Human-enabled Edge Computing: When Mobile Crowd-Sensing meets Mobile Edge Computing

Luca Foschini[1], *Michele Girolami*[2]
[1]*Dipartimento di Informatica: Scienza e Ingegneria (DISI), University of Bologna, Italy*
[2]*ISTI-CNR, Pisa, Italy*
luca.foschini@unibo.it, michele.girolami@isti.cnr.it

1. Introduction

MEC is an architectural model and specification proposal (i.e., by European Telecommunications Standards Institute - ETSI) that aims at evolving the traditional two-layers cloud-device integration model, where mobile nodes directly communicate with a central cloud through the Internet, with the introduction of a third intermediate middleware layer that executes at so-called network edges. This promotes a new three-layer device-edge-cloud hierarchical architecture, which is recognized as very promising for several application domains [1]. In fact, the new MEC model allows moving and hosting computing/storage resources at network edges close to the targeted mobile devices, thus overcoming the typical limitations of direct cloud-device interactions, such as high uncertainty of available resources, limited bandwidth, unreliability of the wireless network trunk, and rapid deployment needs.

Although various MEC solutions based on fixed edges enable an increase of the quality and performance of several cloud-assisted device services, currently there are still several non-negligible weaknesses that affect this emerging new model. First, the number of edges is generally limited because edges are deployed statically (usually by telco providers) and their configuration and operation introduce additional costs for the supported services, such as deployment, maintenance, and configuration costs. Second, once deployed, edges are rarely re-deployed (due to the high re-configuration cost) in other positions and this might result in high inefficiency, e.g., as service load conditions might significantly change dynamically. Finally, some geographical areas might become interesting hotspots for a service only during specific time slots, such as a square becoming crowded due to an open market taking place only at a specific timeslot and day of the week.

At the same time, the possibility to leverage people roaming though the city with their sensor-rich devices has recently enabled Mobile Crowd-Sensing (MCS). In fact, by installing an MCS application, any smartphone can become part of a (large-scale) mobile sensor network, partially operated by the owners of the phones themselves. However, for some high-demanding MCS applications (e.g., a surveillance service that, for security purposes, monitors an environment with smartphone cameras that capture photos/videos of the surroundings and exploits face recognition to trace suspicious users' movements), regular smartphones often have not enough capabilities to timely perform the requested local tasks, in particular if considering their possible immersion in hostile environments with possible frequent intermittent disconnections from the global cloud.

In other words, we claim that there are several practical cases of large and growing relevance where the joint exploitation of MEC and MCS would bring highly significant benefits in terms of efficient resource usage and perceived service quality. However, notwithstanding recent advances in both MEC and MCS, to the best of our knowledge, only a very limited number of seminal works has explored the mutual advantages in the joint use of these two classes of solutions, and they are mostly focused on pure technical communication aspects without considering the crucial importance of having humans as central contributors in the loop [2, 3, 4].

The paper reports some research ideas and findings in a brand new area that we call Human-driven Edge Computing (HEC) defined as a new model to ease the provisioning and deployment of MEC platforms as well as to enable more powerful MEC-enabled MCS applications. First and foremost, *HEC eases the planning and deployment of the basic MEC model*: it mitigates the potential weaknesses of having only Fixed MEC entities (FMEC) by exploiting MCS to continuously monitor humans and their mobility patterns, as well as to dynamically re-identify hot locations of potential interest for the deployment of new edges. Second, to overcome FMEC limitations, *HEC enables the implementation and dynamic activation of impromptu and temporary Mobile MEC entities (M^2EC)* that leverage resources of locally available mobile devices. Hence, a M^2EC is a local middleware proxy dynamically activated in a logical bounded location where people tend to stay for a while with repetitive and predictive mobility patterns [5], thus realizing a mobile, opportunistic, and participatory edge node. Third, given that M^2EC, differently from FMEC, does not implement powerful backhaul links toward the core cloud, *HEC exploits local one-hop communications* *and the store-and-forward principle by using humans* (moving with their devices) *as VM/container couriers* to enable migrations between well-connected FMEC and local M^{2}EC.

2. Boosting Mobile Edge Computing through Human-driven Edge Computing

We refer to the scenario shown in Fig. 1. It extends the usual three-layer device-MEC-cloud hierarchical architecture (based on the interposition of FMEC entities) with the addition of the new M^{2}EC entity. Indeed, the MCS approach combined with the seamless tracking of volunteers (monitoring both their mobility and their performance in terms of completion rates of assigned sensing tasks) allows to: i) identify the optimal locations where people tend to interact. Such locations ease the effective deployment of FMEC and M^{2}EC. Furthermore, it allows to ii) select of those users willing to host M^{2}EC. Such users act as local access points to the hierarchical HEC.

We experienced with the ParticipAct MCS living lab [6] in order to clarify the effectiveness of architecture proposed. We learned from ParticipAct that some locations aggregate people during all the day (such locations are indeed ideal candidates for the FMEC, see E$_1$, E$_2$, and E$_3$ in Fig. 1). At the same time some locations become *active* only during shorter and different timeslots (e.g., P$_1$ and P$_4$ from 9:00AM to 10:30AM, while P$_2$ is frequented only from 4:00PM to 6:00PM). These latest areas, out of the highly frequented people paths, would highly benefit of being served by a local (in time and space) M^{2}EC, while it would be inefficient and overprovisioned to have additional FMEC there (see Fig. 1).

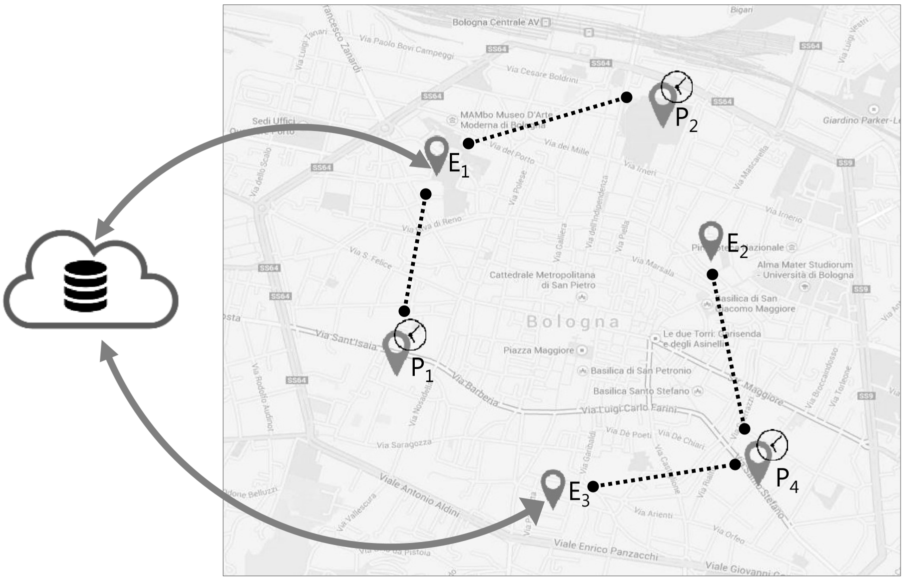


Fig. 1: FMEC, M^{2}EC, and couriers in our HEC model.

Another interesting aspect we learned from the ParticipAct MCS living lab concerns HECs. They exploit opportunistic interactions among devices in order to enable the migration of Virtual Machines (VM)/containers. This feature can be achieved by leveraging human couriers moving from/to different FMECs (see Fig. 1) [7]. In our reference architecture, devices can interact through one-hop *ad-hoc* communications. Such interactions are possible by using short-range network interfaces, such as Bluetooth (i.e., up to 25m), Wi-Fi configured in direct mode (i.e., up to 150m) or the LTE-direct technology (i.e., up to 500m).

Similarly to the paradigm adopted with the MSN, *courier* devices automatically down/upload VM/containers from the FMEC as soon as they are close enough to another device in order to transfer data. In turn, devices can share data gathered from other devices roaming in the same M^{2}EC (see dotted lines in Fig. 1). Refer to Section 3 for the selection criteria of the most suitable human couriers.

Without claiming completeness and due to space limitations, in the following we briefly overview the current state-of-the-art in the main related fields. Focusing on architectural aspects of HEC, the MEC/fog literature has already produced some relevant modeling work and some seminal design/implementation results. Narrowing to efforts close to ours, as reported in [1], some first exploratory research activities have considered cooperation issues between edges and the core, but only a very few works concentrated on the opportunities of having cooperation between devices and the edges. Considering MCS as application scenario, [2] and [3] propose to enhance the MCS process by leveraging intermediate MEC nodes, namely, FMECs, to boost data upload from mobile nodes to the infrastructure [2] and to provide more computing/storage capabilities closer to end mobile devices [3]. A very recent and interesting work, the closest to our HEC concept for what relates to enabling more collaboration between entities co-located at edges is [4]: it proposes not only to have the traditional "vertical" collaboration between devices, MEC, and cloud level, but also an "horizontal" collaboration between entities at the same level via ad-hoc communications; however, it neglects humans and social/mobility effects, namely, there is no idea to dynamically identify and impromptu form M^{2}ECs as in our novel HEC proposal.

Finally, concerning the system and the implementation aspects, only a very few research activities are focused on the migration of VM/container on MEC middleware for mobile services over hostile environments. It is worth to notice that such activities are relevant aspects of modern CPS. Authors of [8] highlight the limitations of traditional live VM migration based on edge devices. They propose a live migration approach in response to client handoff in cloudlets, with less involvement of the hypervisor and, at the same time, by promoting migration to optimal offload sites. Authors also discuss how to adapt the system to the changing network conditions and processing capacity. The work described in [9] presents the *foglets* programming infrastructure. Such infrastructure handles some mechanisms for quality/workload-sensitive migration of service components among fog nodes. Another interesting work is reported in [10]. It proposes the usage of *cloudlets* to support mobile multimedia services and to adjust the resource allocation triggered by runtime handoffs. Concerning the handoff evaluation, the authors of [11] study the handoff conditions in relation to various aspects such as signal strength, bit rate, number of interactions between cloudlets and associated devices. Finally, [12] proposes a multi-agent-based code offloading mechanism. It adopts a reinforcement learning and code blocks migration in order to reduce both execution time and energy consumption of mobile devices. To the best of our knowledge, these papers explore the integrated management of handover operations with VM/container migration. However, none of them considers the possibility of exploiting peoples' devices as storage/VM/container couriers.

3. Mobile Edge Computing extended through the Crowd

We first overview the HEC architecture as well as its main components and functionalities. Then, we present some guidelines and engineering tradeoffs for the selection of FMEC, M^{2}EC, and of the human couriers.

3.1. The Reference Architecture of the HEC Middleware

HEC extends the emerging MEC three-layer hierarchical architecture. In particular, we consider two types of MECs, namely FMEC and M^{2}EC. By focusing on our HEC middleware at mobile devices, we distinguish between regular mobile devices (capable of working only as service clients) and powerful devices (which may be promoted dynamically to host virtualized functions and to serve as M^{2}EC nodes). In our current implementation, we identify a number of powerful devices based on the hardware and software features (ie. tablets or laptops that are locally paired with smartphones). It is worth to notice that the evolution trend of mobile/embedded devices is such that the potential set of mobile nodes that can be promoted to M^{2}EC at runtime is ever increasing. Under this respect, some interesting benchmarks show that also RaspberryPI boards can adequately run OpenStack++ middleware [13]. We consider that our HEC middleware is already installed on such nodes before starting the provisioning of services, even if more sophisticated dynamic mechanisms for HEC middleware download at runtime can be easily integrated. Our HEC middleware implementation fits a wide spectrum of heterogeneous mobile devices, with the only constraint to run Android (iOS version currently under development).

For what concerns the MCS applications, we consider that only highly demanding or group-oriented locality-based MCS tasks are delegated to FMEC and M^{2}EC nodes, possibly based on dynamic considerations (e.g., residual battery energy). At this stage, the MCS tasks that have been already implemented and experimented for execution at HEC nodes are i) video analysis for face recognition and ii) analytics on all or fused monitoring indicators over geographical areas of highest interest and density such as data fusion, history-based processing of temporal series.

3.2. The selection of FMEC and M^{2}EC and Human-enabled VM/Container Migration

Our architecture is configured with a number of FMEC and M^{2}EC. They are selected by analyzing the human mobility over an observation period. Concerning the FMECs, we consider those locations remaining mostly *active* during the whole day. These are locations not subject of mobility changes. To this purpose, we use the DBSCAN algorithm in order to detect clusters of users roaming around the same location [6, 14]. DBSCAN returns K distinct clusters, we filter out some of them, in particular we restrict to $k \leq$ K clusters as FMECs.

The M^{2}EC selection is achieved by spotting those locations of our region becoming *active* only during specific time slots. In fact, our goal is dynamically (re)configure our cloud architecture according to the natural rhythm of a city. To this purpose, we analyze the human mobility only during some temporal slots. In particular, we select those slots characterizing the typical phases of a routinary working day. For each of the slots, we cluster together the positions of the users with a process similarly to the one described for the FMEC. Also for the M^{2}EC selection, we adopted the DBSCAN algorithm. It results with H clusters of which we keep the top $h \leq$ H.

Our process allows setting the parameters k and h according to the mobility and sociality features of the mobility dataset considered. Specifically, a small crowded region can be provisioned with a low number of FMEC, but with a high number of M^{2}EC since crowded area change quickly along the day. Conversely, in a wide depopulated area it could be possible to increase the number of FMEC and, at the same time, reducing the number of M^{2}EC, since mobility changes slowly with the time.

Once FMEC to M^{2}EC are selected, we then consider how to move that among them. To this purpose, we consider humans (i.e., *couriers*), and their mobile devices provisioned with our HEC middleware, as the primary actors that can be involved into the loop. We assume that mobile devices are equipped with different kinds of network interfaces (short, medium and broadband) and of storage capacity. The storage allows devices to store-carry-and-forward data among FMEC and M^{2}EC, as well as it allows replicating data across users joining at the same time the same M^{2}EC For the selection of couriers, we keep track of user mobility and prefers those users that have a more repetitive and predictable behavior: the more a user commutes from a FMES to a M^{2}EC, the more he/she is a good courier candidate.

Since not all the FMECs are connected to all M^{2}ECs during the 24 hours, we consider the possibility of reducing the bandwidth in the cloud-to-FMEC direction and consequently the storage resources at FMECs. To this purpose, the HEC implements a *load balancing* policy. Such policy exploits the knowledge of the mobility and of the connectivity between FMECs and M^{2}EC in order to select which VMs/containers requires to be moved move from the cloud to the FMECs. The load balancing strategy relies on the locality principle according to which VMs/containers are loaded in advance to those FMECs that are more likely to be store-and-forwarded by a courier toward a M^{2}EC.

Also for the sake of briefness and due to paper length limitations, further design/implementation details about our HEC proposal are not reported here because out of the central scope of this paper, which presented the vision and the main design guidelines of our innovative HEC solution. At the current stage, we are working in order to test these ideas through a set of experiments based on the real-world ParticipAct dataset which reproduces the mobility of about 170 students in the Emilia Romagna region (Italy) about 2 years [6].

4. Conclusion

This paper presented HEC, a new architecture model to ease the provisioning and to extend the coverage of traditional MEC approaches by bringing together the best of MEC and MCS. The cornerstone of our proposal lies in the ability to dynamically leverage human sociality and mobility effects to broaden the MEC coverage through the impromptu formation of M^{2}ECs. Those encouraging results are pushing us to further investigate and refine our HEC model and we are currently exploring various related areas. On the one hand, we are working to enable the self-adaptable fine tuning of our HEC middleware to the different dynamics and variations of the city pulse, for instance to the different behaviors that might present along the year, such as working vs. vacation periods, and the week, such as working days vs. weekends. On the other hand, we are investigating innovative techniques in order to reduce the latency of downloading VMs/containers on M^{2}EC nodes via parallelization of I/O and configuration operations.

References

[1] S. Wang et al., “A Survey on Mobile Edge Networks: Convergence of Computing, Caching and Communications”, IEEE Access, vol. PP, no. 99, pp.1-1
doi:10.1109/ACCESS.2017.2685434.

[2] S. K. Datta, R. P. Ferreira da Costa, C. Bonnet and J. Härri, “oneM2M architecture based IoT framework for mobile crowd sensing in smart cities”, in Proceedings of 2016 European Conference on Networks and Communications (EuCNC), 2016, pp. 168-173.

[3] K. M. S. Huq, S. Mumtaz, J. Rodriguez, P. Marques, B. Okyere and V. Frascolla, “Enhanced C-RAN Using D2D Network”, IEEE Communications Magazine, vol. 55, no. 3, pp. 100-107, March 2017.

[4] T. X. Tran, A. Hajisami, P. Pandey and D. Pompili, “Collaborative Mobile Edge Computing in 5G Networks: New Paradigms, Scenarios, and Challenges”, IEEE Communications Magazine, vol. 55, no. 4, pp. 54-61, April 2017.

[5] Pappalardo, L.; Simini, F.; Rinzivillo, S.; Pedreschi, D.; Giannotti, F.; Barabási, A.L., “Returners and Explorers dichotomy in Human Mobility”, Nature Communications, vol. 6, Article 8166, 2015.

[6] G. Cardone, A. Cirri, A. Corradi, L. Foschini, “The ParticipAct Mobile Crowd Sensing Living Lab: The Testbed for Smart Cities”, IEEE Commununications Magazine, vol. 52 , no. 10, pp. 78–85, October 2014.

[7] M. Girolami, S. Chessa, A. Caruso, “On Service Discovery in Mobile Social Networks: Survey and Perspectives”, Computer Networks, vol. 88, pp. 51-71, 2015.

[8] K. Ha et al., “Adaptive VM Handoff Across Cloudlets”, Technical Report CMU-CS-15-113, CMU School of Computer Science, 2015.

[9] E. Saurez et al., “Incremental Deployment and Migration of Geo-Distributed Situation Awareness Applications in the Fog”, in Proceedings of ACM Distributed and Event-based Systems Int. Conf., pp. 258-269, 2016.

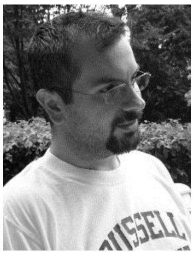
[10] M. Felemban, S. Basalamah, A. Ghafoor, “A Distributed Cloud Architecture for Mobile Multimedia Services”, 2013.

[11] A. Ravi, S.K. Peddoju, “Handoff Strategy for Improving Energy Efficiency and Cloud Service Availability for Mobile Devices”, Wireless Personal Communications, vol. 81, no. 1, pp. 101–132, 2015.

[12] M.G.R. Alam, et al., “Multi-agent and Reinforcement Learning Based Code Offloading in Mobile Fog”, in Proceedings of International Conference on Information Networking (ICOIN), 2016.

[13] P. Bellavista, M. Solimando, A. Zanni, “A Migration-enhanced Edge Computing Support for Mobile Services in Hostile Environments - Lessons Learnt from Platform Implementation and Deployment”, to appear in Proceedings of International Wireless Communications and Mobile Computing Conference, pp. 1-6, 2017.

[14] M. Ester, H. peter Kriegel, J.S, X. Xu, “A density-based algorithm for discovering clusters in large spatial databases with noise”, in Proceedings of International Conference on Knowledge Discovery, AAAI Press, pp. 226–231, 1996.



Luca Foschini graduated from the University of Bologna, Italy, where he received a Ph.D. degree in computer engineering in 2007. He is now an assistant professor of computer engineering at the University of Bologna. His interests include distributed systems and solutions for system and service management, management of cloud computing, context data distribution platforms for smart city scenarios, context-aware session control and adaptive mobile services, and mobile crowd sensing and crowdsourcing.

Contact him at luca.foschini@unibo.it.

Michele Girolami graduated from the University of Pisa, where he received a Ph.D. degree in computer science in 2015. Currently, he is a member of research staff at CNR-ISTI with the Wireless Network Laboratory. His research interests include data forwarding and service discovery in mobile social networks, crowd sensing techniques, and context-aware middleware for smart environments.

Contact him at michele.girolami@isti.cnr.it.