. (a) scenario 1 (b) scenario 2 (c) scenario 3.

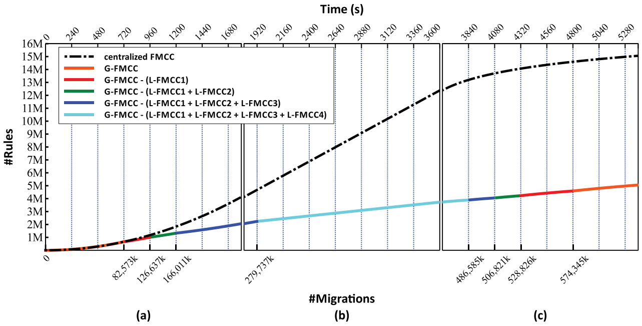


FIGURE 7. The growth in the number of rules managed by G-FMCC and centralized FMCC. (a) scenario 1 (b) scenario 2 (c) scenario 3.

D. SCENARIO 3: OFRMRate DECREASING PHASE

This scenario is characterized by a decreasing number of inter-domain migrations arrivals according to the $M^{D_j}(t)$ function shape of the NHPP model of a domain D_j. Figure 6(c) plots the global G-OFRMRate(t) associated to the G-FMCC and the L-OFRMRate$_i$(t) associated to the L-FMCC$_i$ of domain D_i when the latter is activated. We can see that the application of our elastic approach permits this time to load the G-FMCC when the G-OFRMRate(t) goes under the G-LThr by the L-FMCC(s) NFV removing of less loaded domain(s) and their assignment to the G-FMCC. The Figure 7(c) plots a comparison of the global number of rules managed by the G-FMCC under the application of our elastic control plane framework, and the case with single centralized FMCC. We can observe clearly that our solution permits to preserve resources when the G-LThr is reached by deactivating L-FMCC(s) and approaching thus the case of centralized FMCC architecture.

E. NETWORK DELAY

In this section we will analyse the delay of our approach in terms of the number of d_r exchanged messages. The d_r is a regional long distance message, which is exchanged between two different domains. Accordingly, this kind of message experiences high delay compared with the d_l message, as the latter is a local short distance message.

The Figure 8 compares the number of d_r exchanged messages in the case of G-FMCC under the application of our elastic control plane framework, and the case with single centralized FMCC. We can distinctly observe that the

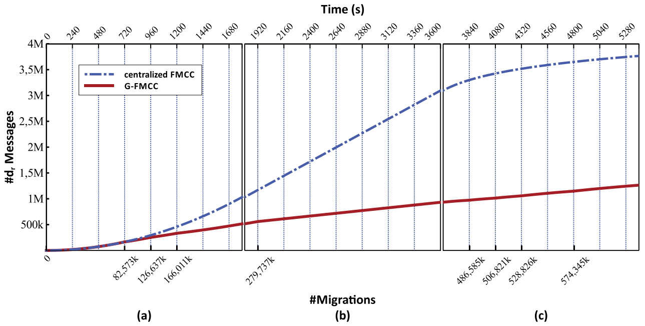


FIGURE 8. Number of d_r messages processed by the system. (a) scenario 1 (b) scenario 2 (c) scenario 3.

distributed architecture DFMCC represented here by the G-FMCC (the most loaded controller) performs better delay performance, and consequently, a faster handling of rules installation in comparison with centralized FMCC architecture in the three phases.

VII. CONCLUSION

In this paper, we have proposed our design of an elastic distributed SDN controller tailored for mobile cloud computing and FMC-based systems. We presented the building blocks of our control plane framework: the performance indicator OFRMRate and the three algorithms (OUA, LDVGA and LNDA). The evaluation results obtained via analysis show that our solution ensures better control plane management, performances maintaining and network resources preservation.

LIST OF ACRONYMS

DFMCC	Distributed Follow Me Cloud Controller
DMAM	Decision Making Application Module
G-FMCC	Global Follow Me Cloud Controller
G-MITab	Global Migration Information Table
H-GTHr	High Global Threshold
IDMD	Inter-Domain Mobility Database
LDVGA	L-FMCC(s) Deployment Vector Generating Algorithm
L-FMCC	Local Follow Me Cloud Controller
L-GTHr	Low Global Threshold
L-MITab	Local Migration Information Table
LNDA	L-FMCC(s) NFV Deployment Algorithm
NHPP	Non-Homogeneous Poisson Process
OFRMRate	OpenFlow Rule Management Rate
OFRs	OpenFlow Rules
OUA	OFRMRate Updating Algorithm

REFERENCES

[1] T. Taleb and K. Hashimoto, "*MS*^{2}: A new real-time multi-source mobile-streaming architecture," *IEEE Trans. Broadcast.*, vol. 57, no. 3, pp. 662–673, Sep. 2011.

[2] 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," *IEEE Commun. Mag.*, vol. 52, no. 2, pp. 131–139, Feb. 2014.

[3] T. Taleb and A. Ksentini, "Follow me cloud: Interworking federated clouds and distributed mobile networks," *IEEE Netw.*, vol. 27, no. 5, pp. 12–19, Sep./Oct. 2013.

[4] T. Taleb and A. Ksentini, "An analytical model for follow me cloud," in *Proc. GLOBECOM*, Atlanta, GA, USA, Dec. 2013, pp. 1291–1296.

[5] A. Ksentini, T. Taleb, and M. Chen, "A Markov decision process-based service migration procedure for follow me cloud," in *Proc. IEEE ICC*, Sydney, NSW, Australia, Jun. 2014, pp. 1350–1354.

[6] R. Bifulco, M. Brunner, R. Canonico, P. Hasselmeyer, and F. Mir, "Scalability of a mobile cloud management system," in *Proc. SIGCOMM Workshop Mobile Cloud Comput. (MCC)*, 2012, pp. 17–22.

[7] T. Taleb, P. Hasselmeyer, and F. G. Mir, "Follow-me cloud: An OpenFlow-based implementation," in *Proc. IEEE GreenCom*, Beijing, China, Aug. 2013, pp. 240–245.

[8] R. Bifulco and R. Canonico, "Analysis of the handover procedure in follow-me cloud," in *Proc. IEEE CLOUDNET*, Paris, France, Nov. 2012, pp. 185–187.

[9] A. Ksentini, T. Taleb, and F. Messaoudi, "A LISP-based implementation of follow me cloud," *IEEE Access Mag.*, vol. 2, pp. 1340–1347, Oct. 2014.

[10] T. Taleb, "Toward carrier cloud: Potential, challenges, and solutions," *IEEE Wireless Commun.*, vol. 21, no. 3, pp. 80–91, Jun. 2014.

[11] A. R. Curtis, J. C. Mogul, J. Tourrilhes, P. Yalagandula, P. Sharma, and S. Banerjee, "DevoFlow: Scaling flow management for high-performance networks," *Comput. Commun. Rev.*, vol. 41, no. 4, pp. 254–265, Aug. 2011.

[12] J. C. Mogul and P. Congdon, "Hey, you darned counters! Get off my ASIC!" in *Proc. 1st Workshop Hot Topics Softw. Defined Netw. (HotSDN)*, New York, NY, USA, 2012, pp. 25–30.

[13] Z. Cai, A. L. Cox, and T. S. E. Ng, "Maestro: A system for scalable OpenFlow control," Dept. Comput. Sci., Rice Univ., Houston, TX, USA, Tech. Rep. TR10-08, 2011.

[14] A. Tootoonchian and Y. Ganjali, "HyperFlow: A distributed control plane for OpenFlow," in *Proc. Internet Netw. Manage. Conf. Res. Enterprise Netw. (INM/WREN)*, Berkeley, CA, USA, 2010, p. 3.

[15] S. H. Yeganeh and Y. Ganjali, "Kandoo: A framework for efficient and scalable offloading of control applications," in *Proc. 1st Workshop Hot Topics Softw. Defined Netw. (HotSDN)*, New York, NY, USA, 2012, pp. 19–24.

[16] M. Yu, J. Rexford, M. J. Freedman, and J. Wang, "Scalable flow-based networking with DIFANE," *SIGCOMM Comput. Commun. Rev.*, vol. 40, no. 4, pp. 351–362, Oct. 2010.

[17] A. Dixit, F. Hao, S. Mukherjee, T. V. Lakshman, and R. Kompella, "Towards an elastic distributed SDN controller," in *Proc. 2nd ACM SIGCOMM Workshop Hot Topics Softw. Defined Netw. (HotSDN)*, New York, NY, USA, 2013, pp. 7–12.

[18] A. Krishnamurthy, S. P. Chandrabose, and A. Gember-Jacobson, "Pratyaastha: An efficient elastic distributed SDN control plane," in *Proc. 3rd Workshop Hot Topics Softw. Defined Netw. (HotSDN)*, New York, NY, USA, 2014, pp. 133–138.

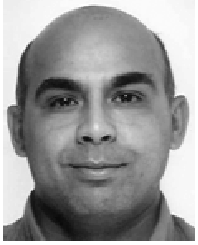
[19] M. F. Bari *et al.*, "Dynamic controller provisioning in software defined networks," in *Proc. 9th Int. Conf. Netw. Service Manage. (CNSM)*, Oct. 2013, pp. 18–25.

[20] M. Chiosi *et al.*, "Network functions virtualisation: An introduction, benefits, enablers, challenges & call for action," SDN OpenFlow World Congr., Darmstadt, Germany, White Paper, Oct. 2012.

[21] A. Aissioui, A. Ksentini, and A. Gueroui, "PMIPv6-based follow me cloud," in *Proc. GLOBECOM*, San Diego, CA, USA, Dec. 2015.

ABDELKADER AISSIOUI received the Engineering degree in computer science from the University of Science and Technology Houari Boumediene, Algeria, in 2004, and the M.Sc. degree in security of contents, networks, telecommunications, and systems from the University of Versailles Saint-Quentin-en-Yvelines, France, in 2007, where he is currently pursuing the Ph.D. degree. His current research interests include mobile networks, mobile cloud networking, network function virtualization, and software defined networking.

ADLEN KSENTINI (SM'14) received the M.Sc. degree in telecommunication and multimedia networking from the University of Versailles Saint-Quentin-en-Yvelines, and the Ph.D. degree in computer science from the University of Cergy-Pontoise, in 2005, with a dissertation on QoS provisioning in the IEEE 802.11-based networks. He is currently an Associate Professor with the University of Rennes 1, France. He is a member of the Dionysos Team with INRIA, Rennes. He is involved in several national and European projects on QoS and QoE support in future wireless and mobile networks. He has co-authored over 60 technical journal and international conference papers. His other interests include future Internet networks, mobile networks, QoS, QoE, performance evaluation, and multimedia transmission. He received the best paper award from the IEEE ICC 2012 and ACM MSWiM 2005. He will be the TPC Chair of the Wireless and Mobile Symposium of the IEEE ICC 2016. He was a Guest Editor of the *IEEE Wireless Communication Magazine* and the *IEEE Communication Magazine*. He has been on the Technical Program Committee of major IEEE ComSoc, ICC/Globecom, ICME, WCNC, and PIMRC conferences.



ABDELHAK (MOURAD) GUEROUI received the M.Sc. degree from the University of Paris 6, and the Ph.D. degree in networking and computer engineering from the University of Versailles Saint-Quentin-en-Yvelines (UVSQ), France. He is currently an Associate Professor of Computer Engineering with UVSQ. His research interests are in the area of wireless networks (WATM, WLAN, and sensors), in particular, performance evaluation and QoS provisioning.

TARIK TALEB (S'04–M'05–SM'10) received the B.E. (Hons.) degree in information engineering and the M.Sc. and Ph.D. degrees in information sciences from the Graduate School of Information Sciences, Tohoku University, Japan, in 2001, 2003, and 2005, respectively. He was a Senior Researcher and 3GPP Standards Expert with NEC Europe Ltd., Heidelberg, Germany. He was then leading the NEC Europe Labs Team working on research and development projects on carrier cloud platforms, an important vision of 5G systems. He was also serving as a Technical Leader of the main work package, Mobile Core Network Cloud, in EU FP7 Mobile Cloud Networking project, coordinating among nine partners, including NEC, France Telecom, British Telecom, Telecom Italia, and Portugal Telecom. Before joining NEC Europe Ltd., and until 2009, he was an Assistant Professor with the Graduate School of Information Sciences, Tohoku University, in a lab fully funded by KDDI, the second largest network operator in Japan. From 2005 to 2006, he was a Research Fellow with the Intelligent Cosmos Research Institute, Sendai, Japan. He has been directly engaged in the development and standardization of the evolved packet system as a member of the 3GPPs' System Architecture Working Group. He is currently a Professor with the School of Electrical Engineering, Aalto University, Finland. His research interests lie in the field of architectural enhancements to mobile core networks (in particular, 3GPPs), mobile cloud networking, network function virtualization, software defined networking, mobile multimedia streaming, inter-vehicular communications, and social media networking. He is a member of the IEEE Communications Society (ComSoc) Standardization Program Development Board. He was a recipient of the 2009 IEEE ComSoc Asia-Pacific Best Young Researcher Award (2009), the TELECOM System Technology Award from the Telecommunications Advancement Foundation (2008), the Funai Foundation Science Promotion Award (2007), the IEEE Computer Society Japan Chapter Young Author Award (2006), the Niwa Yasujirou Memorial Award (2005), and the Young Researcher's Encouragement Award from the Japan Chapter of the IEEE Vehicular Technology Society (2003). Some of his research works have received best paper awards at prestigious conferences. He is an IEEE ComSoc Distinguished Lecturer. As an attempt to bridge the gap between academia and the industry, he founded the IEEE Workshop on Telecommunications Standards: From Research to Standards, a successful event that received the Best Workshop Award by the IEEE ComSoC. Based on the success of this workshop, he has also founded and has been the Steering Committee Chair of the IEEE Conference on Standards for Communications and Networking. He is the General Chair of the 2019 edition of the IEEE Wireless Communications and Networking Conference (WCNC) to be held in Marrakech, Morocco. He is/was on the Editorial Board of the IEEE Transactions on Wireless Communications, the *IEEE Wireless Communications Magazine*, the IEEE Transactions on Vehicular Technology, the IEEE Communications Surveys & Tutorials, and a number of Wiley journals. He serves as the Chair of the Wireless Communications Technical Committee, the largest in the IEEE ComSoC. He served as the Vice Chair of the Satellite and Space Communications Technical Committee of the IEEE ComSoc (2006–2010). He has been on the Technical Program Committee of different IEEE conferences, including Globecom, ICC, and WCNC, and chaired some of their symposia.