# Scalability of a Mobile Cloud Management System

Roberto Bifulco
University of Napoli Federico II
roberto.bifulco2@unina.it

Marcus Brunner
NEC Laboratories Europe
brunner@neclab.eu

Roberto Canonico
University of Napoli Federico II
roberto.canonico@unina.it

Peer Hasselmeyer
NEC Laboratories Europe
peer.hasselmeyer@neclab.eu

Faisal Mir
NEC Laboratories Europe
faisal.mir@neclab.eu

## ABSTRACT

Ubiquitous network access allows people to access an ever increasing range of services from a variety of mobile terminals, including laptops, tablets and smartphones. A flexible and economically efficient way of provisioning such services is through Cloud Computing. Assuming that several cloud-enabled datacenters are made available at the edges of the Internet, service providers may take advantage of them by optimally locating service instances as close as possible to their users. By localizing traffic at the edges of access networks, such an approach may result beneficial for both service and network providers. In this paper we present *Follow-Me Cloud* (FMC), a technology developed at NEC Laboratories Europe that allows transparent migration of services in TCP/IP networks, thanks to the dynamic configuration of a set of coordinated OpenFlow switches located at the edge of the network. In particular, in this paper we analyze the scalability properties of an FMC-based system and propose a role separation strategy based on distribution of control plane functions which enables scale-out of the system. By means of simulation, we prove that the application of the proposed separation strategy results in less state retained by individual OpenFlow controllers and in more effective localization of network traffic.

## Categories and Subject Descriptors

C.2.1 [**Network Architecture and Design**]: Distributed networks

## Keywords

Mobile Cloud Computing, Scalability, Network Management, OpenFlow, Software Defined Networking

## 1. INTRODUCTION

Rich media services that may be accessed anywhere are expected to play a significant role in the mobile apps environment in the next few years, due to their ability to generate significant revenues. These applications have become technically feasible thanks to the ubiquitous availability of multimedia devices and broadband connectivity. Nonetheless, the ability of provisioning such services to large numbers of mobile users is still a technical challenge for service providers. Service provisioning, today, 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*, provided enough bandwidth is available in the end-to-end path connecting clients with servers, rich interactive applications are sensible to other communication metrics, such as delay and jitter. 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. Assuming that several cloud-enabled datacenters are made available at the edges of the Internet, 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 paper, we present *Follow-Me Cloud* (FMC), a technology developed at NEC Laboratories Europe to overcome the current TCP/IP architecture mobility limitations and to support novel Mobile Cloud Computing applications, by providing both the ability to migrate network end-points and to reactively relocate network services depending on users' locations, in order to guarantee adequate performance for the client-server communication and, at the same time, have a precise control on the use of network resources by localizing network traffic generated by applications. The FMC architecture is based on cooperating *FMC controllers* located in the networks of collaborating operators. FMC controllers modify packet forwarding in such a way that location changes of users and of services (e.g., through migration of Virtual Machines) are transparently managed by the network infrastructure, without any need of reconfiguring the end systems. Depending on the users' mobility patterns,

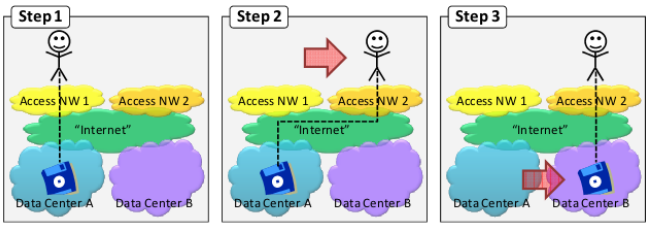


**Figure 1: Follow-Me Cloud Use Case**

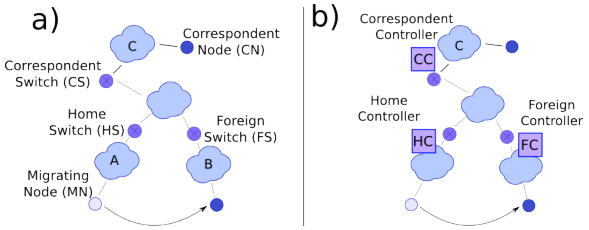


**Figure 2: a. End-points mobility reference scenario; b. FMC distributed architecture**

FMC control-plane decide if, when and where network services have to be migrated.

The coarse flow of actions performed by FMC to manage a migrating service is shown in figure 1. In this simplified use case, initially, a user accesses a dedicated service, e.g., a remote desktop application, in the "home" environment (i.e., usually, his/her office or home). He/She then goes on a trip and, while on the move, accesses the service through an app running in a smartphone. Over time, the smartphone gets connected to the Internet through different mobile network operators. After a change of the terminal's network attachment point, FMC may decide to trigger migration of the service instance to a different network location, e.g. a data center, closer to the new network attachment point. As shown in step 3, the application migrated to the new data center, providing the user with improved Quality of Experience and/or reducing the service provisioning cost for the network operator.

Follow-Me Cloud is implemented using OpenFlow (OF) [1]. It uses the packet filtering and rewriting capabilities of OpenFlow to achieve the seamlessness of migration, and configures network equipment through OpenFlow rules (OFR). OF-enabled networking equipments (i.e., switches) are therefore needed in the network for FMC to work. Nevertheless, OF equipments are only needed at the edges of the network. In this paper we present how FMC achieves network end-points mobility and our solution to solve scalability issues. Cloud services reactive relocation strategies are not discussed here, and will be the topic of future work. The rest of this paper is structured as follows. We discuss related work in section 2. FMC mobility management procedures are presented in section 3, next we discuss the scalability issues of an FMC-based system in section 4, while our solution to support the scale-out of the architecture is presented in section 5 and its evaluation in section 6. Finally, in section 7 we conclude and present future work.

## 2. RELATED WORK

The idea of exploiting the live migration capabilities of modern virtualization technologies for dynamically changing the position of a service instance on a geographic basis has been alredy proposed in the past (e.g. in [6]). Live migration of Virtual Machines is a common practice in virtualized data centers, in which the internal networking infrastructure may be designed as a large single IP subnet. Migration of VMs across the boundaries of a single IP subnet, on the other hand, is not straightforward, as the TCP/IP protocol stack does not provide the needed flexibility in terms of mobility support for network end-points. A technique for the migration of network end-points in a data-center environment by means of a coordinated set of agents is presented in [2]. A first example of the application of OpenFlow to solve network mobility issues is presented in [3], where OpenFlow is used to migrate VMs among different IP subnets, using a network fully composed of OpenFlow switches.

None of these papers present solutions to the control-plane scalability problems that derive from the application of these techniques in large scale networks.

Other approaches, like the *Locator/Identifier Separation Protocol* [4] (LISP), are based on the concept of separation among *locator* and *identifier* addresses. The LISP architecture uses an "alternate topology" to distribute ID/LOC mapping information among routers in the network. LISP does not provide mobility support natively, while it is added in its "mobility extensions" that need direct involvement of mobile devices in the ID/LOC mapping.

MobileIP and MobileIPv6 both enable end-point mobility, but require direct involvement of the moving entities. On the other end, Proxy MobileIP provides mobility without involving end-points, by placing mobility aware proxy devices in the network. In any case, in MobileIP the forwarding of the data packets is usually performed through tunnels and/or triangular routing that are far less efficient in comparison to the FMC solution.

## 3. END-POINTS MOBILITY

Follow-Me Cloud (FMC) enables mobility of network end-points among different IP subnets in a TCP/IP network, both in closed environments, such as data-centers, and on a geographic scale, maintaining all the ongoing network communications of the moving entity active and requiring no modifications to the involved end-points. FMC is applied to a TCP/IP network in which access networks are connected to a "core" network, that provides connectivity among them, through OpenFlow-enabled switches (OFS). Current TCP/IP network architecture uses a single address to both identify and locate a device on the network, making the network unable to support mobility natively. FMC realizes the split of identifier and locator concepts in the edge network, using the OFSes to enforce the splitting in a transparent manner for network end-points. Figure 2.a shows a typical application scenario, with three access networks, and explains also the names used to identify all the network devices involved. Names are assigned from the perspective of a particular migrating node (MN). Using FMC, the MN can migrate from an access network A, to an access network B, without changing it's network configuration (e.g., IP address, Gateway Address). From a network perspective, MN is totally unaware that the access network on which it is residing is changed. All the ongoing communications are kept active, e.g., TCP sessions are not lost. Any correspondent

node, i.e., any node that is on an access network different from A or B and that is communicating with MN, is unaware of the MN location change as well. To provide this result, FMC requires that a new IP address, belonging to the B network, is assigned to MN to work as "locator". The original IP address of MN is still used by MN itself and by any node that is communicating with MN, since it works as "identifier". For any migrated node, the FMC controller (FMC-C) stores the identifier/locator mapping information, that is used to configure involved OFSes with proper OpenFlow rules. The outcome of FMC operations is that each packet destined to a migrated end-point, before traversing the core of the network, is processed to substitute the identifier address with the locator address. Then, the locator address is substituted again with the identifier address, before the packet is delivered to MN. A similar address translation is performed on the source address of packets sent by MN. Hence, the *identifier* address is used to send/receive packets in the edges of the network, while the *locator* address is used in the core of the network, to forward packets to the correct location. Border devices, i.e., OFSes, are in charge of performing the identifier/locator and locator/identifier translations.

# 4. SCALABILITY

FMC uses OpenFlow to provide transparent identifier/locator (ID/LOC in short) splitting, introducing some OFRs to support the redirection of packets to the new location of a migrated node. Even though the used approach is trying to be lightweight, requiring no modifications to the traditional IP routing, when a migration happens, several entities are involved in communication at the network control plane. In particular, FMC controllers need to coordinate their activities, and various OFSes need updated forwarding rules.

In this section we are aiming at assessing the scalability of FMC. Our work is mainly focused on evaluating scalability from the perspective of the number of managed OFRs. We see two main issues: (i) how many rules need to be installed on a particular OFS, and (ii) how many rules must be managed by the FMC controllers. We define the former as a *data-plane* scalability issue and the latter as a *control-plane* scalability issue. *Data-plane* scalability affects the ability of applying FMC when an OFS is involved in many IP address migrations, i.e., when a large number of rules must be installed. An OFS has limited capacity in the number of rules it is able to support, that means, from FMC perspective, that there is a limit on the number of concurrent migrations that can be provided for the network served by that switch. Even if this issue can be a serious problem, and it is worth to be investigated, for the purpose of FMC we assume that it is solvable using a careful partitioning of the network, i.e., reducing the dimension of the network served by a single switch. Clearly, using this *scale-out* approach, we are adding more network devices, hence, we are potentially increasing the work load on the control-plane.

*Control-plane* scalability is, in fact, related to the number of devices and to the total number of OFRs that must be managed by FMC controllers. The number of devices is strictly dependent on the network dimension and on the partitioning level we are applying, e.g., to solve the data-plane scalability issue. The total number of rules is instead directly dependent on the total number of concurrent IP address migrations. To evaluate the total number of generated rules we use the simplistic assumption that there is only one centralized FMC controller[1]. Moreover, we assume that this controller knows in advance on which OFSes rule installation is needed. This way we are able to evaluate the number of rules needed for the packet forwarding redirection, without taking into account implementation-dependent rules. The number of generated rules for the $i$-th IP address migration is:

$$R_i = r_i^{hs} + r_i^{fs} + \sum_j r_{ij}^{cs} \quad (1)$$

where $r_i^{hs}$ and $r_i^{fs}$ are the number of rules installed at the home switch (HS) and foreign switch (FS) for the $i$-th IP address migration, and $r_{ij}^{cs}$ is the number of rules installed at the $j$-th correspondent switch that is exchanging packets with the $i$-th migrated IP address. The number of rules for each migrated IP address is given by the following formulas:

$$r_i^{hs} = \alpha + H_i \quad (2)$$

$$r_i^{fs} = \beta + F_i \quad (3)$$

$$r_{ij}^{cs} = \gamma + C_{ij} \quad (4)$$

The variables $H_i$, $F_i$ and $C_{ij}$ represent the number of nodes from home, foreign and $j$-th correspondent networks that are exchanging packets with the $i$-th migrated IP address. The constants $\alpha$, $\beta$ and $\gamma$ are the fixed number of rules required by FMC for packet redirection per migration[2]. The total number of OFRs managed by the FMC controller is the sum over $i$ of the rules as expressed in (1), plus some rules installed once for each HS or FS:

$$R = 2N + \sum_i R_i \quad (5)$$

where $N$ is the total number of HSes (and since one FS corresponds to each HS, it is also the total number of FSes) involved in IP addresses migrations. From equations (1) and (5) it is clear that the number of rules is directly proportional to the number of concurrent migrations, and to the number of nodes on the HN, FN and CNs that are exchanging packets with migrated addresses.

# 5. DISTRIBUTED CONTROLLER

Providing scalability at the control-plane is a key requirement to make FMC usable in large-scale networks and it is the focus of the work presented in this section. Our aim is to enable the balancing of the FMC operations load among a number of FMC controllers, actually building a FMC distributed controller. Building a distributed controller provides a twofold outcome: apart from scalability, it adds also the flexibility and the distribution of responsibilities required to enable the use of FMC functionalities across administrative boundaries.

The design of the distributed controller follows the principle of distributing knowledge to where it is actually needed. In our case, the needed information at a particular controller is the *identifier/locator* mapping for a given network entity,

[1]The OpenFlow architecture actually suggests a centralized approach to the network control-plane development, through the use of a centralized OpenFlow controller that is connected to a number of OFSes.

[2]In the current implementation $\alpha = 3$, $\beta = 6$ and $\gamma = 3$

while the controllers that need to know about this mapping are the ones that are controlling the HS, the FS and CSes. To manage this information and to distribute it among controllers, we designed the architecture depicted in Figure 2.b. With respect to the migrating node, we identify three different roles among FMC controllers: *Home Controller (HC)* that controls the network to which the *identifier* address belongs to; *Foreign Controller (FC)* that controls the network to which the *locator* address belongs to; *Correspondent Controller (CC)* that controls one or more CSes.

The architecture is flexible enough to enable a single controller to play one, two or all the roles for the same migrating node, e.g., because the same controller is in charge of managing multiple networks. This approach also offers the possibility to adapt the number of controllers used in the network, in order to tackle the actual network load: it is possible to use a scale-out approach, increasing the number of access networks, and consequently reducing the number of nodes per access network. Then, each controller is assigned a number of access networks to manage (the range is from one to all the access networks), with the aim of sharing the overall load.

Since the MN's identifier address belongs to a network managed by HC, this controller is always involved in any MN migration, making it a good candidate to be an "authoritative" repository for the MN mapping information. HC has therefore been selected to be in charge of managing the mapping information for the MN while it is away from its home location.

When MN migrates to a *foreign network* (FN), the HC is notified about the migration and informs the FC that MN is migrating to a network that FC itself is managing. Since FC is in charge of managing the network to which the locator belongs to, we leave to FC the responsibility to generate the locator address for MN[3]. Once the locator is defined, it is sent back to the HC so that both FMC-Cs have the complete information about the ID/LOC mapping to perform the required configurations.

So far, just HC and FC are informed and configured to support the migration. Eventually, a CC will need the information about the ID/LOC mapping, because some endpoints connected to CSes are trying to establish communication with MN. While in the case of HC and FC we know which are the HS and FS, since they are source and destination of the migration, we do not have this information for the CSes. At any point in time, a network entity can be involved in a communication session with MN, and, differently from what we did in section 4, we are not assuming to know in advance who is going to start this communication. To handle this issue, our architecture uses a reactive approach: once a new communication with a CN is established, the involved CC is updated properly. We have to distinguish two cases: (i) the correspondent node starts the communication, and (ii) the migrating node starts the communication.

In the first case, when the CN sends a packet to MN, it always uses the identifier address as the packet's destination. Since the CS does not know yet about the updated ID/LOC mapping, it performs no rewriting on the packet, which is therefore forwarded along the route to MN's home

[3]How the locator address is actually obtained is out of the scope of this work, anyway, it is possible to interact with a network management system, or with a DHCP server to easily generate it.

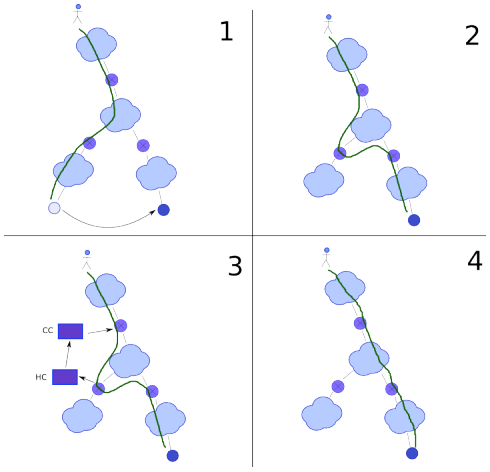


**Figure 3: Overviuew of the eactive update of the identifier/locator mapping at the CC.**

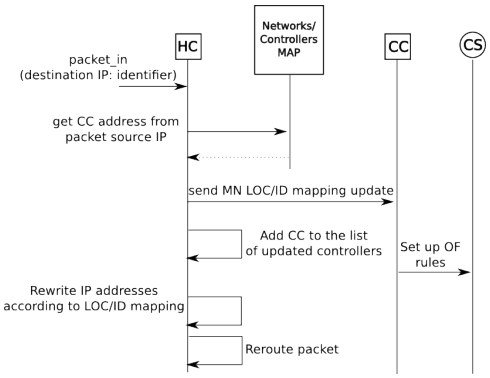


**Figure 4: Reactive update of the identifier/locator mapping at the CC.**

network. Once the packet reaches the HS, it is intercepted and a message to update the ID/LOC mapping at the CC is sent. The packet is then forwarded by substituting the identifier with the locator, using triangular routing. Figure 3 shows an overview of the operations, while figure 4 shows the details of the interactions among HC and CC.

We point out here that it is necessary to store the information of which CC has been updated with the ID/LOC mapping information, since subsequent migrations will trigger an update for all the stored CCs. Otherwise, packets generated by CNs will continue to be directed to the old MN's location.

In the second case, when MN starts the communication with a CN, we have to take into account that the FS is already applying the ID/LOC mapping. Since, on the other hand, the CS does not know yet about the ID/LOC mapping, the packet would reach the CN using the locator address as source, then, CN would use the locator address to talk with the migrated node. If no more migrations happen for MN, there are no problems with this behavior, but, if a new migration happens, then the locator changes, hence, the communication with CN is lost.

Because of this issue, we should ensure that any entity always uses the identifier as destination address. To provide

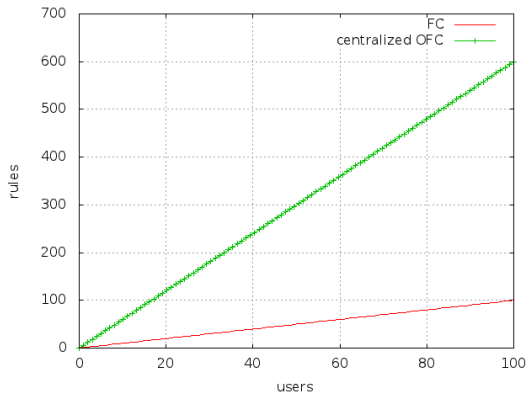


**Figure 5: Number of rules managed by centralized FMC-C and FC, for one migration.**

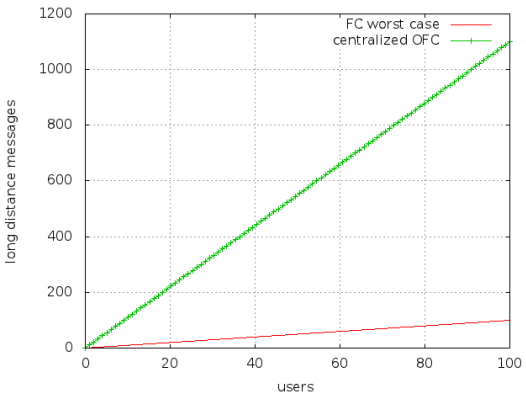


**Figure 6: Number of "long distance" messages sent or received, for one migration.**

the afore mentioned property, FC has to intercept any packet that is sent by the migrated node and is destined to a CN, whose network's FMC controller is still not updated about the ID/LOC mapping for MN. This way, FC is able to send updated ID/LOC information for MN to CC. It is worth to note that the FC does not inform HC about updated CC while MN is still located in its administrative domain. Instead, FC maintains a local list of updated CCs that is sent back to HC when a new MN migration happens.

When MN migrates back to its home network, or to a new network, all the ID/LOC mapping information distributed at different FMC controllers are no longer valid. As for any other MN migration, when HC is informed that MN is migrating, it updates the "old" FC about the MN location change. During this interaction, FC sends its local list of updated CC to HC. Then, HC updates the ID/LOC mapping information on any previously updated CC.

# 6. EVALUATION

We tested our prototype implementation both on a Mininet [5] testbed and on a physical testbed equipped with NEC OpenFlow switches. With regard to the scalability of the distributed architecture, to evaluate the total number of rules we have to modify the $r_i^{fs}$ expression, substituting the definition given in (3) with the following formula, to take into account the reactive update of ID/LOC mapping information:

$$r_i^{fs} = \delta + F_i + J_i \tag{6}$$

where $J_i$ is the number of CNets exchanging packets with the $i$-th migrated node, and $\delta$ is a fixed number of OFRs.

In figure 5 a comparison of the number of rules managed in the case of a centralized FMC controller is compared to the number of rules managed by the most loaded FMC controller of the distributed architecture, i.e., the FC. The number of rules is evaluated for one migration, with a linearly increasing number of nodes that are exchanging packets with MN and that are located at home networks (HNs), foreign networks (FNs), and correspondent networks (CNets).

Taking into account also the OFRs installation time, we have to consider the issue that arises when an FMC controller is located far away from the switch it is controlling, since the network delay has to be added to the installation time. Distributing the architecture enables a better and faster handling of rules installation: even if FMC-related rule installation still needs to face some network delays because of the coordination among the controllers, the majority of actions are performed locally at the controller, that is typically placed close to the network it is controlling.

Assume a scenario in which managed networks are far apart and further assume that a single centralized controller is placed near the FS which is the switch that requires the most rule installation messages. With this deployment, any message exchanged between the controller and HSes/CSes experiences high delay. In the distributed architecture, instead, only inter-FMC controllers messages experience high delay. The number of required "long distance" messages is linearly increasing with the number of nodes in HN, FN and the number of CNets, as shown in figure 6. The figure compares the centralized FMC controller case to the FC of the distributed architecture, in the worst case (i.e., when FC is in charge of updating CCs).

# 7. CONCLUSION AND FUTURE WORK

In this paper we introduced Follow-Me Cloud, a technology that provides mobility features in a TCP/IP network for both users and services, using OpenFlow-enabled equipment at the edges of the network. We presented the implemented mobility technique, the distributed architecture used to support the operations and an early scalability evaluation of the developed prototype. The current implementation and developed test-beds provide a solid base for further research and development. In particular, the logic for deciding when and where to migrate services to needs to be realized. In addition to the original use case of supporting mobile users, it turned out that FMC technology can be used for other scenarios as well, including the general problem of cross-operator service migration and the migration of core mobile network components. These are just two examples of potential use cases for Follow-Me Cloud technology, highlighting its applicability in many domains.


## Acknowledgements

This research has been partly supported by the PRIN SFINGI project funded by the Italian Ministry of University and Research.

## 8. REFERENCES


[1] Openflow - http://www.openflow.org/.

[2] Bifulco, R., Canonico, R., Ventre, G., and Manetti, V. Transparent migration of virtual infrastructures in large datacenters for cloud computing. In *Computers and Communications (ISCC), 2011 IEEE Symposium on* (28 2011-july 1 2011), pp. 179 –184.

[3] Erickson, D., Gibb, G., Heller, B., Naous, J., Underhill, D., Appenzeller, G., Parulkar, G., McKeown, N., and Al., E. A demonstration of virtual machine mobility in an openflow network. *Proceedings of ACM SIGCOMM Demo* (2008), 513.

[4] Farinacci, D., Fuller, V., Meyer, D., and Lewis, D. Locator/ID separation protocol (LISP). Internet-Draft draft-ietf-lisp-13.txt, IETF Secretariat, June 2011.

[5] Lantz, B., Heller, B., and McKeown, N. A network in a laptop: rapid prototyping for software-defined networks. In *Proceedings of the Ninth ACM SIGCOMM Workshop on Hot Topics in Networks* (New York, NY, USA, 2010), Hotnets '10, ACM, pp. 19:1–19:6.

[6] Manetti, V., Canonico, R., Ventre, G., and Stavrakakis, I. System-level virtualization and mobile ip to support service mobility. In *Proceedings of the 2009 International Conference on Parallel Processing Workshops, ICPPW '09* (2009), pp. 243–248.