Cloudlet Network Design Optimization

Alberto Ceselli*, Marco Premoli*, Stefano Secci†
* Dipartimento di Informatica, Università degli Studi di Milano, Italy. Email: first-name.last-name@unimi.it.
† Sorbonne Universités, UPMC Univ Paris 06, UMR 7606, LIP6, F-75005, Paris, France. E-mail: first-name.last-name@upmc.fr.

***Abstract*—Major interest is currently given to the integration of clusters of virtualization servers, also referred to as 'cloudlets', into the access network to allow higher performance and reliability in the access to mobile cloud services. We tackle the cloudlet network design problem for mobile access networks. The model is such that virtual machines are associated with mobile users and are allocated to cloudlets. Designing a cloudlet network implies first determining where to install cloudlet facilities among the available sites, then assigning sets of access points such as base-stations to cloudlets, while supporting virtual machine migrations and taking into account partial user mobility information, as well as the satisfaction of service-level agreements. We present link-path formulations supported by heuristics to compute solutions in reasonable time. We qualify the advantage in considering mobility for both users and virtual machines as up to 40% less cloudlet facilities to install and 40% less virtual machine migrations to execute. We compare two migration modes, bulk and live migration, as a function of mobile cloud service requirements, determining that a high preference should be given to bulk migrations for delay-stringent services such as augmented reality support, while for applications with less stringent delay requirements, live migration appears as largely preferable.**

I. Background

The cloudlet concept was firstly introduced in [1] where is defined as a trusted, resource-rich computer or cluster of computers well-connected to the Internet and available for use by nearby mobile devices. It represents a container for virtual machines (VMs): each connecting user is associated with a VM, which is created by the cloudlet with a process of dynamic synthesis at his first access, with the aim of supporting various low-latency application offloading use-cases such as remote desktop, device offloading, code-partitioning, image and voice recognition for augmented reality.

The cloudlet concept is complemented by the concept of Mobile Cloud Computing 3-tier hierarchical network as presented in [2] and [3]. In the hierarchy the cloudlet is the primal resource for the augmentation of the mobile device capabilities, while a cloud is used as last available resource, or for delay-tolerant resource-intensive applications.

The benefits of cloudlets' usage on users' quality of experience are presented in [4]–[6] where authors compare computing performances of different types of applications on different layers of the 3-tier hierarchy. In [4] authors show that application placement can significantly impact performance and user experience, and in particular that moving applications closer to the users improves their experience. In [5] authors question, by quantitative experimental results, benefits from consolidating computing resources in large data centers when strict latency constraints are required by the applications.

ISBN 978-3-901882-68-5

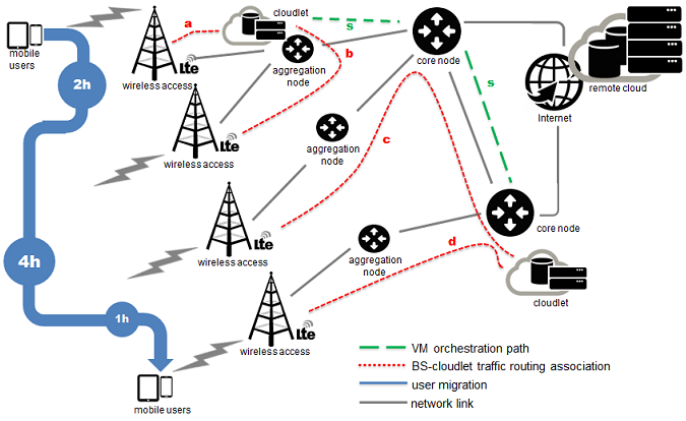


Fig. 1: Example of a simplified cloudlet network.

Considering multi-hop WiFi networks, in [6] authors show that when no more than two wireless hops are used to transfer data the cloudlet-based approach always outperforms the cloud-based one, and that when the maximum number of hops is up to 4 the cloudlet-based approach is still the best one for most of the instances. There is no binding dependence on the nature of the wireless link: even if the seminal idea was to use cloudlet using WiFi, the virtualization architecture is independent of the wireless link.

The hardware technologies for the implementation of cloudlets already exist, thanks to fabrics called "micro data centers" or "modular data center" [7]–[9]. Telecommunication vendors and providers show an increasing interest in such deployments, also referred to as 'mobile edge computing' solutions in industrial fora and standardization bodies (e.g., [28]).

In this paper, we focus on the potential planning of a cloudlet network in mobile access networks, which is, to the best of our knowledge, an untreated problem in the literature. Mobile access networks could be any form of wireless access network disposing of a backhauling wireline infrastructure where cloudlets can be interconnected. In this context, a higher importance is placed on the orchestration of virtual resources: where to place the cloudlets and to which cloudlet assign users is a challenging problem, also because VM orchestration across cloudlet facilities can occur as a function of variations of the load and service level due to user mobility. We investigate three cloudlet network design cases: with no specific support for VM orchestration, with support for VM bulk migrations [13] and with support for VM live migrations [14]. While bulk migrations consist in migrating the whole VM stack including RAM and disk and implies stopping the VM for a long period to transfer it, live migrations can stop

the VM only for a small amount of time required to transfer the most recently used memory, not requiring an entire one-shot disk transfer, but a permanent disk storage synchronization among source and destination locations.

Our contribution is as follows:

- We provide a link-path mixed integer linear programming formulation including a polynomial number of variables to represent location decisions, and an exponential number of them to encode routing ones, solvable by general purpose solvers and mathematical programming heuristics in reasonable time.
- We bring novel and original insights on the planning of cloudlets for mobile access networks. By performing extensive simulations on real 4G cellular network datasets from the Ile-de-France Orange network, we show the trade-off that can be achieved by means of the three approaches and the impact of user mobility on the cloudlet network: as few as 9 to 14 cloudlets can be planned for 180 thousands of users while requiring tight delay guarantees, and as much as 16 cloudlets can instead be planned for delay-stringent applications. We show that there is a sensible gain in the number of cloudlets to install, up to 40%, by including user and virtual machine mobility in the network planning. We do also qualify the efficiency of bulk and live virtual machine migrations, in terms of number of migrations and migrated volume, for three reference mobile cloud services differing in the level of required latency and memory characteristics: augmented-reality, remote desktop and storage box services.
- We report empirical distributions of the dataset features in order to allow the reproducibility of our results.

The paper is organized as follows. Sect. II presents the cloudlet network model and mathematical formulations for the different cases. Sect. III presents the dataset. Sect. IV reports experimental results and Sect. V concludes the paper.

II. Cloudlet Network Model

Accordingly to European Telecommunications Standardization Institute (ETSI)'s Mobile Edge Computing Industry Specification Group [28], the distribution of computing resources into mobile access network should be carefully designed to take into account infrastructure properties. Following the guidelines of white-papers [16]–[19], a broadband access and back-hauling network, such as a cellular network, can be modeled as a two-level hierarchical network: access points on the field are connected to aggregation nodes, which are then connected to core nodes, as depicted in Fig. 1 (for simplicity, we refer in the following to access points as Base Stations, BSs). Cloudlets can reasonably be placed at either field, aggregation or core level, with connections between a BS and its cloudlet potentially crossing twice each level (along uphill and downhill paths).

Various physical interconnection network topologies between BSs, aggregation nodes and core nodes are commonly adopted: tree, ring or mesh topologies, as well as intermediate hybrid topologies. Moreover, with the emergence of 4G, there is a trend to further mesh back-hauling nodes. A variety of network protocol architectures are typically adopted, from circuit-switched optical networks to Ethernet carrier grade and label-switched packet networks. The common denominator of such carrier-grade architectures is the ability to create a virtual topology of links directly interconnecting pairs of nodes at a same level with a guaranteed tunnel capacity. Nowadays, with the convergence towards (Ethernet/MPLS) packet-switching carrier-grade solutions at the expense of legacy (SDH/SONET) circuit-switched approaches, bit-rates for pseudo-Ethernet virtual links is set to giga-Ethernet granularities (i.e., at present typically 1 or 10 Gbps).

In this framework, we believe it is appropriate to model the network as a superposition of stars of virtual links for the interconnection of aggregation nodes to BSs and for the interconnection of core nodes to aggregation nodes, even if nodes can have no physical direct connection. Under the same virtual link provisioning trend, core nodes can be considered as interconnected to each other by a full mesh topology of virtual links, as depicted in Fig. 1. It is worth noting that, as far as we know, partitioning of traffic from one BS to multiple aggregation nodes, and from one aggregation node to multiple core nodes is not the dominating current practice; still, such features would not change significantly the nature of our formulations.

In the following, we give a formal definition of the cloudlet design dimensioning problem, and we propose three variants:

- *Static planning*: neither user mobility nor virtual machine mobility are taken into account when planning cloudlet placement and associations of BSs to cloudlets in a given planning time horizon.
- *Planning with bulk VM migrations*: user mobility is taken into account considering different time-frames in the planning horizon. VM mobility is included in terms of bulk VM transfers across those cloudlet facilities to absorb temporal user load variations.
- *Planning with live VM migrations*: differing from the previous case, live VM migrations are considered, requiring storage synchronization patterns between cloudlet locations.

The last two cases permit to model adaptive VM migrations performed as a function of user mobility and network state variations, that are technically possible via mobile cloud protocol architectures such as the one described in [15].

A. Problem statement

We assume that a set of suitable locations has been identified for hosting a cloudlet facility, and that installation costs, link lengths and capacities, cloudlets capacities, service level requirements and number of users of the system have been estimated by the operator. We report how we set these parameters in our evaluations in Sect. III.

Our models aim at finding simultaneously (i) an optimal network design, including cloudlet placement and assignment of BSs to cloudlets, and (ii) an optimal routing of the traffic from and to the cloudlets, even if the main aim is to provide

strategic insights into optimal design policies rather than an operational planning.

From a practical perspective, placing a cloudlet at a location could mean turning on already installed virtualization servers, and not only physically installing new machines. Similarly, changing BS to cloudlet assignments would in practice correspond to a re-routing of virtual links over the transport network infrastructure, and not physically changing the interconnection network. We consider a solution to be feasible if users' service level agreement is respected. Among feasible solutions, we consider as optimal those minimizing a linear combination of overall installation costs and cloudlet access latency bounds.

Our problem turns out to be hard from both a theoretical and computational point of view. Theoretically, it is strongly NP-Hard, generalizing the traditional uncapacitated facility location problem and its capacitated and single-source variants. Computationally, it is on the cutting edge of those currently under investigation in the facility location literature [26]: state-of-the-art methods are successful when up to two facility levels are considered, but in our models routing optimization, latency bounds and a third location level must be included.

B. Static Planning

We present model's input, output, goal and constraints.

a) Input (problem data): Formally, let B be the set of BS locations. Let I, J and K be the set of sites where aggregation, core nodes and cloudlet facilities can be installed, respectively. Since we assume a superposition of stars as network topology, any BS is connected to a single aggregation node, and each aggregation node to a single core node (as depicted in Fig. 1), while a full mesh is built among cores. Therefore, each BS $s \in B$ can connect to a cloudlet located in $k \in K$ by a set of paths $\bar{S}^{sk}$ (see paths a, b, c and d in Fig. 1). A path $p \in \bar{S}^{sk}$ can traverse multiple sites and we use the notation $j \in p$ to state that site j is traversed by path p.

For each BS $s \in B$, let a_s and b_s be the number of users connected to s and their overall bandwidth consumption. We assume that servicing each user requires the activation of one VM, and therefore a_s represents also the number of VMs needed for BS s. It is worth noting that considering multiple VMs per user (i.e., a generic Infrastructure as a Service) is straightforward and can be easily defined; conversely, sharing a VM by multiple users is not straightforward (and may not correspond to real services); these adaptations are out of scope and left to future work.

Let l_i, m_j, c_k be the fixed cost for activating an aggregation node in $i \in I$, a core node in $j \in J$ and a cloudlet facility in $k \in K$, respectively. Let C denote the number of VMs that each cloudlet can host. Let $d_{i,j}$ and $u_{i,j}$ be the length and bandwidth capacity of each link $(i,j) \in E = (B \times I) \cup (I \times J) \cup (J \times J)$. Finally, let D be a threshold for link length, i.e. the maximum distance allowed between nodes in the network to establish a link, and L be the maximum allowed latency level a user may experience, which we assume to be represented in terms of maximum sum of links' length in a path.

b) Output (decision variables): We introduce two sets of variables. The first set corresponds to location binary variables: x_i take value 1 if an aggregation node is set in $i \in I$. y_j take value 1 if a core node is set in $j \in J$; z_k take value 1 if a cloudlet is set in $k \in K$. The second set corresponds to routing variables: binary variables $r_p^{s,d}$ take value 1 if users in BS $s \in B$ are served by cloudlet in $d \in K$, and the corresponding traffic is routed along path $p \in \bar{S}^{sd}$. Let U be a continuous variable in $[0,1]$, representing the maximum link utilization's percentage on the most congested link.

c) Objective function: The aim of our design problem is to minimize a tradeoff between installation costs and penalties on a worsening of quality of service measured as cloudlet access latency:

$$\min \alpha_0 \cdot \sum_{i \in I} l_i x_i + \alpha_0 \cdot \sum_{j \in J} m_j y_j + \alpha_0 \cdot \sum_{k \in K} c_k z_k + \alpha_1 \cdot \upsilon(U) \quad (1)$$

where $\upsilon(U)$ is a function mapping link utilization to a costs of lower quality of service and α_0, α_1 are user-defined parameters that represent the relative importance of the two (conflicting) objectives.

d) Constraints: The feasible paths are those that satisfy two conditions: (i) no link along the path exceeds the maximum link length and (ii) the overall path length does not yield a violation on the maximum allowed latency, which we express as the function $\lambda(\cdot)$ on the sum of path's link lengths. In order to enforce that only feasible paths are considered, we replace each set $\bar{S}^{dk}$ with the following set:

$$S^{sk} = \{p \in \bar{S}^{sk} : \lambda \left(\sum_{(i,j) \in p} d_{(i,j)} \right) \leq L \wedge d_{(i,j)} \leq D \, \forall (i,j) \in p\}$$

Constraints (2)-(4) impose, respectively, that no path can be selected unless devices are installed in the corresponding aggregation, core and cloudlet sites.

$$\sum_{p \in S^{sk} | i \in p} r_p^{s,k} \leq x_i \ \forall s \in B, \forall k \in K, \forall i \in I \quad (2)$$

$$\sum_{p \in S^{sk} | j \in p} r_p^{s,k} \leq y_j \ \forall s \in B, \forall k \in K, \forall j \in J \quad (3)$$

$$\sum_{p \in S^{sk}} r_p^{s,k} \leq z_k \ \forall s \in B, \forall k \in K \quad (4)$$

Constraints (5)-(7) ensure that each BS is assigned to a cloudlet. Constraints (6) enrich (4) by further imposing that active cloudlets provide at most C VMs. Constraints (7) ensure that U takes the maximum utilization's percentage among all links.

$$\sum_{k \in K} \sum_{p \in S^{s,k}} r_p^{s,k} = 1, \forall s \in B \quad (5)$$

$$\sum_{s \in B} \sum_{p \in S^{s,k}} a_s r_p^{s,k} \leq C z_k \ \forall k \in K \quad (6)$$

$$\sum_{s \in B} \sum_{k \in K} \sum_{p \in S^{s,k} | (i,j) \in p} b_s r_p^{s,k} \leq u_{(i,j)} U \ \forall (i,j) \in E \quad (7)$$

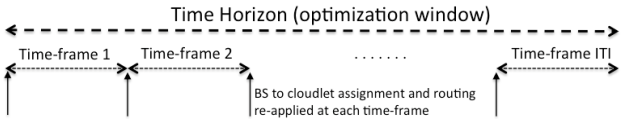


Fig. 2: Representation of the temporal framework.

C. Dynamic planning aware of temporal user & VM mobility

Users move during the day, and connect to different BSs. Hence the load on BSs change, and a re-planning may be needed with new routing maps to re-balance the system. Let T be a set of time-frames, i.e. a partitioning of the planning horizon in periods. We consider the user mobility during the overall given horizon without making assumptions on the users positions in a specific point in time. Let a_s^t and b_s^t be the (average) number of users connected to BS $s \in B$ and their overall bandwidth consumption during time-frame $t \in T$. Let $f_{s's''}$ be the number of users moving from BS $s' \in B$ to BS $s'' \in B$ during time horizon T.

We allow routing decisions to be changed dynamically, replacing for each $s \in B$, $k \in K$ and $p \in S^{sk}$ the variable r_p^{sk} with a set of variables $r_p^{s,k,t}$ for each $t \in T$. The static model is then extended as the following dynamic planning model:

$$\min \ \alpha_0 \sum_{i \in I} l_i x_i + \alpha_0 \sum_{j \in J} m_j y_j + \alpha_0 \sum_{k \in K} c_k z_k + \alpha_1 v(U) \tag{8}$$

$$\text{s.t.} \sum_{p \in S^{sk} | i \in p} r_p^{s,k,t} \leq x_i \ \forall s \in B, \forall k \in K, \forall i \in I, \forall t \in T \tag{9}$$

$$\sum_{p \in S^{sk} | j \in p} r_p^{s,k,t} \leq y_j \ \forall s \in B, \forall k \in K, \forall j \in J, \forall t \in T \tag{10}$$

$$\sum_{p \in S^{sk}} r_p^{s,k,t} \leq z_k \ \forall s \in B, \forall k \in K, \forall t \in T \tag{11}$$

$$\sum_{k \in K} \sum_{p \in S^{s,k}} r_p^{s,k,t} \geq 1, \forall s \in BS, \forall t \in T \tag{12}$$

$$\sum_{s \in B} \sum_{p \in S^{s,k}} a_s^t r_p^{s,k,t} \leq C y_k \ \forall k \in K, \forall t \in T \tag{13}$$

$$\sum_{s \in B} \sum_{k \in K} \sum_{\substack{p \in S^{s,k} \\ | (i,j) \in p}} b_s^t r_p^{s,k,t} \leq u_{(i,j)} U \ \forall (i,j) \in E, \forall t \in T \tag{14}$$

We remark that this model contains $|T|$ copies of each path variable and constraint of model (1) - (7), that are however linked since a single copy of each location variable is kept. By optimizing this model, two important network planning decisions are taken: an optimal placement of aggregation, core and cloudlet facilities, and an optimal assignment pattern of BSs to cloudlet facilities during the time horizon. Indeed, an optimal planning may assign different BSs to different cloudlet facilities during different time-frames.

From an operational point of view, when a user connects during planning horizon T to BSs that are assigned to different cloudlet facilities, two options can be considered: (1) VM bulk migration (the VM is stopped and the whole VM stack, including disk and memory, is transferred to the destination cloudlet) is performed from cloudlet to cloudlet as the user moves; (2) VM live migration (disk is continuously synchronized, only RAM is transferred upon migration, and the VM is stopped only when recently used RAM has to be transferred, see [20], [21]) is activated between *each cloudlet* serving BSs to which the user connects, *synchronizing the associated disk storage*. Option (2) offers some advantages, reducing congestion risks associated to bulk migrations; it can also improve user experience via seamless VM migration, without breaking active connections as explained in [14].

To include in the DP model the user mobility, we introduce variables $g_{s's''}^{k'k''} \in \mathbb{Z}_+$ representing the amount of users connecting to BSs $s' \in B$ and $s'' \in B$ served by cloudlets in sites $k' \in K$ and $k'' \in K$, respectively, through the planning horizon. Let also binary variables w_{sk} take value 1 if BS $s \in B$ is assigned to a cloudlet in $k \in K$ in any time-frame. The following constraints are needed to enforce coherence among these additional variables:

$$\sum_{p \in S^{sk}} r_p^{s,k,t} \leq w_{sk} \ \forall s \in B, \ \forall k \in K, \ \forall t \in T \tag{15}$$

$$g_{s's''}^{k'k''} \geq (w_{s'k'} + w_{s''k''} - 1) f_{s's''} \ \forall s', s'' \in B, \forall k', k'' \in K \tag{16}$$

Indeed taking into account the mobility path of each individual user is usually impractical due to confidentiality of data; constraints (16) allow to consider instead aggregated information.

1) Bulk VM Migration: We model the VM bulk migration option by including the overall number of migrations as a new term in the objective function, and we aim at minimizing it. Formally, we modify the objective function (8) as follows:

$$\begin{aligned} \min \ \alpha_0 \sum_{i \in I} l_i x_i + \alpha_0 \sum_{j \in J} m_j y_j + \alpha_0 \sum_{k \in K} c_k z_k + \\ + \alpha_1 \ v(U) + \alpha_2 \gamma \left(\sum_{k', k'' \in K} \sum_{s', s'' \in B} g_{s', s''}^{k', k''} \right) \end{aligned} \tag{17}$$

where $\alpha_0, \alpha_1, \alpha_2$ are user-defined parameters that represent the conversion multipliers for values of the two (conflicting) objectives, and $\gamma(\cdot)$ is a function mapping user migrations to a costs of lower quality of service. The bulk VM Migration model variation is therefore obtained by the set of equations (9)-(17).

2) Live VM Migration: We model the VM live migration option including explicitly in our model the routing and congestion assessment arising from cloudlet to cloudlet disk storage synchronization traffic. Let $\bar{Q}^{k'k''}$ be the set of synchronization paths, connecting cloudlet facilities installed in $k' \in K$ and $k'' \in K$. Let D^Q and L^Q be the counterpart of D and L for synchronization paths, and let:

$$Q^{k'k''} = \{p \in \bar{Q}^{k'k''} : \lambda \left(\sum_{(i,j) \in p} d_{(i,j)} \right) \leq L^Q \wedge d_{(i,j)} \leq D^Q \ \forall (i,j) \in p\}$$

represents the set of *feasible* synchronization paths between k' and k''. Then, let continuous variables $q_p^{k'k''t} \in \mathbb{R}_+$ represent the amount of synchronization traffic between

cloudlet facilities in $k' \in K$ and $k'' \in K$ routed along path $p \in Q^{k'k''}$ during time-frame $t \in T$. A path $p \in Q^{k'k''}$ can traverse multiple sites and we use the notation $j \in p$ to state that site j is traversed by path p. The following constraints enforce coherence among these additional variables:

$$\sum_{p \in Q^{k'k''}} q_p^{k'k''t} \geq \sum_{\substack{s',s'' \in B \\ |s' \neq s''}} \phi(g_{s',s''}^{k',k''}) \quad \forall k', k'' \in K | k' \neq k'', \forall t \in T \tag{18}$$

$$\sum_{p \in Q^{k'k''} | i \in p} q_p^{k'k''t} \leq M \cdot x_i \quad \forall i \in I, \forall k', k'' \in K, \forall t \in T \tag{19}$$

$$\sum_{p \in Q^{k'k''} | j \in p} q_p^{k'k''t} \leq M \cdot y_j \quad \forall j \in J, \forall k', k'' \in K, \forall t \in T \tag{20}$$

and link utilization constraints (14) become:

$$\sum_{s \in B} \sum_{k \in K} \sum_{\substack{p \in S^{s,k} \\ |(i,j) \in p}} b_s^t r_p^{s,k,t} + \sum_{\substack{k',k'' \in K \\ |k' \neq k''}} \sum_{p \in Q^{k'k''}_{|(i,j) \in p}} q_p^{k',k'',t} \leq \leq u_{(i,j)} \cdot U \quad \forall (i,j) \in E, \forall t \in T \tag{21}$$

$\phi(\cdot)$ in (18) is the function that maps the number of moving users $g_{s',s''}^{k',k''}$ to the amount of synchronization traffic they induce among cloudlets. The live VM migration model variation is therefore obtained by the set of equations (8)-(13), (15)-(16) and (18)-(21).

D. Heuristic resolution algorithms

The presented path-based formulations offer great modeling flexibility and present computational challenges at once. In particular, the number of feasible paths in sets S^{sk} and $Q^{k'k''}$ grows very fast with the network size. In order to obtain good feasible solutions in limited computing time, we implemented the ILP-based heuristics sketched below.

1) we transform variable U into a fixed parameter, on which we perform a parametric analysis. In this way, $v(U)$ is removed from equations (8) and (17); that also solves the problem of converting units of measure in the objective function. Further rationale for this choice is reported in Sect. IV.
2) we fix the location of aggregation devices, and the assignment of BSs to them, by heuristically creating clusters of BSs of limited size and minimum worst-case latency. Our heuristic works as follows: (i) we fix a number F of aggregation nodes to be installed; (ii) we fix a maximum cardinality G of BSs connected to each aggregation node; (iii) we run a PAM k-medoids heuristic [25] on the set of BSs to choose F baricentric ones; (iv) we use such a solution as initialization for a G-capacitated F-center alternating heuristic. This alternating heuristic, in turn, works as follows: (i) fix the locations of aggregation devices, and solve an ILP for assigning the BSs to aggregation devices, forming clusters of BSs where at most G BSs are connected, and minimizing the maximum distance between a BS and the center of its cluster; (ii) choose as new center for each cluster the BS minimizing the maximum distance between all other BSs in the cluster; then iterate from (i), until no more changes in the solution are observed.
3) we fix the x_i variables in our models according to the G-capacitated F-center solution obtained as above, we fix $J = K = \{i \in I : x_i = 1\}$, and we remove from S^{dk} sets all paths in which the BS d is not assigned to the aggregation device of its cluster.
4) a general purpose MILP solver is used to optimize the reduced model. After preliminary experiments, we fixed $F = 50$, $G = 1.3 \cdot \left(\frac{|B|}{F}\right)$, and stopped the final MILP solver optimization after 10 branch-and-bound nodes, or as soon as a duality gap less than 10% is reached.

In addition we have considered a simple heuristic (*Static Planning Heuristic* in the remainder of the paper) to perform dynamic planning without considering user mobility, using a hierarchical resolution strategy:

1) all time-frames are considered separately to get $|T|$ distinct solutions of the model SP described in Sect. II.II, which does not consider user mobility. We assume that the cloudlets selected in one time-frame will be active during all time horizon;
2) we use these $|T|$ solutions to fix variables z_k, $r_p^{s,k,t}$, $w_{s,k}$, $g_{s',s''}^{k',k''}$ in the live VM migration model variations of dynamic planning. Solutions of the model SD don't consider the requirements on the maximum latency of synchronization path, and so we have to compute a new value for $L^{Q'}$ and $D^{Q'}$ that will be equal to the higher distance between two enabled, and fixed, cloudlets. The choice of paths that violate the original values L^Q and D^Q is penalized in the objective function.

III. Dataset

In order to ground our simulations on real data, we used a dataset collected by Orange mobile, France, in the frame of the ABCD project [29]. The dataset comes from the network management tickets generated each 6 minutes and each time a mobile device uses the wireless mobile network for Internet data exchange. The probe assigns the session to the cell identifier of the last used antenna. Data are recorded on a per-user basis and cover a large metropolitan area network, including urban, peri-urban and rural areas.

We had access to data of a single 24-hour period, and we have limited our experiments using the data originating by 606 LTE 4G BSs of the region. The region covered by these BSs has an area of about 931 km^2, with a density of about 0.65 BSs per km^2. The number of users served by the considered BSs is about 180 thousands, generating an overall daily traffic of about 11 TB.

A. Estimation of Model Parameters

Coefficients a_s and b_s for each base station $s \in B$ are drawn by direct queries from the dataset.

Following [22] and [23], we fix $l_i = 0.01$, $m_j = 0.1$, and $c_k = 1$. These costs can be seen as percentage costs, and the

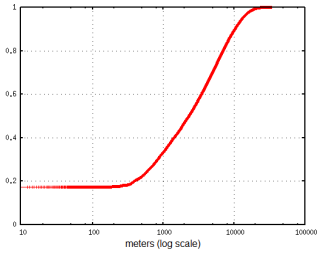


Fig. 3: CDF of traveled distances of user flights.

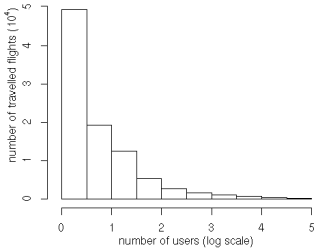


Fig. 4: Histogram of nb. of users covering same flight.

network costs can be estimated as about 10% of the overall cloud data center costs as suggested in [22].

As $d_{i,j}$ values we take the euclidean distances between each pair of BSs $i \in B$ and $j \in B$, as the underlying operator physical topology is not available to us. We fix the bandwidth capacities $u_{(i,j)}$ of each link $(i,j) \in E$ to 1 Gbps in both hierarchical levels. Observing the positioning of the BSs, we fix the maximum link length $D = 15\ km$, corresponding approximately to the radius of the metropolitan region under consideration, and we limit the paths to four hops. Instead of choosing a particular setting for C and L, we perform a parametric analysis on them, as presented in Sect. IV.

B. User Mobility Patterns

Individual user mobility patterns cannot be obtained for confidentiality reasons.

Furthermore, allowing migrations even when a BS is visited infrequently would have a strong negative impact on the overall network load, without significantly improving user experience. Trying to cope with this issue we perform binning on data: for each user we consider the two BSs which are visited more frequently during the planning horizon. These two BSs may represent, for instance, home and work place of users. We consider possible migrations only between these two locations. Technically speaking these data are obtained by creating groups of users and obfuscating individual identifiers on them. Different options may be considered, especially in absence of such data, to estimate mostly visited places [24].

Summarizing, for each pair of BSs s' and s'' let $f_{s',s''}$ be the number of users having s' and s'' as the most frequently visited BSs.

In order to further characterize such user mobility patterns, and to allow third parties to reproduce adequately our findings, we report in Fig. 3 the cumulative distribution function of the distances traveled by users while migrating. We observe that about 20% of users do not move at all during the day and that almost all users move less than the radius of the considered region (i.e. $15km$). Moreover, in Fig. 4, we present a histogram reporting on the x axis ranges for number of users. For each range $[x', x'']$ on the x axis, a bar represents the number of pairs of BSs s' and s'' having $f_{s',s''} \in [x', x'']$. We can conclude that (i) the majority of paths are covered by a small number of users (ii) about 72% of the possible pairs of BSs never appear as most frequent for any user. That is, the mobility is concentrated along a few frequently chosen paths, matching intuition.

IV. Experimental Results

We experimented our algorithms on the presented dataset, considering three cloudlet size cases: tiny cloudlet of $C = 1$ rack, a car parking cloudlet of $C = 4$ racks, and a 2-4 DC-room cloudlet with $C = 40$ racks, considering one rack can host up to 2500 VMs, using values indicated in [8].

Moreover, we consider three reference mobile cloud services, strictly related to the cloud access latency requirements. Considering bottleneck-free back-hauling network ($U \leq 1$) where latency is roughly directly proportional to the traveled distance, we consider $\lambda(\cdot)$ to be the identity function, and three latency bounds L directly proportional to euclidean traveled distances: 'loose', 'mid-level', and 'strict' bounds, corresponding, respectively, to roughly the urban area radius ($15\ km$), $4/5$ and $2/3$ this radius. These three levels of cloudlet access latency can correspond to three reference mobile cloud services: delay-tolerant *storage box services* for the loose case, delay-sensitive *remote desktop services* for the mid-level case, and delay-critical *augmented-reality support services* requiring real-time video or voice recognition. We express these bounds relatively because there is no available public information on absolute cloudlet network latency requirements, despite partial valuable information can be found at [5], [27].

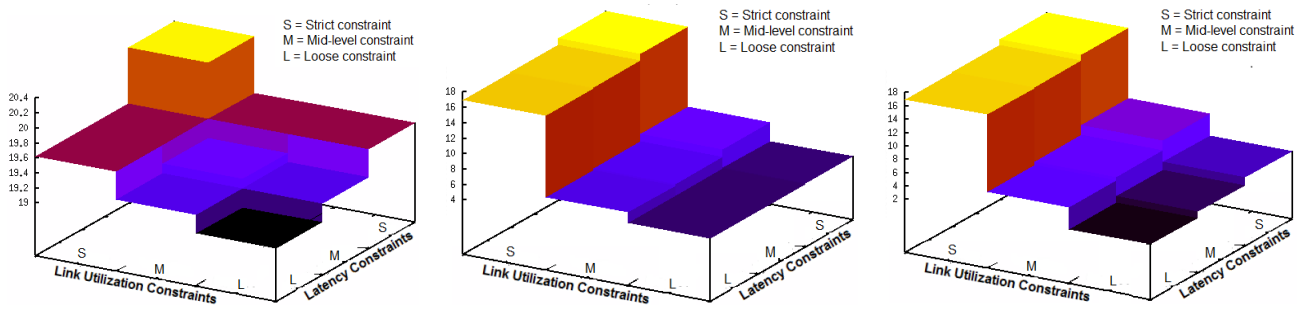
As a common practice in IP traffic engineering, links have a level of over-provisioning so that they are robust against the occurrence of failures, traffic peaks, hence the risk of congestion. As already described, the maximum link utilization's percentage (U) needs to be kept not simply below 100%, but as low as possible in order to better master the congestion risk and guarantee the quality of experience for real-time and interactive services. We evaluate three levels for the maximum link utilization: 'loose', 'mid-level', and 'strict'. The stricter they are, the better interactive services, such as remote desktop and augmented reality, support are expected to be. Storage box TCP-based services are instead fault tolerant, given the commonly non real-time usage of its data.

In the following, we report extensive results for the static planning algorithm, then we investigate the interesting parametric scenarii for the dynamic planning, with bulk migration and live migration, finally comparing the approaches in terms of virtual resource migration volume.

A. Analysis of static planning solutions

For the static planning case (see Sect. II.B), in order to obtain a statistically significant benchmark, we consider 9 time-frames, by averaging the traffic and number of users at each BS over the time-frames: (i) full day, (ii) 0 am to 12 am, (iii) 12:01 pm to 11:59 pm, (iv) 0 am to 6 am, (v) 6 am to 12 am, (vi) 12:01 pm to 6 pm, (vii) 6 pm to 11:59 pm, (viii) 8 am to 8 pm, (ix) 8 pm to 8 am. In this way, we get $9 \cdot 3 \cdot 3 \cdot 3 = 243$ scenarii, combinations of capacity, delay and link utilization bound settings.

As first fitness measure we consider the number of installed cloudlets, as reported in Fig. 5, for the different scenarii (averages over the different scenarii, with an interval of confidence of 99.95%). We can observe that:



(a) Cloudlet Capacity 1 Rack (2500 VMs) (b) Cloudlet Capacity 4 Racks (10k VMs) (c) Cloudlet Capacity 40 Racks (100k VMs)

Fig. 5: Mean number of cloudlets as a function of cloudlet capacity, link utilization and latency bounds (i.c. $= 99.95\%$).

- The case with largest number of installed cloudlets is, trivially, the one with lowest rack capacity: it is between 19 and 20, without changes strengthening delay and utilization bounds. No big difference exists between the 4-rack and the 40-rack cases, while logic suggests a lower number of cloudlets for the 40-rack case: this effect is due to the delay constraints requiring a minimum level of cloud resource geo-distribution. Overall, intermediate size facilities (4 racks) appear as the most appealing option: smaller ones require to install on average one cloudlet every two aggregation nodes, that appears as too much, and larger ones do not reduce the number of required facilities significantly, leading to resource and space waste.
- In terms of maximum link utilization, the number of required cloudlet facilities rapidly grows passing from mid-level bound to strict bound, except for the 1-rack case, likely because of the lower aggregation of traffic with a more distributed cloudlet network.
- In terms of cloudlet access latency, we cannot see clear changes, except for the above consideration for the case of the lowest cloudlet capacity. On average, the solutions show very little sensitivity on the value of L, suggesting that, with static planning unaware of user and VM mobility, a location planning could be pursued without specifically considering different services.

As a second fitness measure, we consider the cumulative distribution function of the cloudlet access path length, as reported in Fig.s 6 and 7. We can note that:

- The lowest cloudlet capacity case allows for very low-delay paths. This can be expected, as when the more facilities are enabled, the more a BS can connect to a nearer cloudlet. The behavior with medium and high capacity is very similar yet much flatter. With the highest capacity, paths are shorter than those with the medium capacity: this may be due to the fact that using a very high capacity, each BS can connect to the nearest cloudlet, while with a more constrained capacity the nearest cloudlet may be already sufficiently loaded.
- With less stringent access delay bounds, trivially the average path length increase, but the distribution of paths does not show striking differences.
- Mid-level and loose bounds on the maximum link utilization yield very similar path length distributions. Instead, for the strict bound, more paths are short: very few aggregation nodes can route traffic on the same links to the cloudlet facilities; this requires to activate many facilities, that in turn allows to assign aggregation nodes to cloudlets that are very near.

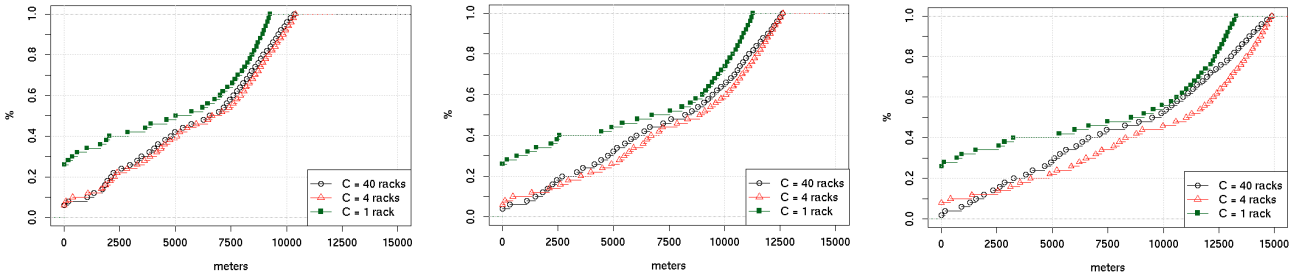
B. *Analysis of dynamic planning solutions*

In a second round of experiments, we tested the behavior of the dynamic models (see Sect. II.C) in the case of two time-frames: from 8 am to 8 pm, and from 8 pm to 8 am. We run simulations for both the bulk and live VM migration cases. We restrict the simulations to the six more interesting scenarii from the static planning results, concentrating on the 4-rack scenarii and discarding the strict maximum link utilization scenarii, which in the previous analysis showed less interesting results.

For the bulk migration model, we set $\alpha_0 = 1$, $\alpha_2 = 1$ and $\gamma(n) = 1/C \cdot n$ to scale the number of migrations by the capacity of a cloudlet: this is justified by the worst-case situation in which enabling a new cloudlet may lead to a number of VM migrations equal to its VM capacity. For the live migrations model, for storage synchronizations we set $\lambda(\cdot)$ to the identity function and $L^Q = 12.75\ km$ ($4/5$ of the urban area radius), i.e. we consider the synchronization as a service that requires a mid-level latency bound; moreover, the mapping function $\phi(\cdot)$ is characterized by the type of mobile cloud service. The size of the disk for *augmented-reality support* VMs, requiring strict latency bounds, is reasonably lower than the one for *remote desktop* VMs, requiring mid-level latency bounds, in turn lower than for *storage box* VMs, requiring a loose level of latency bounds. Considering storage box VMs need the whole disk to be synchronized, we arbitrary set $\phi(\cdot)$ to 100% of the user volume in the time-frame; instead, for remote desktop VMs, only part of the disk is expected to be modified upon user actions, so $\phi(\cdot)$ is set to 70%, while for augmented-reality support VMs, disks are smaller and consequently only small volume need to be synchronized, and $\phi(\cdot)$ is set to 30%.

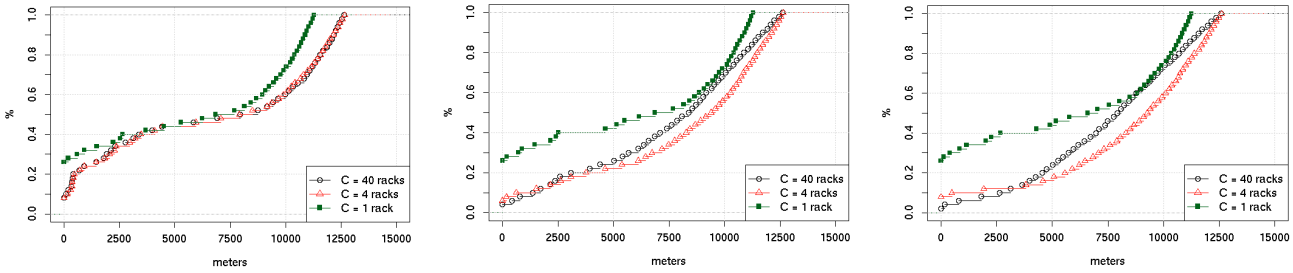
As first fitness measure we consider the number of enabled cloudlets installed, reported in Table I. For the Static Planning Heuristic, it refers to the union of the cloudlets activated in each time-frame.[1] We can measure the disadvantage in static

[1]With an optimality gap g below 10% no additional cloudlets could be opened for most cases (just BS-cloudlet associations could change). The high value given by bulk migration for the *strict latency - mid-level utilization* case is due to the high gap that was possible to obtain.



(a) Strict Delay Sensitivity (b) Mid-level Delay Sensitivity (c) Loose Delay Sensitivity

Fig. 6: Cumulative Distribution Function of paths length as a function of different cloudlet capacities and delay sensitivities



(a) Strict Link Utilization Bound (b) Mid-level Maximum Link Utilization Bound (c) Loose Maximum Link Utilization Bound

Fig. 7: Cumulative Distribution Function of paths length as a function of different cloudlet capacities and link utilizations

planning rather than dynamic planning as the higher number of cloudlets needed: roughly from 15% to 40% more cloudlets are needed with the static planning. No sensible differences between the two types of migration can be noticed. In fact, they do differ in the number and the volume of migrations.

As second fitness measure we consider the expected number of VM migrations generated by the different planning models (it can be seen as a measure of expected incremental point traffic on the network), as reported in Table II, as percentage of the worst-case number of expected migrations.[2] For the Static Planning Heuristic, it corresponds to its independent executions at each time-frame. It becomes striking the advantage of considering multiple time-frames together with user mobility in the same optimization: under static planning, the percentage of VM migrations is always higher than 75%, while using any dynamic planning model the percentage is always less than 22%. The bulk migration model gives a lower expected number of migrations than the live migration model. However, there is an important difference between live and bulk migration in terms of point migrated volume of traffic.

As third fitness measure we consider the traffic volume generated by the different methods as a function of the number of migrations. Let RAM and HD be the capacities of VM's RAM and hard-disk. As live migration does not require disk bulk transfer, the traffic n live migrations generate is given by $RAM \cdot n + \phi(n)$, where $\phi(n)$ is the mapping function to compute the synchronization traffic. Instead, the traffic generated by n bulk migrations is given by $(HD+RAM)\cdot n$.

[2]The expected number of VM migrations is given by values of variables $g^{k'k''}_{s's''}$ giving the number of users moving from s' to s'' associated to k' and k'', respectively. The worst-case maximum number of expected VM migrations is given by $\binom{|T|}{2} \cdot \sum_{s' \in B} f_{s',s'} + |T|^2 \cdot \sum_{s',s'' \in B | s' \neq s''} f_{s',s''}$ having $s' \neq s''$ and $k' \neq k''$.

In Fig. 8 we compare the traffic volume generated by bulk migration and live migration for the loose and mid-level utilization bounds, for the three reference mobile cloud services: as already argued, the augmented reality support service is expected to have the lowest HD capacity and the lowest volume of synchronization traffic, the remote desktop higher values than it, and storage box a higher value than remote desktop. We can notice that there is an even point after which live migration is more interesting than bulk migration; it is easy to see that depends on parameters HD and $\phi(\cdot)$. For augmented reality, the bulk migrations manifests as more effective, which is counter-intuitive; for remote desktop and storage box services, using higher capacity HD and $\phi(\cdot)$, live migration gives advantage in minimizing the traffic volume, despite the number of expected migrations generated by the model is higher.

V. Conclusion

We provided for the first time at the state of the art a comprehensive cloudlet network design framework for mobile access metropolitan area networks. We formally defined the problem, including a planning mode unaware of user and virtual machine mobility, a mode considering bulk migrations and another considering live migrations. We compared the different planning options extensively for scenarii built over real cellular network datasets, and compared them in terms of number of enabled cloudlets and migrated volume, for different traffic engineering and performance goals for reference mobile cloud services. We highlighted the high gain deriving from the consideration of user and virtual machine mobility in the network planning, and determined for which mobile cloud service which planning approach appears as the most appropriate one in terms of migrated traffic volume. We believe the provided insights can stimulate further researches

TABLE I: Number of enabled cloudlets (g is the optimality gap - for the static case is the max of two time-frame gaps).

Bounds on:		*Static Planning Heuristic*			*Dynamic planning - Live Migration*			*Dynamic planning - Bulk Migration*		
Cloudlet access latency:		strict	mid-level	loose	strict	mid-level	loose	strict	mid-level	loose
Link utilization:	mid-level	16 g=.04	14 g=.03	11 g=.10	13 g=.02	12 g=.09	11 g=.09	14 g=.08	11 g=.04	10 g=.03
	loose	9 g=.08	10 g=.10	9 g=.09	6 g=.04	7 g=.24	6 g=.09	18 g=.66	6 g=.18	6 g=.12

TABLE II: Percentage of number of expected VM bulk migration over the worst-case number of expected migrations.

Bounds on:		*Static Planning Heuristic*			*Dynamic planning - Live Migration*			*Dynamic planning - Bulk Migration*		
Cloudlet access latency:		strict	mid-level	loose	strict	mid-level	loose	strict	mid-level	loose
Link utilization:	mid-level	88.98	91.03	76.68	21.51	20.21	20.48	18.66	18.36	21.68
	loose	77.66	88.08	83.44	20.37	19.45	21.44	19.50	17.38	19.39

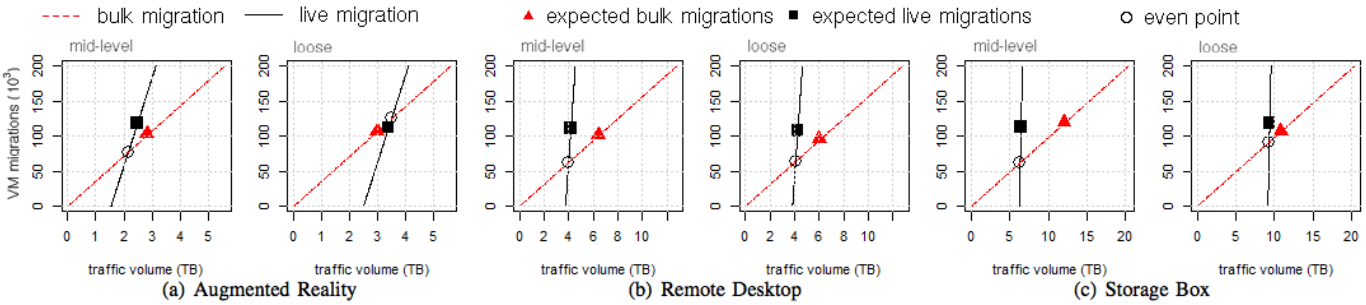


Fig. 8: Traffic volume generated by live and bulk migrations vs the number of migrations (for two link utilization bounds).

in the rising research field of mobile cloud networking in general and mobile edge computing [28] in particular.

ACKNOWLEDGEMENT

This project was funded by the ANR ABCD project (Grant No: ANR-13-INFR-005), the FP7 MobileCloud project (Grant No. 612212), and the EIT ICT-Labs FNS action line. We thank Cezary Ziemlicki and Sandesh Uppor from Orange labs for their support with cellular mobility data retrieval.

REFERENCES

[1] M. Satyanarayanan, P. Bahl, R. Caceres, N. Davies, "The Case for VM-based Cloudlets in Mobile Computing", *IEEE Pervasive Computing*, Vol. 8, no. 4, pp: 14-23, 2009.
[2] Y. Jararweh, et al., "Resource Efficient Mobile Computing Using Cloudlet Infrastructure", in *Proc. of IEEE MSN 2013*.
[3] Elijah project (website): http://elijah.cs.cmu.edu (June 2014).
[4] S. Clinch, et al., "How close is close enough? Understanding the role of cloudlets in supporting display appropriation by mobile users", in *Proc. of IEEE PerCom 2012*.
[5] K. Ha, P. Pillai, G. Lewis, M. Satyanarayanan, "The impact of mobile multimedia applications on data center consolidation", in *Proc. of IEEE IC2E 2013*.
[6] D. Fesehaye, Y. Gao, K. Nahrstedt, G. Wang, "Impact of cloudlets on interactive mobile cloud applications", in *Proc. of IEEE EDOC 2012*.
[7] J. Hamilton, "An Architecture for Modular Data Centers", in *Proc. of CIDR 2007*.
[8] K. Church, A. Greenberg, J. Hamilton, "On delivering embarrassingly distributed cloud services", in *Proc. of Hotnets 2008*.
[9] Myoonet project (website): http://www.myoonet.com (June 2014).
[10] B. Han, P. Hui, A. Srinivasan, "Mobile data offloading in metropolitan area networks", *ACM SIGMOBILE Mobile Computing and Communications Review*, Vol. 14, No. 4, pp: 28-30, 2011.
[11] C. Liu, et al., "The case for re-configurable backhaul in cloud-RAN based small cell networks", in *Proc. of IEEE INFOCOM 2013*.
[12] H. Jiang, et al., "Understanding bufferbloat in cellular networks", in *Proc. of the 2012 ACM SIGCOMM workshop on Cellular networks: operations, challenges, and future design*.
[13] R. Bradford, E. Kotsovinos, A. Feldmann, H. Schiberg, "Live wide-area migration of virtual machines including local persistent state", in *Proc. of ACM CoNext 2007*.
[14] P. Raad, et al., "Achieving Sub-Second Downtimes in Large-Scale Virtual Machine Migrations with LISP", *IEEE Trans. on Network and Service Management*, Vol. 11, No. 2, pp: 133-143, 2014.
[15] P. Raad, S. Secci, D.C. Phung, P. Gallard, "PACAO: Protocol Architecture for Cloud Access Optimization", in *Proc. of IEEE NETSOFT 2015*, April 13-17, 2015, London, UK.
[16] M. Howard, "Using Carrier Ethernet to Backhaul LTE", Infonetics Research, White Paper, 2011
[17] M.A. Alvarez, F. Jounay, T. Major, P. Volpato, "LTE backhauling deployment scenarios", NGMN Alliance, White Paper, 2011.
[18] R. Nadiv, T. Naveh, "Wireless Backhaul Topologies: Analyzing Backhaul Topology Strategies", CERAGON, White Paper, 2010.
[19] "Architectural Considerations for Backhaul of 2G/3G and Long Term Evolution Networks", Cisco, White Paper, 2010.
[20] C. Clark, et al. "Live migration of virtual machines", in *Proc. of USENIX NSDI 2015*.
[21] "KVM live migration", Chapt. 5, Red Hat Enterprise Linux 6 Virtualization Administration Guide.
[22] A. Greenberg, J. Hamilton, D.A. Maltz, P. Patel, "The cost of a cloud: research problems in data center networks", *ACM SIGCOMM computer communication review*, Vol. 39, no: 1, pp: 68-73, 2008.
[23] A. Beloglazov, et al., "Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing", *Future Generation Computer Systems*, Vol. 28, No. 5, pp: 755-768, 2012.
[24] S. Isaacman, et al., "Identifying important places in people's lives from cellular network data", in *Pervasive Computing*. Springer Berlin Heidelberg, pp: 133-151, 2011.
[25] A. Reynolds, et al., "Clustering rules: A comparison of partitioning and hierarchical clustering algorithms", *J. of Mathematical Modelling and Algorithms*, Vol. 5, no. 4, pp: 475-504, 1992.
[26] B. Addis, G. Carello, A. Ceselli, "Combining very large scale and ILP based neighborhoods for a two-level location problem", *European J. of Operational Research*, Vol. 231, no. 2, pp: 535-546, 2013.
[27] V. Bahl, "Mobile Gaming", MobiGames 2012 Keynote Speech, http://research.microsoft.com/en-us/um/people/bahl/Present/Bahl_keynote_mobile_gaming_2012.pdf.
[28] M. Patel, B. Naughton, C. Chan, N. Sprecher, S. Abeta, A. Neal, et al. "Mobile-Edge Computing Introductory Technical White Paper", *European Telecommunications Standards Institute*, https://portal.etsi.org/Portals/0/TBpages/MEC/Docs/Mobile-edge_Computing_-_Introductory_Technical_White_Paper_V1%2018-09-14.pdf, 2014
[29] ANR ABCD Project (website): http://abcd.lip6.fr (November 2014).