Achieving Sub-Second Downtimes in Large-Scale Virtual Machine Migrations with LISP

Patrick Raad, *Graduate Student Member, IEEE*, Stefano Secci, *Member, IEEE*, Dung Chi Phung, Antonio Cianfrani, *Member, IEEE*, Pascal Gallard, and Guy Pujolle, *Senior Member, IEEE*

***Abstract*—Nowadays, the rapid growth of Cloud computing services is stressing the network communication infrastructure in terms of resiliency and programmability. This evolution reveals missing blocks of the current Internet Protocol architecture, in particular in terms of virtual machine mobility management for addressing and locator-identifier mapping. In this paper, we propose some changes to the Locator/Identifier Separation Protocol (LISP) to cope with this gap. We define novel control-plane functions and evaluate them exhaustively in the worldwide public LISP testbed, involving five LISP sites distant from a few hundred kilometers to many thousands kilometers. Our results show that we can guarantee service downtime upon live virtual machine migration lower than a second across American, Asian and European LISP sites, and down to 300 ms within Europe, outperforming standard LISP and legacy triangular routing approaches in terms of service downtime, as a function of datacenter-datacenter and client-datacenter distances.**

***Index Terms*—Virtual machine mobility, Locator/Identifier Separation Protocol (LISP), cloud networking.**

I. Introduction

As a matter of fact, network virtualization has revolutionized datacenter networking. Once solely based on physical server and mainframe interconnections, Cloud datacenters increasingly deploy virtualization servers that host, send and receive virtual machines (VMs), to and from local and distant locations. This evolution raises many networking issues in terms of address continuity and traffic routing. When and how should VMs maintain (or use) the same (or multiple) Ethernet and/or IP addresses upon migration, have been and still are open research questions in Cloud networking. Similar challenges appear with the emergence of advanced services such as Infrastructure as a Service (IaaS) [2], often requiring multiple VMs physically located at different sites to communicate with each other as well as with its users, which keep communicating while moving across datacenters [3].

Submitted on July 10, 2013; revised on October 8, 2013. The associate editor coordinating the review of this paper and approving it for publication was F. De Turck.

P. Raad, S. Secci, D. C. Phung, and G. Pujolle are with Sorbonne Universites, UPMC Univ. Paris 06, UMR 7606, LIP6, F-75005, Paris, France (e-mail: {patrick.raad, stefano.seccim, dung.chi.phung, guy.pujolle}@upmc.fr).

P. Gallard and P. Raad are with Non Stop Systems (NSS), 6, boulevard du Courcerin, 77183 Croissy Beaubourg, France (e-mail: {praad, pgallard}@nss.fr).

A. Cianfrani is with the University of Rome I - La Sapienza, P.za Aldo Moro 5, 00185 Rome, Italy (e-mail: cianfrani@diet.uniroma1.it).

A preliminary version of this article is included in the proceedings of 2013 IEEE/IFIP International Symposium on Integrated Network Management (IEEE/IFIP IM 2013) [1].

Digital Object Identifier 10.1109/TNSM.2014.012114.130517

In virtualization nodes, the hypervisor is a software-level abstraction module essential to concurrently manage several VMs on a physical machine. VM migration is a service included in most hypervisors to move VMs from one physical machine to another, commonly within a datacenter. Migrations are executed for several reasons, ranging from fault management, energy consumption minimization, and quality-of-service improvement. In legacy Cloud networks, VM location was bound to a single facility, due to storage area network and addressing constraints. Eventually, thanks to novel protocols and high-speed low-latency networks, storage networks can span metropolitan and wide area networks, and VM locations can span the whole Internet over very long distances.

Multiple solutions are being tested to make VMs' location volatile [4]–[6]. The main trend is to allow transparent VM migrations by developing advanced functionalities at the hypervisor level [4]. In terms of addressing, the main problem lies in the possibility of scaling from public Clouds and intra-provider Clouds to private and hybrid Clouds, i.e., seamlessly migrating a virtual server with a global IP across the Internet and wide area IP networks. Multiple solutions exist to handle addressing issues, ranging from simple ones with centralized ad-hoc address mapping using MAC-in-MAC or IP-in-IP encapsulation, or a mix of both of them, to more advanced ones with a distributed control-plane supporting VM mobility and location management. Several commercial (non-standard) solutions extend (virtual) local area networks across wide area networks, such as [5] and [6] handling differently layer-2 and layer-3 inter-working.

Among the standards to handle VM mobility and addressing issues, we can mention recent efforts to define a distributed control-plane in TRILL (Transparent Interconnection of a Lot of Links) architecture [7] to manage a directory that pilots layer-2 encapsulation. However, maintaining layer-2 long-distance connectivity is often economically prohibitive, a too high barrier for small emerging Cloud service providers, and not scalable enough when the customers are mostly Internet users (i.e., not privately interconnected customers). At the IP layer, the addressing continuity can be guaranteed using ad-hoc VM turntables as suggested in [8], or Mobile IP as proposed in [9], which however can increment propagation and transmission delays due to triangular routing: the traffic has to pass through the VM original network, before being encapsulated and sent to the new VM location.

More recently, the Location/Identifier Separation Protocol (LISP) [10], mainly proposed to solve Internet routing

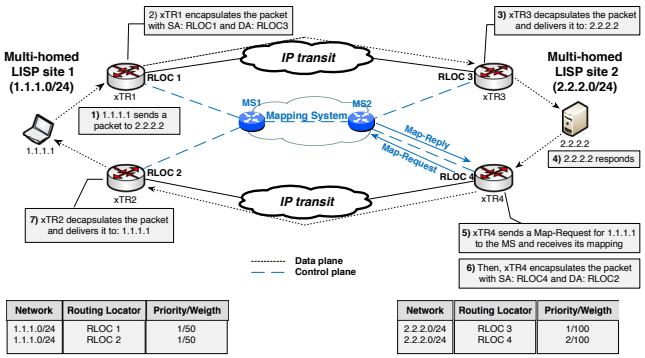


Fig. 1. LISP communications example.

scalability and traffic engineering issues, is now considered for VM mobility and has already attracted the attention for some commercial solutions [11]. In order to efficiently handle locator-identifier mappings, LISP offers a distributed control-plane, decoupled from the data-plane. An advantage of LISP is that it can avoid triangular routing, with encapsulations performed at the first LISP capable IP node. Nevertheless, based on current standards and literature, there are missing functionalities to guarantee low VM migration downtimes with LISP. Moreover, those experiments cannot be reproduced in absence of open source solutions.

The contribution of this paper is the definition and the evaluation of novel LISP functionalities to obtain high performance in large-scale live VM migration. We provide all the elements to reproduce the results, including reference to an open source implementation of our proposition. Our solution is based on the definition of LISP control-plane messages to fast update EID-locator mappings, hence overcoming the long latency of basic LISP mechanisms. We validate and evaluate our solution using the worldwide LISP Beta Network [1] and LISP-Lab [2] nodes, piloting the five LISP sites in four countries worldwide. The paper is organized as follows. Section II briefly presents the background. Section III describes our protocol extension proposition. Section IV reports experimental results. Section V concludes the paper and discusses future works.

II. Background

In this section we describe the state of the art live VM migration networking and we give an overview of LISP.

A. Live VM migration and IP mobility

Live VM migration is a feature introduced in recent hypervisors; it allows moving a running VM between two (physical) host containers without disconnecting the client or application. For most of the hypervisors, live migration is limited to situations in which source and destination hosts look like connected to the same local area network. The main reason is that the machine being migrated needs to keep the same routing view of the network (e.g., gateway, IP subnet) before and after the migration. Alternatively, in some legacy solutions, upon migration the VM changes its IP address, e.g., via the DHCP, to avoid the additional complexity needed to ensure that the origin IP address is not already used in the destination network, and to transfer the routing table; VM's IP readdressing implies, however, long convergence and loss of too many packets.

In order to perform Internet-wide migrations with IP continuity, authors in [9] and [12] propose an IP mobility solution. The logic is implemented in the hypervisor, interacting with the VM before and after its migration to update IP addresses in the VM routing table. While [9] succeeds in bringing lower service downtime compared to [12], the hypervisor has to alter the VM configuration to support the IP mobility feature, which leads to scalability concerns. Moreover, as the authors state, the performance of their solution is expected to worsen in large-scale global live migrations, because of the online signaling nature of the proposition and many-way signaling latencies.

Authors in [13] propose to adapt the Mobile IP (MIP) protocol [14] to pilot Internet-scale VM migrations, implementing it in the hypervisor. Their solution, called HyperMIP, is invoked whenever a VM is created, destroyed or migrated; as in MIP, it involves Home Agents (HA) to keep the connection alive. Whenever a VM changes a location, a tunnel is established between the HA and the source hypervisor to keep the client connected to the VM. The destination hypervisor then destroys the tunnel when the VM registers its new IP address to the HA. However, HyperMIP still introduces an important signaling overhead due to HA tunnel establishment.

Alternatively, to minimize signaling latencies, authors in [8] propose to use an external agent to orchestrate the migration from the beginning to the end, by proactively establishing circuits between the involved containers (source and destination hypervisors) offline, so as to rapidly switch the traffic upon migration, then redirecting the client-VM traffic via dynamic reconfiguration of IP tunnels. They achieve a near second network downtime while migrating machines across wide area networks, with a maximum network downtime around 3.8 seconds. Despite being a more secure approach, with respect to [9], [12] and [13] their solution involves lower-layer technologies, hence can be excessively costly.

B. Layer 2 over Layer 3 overlay tunneling solutions

The above described solutions tackle large-scale VM live migration using Layer 3 tunneling ([9], [12] and [13]), or Layer 3-Layer 1 interaction [8]. More recently, at the IETF, attention has been given to Layer 2 over Layer 3 (L2o3), Ethernet over IP, virtual network overlay solutions, so as to avoid IP reconfiguration to the VM, and service continuity upon migration of VMs across virtualization servers. Virtual eXtensible LAN (VXLAN) [15], Stateless Transport Tunneling (STT) [16], and Network Virtualization using Generic Routing Encapsulation (NVGRE) [17], are recent propositions, already implemented by many commercial stakeholders (e.g., Microsoft, VMWare) and open source virtual switches (e.g., OpenVSwitch), worth being discussed hereafter.

VXLAN [15] is a stateless L2o3 logic that extends the Layer 2 communication domain over IP networks, extending a VLAN broadcast domain thanks to MAC-to-IP encapsulation between hypervisors, even if communicating VMs and

[1]LISP Beta Network (website): http://www.lisp4.net
[2]LISP-Lab platform (website): http://www.lisp-lab.org

endpoints are in different IP segments. Basically, when a VM wants to communicate with another VM on a different host, a 'tunnel endpoint' implemented in the hypervisor receives the frame, verifies that the target VM is on the same VXLAN segment via standard signaling, and then appends an IP address corresponding to the destination tunnel endpoint, and a VXLAN header. Upon reception, the destination tunnel endpoint verifies the packet, decapsulates it and forwards it to the VM target. Therefore, thanks to the VLAN broadcast domain extension, a VM belonging to a VXLAN segment can migrate to a VXLAN endpoint in another IP segment, and its traffic is consequently encapsulated by the source VXLAN endpoint toward the destination VXLAN endpoint.

Functionally, NVGRE [17] is a similar L2o3 tunneling solution, with a different header (VXLAN uses a UDP shim header to easily pass through middle-boxes, while NVGRE does not hence limiting its scope to a single administrative network), and with no specified control-plane to distribute MAC-to-IP mappings (in VXLAN, multicast mechanisms do allow resolving these mappings). Encapsulating Ethernet traffic over IP allows a better bottleneck management thanks to various IP traffic engineering and load-balancing mechanisms. Both VXLAN and NVGRE, however, do not allow typical Ethernet network interface controllers to perform TCP offloading (intermediate fragmentation done by the hardware to boost performances). This is instead allowed by Stateless Transport Tunneling [16] (STT), another stateless L2o3 tunneling protocol, which uses a fake TCP header inside the IP header to allow interface-level TCP optimizations. From a VM mobility and traffic routing perspective, it offers the same encapsulation path than VXLAN and NVGRE, but as NVGRE it has difficulties to pass through Internet middle-boxes and its deployment is also limited to a single administrative domain such as a DC network.

All these technologies (VXLAN, NVGRE, STT) share as reference use-cases intra-DC and inter-DC communications, i.e., between VMs hosted in different virtualization servers potentially in different IP subnets. Therefore, they are not readily applicable to the Cloud access communication use-case, involving an IP user and a VM-based IP server, mainly targeted by our LISP proposition. The user endpoint is typically not a virtualization server and is not connected to the same DC fabric than the server, and can potentially be everywhere in the Internet.

C. Locator/Identifier Separation Protocol (LISP) overview

LISP implements an additional routing level on the top of legacy IP routing protocols, such as the Border Gateway Protocol (BGP), separating the IP location from the identification using Routing Locators (RLOCs) and Endpoint Identifiers (EIDs). An EID is an IP address that identifies a terminal, whereas an RLOC address is attributed to a border tunnelling router. LISP uses a map-and-encap scheme at the data-plane level, mapping the EID address to an RLOC and encapsulating the packet into another IP packet before forwarding through the IP transit. At the control-plane level, multiple RLOCs with different weights and priorities can be associated with an EID: for unipath communications, the least-priority RLOC corresponds to the one to be selected for encapsulation; when a subset or all of the RLOCs have the same priority value, load-balancing is performed on the equal-priority RLOC. RLOC priority and weight are assigned by the destination EID space owner using its LISP routers.

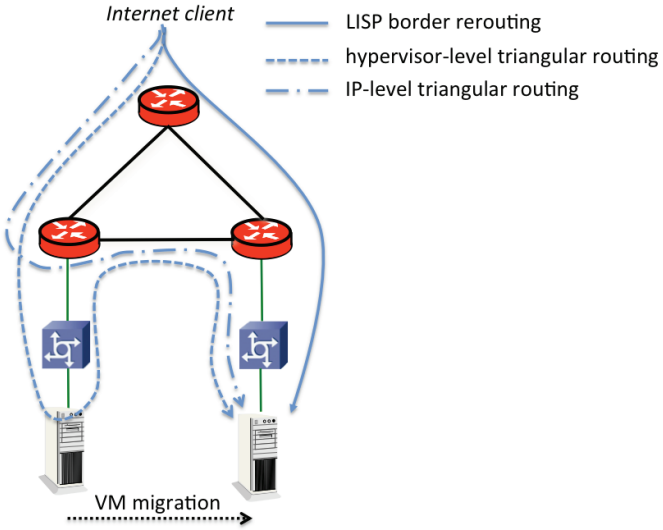


Fig. 2. Triangular routing vs LISP rerouting.

A LISP site is managed by at least one tunneling LISP router (xTR), which has a double functionality: IP packet encapsulation (packet received by a terminal; ingress functionality, or ITR) and packet decapsulation (packet received by the network; egress functionality, or ETR). For a better understanding, consider the example in Figure 1: the traffic from the host 1.1.1.1 to the host 2.2.2.2 is encapsulated by the ITR toward one of the RLOCs (the one with the best priority, i.e., RLOC3), which acts as ETR and decapsulates the packet before forwarding it to its final destination. On the way back to 1.1.1.1, RLOC4's xTR queries the mapping system and gets two RLOCs with equal priorities, hence, performing load-balance as suggested by the weight metric.

In order to guarantee EID reachability, LISP uses a mapping system that includes a Map Resolver (MR) and a Map Server (MS). As depicted in Figure 1, a Map Resolver accepts MAP-REQUESTs from xTRs and handles EID-to-RLOC lookups; a particular MAP-REQUEST message, called SOLICIT-MAP-REQUEST (SMR) can have a flag set (S bit) to solicit a MAP-REQUEST to self by the receiver (passing via the MR). A Map Server receives MAP-REGISTERs from ETRs and registers EID-to-RLOC in the mapping database [18]. The EID-to-RLOC mapping resolutions can be based on two protocols: LISP-ALT (Alternative Topology) [19] and DDT (Delegated Database Tree) [20], the first relying on BGP primitives, the second being inspired by DNS. Due to lack of flexibility, DDT is preferred over LISP+ALT in the LISP Beta Network. It is worth noting that, in a given configuration, if two xTRs, exchanging traffic, use the same MS/MR node, when an xTR sends a MAP-REQUEST for an EID that belongs to the other xTR, the MS/MR does not need to use DDT, hence no additional mapping latency is added to the xTR-MS/MR path latency.

D. Triangular routing solutions vs LISP rerouting

From the IP routing perspective of an Internet client accessing a server running in a VM, the legacy approaches can be considered as triangular routing (or indirect forwarding) solutions – the traffic has to reach the VM source network and/or container before being encapsulated and sent to the new VM location. Triangular routing solutions typically offer higher client-server path latency, than LISP-enabled direct rerouting. A higher path latency implies a higher transfer time, namely for TCP-based connections given that the round-trip-time (RTT) has an impact on TCP acknowledgments reception. Therefore, as far as the LISP tunneling node is implemented closer to the source endpoint than the triangular routing re-encapsulating node, a LISP-based Cloud network certainly outperforms triangular routing solutions in terms of transfer time.

As depicted in Figure 2, the rerouting logic of the above described solutions can be either implemented at the hypervisor level, or at the IP border level (e.g., DC or rack border) at a Mobile IP or similar agent. With LISP, client traffic can be redirected to the new location at the first LISP network ingress point, which can potentially be the client terminal itself (if a solution such as LISP mobile node is used [21]), a client's network provider router, any intermediate router between the client and the VM source DC, or (at last) the VM source DC's egress router if the standard IP path is taken by client traffic and the VM's prefix is announced by DC nodes. In all cases (but the latter that is topologically identical) the path latency offered by LISP is better than the path latency reachable with a non-LISP method alone. In common situations, triangular routing solutions alone add the source DC - destination DC latency to application connections, hence leading to longer forwarding latency, and transfer time, for Cloud access communications.

It is worth stressing that LISP is orthogonal to the existence of emerging hypervisor-level L2o3 triangular routing solution such as VXLAN, NVGRE or STT: LISP reroutes Cloud access user traffic while hypervisor-level mechanisms reroute VM-to-VM communications. It is worth noting that the LISP enhancement we propose in the following to support VM migration is independent of the existence of such inter-VM virtual network overlay mechanisms [3]. Their integration with our LISP-based Cloud access solution could bring advantages in terms of transfer time only for inter-VM communications.

As compared to legacy IP-level triangular routing solution, with our proposition described in the next sections, we can obtain service downtime between 150 and 200 ms, depending on the signaling scope. With respect to the alternative methods at the state of the art described in Section II-A, we can assess that:

- with HyperMIP [13], authors experienced 2 to 4 s of downtime, which is many times more than our approach;

[3]It is worth mentioning that such a possible coexistence seems to become true given that, for instance, VXLAN is based on an IP-UDP-VXLAN-Ethernet encapsulation where the VXLAN shim header has the same size and a similar format than the LISP shim header. Moreover, a LISP mapping interface is currently included in the specifications of the OpenDayLight SDN controller, hence making possible the usage of LISP control-plane features for virtual network overlay protocols such as VXLAN, NVGRE and STT (see: https://wiki.opendaylight.org/view/Project_Proposals:LispMappingService).

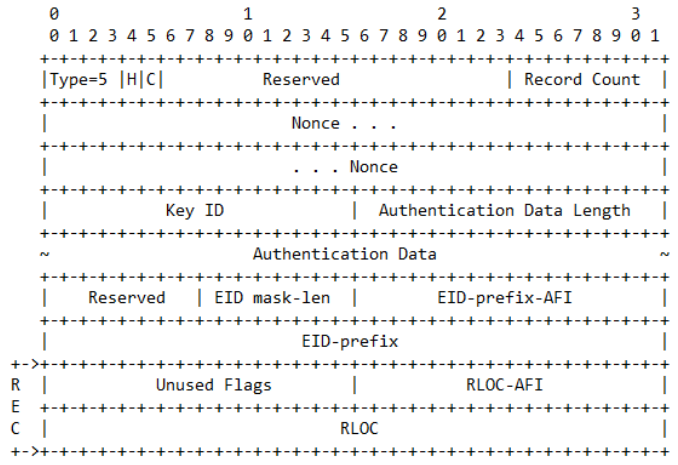


Fig. 3. CHANGE PRIORITY message format.

- similarly in [12] Mobile IPv6 signaling is used to detect VM location change, reaching a minimum overhead around 2500 ms, linearly increasing with the network delay, hence significantly higher than our approach;
- authors in [8] went a step further implementing pro-active circuit provisioning, reaching an application downtime varying between 800 and 1600 ms, which is more than 4 times higher than with our approach.

E. Existing LISP-based Mobility Management Solutions

In a LISP network, the VM can keep the same IP. Two mechanisms at the state of the art can perform this operation. One is a host-based LISP implementation called LISPmob [21]: the host implements a tiny xTR with basic LISP functions, using the network-assigned IP(s) as RLOC(s) and registering mapping updates for its EID with the mapping servers. Essentially conceived for mobile equipment, LISPmob could also be installed in the VM; there would be, however, a problem with most current hypervisors that impose the VM external address to be in the same subnet before and upon migration, which practically limits the LISPmob usability only to situations where source and destination networks are either LISP sites themselves, or layer-2 over wide area network (WAN) solutions. In the first case, a double encapsulation is needed, which could increase mapping latency, overhead and create MTU issues. There may also be scalability issues with a high number of VMs.

Another method to handle VM mobility via LISP is actually implemented in some Cisco products, only partially documented in [11]. The xTR automatically changes the mapping upon reception of outgoing data-plane traffic from an EID that has been registered as mobile node. The solution has an attracting light impact on IP operations, yet it seems to be weak against EID spoofing, and it seems not to have authentication mechanisms. Moreover, in order to guarantee fast mapping convergence, it seems that additional logic would need to be implemented in the VM or in the hypervisor to allow sending outgoing artificial data traffic even if no real outgoing traffic exists.

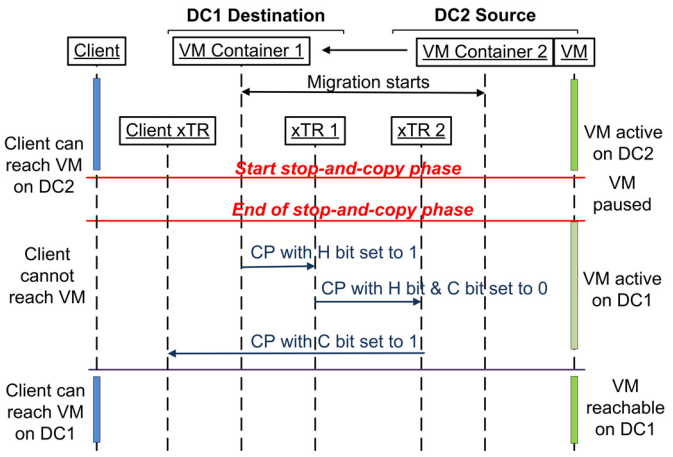


Fig. 4. Example of CP signaling exchange during a VM migration.

III. Proposed LISP-Based VM Migration Solution

We propose a novel solution to support WAN VM live migration exploiting the LISP protocol. We implemented our solution in the open source OpenLISP control-plane implementation [22] [23], which complements the OpenLISP data-plane [24].

As described above, a live migration technique should be able to move a VM keeping its unique EID, from its actual DC to a new DC maintaining all VM connections alive. As a preliminary step, the source and destination DCs have to share the same internal subnet, i.e., the VM unique EID should be routable beyond its RLOC, wherever it is. LISP supports a large number of locators, and does not set constraints on RLOC addressing – i.e., the RLOCs can take an IP address belonging not simply to different subnets, but also to different Autonomous System networks. The current VM location can be selected leveraging on RLOC metrics. We introduce two main enhancements:

- a new LISP control-plane message to speed up RLOC priority update;
- a migration process allowing hypervisor-xTR coordination for mapping system update.

Our solution involves the following network nodes: the source VM container and the destination VM container, both managed by an hypervisor, the VM being migrated from one to the other, LISP border routers at the source DC and at the destination DC, the Cloud user accessing the VM.

In the following, we present the novel LISP control-plane message we introduce for communications between the involved LISP nodes, then we describe the VM migration process, and finally we discuss implementation aspects.

A. Change Priority Message Format

We introduce a new type of LISP control-plane message we call Change Priority (CP). As depicted in Figure 3, we use a new control-plane type field value equal to 5[4], and use two bits to define message sub-types to be managed by both xTR and VM containers' hypervisors:

- **H (Hypervisor) bit**: this bit is set to 1 when the message is sent by the destination hypervisor (the hypervisor that receives the VM), indicating to the xTR that it has just received a new EID. With the H bit set, the record count should be set to 0 and the REC field is empty;
- **C (Update Cache) bit**: this bit is set to 1 when an xTR wants to update the mapping cache of another xTR. With the C bit set, the record count is set to the number of locators and the REC field contains the RLOC information to rapidly update the receiver mapping cache.

The other fields have the same format and function as for the Map-Register message fields [10], i.e., with EID and RLOC fields, a nonce field used to guarantee session controls, and HMAC authentication fields useful to secure the communication (with the important feature that the authentication key used for CP messages can be different than the key used by MAP-REGISTER, provided that the xTR is able to handle different keys as provided in [22] [23]).

[4]A preliminary format of such a new control-plane message has been presented at the first LISP Network Operator Group (LNOG) - http://www.lisp4.net/lnog.

Algorithm 1 CP processing

Ensure: authenticity of CP message
  extract EID from EID-prefix field
  if H bit is set to 1 **then**
    set own locators' priority to 1
    send CP to xTR group with H bit and C bit set to 0
    register mapping to Map Server
  end if
  if H bit and C bit are both set to 0 **then**
    set own locators' priority to 255
    set locators' priority in RLOC field to 1
    send CP with C bit set to 1 to all locators that have requested the VM's EID
    stop registering for EID
  end if
  if C bit is set to 1 **then**
    update mapping cache according to the received message
  end if

B. VM migration process

The LISP mapping system has to be updated whenever the VM changes its location. Before the migration process starts, the xTRs register the VM's EID as a single /32 prefix or as a part of larger EID (sub-)prefix. The involved devices communicate with each other to *atomically* update the priority attribute of the EID-to-RLOC mapping database entries. The following steps describe the LISP-based VM migration process we propose and demonstrate.

1) The migration is initialized by the hypervisor hosting the VM; once the migration process ends, the destination hypervisor (the container that receives the VM) sends a CP message to its xTR (also called *destination xTR*) with the H bit set to 1, and the VM's EID in the EID-prefix field.
2) Upon reception, the *destination xTR* authenticates the message, performs an EID-to-RLOC lookup and sets

the highest priority to its own locators in the mapping database with a MAP-REGISTER message. Then, it sends a CP message, with H and C bits set to 0, to update the mapping database of the xTR that was managing the EID before the migration (also called *source xTR*).

3) Before the VM changes its location, the *source xTR* keeps a trace file of all the RLOCs that have recently requested it (we call them *client xTRs*), i.e., that have the VM RLOCs in their mapping cache.
4) When the *source xTR* receives the CP message from the *destination xTR*, it authenticates it and updates the priorities for the matching EID-prefix entry in its database.
5) In order to redirect the client traffic, there are two different client-redirection possibilities, whether the *client xTR* is a standard router not supporting CP signaling (e.g., a Cisco router implementing the standard LISP control-plane [10]), or an advanced router including the CP logic (e.g., an OpenLISP router with the control-plane [22] [23]).
   - For the first case, the *source xTR* sends a SMR to standard *client xTRs*, which triggers mapping update as of [10] (MAP-REQUEST to the MR and/or to the RLOCs, followed by a MAP-REPLY to the xTR).
   - For the second case, in order to more rapidly redirect the traffic to the VM's new location (*destination xTR*), the *source xTR* sends a CP message with C bit set to 1 directly to all the *client xTRs*, which will therefore process it immediately (avoiding at least one client xTR-MR round-trip-time).
6) Upon EID mapping update, the *client xTRs* update their mapping cache and start redirecting the traffic to the VM's new routing locator(s).

All in all, updating the mapping database of the nodes involved in a VM migration requires two compulsory message exchanges (one message that notifies the destination DC about the migration process and another one that is used to notify the source DC), and optionally a number of additional messages equal to the number of clients that are communicating with the VM to inform the xTR clients' about the updates. Considering only the location update messages for the LISP mapping system does not make our solution heavier than triangular routing solution (e.g., Mobile IP). Additional signaling messages are generated with respect to triangular routing solutions if VM clients's mapping updates are considered. The limited increase of control messages is indeed counterbalanced by more significant benefits, in terms of resiliency and convergence, of our solution with respect to triangular routing ones.

It is worth noting that our solution fully relies on control-plane features. It is our methodology choice to avoid mixing data-plane and control-plane functions (for example proposing a specific usage of Map-Versioning or Locator Status Bit field in the data-plane [25]). The main advantage of creating network control functions disjoint from the data-plane is the possibility to program the control-plane independently of the forwarding logic, hence to implement advanced and personalized functionalities. This separation respects the current design trend in networking called Software Defined Networking [26]. Thanks to that, new functionalities can be added rapidly to the OpenLISP control-plane and allow rapid deployment even using pre-existing basic data-plane elements.

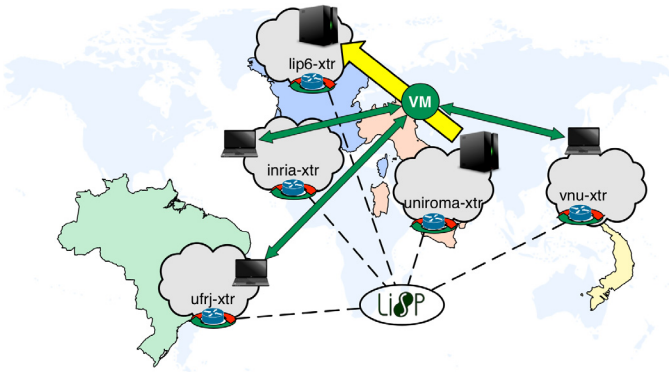


Fig. 5. LISP testbed topology.

C. Implementation aspects

The proposed solution has been implemented using open-source software (i.e., OpenLISP [22] [23]), and its implementation involves both the hypervisor and the xTR sides.

1) On the hypervisor: we integrated a new function that interacts with LIBVIRT (a management kit handling multiple VMs in the KVM hypervisor) [27] to trigger CP message generation. When a live migration starts, the hypervisor creates a "paused" instance of the VM on the destination host. Meanwhile, LIBVIRT monitors the migration phase from the start to the end. If the migration is successfully completed, LIBVIRT checks if the VM is running on the target host and, if yes, it sends a CP message to its xTR on the UDP LISP control port 4342. The VM EID is included in the EID-prefix field.

2) On the xTR: we implemented the Algorithm 1 function in the OpenLISP control-plane [22]. While the OpenLISP data-plane logic runs in the kernel of a FreeBSD machine, the control-plane runs in the user space. The control-plane has a new feature to capture control-plane message type 5 and the logic to handle CP signaling[5].

3) A signaling example: upon receiving a client request, or as triggered by a consolidation engine, a VM needs to be migrated to another public DC. As in the Figure 4 example, VM Container 2 starts migrating VM from DC2 to DC1 while the Client is still connected. When the migration reaches the so called stop-and-copy phase (i.e., the phase dedicated to transfer the "dirty pages", which are pages updated too frequently to be transferred while the VM runs [28]), the VM stops and begins transferring its last memory pages. Meanwhile, the Client loses the connection, but keeps directing the traffic to DC2.

The hypervisor on VM Container 1 detects that VM is now successfully running, indicating the end of the migration. Then the VM Container 1 announces that the VM has changed its location by sending to xTR 1 a CP message with the H bit

[5]It is worth noting that type 5 is currently not allocated to any standard message in LISP standardization documents.

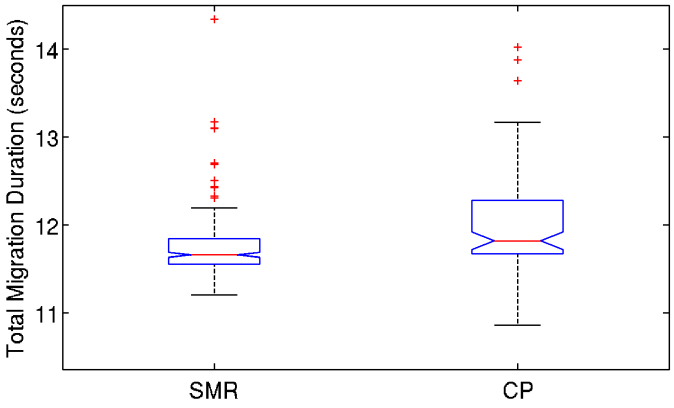


Fig. 6. Total migration duration (boxplot statistics).

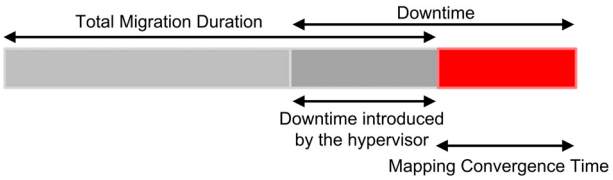


Fig. 7. Migration duration and downtime composition.

set. Upon reception, xTR 1 sends a CP with H bit and C bit set to 0 to notify xTR 2 about the new location of VM: xTR 1 updates the priorities for VM's EID entry in its database.

When xTR 2 receives the CP message, it matches the EID-prefix to the entries within its mapping database, and modifies the priorities accordingly, then it stops registering VM's EID. As mentioned in Section III-B, xTR 2 keeps a trace file of all the locators that recently requested the VM's EID. In this example, only one client is communicating with VM, so xTR 2 sends a CP message with C-bit set to the Client's xTR.

Finally, the Client's xTR receives the CP message, maps VM's EID, and updates its cache, then starts redirecting Client's traffic to VM's new location (DC1).

IV. Testbed Evaluation

We performed live migrations of a FreeBSD 9.0 VM, with one core and 512 MB RAM (corresponding to a typical service VM like a lightweight web server), from UROMA1 (Rome) to LIP6 (Paris), using KVM [29] as hypervisor. Please note that the size of the VM has no direct impact on live migration downtime for a given service type; different services may have a more or less intensive usage of memory pages, so that the stop-and-copy phase duration may have a more or less important impact on the downtime.

Figure 5 gives a representation of the testbed topology. As distributed storage solution, we deployed a Network File System shared storage between source and destination host containers. Hence, only RAM and CPU states are to be transferred during the live migration. The VM containers are Ubuntu 12.04 servers, dual core, with 2048 RAM and using KVM and Libvirt 0.9.8.

We measured node reachability by 20 ms spaced pings from different clients: distant ones at VNU (Hanoi, Vietnam), UFRJ (Rio de Janeiro, Brazil) LISP sites, and a close one at the

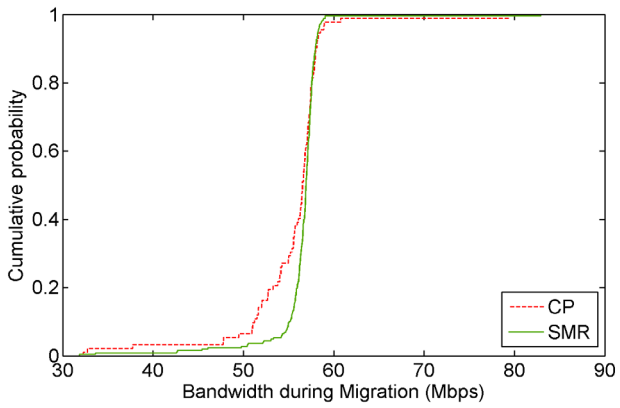


Fig. 8. Bandwidth during migration with SMR and CP approaches.

INRIA (Sophia Antipolis, France) LISP site. It is important to mention that:

- the clocks on all LISP sites were synchronized to the same Network Time Protocol (NTP) stratum [30], so that a same VM migration can be monitored concurrently at the different client sites;
- all LISP sites' xTRs register to a same MS/MR located in Denmark (www.lisp4.net), hence avoiding the DDT latency in the mapping convergence (as already mentioned in Section II-C).

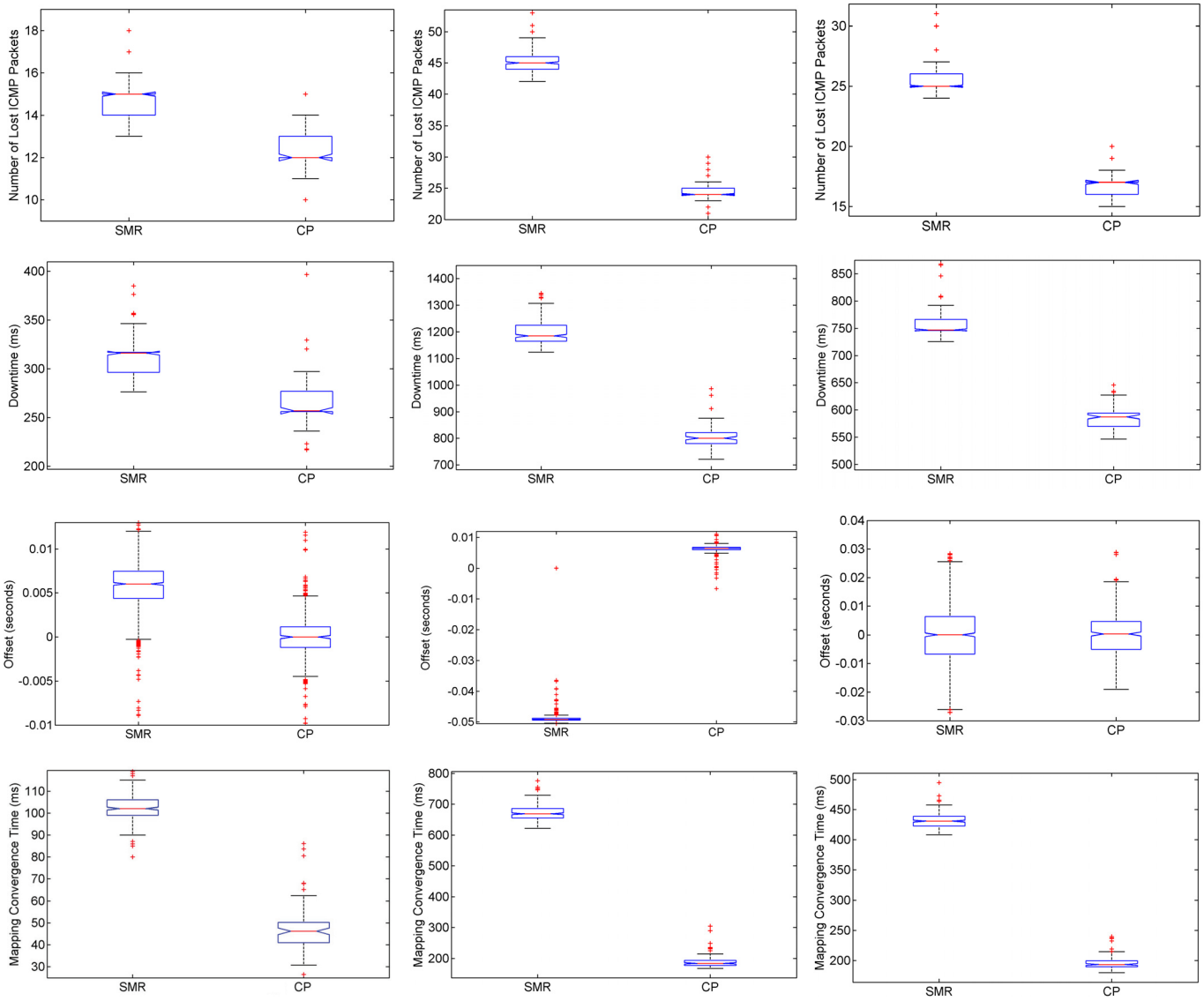
The latter is a possibility left to the datacenter manager, depending on the quality of the DDT architecture the Cloud provider could choose to which MS/MR to connect both the client and the Cloud networks. In our experimentations, we chose so also to get around some of the instabilities on the Asian (APNIC) MS/MR.

We performed hundreds of migrations from the UROMA1 site to the LIP6 site, over a period of three months at different times of the day, with two migrations per hour, to obtain a statistical population representative enough to capture the bandwidth, latency and routing variations of the Internet paths. We measured the experienced bandwidth; we report its experimental distribution in Figure 8, where we can see that most of the migrations experienced between 50 and 60 Mbps.

We used the two possible inter-xTR mapping update modes with the proposed control-plane enhancement: SMRs to simulate standard client xTRs, and CP to encompass the case with enhanced xTRs at client LISP sites. Given the Internet wideness of the testbed, both the bottleneck bandwidth and RTTs were floating, depending by the time and day, hence we did a statistical evaluation as described hereafter. The average measured RTTs between each site during the migration are reported in Table I; having both close and far clients' sites allows us to precisely assess the migration performance.

In order to experimentally assess the relationship between different time components and network situations, we measured the following different parameters:

- number of lost packets for each client (i.e., the number of ICMP messages that are lost on each client during migration);
- mapping convergence time for each client: the time between the transmission of CP by the hypervisor and the mapping cache update on each client.



(j) INRIA (k) VNU (l) UFRJ

Fig. 9. Boxplot statistics of experimentation parameters (lost packets, downtime, offset, mapping convergence) for the three LISP sites (INRIA, VNU, UFRJ).

TABLE I
AVERAGE MEASURED RTT DURING MIGRATIONS

LISP Sites	Average RTT
LIP6-UROMA1	30.47 ms
LIP6-VNU	299.86 ms
LIP6-INRIA	16.47 ms
LIP6-UFRJ	246.07 ms
UROMA1-VNU	321.27 ms
UROMA1-INRIA	27.27 ms
UROMA1-UFRJ	259.13 ms

- downtime perceived by each client: the time during which the client could not communicate with the VM;
- total migration duration;
- inter-container bandwidth during migration;
- offset for each client: the difference between the clocks on each client and the NTP server;
- RTT between the VM and the clients.

For the sake of completeness, we report in Figure 6 statistics about the total migration duration. It has a median around 11.75 s for both signaling modes. It includes the downtime introduced by the hypervisor (stop-and-copy phase), not including the mapping convergence downtime component. As depicted in Figure 7 and as of previous arguments, it is worth underlining that one should expect that the overall downtime is greater or equal than the downtime introduced by the hypervisor (the stop-and-copy phase duration to transfer the dirty pages as already mentioned[6]) plus the mapping convergence time. Therefore, the mapping convergence time reflects our protocol overhead, which is differently affected by the RTT between LISP sites (Table I) depending on the client xTR support of CP signaling.

In order to characterize absolute service downtimes suf-

[6]With standard tools, we cannot really control the stop-and-copy duration since it depends on the volume of the last memory pages to be transferred that, as already mentioned, depends on the running application, and in some cases also on the number of connected users.

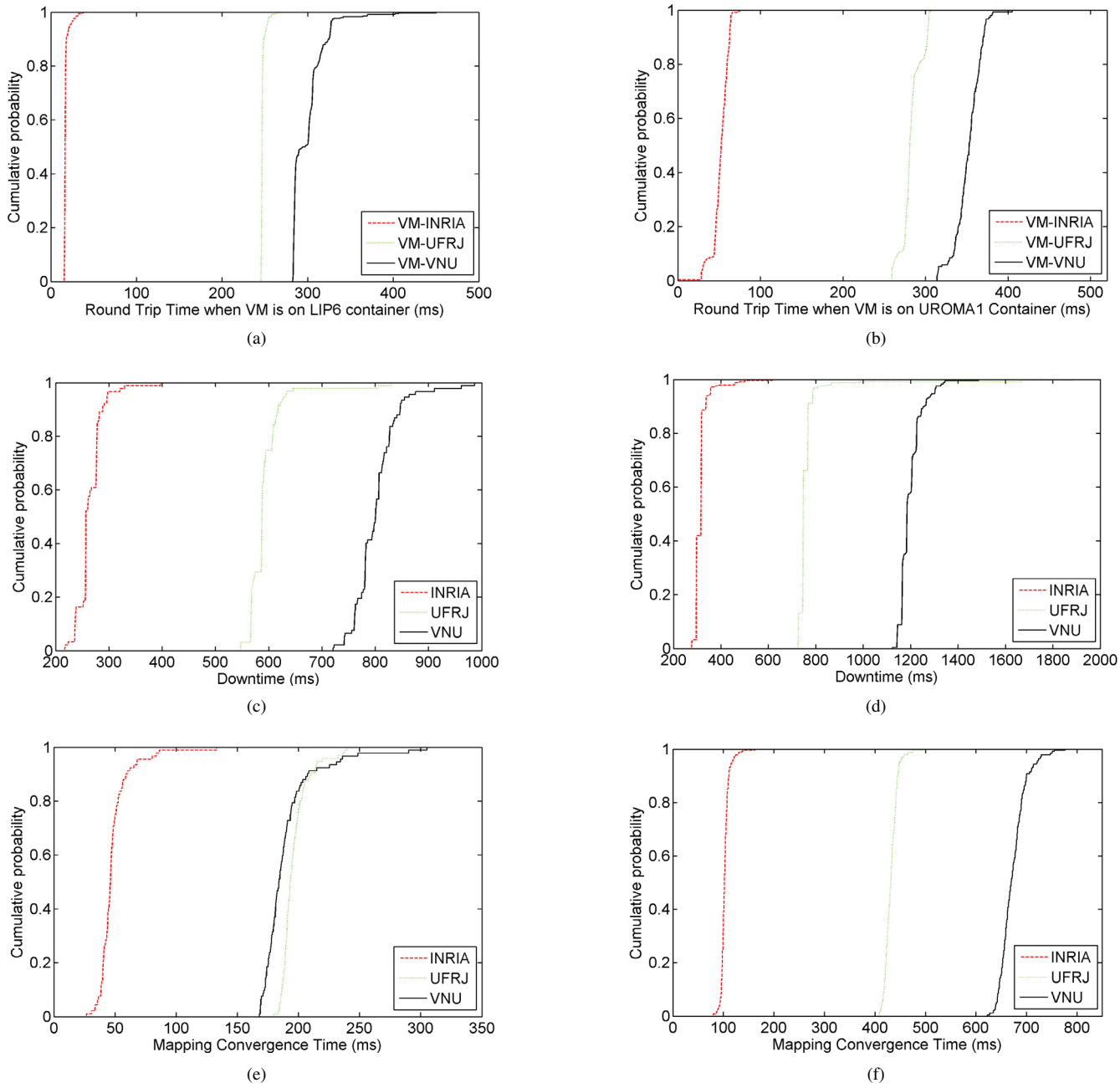


Fig. 10. Cumulative Distribution Functions of different migration parameters.

fered by clients, Figure 9 reports the boxplots (minimum, 1^{st} quartile, median with the 95% confidence interval, 3^{rd} quartile, maximum, outliers) of the obtained number of lost packets, offset, downtime, and mapping convergence time. We measured the results with the two possible modes for inter-xTR mapping update, using SMR signaling and using CP signaling.

Using SMR *signaling:* as explained in Section III-B, as of LISP standard control-plane, the SMR message is sent by an xTR to another to solicit mapping update for a given EID. Upon reception of a SMR, the target xTR sends a MAP-REQUEST to mapping system, followed by a MAP-REPLY. The overall SMR signaling time should therefore be lower bounded by one and a half the RTT between the two xTRs, which impacts the mapping convergence time and hence the service downtime. As of our experimentations, we obtained a median downtime of about 320 ms for the INRIA client, 1.2s for VNU (Figure 9). This large gap between the downtimes of close and distant clients can be explained not only by the distance that separates each client from the VM, impacting the propagation delay (see Table I), but also by the fact that the Map Resolver is closer to INRIA than to VNU and UFRJ, as mentioned in Section IV. We find this gap also in the number of lost ICMP packets, two to three times higher for distant clients than for close ones (Figure 9).

Using CP *signaling:* as explained in Section III-B, using CP signaling the mapping convergence time can be decreased of at least one RTT between xTRs, with an authenticated one-

way message that directly updates xTR cache upon reception. For the INRIA client, we obtain a median downtime of 260 ms gaining a few dozens of ms, whereas we could gain 200 ms for VNU and 400 ms for UFRJ. Moreover, we notice that the number of lost ICMP packets for distant clients has exponentially decreased. This important decrease is due to the fact that xTRs have no longer to pass via the Map Resolver to update their mapping cache. Finally, Figures 9(j),9(k), and 9(l) show that the mapping convergence time component of the downtime decreases with CP signaling for all cases. While it is roughly between one-third and one-half the downtime with SMR signaling, it falls to between one-sixth and one-third with CP signaling, and this ratio is higher for distant clients. This implies that the hypervisor downtime (stop-and-copy phase) is less sensible to the RTT than the mapping convergence is (likely, the last page transfer profits from an already established TCP connection with an already performed three-way handshake).

It is worth noticing that these measurements may suffer from a small error due to clock synchronization. As mentioned above, the xTRs have synchronized clocks over the same NTP stratum. The median of the offsets is represented in Figure 9. While it is negligible for close clients, it is of a few dozens of ms for distant clients, however less than the 5% of the downtime.

A more precise insight on the simulation parameters is given by the cumulative distribution functions (CDFs) in Figure 10. The RTT CDFs show us that, from an Internet path perspective, VNU appears as more distant than UFRJ from the VMs, with a similar RTT gap before and after the migration. With respect to the downtime, the relative gap between VNU and UFRJ clients with CP and SMR is similar. In terms of mapping convergence time, the VNU-UFRJ gap changes significantly, the reason is likely due to the RTT amplification with SMR.

V. Conclusion

In this paper, we propose a novel LISP-based solution for VM live migrations across geographically separated datacenters over wide area IP networks. We tested it via the global LISP testbed. We can summarize our major contributions as follows:

- we defined and implemented a new type of LISP control-plane message to update VM location upon migration, with the interaction between hypervisors and LISP routers[7];
- we performed extensive (hundreds) Internet-wide migrations between LISP sites (LIP6 in Paris, UROMA1 in Rome) via the LISP testbed, including the case of clients close to source and destination containers (INRIA - Sophia Antipolis), and the case of distant clients (VNU - Hanoi, UFRJ - Rio de Janeiro);
- by exhaustive statistical analysis on measured relevant parameters and analytical discussions, we characterized the relationship between the service downtime, the mapping convergence time and the RTT;
- we showed that with our approach we can easily reach sub-second downtimes upon Internet-wide migration, even for very distant clients.

As a future work, we are interested in tackling other open issues generally related to VM mobility management with LISP. From a scalability perspective, the amount of VM EIDs that might be registered, with thousands of non-aggregated VM EIDs, may be huge and may generate excessive signaling. To circumvent this issue, we are interested in defining new LISP control-plane functionalities allowing modular mapping registrations. From a VM migration performance perspective, we are interested in investigating how the LISP-enabled path diversity can be offered to Cloud servers to augment the user's quality of experience; preliminary results are presented in [31].

[7]The LISP control-plane code with related functionalities is publicly available, see [22] [23]. Part of the CP signaling logic was implemented into LIBVIRT.

Acknowledgment

The authors would like to thank Damien Saucez from INRIA Sophia Antipolis, Giulio Colombo from the University of Roma I - La Sapienza, and Rodrigo de Souza Couto from Universidade Federal do Rio de Janeiro (UFRJ) for running LISP site clients probing.

This work was partially supported by the ANR LISP-Lab project (http://www.lisp-lab.org - Grant No: ANR-13-INFR-0009), the "Investissement d'Avenir" NU@GE project (http://www.nuage-france.fr), the FUI 15 project RAVIR (http://www.ravir.io) and the EIT ICT-Labs Future Networking Services action line (http://www.eitictlabs.eu).

References

[1] P. Raad *et al.*, "Achieving sub-second downtimes in Internet virtual machine live migrations with LISP," in *Proc. 2013 IEEE/IFIP Int. Symposium on Integrated Network Management.*

[2] S. Bhardwaj, L. Jain, and S. Jain, "Cloud computing: a study of infrastructure as a service (IAAS)," *Int. J. Engineering and Inf. Technol.*, vol. 2, no. 1, pp. 60–63, 2010.

[3] Q. Duan, Y. Yan, and A. V. Vasilakos, "A survey on service-oriented network virtualization toward convergence of networking and cloud computing," *IEEE Trans. Network and Service Management*, vol. 9, no. 4, pp. 373–392, 2012.

[4] M. Nelson *et al.*, "Fast transparent migration for virtual machines," in *Proc. 2005 USENIX Annual Technical Conference*, pp. 25–25.

[5] S. Setty, "vMotion architecture, performance, and best practices in VMware vSphere 5," VMware, Inc., Tech. Rep., 2011.

[6] Cisco, "Cisco overlay transport virtualization technology introduction and deployment considerations," Cisco Systems, Inc., Tech. Rep., Jan. 2012.

[7] L. Dunbar *et al.*, "TRILL edge directory assistance framework," RFC 7067, Nov. 2013.

[8] F. Travostino *et al.*, "Seamless live migration of virtual machines over the MAN/WAN," *Future Generation Computer Systems*, vol. 22, no. 8, pp. 901–907, Oct. 2006.

[9] H. Watanabe *et al.*, "A performance improvement method for the global live migration of virtual machine with IP mobility," in *Proc. 2010 ICMU.*

[10] D. Lewis *et al.*, "Locator/ID Separation Protocol (LISP)," RFC 6830, Jan. 2013.

[11] Cisco, "Locator ID Separation Protocol (LISP) VM mobility solution," Cisco Systems, Inc., Tech. Rep., 2011.

[12] E. Harney *et al.*, "The efficacy of live virtual machine migrations over the internet," in *Proc. 2007 Int. Workshop on Virtualization Technology in Distributed Computing*, p. 8.

[13] Q. Li *et al.*, "Hypermip: hypervisor controlled mobile IP for virtual machine live migration across networks," in *Proc. 2008 IEEE High Assurance Systems Engineering Symposium*, pp. 80–88.

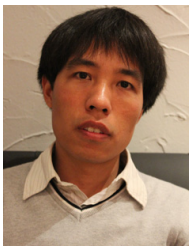
[14] C. Perkins, "IP mobility support for IPv4," IETF RFC 3344, 2002.

[15] T. Sridhar, L. Kreeger, D. Dutt, C. Wright, M. Bursell, M. Mahalingam, P. Agarwal, and K. Duda, "VxLAN: a framework for overlaying virtualized layer 2 networks over layer 3 networks," draft-mahalingam-dutt-dcops-vxlan-04, May 2013.
[16] E. B. Davie and J. Gross, "Stateless transport tunneling protocol for network virtualization (STT)," draft-davie-stt-04, Sept. 2013.
[17] M. S. et al., "NVGRE: Network Virtualization using Generic Routing Encapsulation," draft-sridharan-virtualization-nvgre-03, Aug. 2013.
[18] V. Fuller and D. Farinacci, "LISP map server interface," RFC 6833, Mar. 2012.
[19] D. Lewis *et al.*, "LISP Alternative Topology (LISP+ALT)," RFC 6836.
[20] D. Lewis and V. Fuller, "LISP Delegated Database Tree," draft-fuller-lisp-ddt-04, Sept. 2012.
[21] C. White *et al.*, "LISP Mobile Node," draft-meyer-lisp-mn-09, July 2013.
[22] D. C. Phung, S. Secci, D. Saucez, and L. Iannone, "The OpenLISP Control-Plane Architecture," *IEEE Network Mag.*, vol. 28, no. 2, Mar. 2014.
[23] Website, "OpenLISP control plane." Available: http://github/lip6-lisp/control-plane
[24] L. Iannone *et al.*, "OpenLISP: an open source implementation of the Locator/ID Separation Protocol," *2009 ACM SIGCOMM*, demo paper.
[25] L. Iannone, D. Saucez, and O. Bonaventure, "Locator/ID Separation Protocol (LISP) Map-Versioning," RFC 6834, Jan. 2013.
[26] Software defined networking: the new norm for networks, white paper, ONF, Apr. 2012.
[27] "libvirt: the virtualization API." Available: http://libvirt.org/
[28] C. Clark *et al.*, "Live migration of virtual machines," in *Proc. 2005 Conference on Networked Systems Design & Implementation—Vol. 2*, pp. 273–286.
[29] A. Kivity *et al.*, "KVM: the Linux virtual machine monitor," in *Proc. 2007 Linux Symposium*, vol. 1, pp. 225–230.
[30] D. Mills, "Internet time synchronization: the network time protocol," *IEEE Trans. Commun.*, vol. 39, no. 10, pp. 1482–1493, 1991.
[31] M. Coudron, S. Secci, G. Pujolle, P. Raad, and P. Gallard, "Cross-layer cooperation to boost multipath TCP performance in cloud networks," in *Proc. 2013 IEEE Int. Conference in Cloud Networking.*

Patrick Raad obtained a computer science degree from the Lebanese University in 2011, and the M.Sc. degree in networking from UPMC, France, in 2012. Since Nov. 2012, he enrolled in an industrial Ph.D. program with UPMC and Non Stop Systems (NSS). His current interests include Internet routing and Cloud Networking.

Stefano Secci is an Associate Professor at the University Pierre and Marie Curie (UPMC - Paris VI, Sorbonne Universites), since 2010. He received a "Laurea" degree in Telecommunications Engineering from Politecnico di Milano, in 2005, and a dual Ph.D. degree in computer networks from the same school and Telecom ParisTech, in 2009. He also worked as a Research Fellow at NTNU, George Mason University, Ecole Polytechnique de Montreal, and Politecnico di Milano, and as a Network Engineer with Fastweb Italia. His works mostly cover network optimization, protocol design, Internet routing and traffic engineering. His current research interests are about Internet resiliency and Cloud networking.

Dr. Secci has been member of many Technical Program Committees of many leading conferences (NOMS, CLOUDNET, ICC, GLOBECOM, WCNC, VTC, Networking), TPC co-chair of IEEE CLOUDNET 2012, chair of the Cloud Networks track at IEEE GLOBECOM 2014, and referee for the Italian Ministry of Research and University. He is an associate editor for *IEEE Communications Surveys and Tutorials* since 2012, for *Journal of Network and Systems Management* (Springer) since 2013, and guest editor for the 2013 *Computer Networks* (Elsevier) special issue "Communications and Networking in the Cloud." He is Vice-Chair of the Internet Technical Committee (ITC), joint between the IEEE Communication Society and the Internet Society (ISOC), since 2013.



Dung Chi Phung received a M.Sc. degree from the Vietnam National University (VNU) at Hanoi, Vietnam, where he worked as a campus network engineer. He is actually on leave of absence from VNU and works as a research engineer at Sorbonne Universites, UPMC Univ Paris 06, UMR 7606, LIP6, France.

Antonio Cianfrani received his "Laurea" degree in Telecommunications Engineering in 2004 and the Ph.D. in Information and Communication Engineering in 2008, both from the University of Rome "La Sapienza." He is an Assistant Professor at the DIET Department of the University of Rome "La Sapienza." Dr. Cianfrani was involved in many European and Italian research projects; since March 2012 he is the University of Rome coordinator of the GreenNet (Greening the Network) project funded by the Italian Ministry of Research and Education under the FIRB (Futuro in Ricerca) program. His field of interests includes routing algorithms, network protocols, performance evaluation of Software Routers and optical networks. His current research interests are focused on green networks and future Internet architecture.

Pascal Gallard obtained a computer science degree from Rennes University in 2001, and a Ph.D. degree in computer from INRIA and Rennes University in 2004. He was the cofounder of Kerlabs, a company on system virtualization, where he worked from 2006 to 2010. Since 2011 he is a research and development director at Non Stop Systems (NSS), an SME on Cloud computing and virtualization.

Guy Pujolle received the Ph.D. and "Thèse d'Etat" degrees in Computer Science from the University of Paris IX and Paris XI, on 1975 and 1978, respectively. He is currently a Professor at the UPMC, and a member of the Institut Universitaire de France. He spent the period 1994-2000 as Professor and Head of the computer science department of Versailles University. He was also Professor and Head of the MASI Laboratory (Pierre et Marie Curie University), 1981-1993, Professor at ENST (Ecole Nationale Supérieure des Télécommunications), 1979-1981, and member of the scientific staff of INRIA, 1974-1979. He is currently an editor for the International Journal of Network Management, WINET, and the editor-in-chief of *Annals of Telecommunications*. He was in charge of a large number of European and French projects.