# Move with Me: Scalably Keeping Virtual Objects Close to Users on the Move

Roberto Bruschi[2], Franco Davoli[1,2], Paolo Lago[1,2] and Jane Frances Pajo[1,2]
[1]DITEN, University of Genoa, Genoa, Italy
[2]CNIT S3ITI National Laboratory, Genoa, Italy
roberto.bruschi@cnit.it, franco.davoli@unige.it, {paolo | jane.pajo}@tnt-lab.unige.it

***Abstract*—The upcoming Cloud-Fog interplay is expected to grant service providers more degrees of freedom in the implementation and management of their service portfolios. However, recent advances in Mobile Internet have developed a growing need to support user mobility – as users move, the Fog counterparts of services that require close proximity may call for migration(s) to meet the desired quality of service (QoS). With the state-of-the-art virtualization technologies, next-generation Cloud/Fog services are being implemented in modular software (i.e., as a graph/chain of portable virtual objects (VOs)) that can be migrated around the Telco infrastructure, and yet scalability is still an open issue, especially with the inter-datacenter bulk live migration of VOs. In this perspective, a VO clustering and migration policy that jointly considers user proximity and inter-VO affinity is proposed to scalably support user mobility, while allowing service differentiation among users. Results confirm that introducing migrations improve the QoS to always meet or exceed the requirements, as compared to static service placement, and considering VO clusters as aggregate entities will initiate around $40\%$ less migrations, on average – an improvement that increases with inter-VO affinity and could potentially simplify service management when supporting user mobility.**

## I. Introduction

The emerging Fog paradigm is expected to bring Cloud-like services at different levels of user proximity, as small to medium-sized computing facilities (e.g., street cabinets [1], micro- and container-based datacenters [2], mobile base stations [3], among others) join in. This upcoming Cloud-Fog interplay will grant service providers more degrees of freedom in improving either the quality of service (QoS) of service components or the quality of experience (QoE) of the end-users, as necessary.

In the recent years, more and more devices (of increasingly heterogeneous capabilities) are connecting to the Internet, and the numbers are expected to grow to 29 billion by 2022 [4]. Fog nodes can provide a wide-range of services to improve the performance and augment the capabilities of these devices (i.e., from providing intelligence to dumb devices, to offloading smart ones), when the Cloud is located too far for the required QoS. Works like [5] and [6] demonstrate the improvements in latency and network usage that can be achieved by pooling Cloud and Fog resources.

State-of-the-art Cloud services have been starting to embrace modular software design, and can be viewed as graphs/chains of software components referred to as virtual objects (VOs) [7] hereinafter. A similar scenario is expected in the Fog domain, with the exception that the VOs will have heterogeneous user proximity requirements (e.g., home environment and Content Delivery Network (CDN) virtualization use cases [8], mobile augmented reality applications [9], among others).

An open issue with the Fog scenario, brought by recent advances in Mobile Internet, regards user mobility support – as users move, the Fog counterparts of services that require close proximity may call for migration(s) to meet the Service Level Agreement (SLA). At the same time, correlation/chaining among VOs (i.e., inter-VO affinity) must be taken into account in making such migration decisions. Depending on both user proximity and inter-VO affinity, (bulk) VO migration(s) may be initiated with user mobility, which should be performed with minimal/no service disruption for seamless user experiences. Although (live) migration support [10] in state-of-the-art virtualization technologies enables portability of VOs around the Telco infrastructure with unprecedented simplicity, scalably realizing the ensuing network re-configurations is of utmost importance, especially for inter-datacenter bulk live migration of (an increasingly large number of) chained VOs.

A number of recent works gave different contributions to this user-service mobility problem. For instance, the *Follow Me Cloud* framework [11] proposed full/partial "service migration" by initiating/replicating VOs based on migration costs vs. QoS/QoE trade-off, while the authors in [12] considered live migration, taking into account the dynamic user access patterns and migration amortization in the decision. As regards bulk live migration optimization, [13] and [14] focused on migration bandwidths and remapping of correlated VOs, respectively. However, they either consider only a single VO (or a small set of VOs) or neglect the inter-VO affinity, and more importantly, they did not study the scalability aspect of wide-area network re-configurations during the migrations.

In a user-centric perspective, the *INPUT* platform for personal Cloud services supports seamless (bulk) live migration of user-owned VOs based on QoS/QoE [8]. Its reference architecture, *OpenVolcano* [15], seeks to cluster VOs with similar QoS/QoE requirements, and consider each cluster as an aggregate entity in the wide-area network in order to minimize the number of re-configuration operations (i.e., in terms of frame forwarding rule updates) [16]. Building on this, we look into how the VOs can be clustered, by jointly considering user proximity and inter-VO affinity.

978-1-5386-3180-5/18/$31.00 

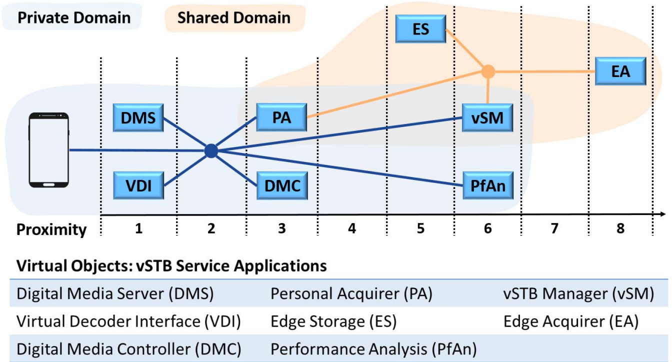



Fig. 1: Example based on the vSTB use case in [8].

This paper proposes a proximity- and affinity-aware clustering and migration policy for user-centric VO networks to scalably support user mobility in a Cloud-Fog environment, while allowing service differentiation among users. A series of numerical evaluations are conducted on a graph-based logical topology to evaluate the performance of the approach, providing insights on the QoS improvement and service management simplification offered by cluster migration.

Fig. 1 illustrates our conceptual framework in an example based on the *virtual Set-Top-Box (vSTB)* use case evaluated in [8], considering the vSTB *service applications* as the user's VOs. Both the user's private domain (e.g., personal network) and the third parties' shared domain (e.g., back-end networks) are indicated through a multi-point link model, and the VOs' QoS/QoE requirements through user proximity levels; note that VO clusters with lower proximity levels may require migrations more often than those with higher proximity levels.

The remainder of this paper is organized as follows. Section II describes a user-centric VO network and the metrics considered for user mobility support. Section III provides details on the proposed VO clustering and migration approach. Numerical results are then presented in Section IV, and finally, conclusions are drawn in Section V.

## II. A User-centric Virtual Object Network

We consider a scenario where users own a set of Cloud- and/or Fog-hosted VOs with varying user proximity and inter-VO affinity requirements. Such requirements must be taken into account as users move around throughout the day in order to keep the desired QoS/QoE.

Each user $u$ is associated to a set of VOs $V_u$ that can be placed in a distributed and dynamic fashion within the set of Telco (in-network) datacenters $D$. A high-level view of a user's VO network and connectivity at a certain time instant is illustrated in Fig. 2. As $u$ moves from one access point to another in the succeeding time instants, migrations may be necessary to meet the close proximity requirements of some VOs. In addition, such VOs may be tightly coupled to other VOs with loose proximity requirements (i.e., as service chains). Hence, both user proximity and inter-VO affinity will be considered in the proposed VO cluster migration. More details on these metrics will be discussed in the following sub-sections.

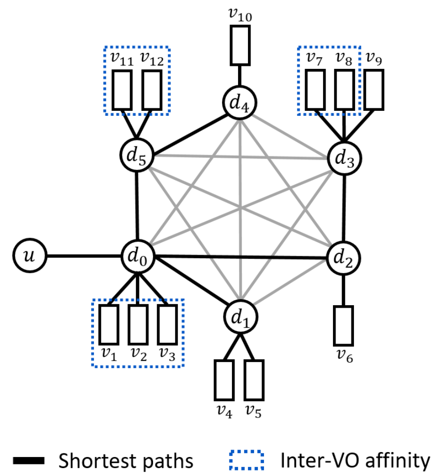



Fig. 2: High-level view of a user's VO network and connectivity at a certain time instant.

### A. User Proximity

User proximity can be measured in terms of different QoS parameters (e.g., path lengths, latencies, available bandwidth, etc.). In this work, we consider the path lengths from user $u$'s access device to the subset of datacenters $D_u \subseteq D$ currently hosting his/her VO network, supposing that latencies and bandwidths are already represented in the link weights; for instance, larger weights can be assigned to high latency/low bandwidth links, resulting in "longer" path lengths.

We define *proximity levels* using a range of indexes $\{\mathbf{1}, \ldots, \mathbf{P}\}$ based on user $u$'s proximity requirements given as $\Delta_u = \{\Delta_u(i), i = 1, \ldots, |V_u|\}$, and a subscription-based parameter $P_u$. The latter specifies the maximum number of proximity levels allowed by $u$: $\mathbf{P} = P_u$, where $P_u \leq |V_u|$. Each index is then mapped to a range of path lengths, with index $\mathbf{1}$ corresponding to the level requiring the closest proximity. Note that users with premium subscriptions can invoke smaller $P_u$ values to have less proximity levels with longer range intervals.

In more detail, we suppose that $\Delta_u$ and $P_u$ are specified in the SLA. Among the allowable path lengths, let $L_{min} = min(\Delta_u)$ and $L_{max} = max(\Delta_u)$, corresponding to the VOs with the tightest and loosest requirements, respectively. The $p$-th range, $p = 1, \ldots, P_u$, is then given by

$$[\,r_{min}(p),\; r_{max}(p)\,] = [\,L_{min}{+}(p{-}1){\cdot}R,\; L_{min}{+}p{\cdot}R\,] \quad (1)$$

where $R = (L_{max} - L_{min})/P_u$. VOs that fall on the $p$−th range will have a proximity level $\mathbf{p}$.

### B. Inter-VO Affinity

The concept of inter-VO affinity is somehow analogous to the ETSI NFV's "affinity/antiaffinity rules" that define whether a certain (sub)set of resources are placed in proximity to one another (e.g., sharing the same physical NFV infrastructure node) [17]. Here, the VOs are the users' resources, which may have proximity requirements not only towards users, but among one another as well. We refer to the latter as 'affinity' hereinafter to distinguish the two metrics.

In general, multiple *affinity levels* can also be defined through a range of indexes by considering inter-VO traffic. However, such interactions may not be directly specified in the SLA, requiring more advanced learning mechanisms to be extracted from each user's VO network [18]. For the sake of simplicity, but without losing generality, we only consider two levels in this paper – i.e., the distance $\delta_{(i,j)}$ between any pair of user $u$'s VOs $(v_i, v_j) \in V_u$, $i \neq j$, is either $0$ or $\infty$, leaving multi-level affinity to future work. This means that pairs of VOs with 0 distance must be placed in the same datacenter, while the rest can be placed in any datacenter $d \in D$, provided that their user proximity requirements are met. User $u$'s affinity requirements are given as $\delta_u = \{\delta_{(i,j)}, i, j = 1, \ldots, |V_u|, i \neq j\}$.

## III. Proximity- and Affinity-aware Cluster Migration

In this section, we introduce a novel VO clustering and migration policy that supports user mobility by jointly considering user proximity and inter-VO affinity requirements. Additionally, it adopts a subscription-based clustering parameter (that could vary among users) to allow service differentiation.

Suppose that for each user $u$, the proximity ($\Delta_u$) and affinity ($\delta_u$) requirements, as well as the subscription-based parameter ($P_u$), are given. Firstly, a two-step VO clustering is performed by:

**S1:** considering the inter-VO affinity to obtain an initial set of clusters $\hat{C}$; and

**S2:** considering the user proximity to obtain the final set of clusters $C$.

Then, each cluster $c \in C$ is dynamically placed according to the minimum proximity requirement $min(\Delta_u^c)$ among VOs in $c$ and $u$'s current access point. More details on the process are discussed in the following sub-sections.

### A. VO Clustering

From the given inter-VO affinity requirements, VO pairs $(v_i, v_j) \in V_u$, $i \neq j$, with $\delta_{(i,j)} = 0$ are first clustered together. At the end of this step, we obtain $\hat{C}$ initial clusters.

Now, from this initial clustering and the user proximity requirements, the second step starts by identifying the minimum requirements $min(\Delta_u^{\hat{c}})$ of each cluster $\hat{c} \in \hat{C}$. Then, the range intervals of the $P_u$ proximity levels are obtained by adapting Eq. (1) to consider clusters instead of individual VOs – i.e., by letting $L_{min} = min_{\hat{c} \in \hat{C}}\{min(\Delta_u^{\hat{c}})\}$ and $L_{max} = max_{\hat{c} \in \hat{C}}\{min(\Delta_u^{\hat{c}})\}$. This allows merging of clusters $\{\hat{c}\} \subseteq \hat{C}$ with $\{min(\Delta_u^{\hat{c}})\}$ falling on the same range. At the end of this step, we obtain the final $C$ clusters and their corresponding minimum proximity requirements $\{min(\Delta_u^c),\ \forall c \in C\}$. It is important to note that $|C| \leq min(|\hat{C}|, P_u)$ in all cases.

VOs in each cluster $c \in C$ can now be considered as an aggregate entity, in effect simplifying network management during inter-datacenter bulk live migrations for user mobility support.

**Algorithm 1** Cluster Migration at Time Instant $t$

**In:** $C$, $\{min(\Delta_u^c),\ \forall c \in C\}$, $\{D_u^c(t-1),\ \forall c \in C\}$, $\mathbf{u} \mapsto \mathbf{ac}(t)$
$\{L_c,\ \forall c \in C\} \leftarrow distances(\mathbf{u}, \{D_u^c(t-1)\})$
$\{D_u^c(t),\ \forall c \in C\} \leftarrow \{\}$
**for** $c \in C$ **do**
  **if** $L_c > min(\Delta_u^c)$ **then**
    $S \leftarrow shortestpath(\mathbf{ac}(t), D_u^c(t-1))$
    $L_S \leftarrow distances(\mathbf{u}, S)$
    **for** $i = 0$ to $|S| - 1$ **do**
      **if** $L_S(|S| - i) \leq min(\Delta_u^c)$ **then**
        $D_u^c(t) \leftarrow S(|S| - i)$
        **break**
      **end if**
    **end for**
  **else**
    $D_u^c(t) \leftarrow D_u^c(t-1)$
  **end if**
**end for**
**Out:** $\{D_u^c(t)\}$

### B. Cluster Migration

In this work, we assume that each datacenter $d \in D$ has enough resources for hosting VOs, focusing on the QoS improvement achieved by allowing VO clusters to "move with the user," when necessary. Particularly, as user $u$ moves around throughout the day – e.g., from home to work or to do some errands, etc., and then, back home – some of the clusters' proximity requirements may be violated at some point, necessitating migrations in order to keep the desired QoS.

Suppose that at a time instant $t$, the network detects that $u$'s access point changed from $\mathbf{ac}(t-1)$ to $\mathbf{ac}(t)$, and $\{D_u^c(t-1),\ \forall c \in C\}$ is the previous placement of the clusters (i.e., the datacenter locations that meet $\{min(\Delta_u^c)$, $c = 1, \ldots, C\}$ when $u$ was connected to $\mathbf{ac}(t-1)$). Algorithm 1 summarizes how migrations are initiated at such time instants, given the VO clustering results in Subsection III-A and the shortest-path lengths $\{L_c,\ \forall c \in C\}$ from $u$'s device $\mathbf{u}$, via $\mathbf{ac}(t)$, to the previous placement $\{D_u^c(t-1)\}$, and how the new placement $\{D_u^c(t),\ \forall c \in C\}$ is obtained.

For a given cluster $c$, a migration is only initiated if $L_c$ exceeds $min(\Delta_u^c)$). In such a case, the shortest path $S$ between $\mathbf{ac}(t)$ to $c$'s previous location $D_u^c(t-1)$ is obtained, as well as the corresponding path lengths $L_S$ from each of its hops to $\mathbf{u}$. Starting from the hop closest to $D_u^c(t-1)$, the first one that satisfies $min(\Delta_u^c)$ is chosen as $c$'s new location $D_u^c(t)$.

## IV. Numerical Results

Generally, VOs can be hosted in the Cloud and/or Fog domains; hence, we classify datacenters as: a) cloud (**cl**), b) transit/aggregation (**t/a**) or c) access (**ac**) nodes. Through a simulation framework implemented in Matlab, we consider a scaled-down, city-wide Telco infrastructure with 30 datacenters: 2 are **cl** nodes (e.g., like Telecom Italia's Sparkle nodes in Milan [19]), while the rest are **t/a** and **ac** nodes generated according to the probability mass function

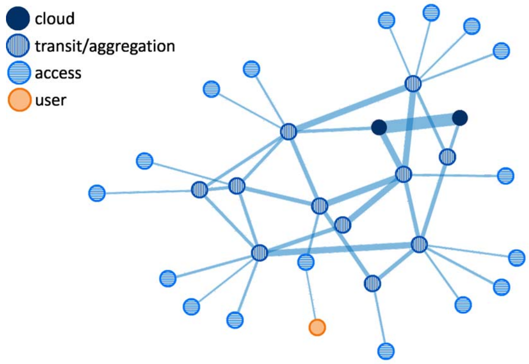



Fig. 3: Graph-based logical topology of a scaled-down city-wide infrastructure.

$\mathcal{P} = \{0.4, 0.6\}$, respectively. The logical interconnections $E$ among these nodes are randomly generated to form a graph $G(D, E)$ – except for the 2 **cl** nodes that are supposed to be part of the nationwide network backbone.

The links interconnecting any pair of datacenters $(d_n, d_m) \in D$, $n \neq m$, are characterized by their corresponding weights $\{w_{(n,m)}\}$. As previously noted, one way to incorporate the latency/bandwidth aspect in a QoS evaluation based on path lengths is by assigning higher weights to high latency/low bandwidth links. With this in mind, the link between the 2 **cl** nodes has weight set to '1', links between **cl** and **t/a** nodes or between 2 **t/a** nodes have weights drawn from the discrete uniform distribution $\mathcal{U}\{2, 4\}$, while those interconnecting **t/a** and **ac** nodes from $\mathcal{U}\{5, 7\}$. Finally, the link between a user $u$'s device **u** and an **ac** node has weight drawn from $\mathcal{U}\{8, 10\}$. Fig. 3 shows an example of a graph-based logical topology generated in such fashion, where the widths of edges decrease with increasing link weights.

We further suppose that a user $u$ has 20 VOs with proximity requirements drawn from $\mathcal{U}\{10, 30\}$ – this range of values is chosen based on the link weights to cover different user proximity cases. Particularly, VOs are uniformly generated such that some require to be on or close to the current **ac** node, some with '*don't care*' proximity, while others are somewhere in between these two extremes. Moreover, the inter-VO affinity is specified in terms of percentage (i.e., 0%, 5% and 10%, in this work), which corresponds to the percentage of VO pairs $(v_i, v_j) \in V_u$, $i \neq j$, with $\delta_{(i,j)} = 0$.

In this evaluation, we first take a look at the behaviour of the number of VO clusters as certain proximity and affinity parameters are varied. Then, considering both static and dynamic user cases, we compare the differences between the required (SLA) and actual path lengths from user $u$ to his/her VOs', with and without migrations. Additionally, for the dynamic user case, we take a look at the number of migrations initiated in terms of VOs and clusters, to provide insights on the network management simplification obtained with cluster migration.

Statistical significance in the results are illustrated through first-order statistics, quartiles and 95% confidence intervals obtained from 20 simulation runs of varying seeds.

### A. Number of Clusters

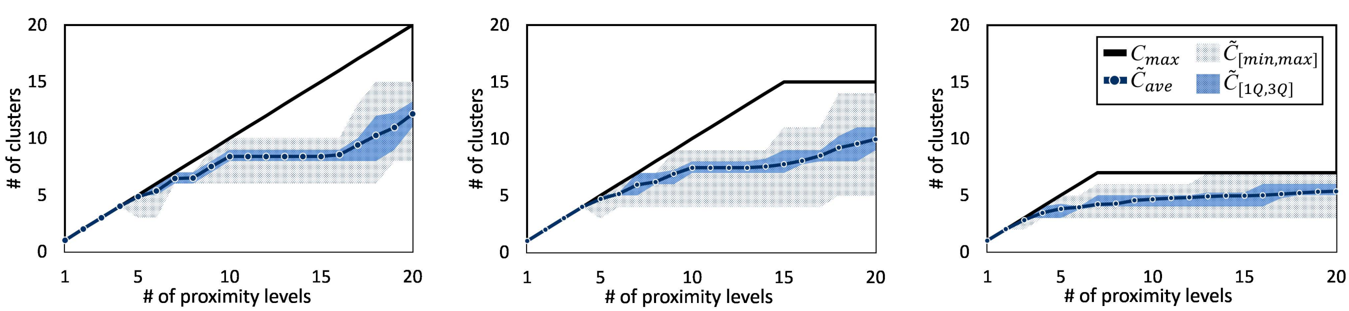
Recall that $C \leq C_{max}$, where $C_{max} = min(\hat{C}, P_u)$. This simply means that the number of clusters obtained neither exceeds the number of initial clusters (defined only by inter-VO affinity) nor the number of proximity levels allowed by the user.

Fig. 4 shows the number of clusters obtained in the simulation runs, indicating first-order statistics ($\tilde{C}_{ave}$, $\tilde{C}_{[min,max]}$) and quartiles ($\tilde{C}_{[1Q,3Q]}$), in comparison with $C_{max}$. A generally stable increasing trend is observed in the number of clusters as the number of proximity levels increases. The impact of inter-VO affinity can also be observed as the $C_{max}$ curves flatten with increasing percentage of VO pairs with affinity among them, generating lesser number of clusters.

### B. Static User

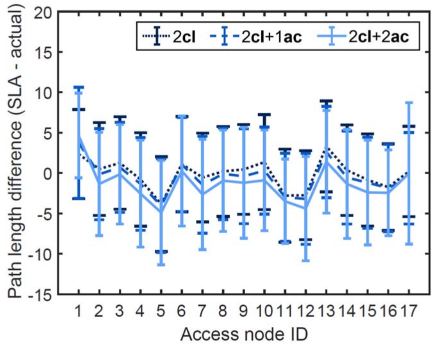
For the static user case, we compare the required (SLA) and actual path length differences when user $u$ accesses his/her VO network via the node **ac**-$x$, $x = 1, \ldots, 17$, with and without migrations, supposing that the VOs are independent of one another.

Without migrations, clusters are randomly placed among: a) 2 **cl** nodes, b) 2 **cl** and 1 **ac** nodes, or c) 2 **cl** and 2 **ac** nodes, to simulate the traditional Cloud scenario and the Cloud-Fog interplay with 1 (e.g., at Home) or 2 (e.g., at Home and at Work) Fog nodes, respectively. Here, we suppose that **ac**-1 is at user $u$'s Home, while **ac**-17 is at Work. Fig. 5a shows that

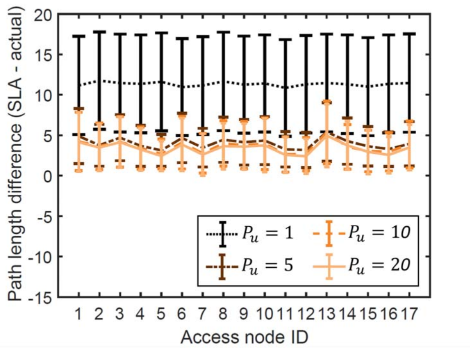



(a) 0% (b) 5% (c) 10%

Fig. 4: Number of clusters for varying number of proximity levels and percentage of VO pairs with affinity among them.

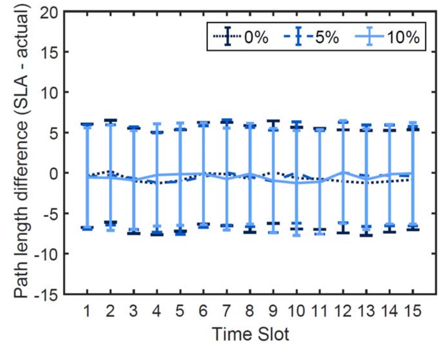



(a) without migration

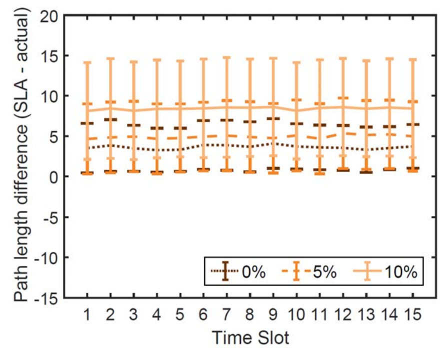



(b) with migration

Fig. 5: Required and actual path length differences with and without migrations for the static user case.




(a) without migration




(b) with migration

Fig. 6: Required and actual path length differences with and without migrations for the dynamic user case.

the traditional Cloud case has generally better performance due to the VOs' central location. Path length improvements in the Cloud-Fog cases are only observed when $u$ is at Home or at Work, with close proximity to VOs placed in node **ac**-1 or **ac**-17. In all three cases, some SLA violations are observed, as indicated by the negative path length differences.

Now, by introducing migrations, SLA specifications are always met, as shown in Fig. 5b. It can also be observed how the subscription-based parameter ($P_u$) impacts the path length improvements. For instance, users with premium subscriptions can invoke $P_u = 1$ so that the network will consider an entire VO network as one cluster that follows its user according to the minimum proximity requirement.

### C. Dynamic User

For the dynamic user case, we compare the required and actual path length differences as user $u$ accesses his/her VO network via the node $\mathbf{ac}(t)$, at time instants $\{t = 1, \ldots, T\}$, with and without migrations. We suppose to have $T = 15$ time slots (e.g., considering 1-hr. granularity from 7:00 to 22:00), during which the user $u$'s access point changed from $\mathbf{ac}(t-1)$ to $\mathbf{ac}(t)$.

In the case of migrations, we fix $P_u = 20$ to maximize the number of clusters and study the impact of inter-VO affinity on the path length improvements. We also take a look at the number of migrations initiated in terms of VOs and clusters to get a hint on how user mobility support can be simplified by considering VO clusters as aggregate entities during migrations.

*1) Path lengths:* In the case of no migrations, we only consider the traditional Cloud case since it generally gave better performance than the other cases, as previously seen. Fig. 6a shows that the actual path lengths do not vary much with the inter-VO affinity since clusters are placed in either of the 2 **cl** nodes anyway, and as before, some SLA violations are observed.

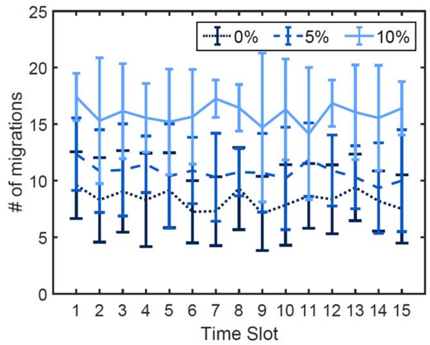
The impact of inter-VO affinity on the path lengths is more evident when migrations are introduced, as illustrated in Fig. 6b. Since lesser number of clusters are generated with increasing percentage of VO pairs with affinity among them, cluster sizes will increase for a given number of VOs. This means that more VOs will be carried over by the same (possibly, tighter) proximity requirement, and hence, the greater path length improvements.

*2) Number of migrations:* Fig. 7 shows the number of migrations generated by the proposed approach, in terms of VOs and clusters, when supporting user mobility.

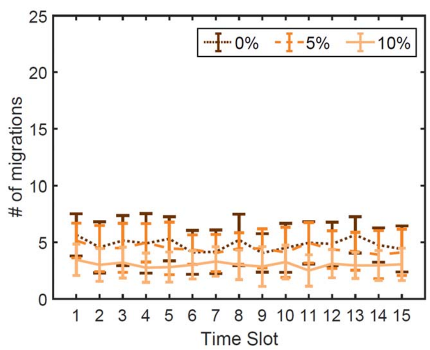
In the case where VOs are independent of one another, considering VO clusters as aggregate entities will initiate around $40\%$ less migrations, on average, and such improvement increases with inter-VO affinity. For instance, when $10\%$ of VO pairs have affinity among them, up to over $80\%$ improvement is achieved.

## V. Conclusion

The upcoming Cloud-Fog interplay is expected to grant service providers more degrees of freedom in the




(a) VO migrations




(b) cluster migrations

Fig. 7: Number of migrations in terms of VOs and clusters for the dynamic user case.

implementation and management of their service portfolios. With the state-of-the-art virtualization technologies, services can be implemented in modular software as a graph/chain of portable VOs that can be dynamically migrated around the Telco infrastructure.

In this perspective, a proximity- and affinity-aware clustering and migration policy for user-centric VO networks is proposed to scalably support user mobility. Additionally, a subscription-based proximity ranging parameter is adopted to allow service differentiation among users. Results show how the number of clusters generated by the policy vary with this parameter and the inter-VO affinity. Some SLA violations are observed with static service placement, and introducing migrations improves the QoS to always meet or exceed the requirements. Moreover, considering VO clusters as aggregate entities will initiate around $40\%$ less migrations, on average – an improvement that increases with inter-VO affinity and could potentially simplify service management when supporting user mobility.

For future work, we would like to extend the policy to cover multiple affinity levels, as well as add constraints on the available capacities among datacenters.

## Acknowledgment

This work was supported by the INPUT (In-Network Programmability for next-generation personal cloUd service supporT) project, funded by the European Commission under the Horizon 2020 Programme (Grant no. 644672).

## References

[1] M. Yannuzzi, F. van Lingen, A. Jain, O. L. Parellada, M. M. Flores, D. Carrera, J. L. Pérez, D. Montero, P. Chacin, A. Corsaro, and A. Olive, "A New Era for Cities with Fog Computing," *IEEE Internet Comput.*, vol. 21, no. 2, pp. 54–67, Mar. 2017.

[2] B. Kim and B. Lee, "Integrated Management System for Distributed Micro-Datacenters," in *Proc. 18th Int. Conf. Adv. Commun. Technol. (ICACT)*, PyeongChang, Korea, Jan. 2016, pp. 466–469.

[3] D. Sabella, A. Vaillant, P. Kuure, U. Rauschenbach, and F. Giust, "Mobile-Edge Computing Architecture: The role of MEC in the Internet of Things," *IEEE Consum. Electron. Mag.*, vol. 5, no. 4, pp. 84–91, Oct. 2016.

[4] "On the Pulse of the Networked Society," Ericsson Mobility Rep., Nov. 2016. [Online]. Available: https://www.ericsson.com/assets/local/mobility-report/documents/2016/ericsson-mobility-report-november-2016.pdf

[5] M. Taneja and A. Davy, "Resource Aware Placement of IoT Application Modules in Fog-Cloud Computing Paradigm," in *Proc. 2017 IFIP/IEEE Symp. Integr. Netw. Service Manag. (IM)*, Lisbon, Portugal, May 2017, pp. 1222–1228.

[6] A. Yousefpour, G. Ishigaki, and J. P. Jue, "Fog Computing: Towards Minimizing Delay in the Internet of Things," in *Proc. 2017 IEEE Int. Conf. Edge Comput. (EDGE)*, Honolulu, HI, Jun. 2017, pp. 17–24.

[7] M. Nitti, V. Pilloni, G. Colistra, and L. Atzori, "The Virtual Object as a Major Element of the Internet of Things: A Survey," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 2, pp. 1228–1240, Q2 2016.

[8] R. Bruschi, F. Davoli, P. Lago, A. Lombardo, C. Lombardo, C. Rametta, and G. Schembra, "An SDN/NFV Platform for Personal Cloud Services," *IEEE Trans. Netw. Service Manag.*, 2017, DOI: 10.1109/TNSM.2017.2761860.

[9] T. Braud, F. H. Bijarbooneh, D. Chatzopoulos, and P. Hui, "Future Networking Challenges: The Case of Mobile Augmented Reality," in *Proc. 37th Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Atlanta, GA, USA, Jun. 2017, pp. 1796–1807.

[10] "Virtualization Essentials," VMware E-book, 2014. [Online]. Available: http://www.vmware.com/content/dam/digitalmarketing/vmware/en/pdf/e-book/gated-vmw-ebook-virtualization-essentials.pdf

[11] T. Taleb, A. Ksentini, and P. Frangoudis, "Follow-Me Cloud: When Cloud Services Follow Mobile Users," *IEEE Trans. Cloud Comput.*, 2017, DOI: 10.1109/TCC.2016.2525987.

[12] N. Tziritas, S. Khan, T. Loukopoulos, S. Lalis, C. Z. Xu, K. Li, and A. Zomaya, "Online Inter-Datacenter Service Migrations," *IEEE Trans. Cloud Comput.*, 2017, DOI: 10.1109/TCC.2017.2680439.

[13] W. Cerroni and F. Esposito, "Optimizing Live Migration of Multiple Virtual Machines," *IEEE Trans. Cloud Comput.*, 2017, DOI: 10.1109/TCC.2016.2567381.

[14] G. Sun, D. Liao, D. Zhao, Z. Xu, and H. Yu, "Live Migration for Multiple Correlated Virtual Machines in Cloud-based Data Centers," *IEEE Trans. Services Comput.*, 2017, DOI: 10.1109/TSC.2015.2477825.

[15] "Open Virtualization Operating Layer for Cloud/fog Advanced NetwOrks (OpenVolcano)." [Online]. Available: http://openvolcano.org

[16] R. Bruschi, F. Davoli, P. Lago, and J. F. Pajo, "A Scalable SDN Slicing Scheme for Multi-domain Fog/Cloud Services," in *Proc. 2017 IEEE Conf. Netw. Softwarization (NetSoft)*, Bologna, Italy, Jul. 2017, DOI: 10.1109/NETSOFT.2017.8004244.

[17] "Network Functions Virtualisation (NFV); Management and Orchestration; Vi-Vnfm Reference Point – Interface and Information Model Specification," ETSI NFV ISG Spec., 2016. [Online]. Available: http://www.etsi.org/deliver/etsi_gs/NFV-IFA/001_099/006/02.01.01_60/gs_NFV-IFA006v020101p.pdf

[18] X. Meng, V. Pappas, and L. Zhang, "Improving the Scalability of Data Center Networks with Traffic-aware Virtual Machine Placement," in *Proc. 2010 Int. Conf. Comput. Commun. (INFOCOM)*, San Diego, CA, USA, Mar. 2010, DOI: 10.1109/INFCOM.2010.5461930.

[19] "Sparkle. The World's Communication Platform." [Online]. Available: http://www.tisparkle.com/