Received July 21, 2014, accepted August 21, 2014, date of publication September 25, 2014, date of current version November 24, 2014.

Digital Object Identifier 10.1109/ACCESS.2014.2360352

A LISP-Based Implementation of Follow Me Cloud

ADLEN KSENTINI[1], (Senior Member, IEEE), TARIK TALEB[2], (Senior Member, IEEE), AND FAROUK MESSAOUDI[1]
[1]Institut de Recherche en Informatique et Systèmes Aléatoires, University of Rennes 1, Rennes 35065, France
[2]Aalto University, Espoo 02150, Finland

Corresponding author: A. Ksentini (aksentin@irisa.fr)

ABSTRACT The follow me cloud (FMC) concept enables service mobility, wherein not only content/data, but also services follow their respective users. The FMC allows mobile users to be always connected via the optimal data anchor and mobility gateways to access their data and services from optimal data centers. The FMC was initially designed to support user mobility, particularly in 3rd Generation Partnership Project (3GPP) networks. In this paper, FMC is further tailored to support mobile users connected from other network types, such as public WiFi or Asymetric Digital Subscriber Line (ADSL) fixed networks. Indeed, this paper presents an implementation of FMC based on local/identifier separation protocol (LISP), whereby the main goal is to render FMC independent from the underlying technology. To simplify further the deployment, all FMC entities (including LISP entities) are virtualized considering the network function virtualization principle.

INDEX TERMS Follow me cloud, network function virtualization (NFV), LISP, QoS, testbed.

I. INTRODUCTION

Thanks to the flexibility and elasticity it offers, cloud computing has been gaining lots of ground in the last few years. It is a fast rising business market with well-known players (e.g., Amazon, Microsoft, Google, and VMWare) and many emerging ones (e.g., Verizon Terremark and Saleforce). The size of its market is in the order of hundred of billions of US dollars. Moreover, many companies are using or intend using cloud services, by moving their applications and services to the cloud. Depending on the service needs, cloud computing offers three main models, namely Infrastructure-, Platform-, and Software-as a Service – (IaaS, PaaS, and SaaS). Open source initiatives are not outside this wave of cloud computing. Indeed, many open source solutions have emerged. Notable examples are KVM, Xen, OpenVZ, OpenStack, CloudStack, and Smart OS.

The tremendous growth in cloud business has pushed cloud providers to consider deploying more regional Data Centers (DCs) [1], [2], moving the architecture form a centralized one to a distributed one. Distributed cloud networks, namely federated cloud, consist of multiple regional DCs, which are geographically distributed and interconnected. Thus, cloud services can be placed as nearby as possible to end users, which may ultimately improve the Quality of Experience (QoE) of the offered cloud services.

On the other hand, users are connected to their cloud services from laptops, tablets and smartphones, often while they are on the move and that is through different access networks (mobile, WiFi, small cell networks, or a combination thereof [23]). To ensure mobile users acceptable QoE, they need to be always granted optimal end-to-end connectivity to their cloud services during the entire course of the service usage. Indeed, as users are mobile and frequently change their data anchor access routers, it is very likely to have users connected to an optimal data anchor router [3], [4] but accessing a cloud service hosted at a geographically distant DC. To solve this issue, the authors have introduced in [5] the Follow Me Cloud (FMC) concept, which enables mobile users to be always connected via optimal data anchor and mobility gateways and access their data and cloud services from optimal DCs, i.e. geographically/topologically nearest (or DC optimality defined in any other metric such as load, processing speed and service type [6]). As described in [5], the FMC concept consists of several modules, which aim at ensuring optimal end-to-end connection to the cloud by migrating (when necessary [7], [8]) Virtual Machines (VMs) between DCs according to users' mobility and operators' policies. For more details on the FMC concept, its evaluation and VM migration algorithms, the interested readers may refer to [7] and [8].

FMC is initially designed to support mobile users of 3GPP mobile networks, whereby user mobility and data delivery are handled by specific mobile Core Network elements, such as Mobility Management Entity (MME), Serving Gateways (S-GWs) and Packet Data Network Gateways (PDN-GWs). It is therefore somehow restricted to the

3GPP architecture. The key objective of this paper is to extend the FMC concept to support mobile users that access any radio access technology to connect to the cloud. In this vein, there are two key challenges that have to be addressed: (i) managing user mobility and (ii) migrating the cloud service, mainly VM, between DCs. Concerning the first challenge, there are several ways to manage user mobility at the IP level. Mobile IP, Mobile IPv6 and Distributed Mobility Management (DMM) are notable examples [24]. Regarding service migration, alternatively VM migration, there are two separate issues to handle. The first one relates to the transfer of VM between DCs. VM migration is currently easily possible in case of intra-DC migration. It has become also technologically possible even between geographically separated DCs. Most of the recent hypervisors (e.g., KVM and XEN) are able to move VMs between DCs. However, service migration, through VM migration and during runtime, is still challenging. Indeed, VM relocation requires the change of IP addresses, which complicates maintaining the service continuity during migration. Several solutions have been proposed to address the VM relocation issue at the network level at either level 2 or level 3. The former takes place in case there is Ethernet continuity between DCs; Transparent Interconnection of Lots of Links (TRILL) and Shortest Path Bridging (SPB) can be used. The latter can be supported by Local/Identifier Separation Protocol (LISP). For mixed L2/L3 VM relocation, virtual extensible LAN (VXLAN), Network Virtualization using Generic Encapsulation (NVGRE), or Stateless Transport Tunneling (STT) can be used.

To extend the usability of the FMC concept beyond 3GPP mobile networks, LISP represents the best alternative and that is for several reasons as summarized below:

- It can be used to manage user mobility and avoid triangular routing, which represents the main drawback of Mobile IP (and IPv6);
- It can be implemented at either IP router or data anchor gateways (i.e., PDN-GW in case of 3GPP networks);
- With few modifications, it can efficiently ensure VM relocation in real time with negligible downtime [9].

On the other hand, one of our objectives is to propose a LISP-based FMC in a complete virtual environment using the trendy concept of Network Function Virtualization (NFV) [10], [11], [27]. Indeed, NFV 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 domain. To achieve this objective, we implemented all LISP protocol elements using ClickOS [12], a minimalist and efficient Operating System (OS) dedicated to support NFV, implemented on top of XEN and integrating a software-based router (namely, Click modular routers).

The remainder of this paper is structured as follows. Section II describes some research work related to LISP, NFV, and service migration in the cloud. The LISP-based FMC architecture and its virtualization based on NFV are described in Section III. Section IV presents details on the envisioned implementation and discusses some performance results. The paper concludes in Section V.

II. RELATED WORK

A. SERVICE MIGRATION

Distributed Clouds, or federated cloud, refer to the connection of geographically distributed DCs together into a common resource pool, to deliver a variety of cloud services. Upon reception of a service request, one of these DCs will be chosen to deliver the requested service over the underlying transport network to end-users. The distribution of cloud computing resources over different locations in the network is beneficial for different reasons such as increasing availability, reducing bandwidth cost, and reducing latency by locating resources nearby users. To efficiently handle user requests, there is a need to define a cloud management procedure. This procedure directs service requests from users to the optimal DC, which satisfies user constraints (cost), optimizes the usage of network resources (load balancing) and ensures application's Quality of Service (QoS)/QoE. Furthermore, this cloud management procedure must be able to migrate all or portions of services between DCs if one of the selected criteria is no more satisfied (e.g. degradation in QoS). Obviously, redirecting user requests to geographically nearby DCs seems to be an efficient solution. However, for popular services (i.e., over a certain region), redirecting all service requests to the geographically nearest DC can overload this latter, potentially causing a degradation in QoS/QoE. Therefore, more sophisticated solutions need to be used for cloud resource management.

In [13], a cloud management middleware is proposed to migrate part of a user's service (constituted by a set of VMs) between DC sites in response to workload change at the DCs. Based on workload monitoring at each DC, the middleware initiates VM migration in order to move application components (geographically) closer to the client. Volley [14] is an automatic service placement for geographically distributed DCs based on iterative optimization algorithms. Volley migrates services to new DCs, if the capacity of a DC changes or the respective users change their locations: it chooses a DC nearby users' new locations. The authors in [15] propose a DC selection algorithm for placing a VM requested by a user such that it minimizes the maximum distance between any two DCs. The DC selection problem was formulated as a sub-graph selection problem. The demonstrator described in [16] shows how services can be placed according to information retrieved from an ALTO (Application-Layer Traffic Optimization) network server.

However, most of the above mentioned research works focus rather on issues related to the VM migration process than on issues related to the VM mobility management, particularly, when migration is done between two IP domains (i.e., in a Wide Area Network – WAN). Solutions, described in [17]–[19] integrate IP mobility solution (Mobile IP) directly with the hypervisor. While in [17] and [18], the hypervisor interacts with VM before and after its migration to update IP addresses in the VM's routing table, in [20] the hypervisor (called HyperMIP) is called each time a VM is created, destroyed or migrated. Although these solutions achieve the goal of live migration of VMs, they fail in terms of performance, as long downtimes can be experienced, particularly in case of [17] and [18].

Recently, authors in [9] have achieved a sub-second downtime when using LISP to migrate large-scale VMs. They modified LISP to support VM migration and ensure rapid redirection of traffic to reduce the downtime. However, as it will be discussed later, the solution proposed in [9] considers that the hypervisor has to generate LISP messages to ensure VM transparent mobility. Such mechanism cannot be easily integrated into the Hypervisor. Indeed, it depends on the envisioned implementation and is not easy to integrate into the network operator domain.

B. LISP

The traditional IP addressing approach associates both location and identity to a single IP address space, making mobility a very challenging task as identity and location are integrated together. LISP separates between the location and identity by using Routing Locators (RLOCs) and Endpoint Identifiers (EIDs). Both RLOC and EID could be an IP address. EIDs should not be used as RLOCs, since the latter is needed to forward packets in the Internet, while EIDs are local to an IP subnet. LISP uses a mapping and encapsulation scheme at the data-plane level, by mapping the EID address to a RLOC and encapsulating the packets into other IP packets before forwarding them through the IP transit. Usually, a LISP site is managed by at least one tunneling LISP router (xTR), having two functionalities: IP packet encapsulation (packet received by a terminal; ingress functionality, or ITR) and packet decapsulation (packet received by the network; egress functionality, or ETR).

In order to guarantee EID reachability, LISP uses a mapping system that includes a Map Resolver (MR) and a Map Server (MS), and a cache table at each xTR. When a station has a packet to send, the EID of the remote station is used in the destination address. Once reaching the ITR (ingress part of xTR), the latter encapsulates the sent packet by adding three headers (LISP, UDP, and IP) and fixing the fields "*Source Routing Locator*" and "*Destination Routing Locator*" of the LISP header with source xTR RLOC address and the destination xTR RLOC, respectively. The mapping between EID and the corresponding destination xTR RLOC is firstly searched in the local cache. If the mapping does not exist, a *Map_Request* is sent to the Map Resolver, which replies with a *Map_Reply* if the mapping is found; otherwise it redirects this request to the Map Server. The latter searches in its local database to find a xTR that would correspond to this EID, and replies with a *Map_Reply* if it exists. Otherwise, it replies with a *Negative_Map_Reply.* It is worth noting that the Map Server receives *Map_Registers* from ETRs and registers EID to RLOC mapping in the mapping database.

Thanks to the separation between location and identifier, LISP represents an interesting solution to support station mobility, avoiding the mobile IP drawback (e.g., triangular routing). Indeed, a station can move from one location to another location without changing its EID. Only RLOC has to be updated at MS/MR. Furthermore, with few modifications, LISP can ensure VM migration with short downtime [9]. For the above reasons and more, LISP is chosen to extend the FMC concept to support mobile users of non-3GPP networks.

C. NETWORK FUNCTION VIRTUALIZATION (NFV)

Network Function Virtualization (NFV) aims at decoupling the software part from the hardware part of a carrier network node, using virtual hardware abstraction techniques. The goal is to run network functions as software in standard VMs on top of a virtualization platform in a general-purpose multi-service multi-tenant node (e.g. a carrier grade blade server). Appropriate Software Defined Networking (SDN) technologies can be employed to interconnect different NFVs on different VMs in the same DC or across multiple DCs. NFV would give high degree of flexibility to network operators in the deployment of their resources on the cloud. The technologies enabling the virtualization of network functions are currently at an early stage. Standardization activities are also ongoing in ETSI and IRTF [21], [22], where recent groups on network function virtualization have been launched. Note that the NFV ETSI group is supported by leading telecom operators and equipment vendors. It has already published different documents to build the basis of the NFV architecture and system.

As a technology enabler for NFV, the ClickOS initiative [12] has been proposed. It is based on open source tools. It is a minimal OS based on XEN software platform optimized for middlebox processing. Hereby, middlebox refer to all hardware-based network appliances used to run a specific network function (e.g., firewall, Intrusion Detection System – IDS, and Network Address Translation – NAT.). ClickOS includes a software modular router, namely Click, that processes packets and acts as a router or a firewall. As one of the challenges of NFV is to enable processing of packets as fast as in dedicated hardware-based solutions, ClickOS leverages the XEN I/O subsystem by changing the back-end switch, virtual network devices, and back/front end-drivers. The results presented in [12] show that ClickOS is able to forward packets at around 30Gbps, proving that NFV could achieve a performance similar to that of dedicated hardware-based solutions.

To take advantage of the flexibility offered by NFV, and to ease the deployment of FMC, particularly in the cloud domain, we envision implementing LISP elements on top of ClickOS. Virtualizing the LISP elements, especially xTR, requires high performance forwarding system, which is now enabled by ClickOS. All LISP protocol elements were implemented by extending the Click software router. The xTR are

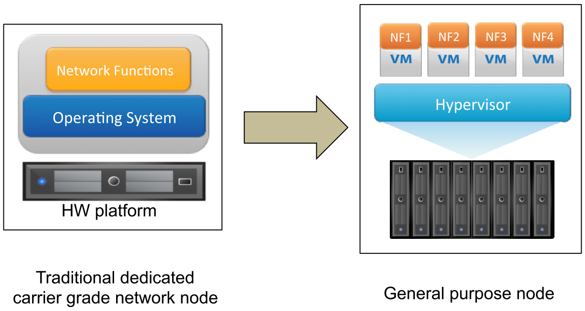

FIGURE 1. **The envisioned NFV principle.**

Click modules that implement the LISP functionality, such as encapsulation and EID-to-RLOC mapping and forwarding. The MR and MS elements are regrouped in one entity, and implemented in a standalone ClickOS VM. It shall be noted that MR and MS do not require a high packet processing capability. Indeed, they may be implemented on MiniOS by adding a database management system to maintain the EID-to-RLOC mapping of all the LISP architecture. Besides the LISP elements, we also implemented the FMC controller, as described in [5], on ClickOS.

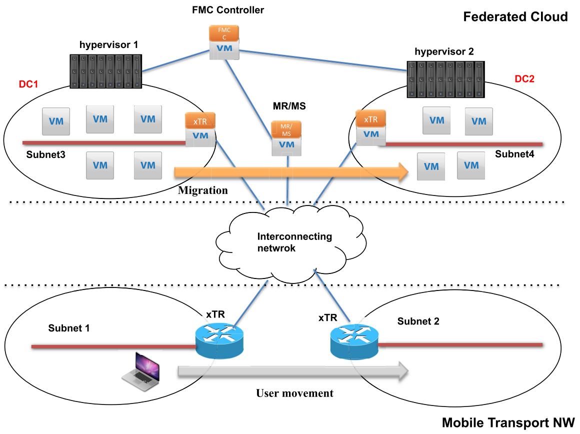

FIGURE 2. **The envisioned LISP-based follow me cloud architecture.**

III. LISP-BASED FMC

A. ARCHITECTURE

Fig. 2 depicts the envisioned LISP-based FMC architecture. The FMC architecture consists of two main domains: (i) the cloud domain and (ii) the mobile transport network domain. The cloud domain, in turn, consists of a number of geographically distant DCs, forming a federated cloud. Each DC is connected to the Internet through a xTR router. The Mobile domain contains IP subnets, also interconnected through xTR routers. The architecture also comprises a LISP MR/MS element. As mentioned earlier, LISP entities (MR/MS and xTRs) are developed as Virtualized Network Functions (VNFs) and deployed on VMs in the cloud. Here, the xTR routers of the mobile transport network domain could be VNFs or classical hardware routers. Besides the usual LISP entities, the architecture also comprises the FMC Controller element in the form of a VNF. As described in [5], FMCC is in charge of tracking user mobility; deciding on VM migration, triggering it if needed, and selecting the target DC.

In this paper, the scenario we envision and want to implement is depicted in Fig. 2 and is described as follows. A mobile user, with a service hosted in DC1 and initially connected to subnet1, moves to subnet2. The xTR router of subnet 2 notifies MR/MS about this movement. MR/MS accordingly updates its cache and informs the FMC controller about the new location of the user. As specified in [5], [7], and [8], the FMC controller uses some intelligence to decide whether to migrate or not the user's service to a new DC corresponding to the new location of the user. If the FMC controller decides migrating the service, intuitively the VM hosting the service, it asks Hypervisor 1 to launch the procedure. Since the VM is migrated to Hypervisor 2, the xTR router of subnet 4 is informed about this VM migration. It accordingly informs MR as well as the xTR router of subnet 3 (and also other xTR routers involved in communication with the VM) about this VM migration. Finally, the MR/MS resolver updates its cache and notifies the FMC controller about this change. The FMC controller then considers that the VM migration has been done with success.

B. SERVICE CONTINUITY

To ensure service continuity, a LISP-assisted live migration of services should be capable to: (i) maintain VM EID when migrating it from its current DC to the target DC; (ii) update RLOC of the target xTR router to include the VM's EID; (iii) inform the MR server and all xTR routers involved in a communication with the migrated VM to update RLOC of the migrated VM; and (iv) inform the old xTR router to erase the VM EID from its cache.

LISP does not impose any constraints on the EID and RLOC identifiers, where IP addresses are usually used. In this work, we assume that EID is the first IP address obtained by a VM, and that RLOC is the IP address of the corresponding xTR router. Furthermore, we consider that a VM's EID is registered at the initial xTR router with /24 (or any large prefix of IP subnet). EID is mapped to RLOC of the source xTR router at MR/MS as well as at the caches of xTR routers communicating with the VM. As mentioned earlier, when the FMC controller triggers a VM migration request to the source hypervisor, the latter migrates the VM to the target DC. When the VM is migrated to the target hypervisor, EID of the migrated VM should be maintained and the xTR router has to be informed about this migration. Different approaches exist to inform the target xTR router about the migration of a VM. In one approach [20], the xTR router becomes aware of the new VM until the VM indeed initiates communications, i.e. by finding that the source IP (migrated VM's EID) is not belonging to its IP subnet. Although this solution does not require any signaling messages, it can break the current VM connection and hence does not ensure service continuity. In fact, if the VM has no packet to transmit, the current xTR router communicating with the VM may continue using old RLOC. An alternative to this approach was proposed in [9], whereby LISP is used in the control plane to inform the source and target xTR routers about the success of a VM migration. In this solution, a new message, dubbed, LISP Change Priotity (CP) message, is introduced. The CP message allows: (i) the target hypervisor to inform the target xTR router about the migration of a new VM to the target DC (including the VM's EID) and (ii) to update the cache (RLOC-EID mapping) of other xTR routers. However, this solution requires modifying the hypervisor, making it hard to implement in real-life as the hypervisor software is independent from the operator (as well as the LISP domain). In this paper, we consider another approach, wherein the FMC controller informs both xTR routers (handling the involved-DC domains) about the change in the VM's RLOC. Indeed, as the FMC controller is integrated within the LISP domain (it already communicates with MR/MS to track users' location), it could easily know the xTR router handling a DC domain. This could be obtained by sending a message to MR/MS to know the xTR router handling the DC's IP domain.

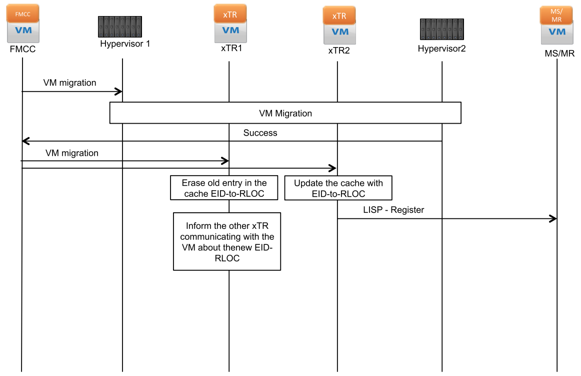
FMCC
VM
FMCC
Hypervisor 1
xTR
VM
xTR1
xTR
VM
xTR2
Hypervisor2
MS/
MR
VM
MS/MR
VM migration
VM Migration
Success
VM migration
Erase old entry in the cache EID-to-RLOC
Update the cache with EID-to-RLOC
LISP - Register
Inform the other xTR communicating with the VM about thenew EID-RLOC

FIGURE 3. **Message exchange sequence for a VM migration.**

Once the xTR router becomes aware of the reception of a new VM, it sends a message to the MR/MS to update its RLOC by including the migrated VM's EID. In this case, the migrated VM's EID is in the form of the initial IP address but with /32 prefix. Therefore, RLOC of the target xTR router is mapped to both its subnet prefix and VM's EID prefix (/32). Furthermore, the former xTR router erases the old EID-to-RLOC entry from its cache. To speed up the traffic redirection, the source xTR router uses a new LISP message (as in [9]) to inform the other xTR router which was communicating with the concerned VM so it accordingly updates the VM's RLOC. The xTR router should maintain for each active connection a list of xTR routers involved with them. Fig. 3 depicts the complete sequence of messages exchanged to migrate a VM between two DCs in the envisioned LISP-based FMC implementation.

C. SERVICE MIGRATION ALGORITHM

The possible need for a FMC service migration can be intuitively noticed when a mobile user changes his xTR anchor gateway (i.e., followed by a RLOC update), i.e., changes his RLOC (IP address of its attached xTR). A change of the RLOC associated to an EID should be notified to the FMC controller and this operation can be handled by MR/MS. Once deemed appropriate, the FMC controller takes the decision to migrate the VM running the user service [5], [7], [8]. This decision may be based on the service type (e.g., an ongoing video service with strict QoS requirements may be migrated) [13], [25], content size, task type of the service (e.g., Emergency warning services and delay-sensitive measurement reporting services have to be always migrated to the nearest DC) [26], and/or user class. It is worth noting that the service migration decision relies on several attributes/criteria that could be conflicting and may depend on users' expectations on the service (e.g., QoS/QoE and cost) and network/cloud provider policies (e.g., at each RLOC update, load balancing, maximize using the DC resources). An estimate of the cost/overhead to be possibly incurred shall be compared against benefits to the cloud in terms of traffic distribution and also to end users in terms of QoE [7], [8].

In this paper, the service migration algorithm is based on the algorithm proposed in [7], using MDP (Markov Decision Process) with few adaptations. We assume that mobile users randomly move between IP networks, and each time a mobile user crosses a new domain (i.e., RLOC update), an action has to be taken, consisting in migrating or not a service. A reward is associated with each action. This reward is based on a specific defined function that captures the tradeoff between the migration cost and user's expected QoE.

With the trigger destined to the source hypervisor, the FMC controller includes information on the selected target DC with the right service and right content to serve the mobile user in its new location and to initiate the service migration. As a service may consist of multiple cooperating sessions and pieces, the decision has to be made indicating whether the service has to be fully or partially migrated, and that is while considering the service migration cost; e.g., cost associated with the initiation of a new VM at the target DC, cost (if any) associated with the release of resources at the source DC, and cost associated with the bandwidth consumption due to traffic to be exchanged between the DCs (hypervisors) and also the FMC controller. As mentioned earlier, the FMC controller is also in charge of informing the xTR routers managing both DCs' domains about the new RLOC of the migrated VM. Therefore, once the decision of service migration is taken, the FMC controller generates a LISP MAP REQUEST message to MR/MS seeking the xTR routers in charge of both DCs' domains. Fig. 4 shows the messages exchanged between MR/MS, the FMC controller, and other entities to decide on a VM migration.

IV. RESULTS

As mentioned earlier, the implementations of the entities forming the envisioned LISP-based FMC architecture are made as VNFs running on top of Click Routers (i.e., on ClickOS VMs) and deployed on XEN. DCs are emulated using KVM, as it allows easy migration of VMs between hypervisors. In fact, KVM migrates only the RAM (Random

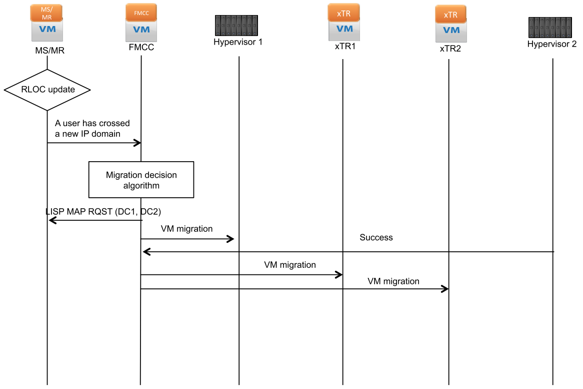

FIGURE 4. Message exchange sequence to decide on a VM migration.

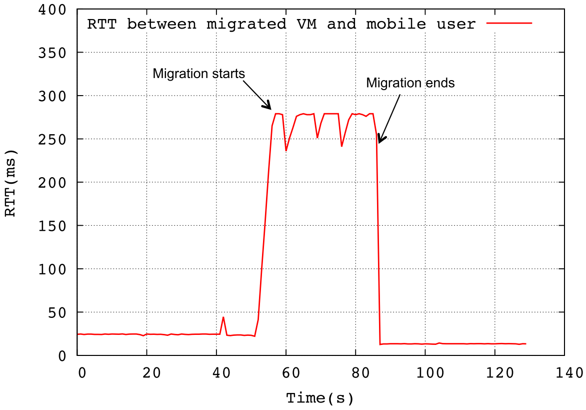

FIGURE 5. The RTT between the migrated VM and mobile user.

Access Memory) content between the involved DCs. Moreover, the only requirement to achieve a VM migration is to have a shared network storage (NFS – Network File System) between the hypervisors. The emulated scenario corresponds to the architecture depicted in Fig. 2. We consider a mobile user moving between two IP domains, and whenever it enters into a new domain, a VM migration is triggered. Several tests were launched by modifying the Round Trip Times (RTTs) between DCs and between the FMC controller and the xTR routers. The metrics we measured are as listed as follows:

- The downtime duration, which corresponds to the time when the VM is not available.
- RTT between the mobile user and the remote VM.
- The bandwidth required to migrate a VM between DCs.
- The migration duration, which corresponds to the total time elapsed since a VM migration is triggered till the VM is fully migration.

The migrated VM is running a Ubuntu 8.04, with 1GBytes of RAM and one core processor. DCs (KVM managers) run on Ubuntu14.04 with 8 GBytes of RAM and one core processor.

Fig. 5 plots the instantaneous RTT between the migrated VM (service) and the mobile user. This result is obtained using PING messages between the mobile user and the remote VM. Here, (i) RTT between the DCs is set to 10ms; (ii) RTT between the FMC controller and xTR1 is about 1ms; and (iii) RTT between the FMC controller and xTR2 is about 10ms. In the first part of the testbed run, the mobile user is connected to DC1, where the measured RTT is around 12ms, representing a good quality connection. From t = 55s, the mobile user moves to another network (i.e., a new IP domain) that degrades the quality, and the measured RTT increases to around 250ms. At this moment, the FMC controller decides launching a VM migration to move the user's service from DC1 to DC2, as the latter is deemed to be an optimal one. The VM migration starts at t = 58s and ends at t = 84s, yielding a migration duration of 26 seconds. The downtime of the VM is around 7.5ms. This downtime is not shown in Fig. 5 as the time is scaled by seconds. This downtime is mainly due to the fact that the FMC controller waits until the VM migration is completed before notifying both xTRs about the change in VM's EID. Furthermore, the ICMP (Internet Control Message Protocol) echo reply continues to be sent from the VM while instantiated on DC1, i.e. the KVM hypervisor maintains the VM active in DC1 since the migration is not complete. Accordingly, the downtime is mainly caused by the RTT between the DC and the FMC controller and the RTT between the FMC controller and the xTR routers, as also shown in Fig. 6. From t = 87s, the mobile user is served by DC2 with a short RTT in the vicinity of 3ms, as DC2 represents the optimal DC for connecting the mobile user to its service.

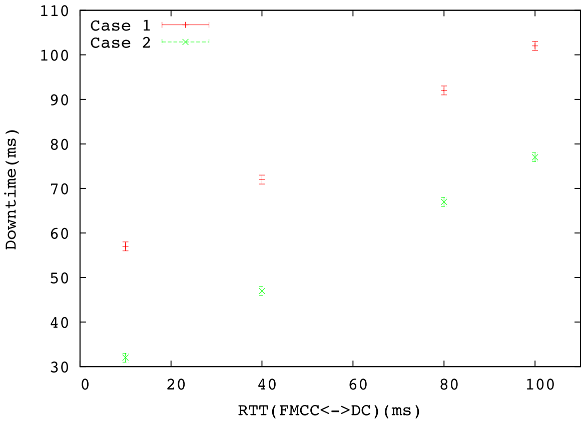

FIGURE 6. The downtime duration.

Fig. 6 shows the downtime duration when the VM is migrated from DC1 to DC2 for different RTTs between the FMC controller and the DC (more specifically DC2). Each point on the graphs represents the average value of several tests. We considered two configurations: (i) case 1: the RTTs between the FMC controller and xTR1 and xTR2 are set to 100ms and 10ms, respectively; (ii) case 2: the RTTs between the FMC controller and xTR1 and xTR2 are set to 50ms and 50ms, respectively. We clearly remark that the downtime duration is proportional to the RTT between the FMC controller and the target DC. This is intuitive as the longer the time needed to have the information from the DC about the success of a VM migration, the longer the time to accordingly inform the xTR routers and hence to redirect the traffic to xTR2. It is worth noting that we can draw the same conclusion for the impact of the RTT between the FMC controller and the xTR routers on the downtime. In Fig. 6, the maximum downtime experienced is 100ms (obtained in Case 2), which remains minimal and without much noticeable impact on the service quality. Clearly, the downtime is mainly caused by the LISP mobility management process and its capacity to rapidly inform the xTR routers about the VM migration. Indeed, as depicted in Fig. 3, in the envisioned LISP-based FMC solution, the downtime represents the time taken by the DC to inform the FMC controller and the time taken by the controller to subsequently inform the xTR routers. The size of the VM has no major impact on the downtime since KVM activate the VM in DC2 only after the migration is completed, which confirms the observations made in [9].

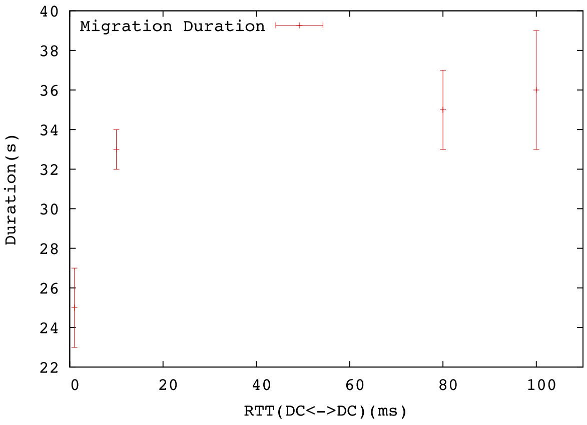

FIGURE 7. **The VM migration duration.**

Fig. 7 plots the time duration for migrating a VM from DC1 to DC2 and that is for different values of the RTT between the DCs. Each point on the graph represents an average value for several tests. Here, the link bandwidth is set to 100 Mbps. We observe that the VM migration duration becomes practically independent from the RTT between DCs, when the latter exceeds 10ms. This is attributable to the fact that the VM migration is based on TCP (Transmission Control Protocol) which is impacted more by the link bandwidth than by the link RTT. Certainly, RTT has an impact, but its impact remains relatively small, in comparison to the impact of the link bandwidth, as depicted in Fig. 7.

TABLE 1. **The bandwidth used for VM migration.**

	Maximum	Mean	Minimum
Bandwidth (Mbit/s)	92.938	86.335	68,449

Tab. 1 shows an example of the bandwidth used for migrating the VM from DC1 to DC2. Here, the link bandwidth is also set to 100 Mbps and its RTT is about 10ms. The showed results represent the average, the maximum and minimum used bandwidth for several tests. The objective beneath Tab. 1 is to show the cost of VM migration in terms of the consumed bandwidth. Clearly, we notice that the required bandwidth (around 80 Mbps) does not represent an issue nowadays, as DCs are currently interconnected with high-speed data link. Tab. 1 results indeed demonstrate that the cost of migrating a VM with the considered specifications is not a major technical issue, particularly when using hypervisors such as KVM, which migrates only a part of VMs.

V. CONCLUSION

In this paper, we presented a LISP-based implementation of the FMC concept. This version of FMC enables the concept of intelligent service mobility for mobile users not only of 3GPP networks, but also of other non-3GPP access networks such as WiFi and small cell networks. Thanks to the nice 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 instantiated by the ClickOS framework. The results obtained from a real-life testbed of the envisioned LISP-based FMC architecture showed that the architecture achieved its main design goals, transferring users' services in the order of milliseconds, and accordingly improving users' QoE as in FMC, optimal DCs are always selected for cloud service delivery. Whilst the obtained results are highly encouraging, issues pertaining to scalability of caches at FMC controller, MR/MS, and xTRs deserve further investigations. This defines one future research direction for the authors with relevance to FMC.

REFERENCES

[1] R. Miller. (May 2012). Solar-powered micro data center at rutgers. Data Center Knowledge. [Online]. Available: http://www. datacenter-knowledge.com/archives/2012/05/31/solar-powered-micro-data-center-at-rutgers/

[2] R. Miller. (Jul. 2012). AOL gets small with outdoor micro data center. Data Center Knowledge. [Online]. Available: http://www. datacenterknowledge.com/archives/2012/07/06/aol-micro-data-centers/

[3] T. Taleb and A. Ksentini, "VECOS: A vehicular connection steering protocol," *IEEE Trans. Veh. Technol.*, doi: 10.119/TVT.2014.2327241.

[4] T. Taleb, K. Samdanis, and A. Ksentini, "Supporting highly mobile users in cost-effective decentralized mobile operator networks," *IEEE Trans. Veh. Technol.*, vol. 63, no. 7, pp. 3381–3396, Sep. 2014.

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

[6] M. Bagaa, T. Taleb, and A. Ksentini, "Service-aware network function placement for efficient traffic handling in carrier cloud," in *Proc. IEEE WCNC*, Istanbul, Turkey, Apr. 2014, p. 68.449.

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

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

[9] P. Raad, S. Secci, D. C. Phung, A. Cianfrani, P. Gallard, and G. Pujolle, "Achieving sub-second downtimes in large-scale virtual machine migrations with LISP," *IEEE Trans. Netw. Service Manage.*, vol. 11, no. 2, pp. 133–143, Jun. 2014.

[10] T. Taleb *et al.*, "EASE: EPC as a service to ease mobile core network," *IEEE Netw. Mag.*, Aug. 2014.

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

[12] J. Martins *et al.*, "ClickOS and the art of network function virtualization," in *Proc. 11th USENIX Symp. NSDI*, Seattle, WA, USA, Apr. 2014, pp. 459–473.

[13] B. Malet and P. Pietzuch, "Resource allocation across multiple cloud data centres," in *Proc. 8th Int. Workshop MGC*, Bangalore, India, 2010, Art. ID 5.

[14] S. Agarwal, J. Dunagan, N. Jain, S. Sariou, A. Wolmann, and H. Bhogan, "Volley: Automated data placement for geo-distributed cloud services," in *Proc. 7th Symp. Netw. Syst. Design Implement. (NSDI)*, San Jose, CA, USA, 2010, pp. 17–32.

[15] M. Alicherry and T. V. Lakshman, "Network aware resource allocation in distributed clouds," in *Proc. IEEE INFOCOM*, Orlando, FL, USA, Mar. 2012, pp. 963–971.

[16] M. Steiner *et al.*, "Network-aware service placement in a distributed cloud environment," in *Proc. ACM SIGCOMM*, Helsinki, Finland, Aug. 2012, pp. 73–74.

[17] H. Watanabe, T. Ohigashi, T. Kondo, K. Nishimura, and R. Aibara, "A performance improvement method for the global live migration of virtual machine with IP mobility," in *Proc. 5th ICMU*, 2010, pp. 194–199.

[18] E. Harney, S. Goasguen, J. Martin, M. Murphy, and M. Westall, "The efficacy of live virtual machine migrations over the internet," in *Proc. 2nd Int. Workshop Virtualization Technol. Distrib. Comput.*, 2007, Art. ID 8.

[19] Q. Li, J. Huai, J. Li, T. Wo, and M. Wen, "HyperMIP: Hypervisor controlled mobile IP for virtual machine live migration across networks," in *Proc. 11th IEEE High Assurance Syst. Eng. Symp. (HASE)*, Dec. 2008, pp. 80–88.

[20] "Locator ID separation protocol (LISP) VM mobility solution," Cisco Systems, Inc., San Jose, CA, USA, Tech. Rep., 2011.

[21] IRTF NFV Group. [Online]. Available: http://trac.tools.ietf.org/group/irtf/trac/wiki/nfvrg, accessed Sep. 24, 2014.

[22] ETSI NFV Group. [Online]. Available: http://www.etsi.org/technologies-clusters/technologies/nfv, accessed Sep. 24, 2014.

[23] T. Taleb and A. Ksentini, "QoS/QoE predictions-based admission control for femto communications," in *Proc. IEEE ICC*, Ottawa, ON, Canada, Jun. 2012, pp. 5146–5150.

[24] T. Taleb, A. Jamalipour, Y. Nemoto, and N. Kato, "DEMAPS: A load-transition-based mobility management scheme for an efficient selection of MAP in mobile IPv6 networks," *IEEE Trans. Veh. Technol.*, vol. 58, no. 2, pp. 954–965, Feb. 2009.

[25] T. Taleb and A. Ksentini, "On efficient data anchor point selection in distributed mobile networks," in *Proc. IEEE ICC*, Budapest, Hungary, Jun. 2013, pp. 6289–6293.

[26] L. Karim, N. Nasser, and T. Taleb, "Dynamic multilevel priority packet scheduling scheme for wireless sensor network," *IEEE Trans. Wireless Commun.*, vol. 12, no. 4, pp. 1448–1459, Apr. 2013.

[27] T. Taleb, A. Ksentini, and A. Kobbane, "Lightweight mobile core networks for machine type communication," *IEEE Access Mag.*, Sep. 2014.

ADLEN KSENTINI (SM'14) is currently an Associate Professor with the University of Rennes 1, Rennes, France. He is a member of the INRIA Rennes team Dionysos. He received the M.Sc. degree in telecommunication and multimedia networking from the University of Versailles, Versailles, France, and the Ph.D. degree in computer science from the University of Cergy-Pontoise, Cergy-Pontoise, France, in 2005, with a dissertation on QoS provisioning in IEEE 802.11-based networks. His other interests include future Internet networks, mobile networks, QoS, QoE, performance evaluation, and multimedia transmission. 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. He was a recipient of the Best Paper Award from the 2012 IEEE International Conference on Communications, and the 2005 ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems. He is the Technical Program Committee Chair of the IEEE Third Workshop on Standards on Telecommunication (collocated with Globecom 2014), and the Workshop Chair of the 2014 ACM/IEEE International Conference on Heterogeneous Networking for Quality, Reliability, Security, and Robustness. 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 Communications Society conferences, ICC/Globecom, ICME, WCNC, and PIMRC.

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 science from Tohoku University, Sendai, Japan, in 2001, 2003, and 2005, respectively. He is a Faculty Staff with the School of Engineering, Aalto University, Espoo, Finland. He has been a Senior Researcher and the 3G Partnership Project Standardization Expert with NEC Europe Ltd., Heidelberg, Germany. He led the NEC Europe Labs Team, working on research and development projects on carrier cloud platforms. Prior to his work at NEC until 2009, he was an Assistant Professor with the Graduate School of Information Sciences, Tohoku University. His research interests include architectural enhancements to mobile core networks (in particular, 3GPP), mobile cloud networking, mobile multimedia streaming, congestion control protocols, handoff and mobility management, inter-vehicular communications, and social media networking. He has also been directly involved in the development and standardization of the evolved packet system as a member of 3GPP's System Architecture Working Group.

Dr. Taleb is a Distinguished Lecturer of the IEEE Communications Society (ComSoc) and a Board Member of the IEEE ComSoc Standardization Program Development Board. He is serving as the Vice Chair of the Wireless Communications Technical Committee, the largest in the IEEE ComSoC. He also served as the Secretary and Vice Chair of the Satellite and Space Communications Technical Committee of the IEEE ComSoc (2006–2010). As an attempt to bridge the gap between academia and industry, he founded and has been the General Chair of the IEEE Workshop on Telecommunications Standards: From Research to Standards, which is a successful event that received the Best Workshop Award from the IEEE ComSoC. He has been on the Technical Program Committee of different IEEE conferences, including the IEEE Global Communications Conference, the IEEE International Conference on Communications, and the IEEE Wireless Communications and Networking Conference, and he has chaired some of their symposia. He is/was on the Editorial Board of the *IEEE Wireless Communications Magazine*, the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, the IEEE COMMUNICATIONS SURVEYS AND TUTORIALS, and a number of Wiley journals. He was a recipient of the IEEE ComSoc Asia Pacific Best Young Researcher Award in 2009, the TELECOM System Technology Award from the Telecommunications Advancement Foundation in 2008, the Funai Foundation Science Promotion Award in 2007, the IEEE Computer Society Japan Chapter Young Author Award in 2006, the Niwa Yasujirou Memorial Award in 2005, and the Young Researcher's Encouragement Award from the Japan Chapter of the IEEE Vehicular Technology Society in 2003. Some of his research work have received best paper awards at prestigious conferences.

FAROUK MESSAOUDI is currently pursuing the Ph.D. degree with the University of Rennes 1, Rennes, France. He received the Engineering degree in computer science from the National High School in Computer Science, Algiers, Algeria, and the M.Sc. degree in embedded systems and information processing from the University of Paris-Sud, Orsay, France. His interests include network virtualization, future Internet networks, and green networks.