



HAL
open science

Microcities: a Platform based on Microclouds for Neighborhood Services

Ismael Cuadrado Cordero, Felix Cuadrado, Chris Phillips, Anne-Cécile Orgerie, Christine Morin

► **To cite this version:**

Ismael Cuadrado Cordero, Felix Cuadrado, Chris Phillips, Anne-Cécile Orgerie, Christine Morin. Microcities: a Platform based on Microclouds for Neighborhood Services. ICA3PP 2016 - International Conference on Algorithms and Architectures for Parallel, Dec 2016, Granada, Spain. pp.192-202, 10.1007/978-3-319-49583-5_14. hal-01362532

HAL Id: hal-01362532

<https://inria.hal.science/hal-01362532>

Submitted on 8 Sep 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Microcities: a Platform based on Microclouds for Neighborhood Services

Ismael Cuadrado-Cordero¹, Felix Cuadrado², Chris Phillips²,
Anne-Cécile Orgerie³, and Christine Morin¹

¹ Inria, France,

Email: {ismael.cuadrado-cordero,christine.morin}@inria.fr

² Queen Mary University of London, UK

Email: {felix.cuadrado,chris.i.phillips}@qmul.ac.uk

³ CNRS, IRISA, France

Email: anne-cecile.orgerie@irisa.fr

Abstract. The current datacenter-centralized architecture limits the cloud to the location of the datacenters, generally far from the user. This architecture collides with the latest trend of ubiquity of Cloud computing. Distance leads to increased utilization of the broadband Wide Area Network and poor user experience, especially for interactive applications. A semi-decentralized approach can provide a better Quality of Experience (QoE) in large urban populations in mobile cloud networks, by confining local traffic near the user while maintaining centralized characteristics, running on the users and network devices. In this paper, we propose a novel semi-decentralized cloud architecture based on microclouds. Microclouds are dynamically created and allow users to contribute resources from their computers, mobile and network devices to the cloud. Microclouds provide a dynamic and scalable system without an extra investment in infrastructure. We also provide a description of a realistic mobile cloud use case, and its adaptation to microclouds.

1 Introduction

The wide uptake of Cloud architectures have caused global IP traffic to increase five fold [1], catalyzed by the ubiquity of mobile devices. According to Cisco, global IP traffic is envisioned to increase threefold over the next five years, with mobile wireless traffic exceeding wired traffic by 2016 [1]. This adds to the limitations of mobile devices in terms of resources and connectivity [2], bringing new challenges to the Cloud. In order to address these issues, a new paradigm is emerging: Mobile Cloud computing. This paradigm improves resource-hungry mobile services such as Internet data sharing [3], wearables [4] or augmented reality [5], by offloading data and computation into the Cloud [2].

This offloading of computation requires high-speed connectivity between the clients and the Cloud. However, highly geographically-centralized Cloud architectures cannot properly handle it. Fernando et al. argue that considering access fees, latency, bandwidth and energy consumption of wireless connectivity, a Mobile local Cloud - constrained to the location of the user - is a better alternative

than a remote one [2]. Local Clouds also provide locality-awareness and support for latency-critical interactions, thus a better Quality of Experience (QoE).

In this paper, we go a step further into Mobile Cloud computing by integrating the mobile devices themselves into the Cloud. Our proposed architecture introduces a flexible, semi-decentralized and yet efficient solution for locality-related applications. We build a local cloud on top of static private (i.e. computers) and public infrastructures (i.e. networking equipment); and the users' mobile devices spread across a defined area. This local cloud is managed by lightweight mechanisms which handle the dynamism of users who can appear/disappear and move. We eliminate the need for dedicated infrastructures, i.e. datacenters, and provide a dynamic environment where multiple services coexist.

Our contributions are: 1) a semi-decentralized mobile cloud architecture based on microclouds and 2) a realistic mobile cloud use case for smart cities.

The remainder of this paper is structured as follows. Section 2 provides the context and motivation and establishes the case study. In Section 3 we describe our design and implementation. Finally, Section 4 highlights our conclusions and draws directions to future work.

2 Motivation and Scenario

2.1 Context and Motivation

The use of centralized datacenters is, nowadays, the most realistic approach regarding the deployment of heavy computation services. This architecture relies on a robust communication infrastructure between distant clients, obtaining a computing power otherwise unattainable. However, centralized architectures suffer downsides such as traffic delays and scalability issues related with the physical constraints of datacenter resources. Moreover, it forces other actors involved in the communication (such as Internet Service Providers) to oversize their infrastructure in consequence [6].

As studied in [7], [8] and [9], information propagation in real-life is usually not distant. This is because interactions in the Internet are conditioned by our interactions as a society. Groups of interest are generally geographically constrained. This is the case with sports teams (whose fans are generally located in the same area), departments in a company (whose users are located in the same building) or Geographic Information Systems [10].

This situation is, by design, approached by Cloud computing, as it is described as a versatile and ubiquitous system. However, in reality, Cloud platforms run on large centralized datacenters, which provide the needed infrastructure. Yet, since existing infrastructures cannot effectively host ever increasing demands, datacenters need to be expanded, which is costly and requires advanced planning. In fact, this situation has already been anticipated by Cloud providers, which started to balance the connection between datacenters around the globe.

The use of centralized systems in users' inclusive - citizens are both providers and consumers of information - and non-heavy computation services with low

propagation is, thus, inappropriate. A distributed approach, where the flow of information is not only produced and consumed, but also processed in the same area is, in this case, desirable. The main advantages of distributed approaches in this scenario are low latency, scalability, and adaptability. This kind of approach fits the mobile clouds created in smart cities initiatives all around the world (Santa Cruz, Amsterdam, Barcelona, etc.) where local authorities deploy wireless platforms to manage traffic or emergency situations.

Even if the proportion of geographically constrained traffic is difficult to estimate, traffic characterization in different areas can be found in literature. For example, in [11] the authors evaluate the consumed and generated traffic in a rural African village. In this scenario, the authors show that most of the generated and consumed traffic is of a local scope, with web and social networking services being the most utilized ones. In addition, in [12] the authors characterize usage of a freely available outdoor wireless Internet in California (USA). Their results show a peak of smartphone connections in transit areas, while in residential areas the connections are more balanced between static and mobile. Commercial areas show a higher activity than either transit or residential areas.

In this paper, we propose a semi-decentralized Cloud architecture for localized communities such as neighborhoods. We propose a mobile Cloud case study based on smart cities initiatives combining mobile and static devices - other existing Cloud infrastructures provide either a mobile peer-to-peer network [3] or local clouds with connectivity to remote cloud servers [13]. We expect that, through the use of microclouds, latency is reduced - compared to centralized systems - while providing a robust, elastic and adaptive platform of services. We also expect that traffic in datacenters, and network providers' broadband utilization and transit costs are reduced.

2.2 Related Work

Two main approaches in the literature are of interest to the case study.

Centralized Clouds are based on a specific infrastructure to which all clients connect. An issue commonly attributed to these architectures is the excessive distance between clients and the computation infrastructure. Adaptability is also a problem, because once the infrastructure has been allocated, it is expensive and complicated to extend. To provide a service "closer to the user", which is handles better the increase in demand of computation, Cloud providers - such as Google [14] or Amazon [15] - disperse their datacenters around the globe.

Centralized local clouds reduce the distance between the user and the infrastructure. In [16], the authors describe an environment where mobile devices outsource their computation to a Cloudlet - domestic servers which provide cloud services to a relatively small set of users. It reduces the computational load of the devices, the replication of data, and the delay, and provides a service adapted to users' needs. However, CloudLets require a, sometimes, prohibitive investment (which rockets if its computational resources need to be scaled), they are rigid and the dependence on the infrastructure ties the user to the system. Since

CloudLets are motivated by the need of domestic solutions in relatively dynamic environments, they are not easily integrated with external infrastructures. Moreover, when a users connect to a CloudLet from the outside - for instance from a workplaces - the communication is done through the Internet. This situation causes the same problems than datacenter centralized solutions.

Distributed Clouds are not deployed on a single infrastructure but among several independent nodes, providing a more robust and adaptable system compared to a lone infrastructure that offers a single point of failure. Representative examples of distributed cloud systems are Content Distribution Networks (CDNs) [17] and Peer-To-Peer (P2P) Clouds [18], [19] - systems where individual computers are distributed across the globe and connected through the Internet hosting VMs for parallel computation, content distribution or storage. A thorough review of distributed technologies is shown in [20]. One of the more praised benefits of distributed approaches is the low latency experienced by users.

Distributed approaches also have disadvantages, such as network restrictions (they are rather network demanding) and excessive replication of content. As a solution, the union of distributed and centralized architectures (semi-decentralized) is also covered in literature. In [21], authors propose the replication of data in the Internet Provider's datacenter in order to reduce latency and energy consumption. However, this system still suffers from higher delay than a totally centralized approach, in addition to an excessive replication (for example, in between users connecting to different providers).

2.3 Scenario: Neighborhood Services

Neighborhood-related applications are a good example of geographically localized services. In a neighborhood, many services are only of interest to the community, like street works, water or electricity cuts or local store information (stocks, opening hours, etc.). These networks are heterogeneous - comprising both mobile and fixed nodes provided by both citizens and city infrastructures that serve up and consume information [22]. Social networks such as [23] and [24] - where users share information of interest only with their neighbors - are examples of neighborhood-oriented applications. A system appropriate for this environment should adjust to the following characteristics:

Consistency: Data can be classified in *Announcements*, which are immutable and distributed through the network to inform of an event; and *collaborative/interactive information* (shared documents, forums, etc.), which have several contributors and are prone to version conflicts and extensive replication in non-centralized systems (broadband utilization in multipurpose networks is very dynamic, and information may flood the network if not handled correctly). Therefore, no replication of data and/or management is allowed, as it saturates the network and provides poor consistency in very dynamic situations. This characteristic automatically discards distributed Clouds defined in 2.2 from the possible solutions, given that those require replication of data as a design principle.

Adaptability and Virtualization: The workload inside a neighborhood is very dynamic, specially when many clients are connected through mobile devices. Also, different neighborhoods show differences in communication patterns and infrastructural support (for example, in residential neighborhoods the number of active users would boost out of working hours, on the contrary to business ones). A suitable platform should dynamically adapt its topology to the characteristics of the neighborhood. Furthermore, adaptability is linked with virtualization. Virtualization allows migration and robustness and enhances the *isolation* and *coexistence* of different services in different neighborhoods.

Versatility: The main characteristic of the proposed use case is the heterogeneity of its components. They are either private (laptops, smartphones, domestic servers and routers, etc.) or public (ISP's routers, smart-city infrastructure, etc). The common characteristic of all these components is that they share an application layer, and thus services can be run on them [25].

Network Orientation, Locality Awareness and Asynchronous Communication: Since a neighborhood exists within a relatively small geographical area, the system should take into account the physical location of the nodes to obtain the best possible utilization of the network. Knowing the location of the nodes, the platform is able to manage *information dissemination*, adapting to the *available bandwidth*. Due to the interactive nature of *real-time information*, disruptive behavior (asynchronous communication) is not desirable.

3 Our Solution: Microcities

We propose a semi-decentralized platform oriented to the use of shared services in Mobile Clouds. Our platform extends the concept of microcloud proposed in [26] in order to support interactivity of multiple services with no replication in very dynamic environments. A microcloud is an overlay topology that connects independent users employing the same service. For each service a microcloud is created. As a consequence, multiple microclouds coexists (and may overlap) on the same network. For each microcloud, a Light Virtual Machine or LVM - a type of operating-system-level virtualization which runs the service is hosted in one of the involved devices. Using the LVM only one copy of each service is kept, avoiding conflicts. On the other hand, while microclouds distribute the computation across the network. Our solution makes use of existing resources and does not require extra infrastructure or investment, such as a datacenter.

3.1 Architecture Design

Figure 1 describes our use case. In Figure 1a, all devices participating in the platform are represented. A public infrastructure device is any device belonging either to the city council or a private company which offers public access. It includes network devices - such as datacenter routers or domestic routers rented to its users - and specific purpose hardware - such as smart-cities infrastructures. Private devices may be static - such as laptops or personal computers - or mobile

- such as smartphones or tablets. In Figure 1b, Microcloud A and B represent different services, overlapping on the same network and, sometimes, on the same devices. We distinguish the following roles, all transparent to the end user.

Base Manager (BM): This role is taken by a resilient and trustworthy device, to manage all microclouds in a given neighborhood (including the creation/deletion of microclouds and failure management). It can request microclouds to change their topologies, if it determines that one or more microclouds can be better arranged (or they can be merged/split). It is also used as a service repository. In each neighborhood, the BM is a unique global service.

Service Manager (SM): Controls the topology, distributes other roles and manages failures in one microcloud. This role is assigned by the base manager depending on the nodes' computing capacity and robustness. Only one service manager exists per microcloud, and it is assigned to a stable, static node.

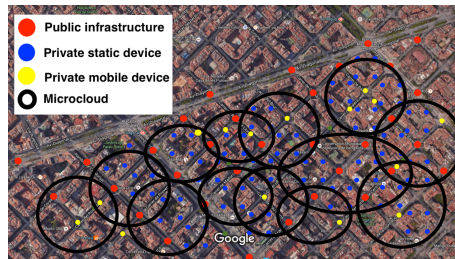
Service Provider (SP): Hosts the LVM in a microcloud. This role is assigned by the SM based on the minimum delay between the service provider and every client, reliability in time and hardware capabilities, and it is unique in the microcloud. It is also preferably assigned to a stable, static node. When the SM detects a more efficient position for LVM, both LVM and SP role are migrated.

Client: Consumer of information. Before using the service, the client starts the join process described in Section 3.2.

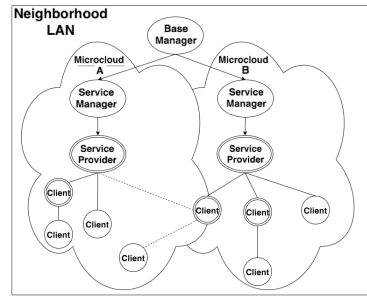
A physical node may participate in two or more microclouds, holding different roles in each one. No assumption has been made about the communication links used between nodes. Therefore, a node may be either static or mobile.

3.2 Join and Detach Processes

Join/rejoin process is depicted in Figure 2a. When a user launches the client service, it contacts the BM and an API lists all the available services. Then, the user selects the service to be joined and the process starts. When the client needs to rejoin, due to detachment, reconfigurations triggered by the *BM* or the



(a) Microclouds operating within a neighborhood (Google Maps)



(b) Logical distribution of microclouds

Fig. 1: Neighborhood overlay services (microclouds)

mobility of the client, the same process occurs. The process is the same, but is initiated without a user request. This process uses the DEEPACC protocol presented in [26] to find the fastest path between the client and the *SP*.

1. The client obtains the address of the *SM* through a request to the base manager. After that, a join request is sent towards the *SM*, which answers with the address of the *SP*. Then, the client runs the DEEPACC protocol.
2. Best route: In DEEPACC protocol, a discovery message is sent through every possible path. In every node, the message is captured, processed and updated with the current Round-Trip Time (RTT); and forwarded.
3. Once the *SP* obtains all the possible paths, those are sent to the service manager. The *SM* uses a Branch and Bound algorithm [26] to plan the microcloud's topology and communicates this topology to the *SP* and the *BM*. It also keeps updated information about possible routes in the microcloud.
4. The *SP* sends an acceptance message to the client with its address and a list of the nodes in the route through the chosen route. Before forwarding the message, every node that intercepts it, updates its routing table and extracts itself from the list until the client receives the message.

Detachment process follows the process depicted in Figure 2b. This is launched once a user detaches from the service (properly or due to a failure in communications). This detachment is either processed or discovered and communicated to the *SM*, which restructures the network without the failing client.

3.3 Migration of LVMS

The system shall ensure the QoE for all clients. When the topology of the network changes, the configuration of the microclouds dynamically adapts. Topology changes may be caused by new clients joining the network, existing clients leaving it or changes in the connections between clients (due to mobility of nodes or either BM or SM started reconfiguration). The dynamic nature of the use case makes unrealistic an approach where all clients always find the best route to the SP. Such an approach would introduce an exponential computation time when the network grows, affecting the QoE. The best solution is using efficiency heuristics. Thus, reconfigurations would be launched to keep efficiency over a minimum threshold, set experimentally. There exist two heuristics:

Global optimization heuristic: Since every microcloud is independent, their best configuration may result in a global inefficiency. On the other hand, the BM has information about every possible route in the system, as it is transmitted to it during the join process. The BM accepts every configuration which provides, at least, a minimum efficiency. This efficiency is calculated by the number of nodes connecting clients and SP which are not clients of the microcloud, for example, network infrastructure nodes; and the size of the microclouds. If the BM detects a configuration under the minimum efficiency threshold, it requests a redistribution to one or more microclouds. This redistribution may imply changing routes between clients and SP, split/merge between microclouds.

Local optimization heuristic: Inside a microcloud, the SM accepts every configuration providing a minimum efficiency (in terms of clients' QoE), to reduce the number of redistributions needed. If this threshold is surpassed, then a redistribution - new routes or SP migration - is requested inside the microcloud.

3.4 Failure Management

Four different roles may fail in our system: BM, SM, SP and client. In the case of the BM, a failure would not stop the services already running, but it would the system would not be checked for inefficiency. Furthermore, no new service or node would be able to join while the BM is unavailable. Traditional high availability and fault tolerance approaches can be used in this case. Also, the rejoin process would be affected, as the BM is unavailable. In this case, the clients would launch DEEPACC targeting their last known SM. If responsive, the process would proceed. If not, the rejoin is impossible until the BM is available.

A failure in a client only affects itself. Traditional high availability and fault tolerance approaches are enough in this case.

In the case of a failure of the SM, new users would not be able to join the microcloud, nor former users to rejoin it. Also, the inefficiency would go unchecked. To mitigate its effects, the SM is replicated across different nodes in the microcloud (backup nodes) following a hierarchical structure transparent to the user. If the SM is confirmed to have failed (by the *BM*, which cannot contact it), its role is taken by the next node in the hierarchy.

In the case of a SP failure, service would be completely stopped. As for the SM, this role is replicated in several nodes. If confirmed to have failed (by several clients which cannot contact it, which send a message to the SM), then the SM starts the next SP in the hierarchy. The LVM is relaunched in this node (to do so, the nodes selected as backups periodically retrieve a snapshot of the LVM).

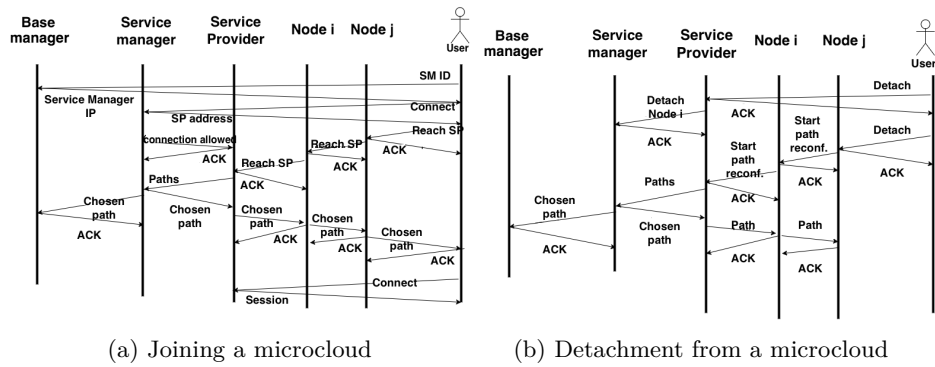


Fig. 2: Protocol description

To determine connectivity and/or physical failures a keepalive protocol is used. That way, every client in a microcloud is responsible for the state of its connection with its main neighbor (the first hop between the node and the *SP*), running a periodic check. If a link fails, a new join process is launched by the client or clients which detected the failure. Finally, the split of microclouds offers higher resilience - as a node crash would affect a smaller number of nodes if a large microcloud is split in several smaller ones.

4 Conclusions and Future Work

In this paper we proposed a microcloud-based platform, a semi-decentralized approach for managing services used in a limited geographical area - such as neighborhoods - in the context of Mobile Clouds. Our design is semi-decentralized and network-oriented, that is, it considers the network as a participating entity and profits from it. Our approach exploits network resources to reduce unnecessary data transportation over long distance networks, running computation on the nodes participating from the communication (personal devices, network equipment and/or specific purpose hardware such as smart-cities networks). We described the expected benefits of using our architecture over the dominant datacenter-centralized approach in terms of QoE.

As we understand that one main concern for the adoption of our system is the unfair distribution of load, where some devices run computation while other are free riders, we are currently working on a pricing model for microclouds, as an extra incentive to adopt a microclouds-based system. The exchange of computational time for monetary revenue would boost the uptake of microclouds. Service creators would benefit from the open market and the increase in competition, while clients would receive a faster and more personalized service, together with financial benefits from making their computational resources available for hosting or relaying service information. Together with this, we plan to work on mobile congestion issues to refine our protocol and address security issues.

References

1. Cisco, "The zettabyte era: Trends and analysis," White paper, 2015.
2. N. Fernando, S. W. Loke, and W. Rahayu, "Mobile cloud computing: A survey," *Future Generation Computer Systems*, vol. 29, no. 1, pp. 84 – 106, 2013.
3. E. Marinelli, "Hyrax: Cloud Computing on Mobile Devices using MapReduce," Master's thesis, Carnegie Mellon University, 2009.
4. O. Amft and P. Lukowicz, "From Backpacks to Smartphones: Past, Present, and Future of Wearable Computers," *IEEE Pervasive Computing*, vol. 8, no. 3, pp. 8–13, July 2009.
5. X. Luo, "From Augmented Reality to Augmented Computing: A Look at Cloud-Mobile Convergence," in *IEEE International Symposium on Ubiquitous Virtual Reality (ISURV)*, July 2009, pp. 29–32.
6. M. Bertier et al, "Beyond the Clouds: How Should Next Generation Utility Computing Infrastructures Be Designed?" in *Cloud Computing*, ser. Computer Communications and Networks, Z. Mahmood, Ed. Springer, 2014, pp. 325–345.

7. S. Scellato, C. Mascolo, M. Musolesi, and J. Crowcroft, "Track globally, deliver locally: Improving content delivery networks by tracking geographic social cascades," in *International Conference on World Wide Web (WWW)*, 2011, pp. 457–466.
8. D. Hristova, M. Musolesi, and C. Mascolo, "Keep your friends close and your facebook friends closer: A multiplex network approach to the analysis of offline and online social ties," *CoRR*, vol. abs/1403.8034, 2014.
9. M. Cha, A. Mislove, and K. P. Gummadi, "A measurement-driven analysis of information propagation in the flickr social network," in *International Conference on World Wide Web (WWW)*, 2009, pp. 721–730.
10. P. A. Longley, M. F. Goodchild, D. J. Maguire, and D. W. Rhind, *Geographic information science and systems*. John Wiley & Sons, 2015.
11. D. L. Johnson, E. M. Belding, and G. van Stam, "Network traffic locality in a rural african village," in *International Conference on Information and Communication Technologies and Development (ICTD)*, 2012, pp. 268–277.
12. M. Afanasyev, T. Chen, G. Voelker, and A. Snoeren, "Usage Patterns in an Urban WiFi Network," *IEEE/ACM Trans. on Networking*, vol. 18, no. 5, pp. 1359–1372, 2010.
13. M. Kristensen, "Scavenger: Transparent development of efficient cyber foraging applications," in *IEEE International Conference on Pervasive Computing and Communications (PerCom)*, March 2010, pp. 217–226.
14. GoogleTM. Google cloud. [Online]. Available: <https://cloud.google.com/>
15. AmazonTM. Amazon cloud infrastructure. [Online]. Available: <http://aws.amazon.com/about-aws/global-infrastructure/>
16. M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies, "The case for vm-based cloudlets in mobile computing," *IEEE Pervasive Computing*, vol. 8, no. 4, pp. 14–23, Oct 2009.
17. S. Bakiras and T. Loukopoulos, "Combining replica placement and caching techniques in content distribution networks," *Computer Communications*, vol. 28, no. 9, pp. 1062 – 1073, 2005.
18. Z. Peng, H. Ting-lei, L. Cai-xia, and X. Wang, "Research of P2P architecture based on cloud computing," in *Int. Conf. on Intelligent Computing and Integrated Systems (ICISS)*, Oct 2010, pp. 652–655.
19. K. Xu, M. Song, X. Zhang, and J. Song, "A Cloud Computing Platform Based on P2P," in *Int. Symp. on IT in Medicine Education (ITIME)*, 2009, pp. 427–432.
20. O. Babaoglu and M. Marzolla, "The people's cloud," *IEEE Spectrum*, vol. 51, no. 10, pp. 50–55, October 2014.
21. V. Valancius, N. Laoutaris, L. Massoulié, C. Diot, and P. Rodriguez, "Greening the internet with nano data centers," in *International Conference on Emerging Networking Experiments and Technologies (CoNEXT)*, 2009, pp. 37–48.
22. IEC, "Orchestrating infrastructure for sustainable Smart Cities," International Electrotechnical Commission, Tech. Rep., 2015.
23. (2010) Nextdoor. [Online]. Available: <https://nextdoor.com/>
24. (2010) Goneighbour. [Online]. Available: <http://www.goneighbour.org/>
25. C. G. Kim and K. J. Kim, "Implementation of a cost-effective home lighting control system on embedded linux with openwrt," *Personal and Ubiquitous Computing*, vol. 18, no. 3, pp. 535–542, 2014. [Online]. Available: <http://dx.doi.org/10.1007/s00779-013-0671-1>
26. I. Cuadrado Cordero, A.-C. Orgerie, and C. A. Morin, "GRaNADA: A Network-Aware and Energy-Efficient PaaS Cloud Architecture," in *IEEE International Conference on Green Computing and Communications (GreenCom)*, Dec. 2015.