FOG-oriented Joint Computing and Networking: the GAUChO Project Vision

FOG-oriented Joint Computing and Networking: the GAUChO Project Vision

Francesca Nizzi, Tommaso Pecorella,
Alessio Bonadio, Francesco Chiti,
Romano Fantacci
Dpt. of Information Engineering
Università di Firenze
Firenze, Italia
name.surname@unifi.it

Daniele Tarchi, Walter Cerroni
Dpt. of Electrical, Electronic, and Information Engineering
University of Bologna
Bologna, Italy
name.surname@unibo.it

***Abstract*—This paper presents a novel architectural principle for distributed and heterogeneous systems integrating Fog Computing and Networking approaches, which has been proposed within the "Green Adaptive Fog Computing and Networking Architecture" (GAUChO) project, funded by the MIUR Progetti di Ricerca di Rilevante Interesse Nazionale (PRIN) Bando 2015 - grant 2015YPXH4W-004. In particular a modular and flexible platform has been designed and developed, supporting low-latency and energy-efficiency applications as well as security, self-adaptation, and spectrum efficiency by means of a strict collaboration among devices. Specifically, the focus here is on the design of an integrated protocol architecture supporting mobile Fog-oriented services, and the developed Fog computing testbeds.**

***Index Terms*—Fog Computing; Fog Networking; Mobile Applications; Testbeds.**

I. The GAUChO Project: Overview and Challenges

In recent years, the Cloud Computing (CC) paradigm [1], [2] has become very popular by providing customers and enterprises with an ubiquitous on-demand network access to remote computing and storage platforms, or even services that can be rapidly provisioned and released with minimal design and operational effort. However, several constraints in terms of requested bandwidth and real time processing suggest to move processing as *close* as possible to data generation units (i.e., directly at the end-device or cluster of end-devices suitably formed) so as to minimize the "data production to decision making" latency.

In this perspective, Fog Computing (FC) is an emerging paradigm that extends CC towards the *edge* of the network [3]–[5]. In particular, FC refers to a distributed computing infrastructure confined on a limited geographical area in which some applications/services run directly at the network edge in smart end-devices. The goal of FC is to improve efficiency and reduce the amount of data that needs to be transported to the Cloud for massive data processing, analysis and storage. Furthermore, the design of efficient solutions within FC also requires investigate a novel communication/networking paradigm, called Fog Networking (FN), in order to meet specific configurability, adaptability, flexibility and energy/spectrum-efficiency constraints. To this purpose, self-adaptive design solutions [6], where the application autonomously adapts to follow context changes need to be introduced.

Generally speaking, FN leverages past experience in wireless communication and networking research, and fuses the latest advances in devices and network systems in the ecosystem of computing and networking. As a consequence, effective and efficient FN methodologies are of paramount importance to meet specific requirements. Despite FN exhibits promising technical features, new challenging issues are raised at the same time. In particular, FN has to guarantee the data delivery reliability among edge entities as well as to efficiently support end-devices mobility and service continuity. Device-to-Device communications [7]–[9] and, recently, the new concept of mmWave communication technology [10] are gaining importance in the literature since they appear to be promising solutions yielding efficient FN capabilities [11]. However, it appears from the existing literature, that significant research efforts have to be devoted to identify and tailor solutions to specific FN needs and to pursue an efficient and functional networking and computing capabilities integration in a same FC+FN platform. This is corroborated by the fact that generally FC and FN are designed and developed independently each other, hence not being able to fully exploit the complementary characteristics.

The "Green Adaptive Fog Computing and Networking Architecture" (GAUChO) project, funded by the MIUR Progetti di Ricerca di Rilevante Interesse Nazionale (PRIN) Bando 2015 - grant 2015YPXH4W-004 aims at designing a novel distributed and heterogeneous architecture able to functionally integrate and jointly optimize FC and FN capabilities in the same platform. The joint FC+FN architecture, representing the overall outcome of the project, aims at supporting low-latency and energy-efficiency as well as security, self-adaptation, and spectrum efficiency by means of a strict collaboration among end-devices and FC+FN units in a same integrated platform. Additionally, the development of suitable analytic methods and definition of appropriate techniques will enable extra relevant characteristics of the FC+FN platform including ubiquity, decentralized management, cooperation, proximity to

end users, dense geographical distribution, efficient support for mobility and real-time applications. To achieve this goal, the GAUChO project foresees to address several relevant and challenging research topics that require skills and knowledge in different scientific fields. The scientific contributions that the project aims at achieving, are significant and relevant in several aspects, such as:

1) Efficient schemes for the coordinated management of resources and interference in heterogeneous wireless communication systems;
2) Joint optimization of communication and computing capabilities to support energy-efficient management and self-reconfigurations;
3) A learning modality permitting software agents to detect variations within the integrated FC+FN platform;
4) Model-free fault diagnosis systems and comprehensive methodology integrating intelligence-based mechanisms for optimally managing energy consumption, detecting changes in environment/system under inspection, and evaluating and mitigating the possible occurrence of faults affecting the end-devices computing units.

The GAUChO project is expected to have a significant technological impact on many up-to-date and relevant families of technologies, such as Smart Wireless Sensor Networks (SWSNs), Smart Objects of Internet-of-Things and Intelligent Embedded Systems. All in all, the project will allow the FC+FN paradigm to enter into a new phase where real-world problems emerging from complex applications are addressed and effectively solved. It is expected that project outcomes will move from basic research to mass production in years 2018-25 providing significant economic benefits to masses of potential end users, together with the added value of the offered application services, limited cost of the communication and processing infrastructure.

In this papers, several specific aspects of the GAUChO project are described: in Sec. II the design of a distributed consensus sensing application for a mobile scenario is presented by joining integrating vehicular networking and blockchain approaches. Besides, in Sec. III a couple of Fog computing testbeds under development is presented, in order to prove the effectiveness of the proposed vision. Finally in Sec. IV the conclusions are drawn.

II. Integrated Architecture for Mobile Fog Applications

Intelligent Transport Systems (ITS), which could be alternatively expressed as "the application of information and communications technologies to the planning and operation of transportation systems", is expected to face numerous transport challenges, related to:

- Improve road safety and security
- Alleviate rising road congestion
- Enhance public transport
- Reduce the environmental impacts of mobility
- Support (semi)-automated car.

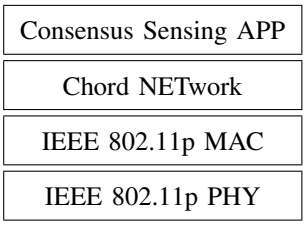

Fig. 1. Protocol Stack.

From a networking perspective, this vision is strictly related to the so called Vehicular Ad hoc NETworks (VANETs), that are a special kind of Mobile Ad hoc NETworks (MANETs) composed by vehicles as well as access points located at the edges of the roads (RSU) [12]. The reference network standard for VANETs is the IEEE 1609/WAVE (Wireless Access in the Vehicular Environment), which offers two modes of operation: vehicle-to-vehicle (V2V) and RSU-to-vehicle (R2V) communications [13]. WAVE is composed by two categories of standards: (i) 802.11p for PHY and MAC layers and (ii) IEEE 1609 for security, network management as well as other aspects of VANETs. With the emerging of the fifth generation (5G) mobile communication systems and software defined networks (SDNs), VANETs performance can be greatly improved by relaying on an abstract flexible radio interface called vehicle-to-everything (V2X) [14].

Moreover, a FC approach can further enhance ITS capabilities, enabling it to achieve its objectives. To improve the knowledge of streets and vehicles, local information with strict latency constraints is particularly relevant. Thanks to the advantages of edge location, FC can fulfill VANETs requirements, increasing safety services and improving traffic management [15].

Originally conceived as a digital payment framework, blockchain (BC) technology enables participants to read from and update to a common shared ledger (or blockchain) whose *state* is collectively maintained by the network in a *decentralized* fashion [16]. BC is updated via the consensus protocol that ensures a common, unambiguous ordering of transactions and guarantees the integrity and consistency of the blockchain.

When applied to the VANET domain, BC has to take into account some specific challenges, including, in particular the fast time varying topology. As a consequence, adaptable and robust mobile architectures are needed. We propose an integrated approach, managing the network topology set-up and maintaining, together with the information dissemination and validation in a decentralized and autonomous way.

A. System Model

The functional reference model, shown in Fig. 1, adopted in designing our integrated approach is comprised of:

- Physical and Data Link Layers, as specified in IEEE 802.11p;
- Network Layer's functionalities, provided by Chord protocol [17];

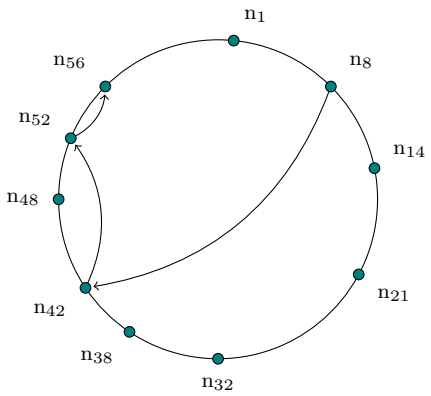

Fig. 2. Chord's lookup.

- Consensus Sensing (CS) Application designed according to BC technology.

Chord [17] is a protocol and an algorithm for distributed hash tables (DHT), used to realize a decentralized peer-to-peer (P2P) overlay network.

One of the main characteristics of Chord is its efficiency in resolving lookups. Indeed, it needs only $O(\log N)$ messages to reach any node in the network, as it depicted in Fig. 2. Moreover, each node maintains information only about $O(\log N)$ other node, so that node *join* and *leave* events only imply no more than $O(\log^2 N)$ messages.

In our approach Chord provides a *dynamic* and *distributed* address resolution table, which adapts to join and leave events. The APP layer directly uses services provided by Chord protocol and only sees the end-to-end (E2E) connections between nodes.

We consider a typical situation of traffic congestion, characterized by slowing speeds and vehicular queuing. This can occur either due to a crash or even to red traffic light. In this scenario, it is particularly relevant that the involved vehicles get the context awareness by means of a distributed query about the road/traffic conditions. Thus, vehicles are requested to arrange the convoys into a network and then start the CS application. This is dictated by the need of achieving a distributed decision, which considers all the information originated by each vehicle.

Employing a CS applications designed according to the BC technology allows all the participants to record all the observations from vehicles. This guarantees a *common view* of the context, and assures the integrity and the consistency of the information and its non ambiguous ordering.

The overall protocol we propose consists of two phases:

- The first phase is characterized by the formation of the Chord P2P network;
- In the second phase vehicles start a CS application.

B. Results

The proposed integrated protocol performance has been validated by means of numerical simulations. We implemented our protocol on a integrated VANET simulator widely adopted in the research community: VeiNS (Vehicle in Network Simulator).

As shown in Fig. 3, we evaluate the overall number of packets sent by each node. Here we consider only messages designed to maintain the Chord network's consistency (`P2P Messages`, P2PMs) and those which carry the information vehicles collect (`Observation Messages`, OMs), which are the most relevant in our scenario. This simulation aims at obtaining a network of 35 nodes and stops when the validation of a block is reached.

We can observe the presence of two different network, wherein nodes present the same tendency. This is due to the signaling of Chord protocol: for each node joining the network, some routing information needs to be updated. In particular, we can observe that the first nodes to join the network are those that send more messages, whilst the overhead gradually decreases for the other ones. We can also point out that the first community completed the network forming process, and its nodes start to send OMs. Conversely, the other group has not yet formed and no application messages are sent. Moreover, the number of P2PMs is much higher than OMs, this highlights that the formation of the Chord network is the more critical phase.

The proposed approach pointed out good performance in terms of consensus making overhead, latency and scalability, while the higher cost is due to P2P network formation via Chord protocol which in turns allows a highly efficient and resilient topology control.

III. Gaucho Fog Computing Testbeds

In order to prove the effectiveness of the proposed system and techniques, two suitable testbed implementations are under development.

A. University of Bologna Testbed

The goal of the Gaucho Fog Computing testbed in Bologna is to implement a proof of concept solution for generic fog services by using off-the-shelf devices such as System-on-Chip (SoC), System-on-Module (SoM) and Android devices.

To this aim the goal of the testbed is twofold. On one side we have to define first a service models for a fog network environment, and then a suitable networking architecture for coping with a flexible edge environment such as that implemented in a fog network.

A Fog Network can be considered as an extension toward the edge of the cloud computing paradigm, and hence it could support some of the XaaS models, e.g., Software-as-a-Service (SaaS), Platform-as-a-Service (PaaS) and Infrastructure-as-a-Service (IaaS).

In a cloud computing implementation a SaaS model allows to use a software application instance on a cloud server. This principle, if extended to the fog environment, allows to implement a model where a node (i.e., the server) provides a specific fog service through a specific interface to another node (i.e., the client); this means that both nodes should have implemented the same application/service. The exchanged

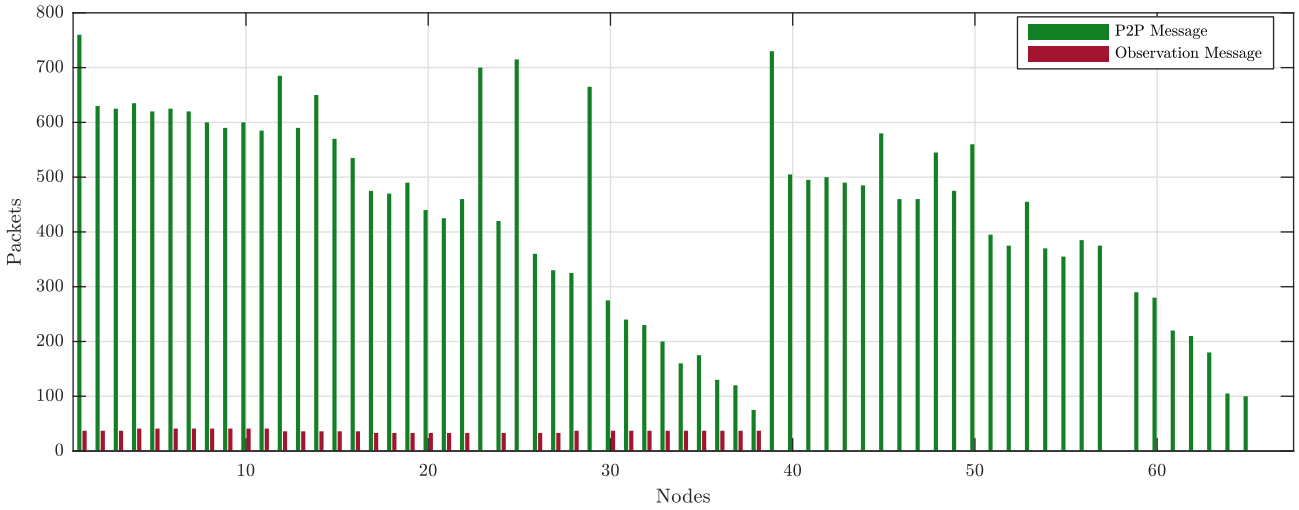

Fig. 3. Packets sent by each node.

data are limited to the input parameters/output results from one node to another.

In a cloud computing implementation, a PaaS model allows to use a platform/OS instance for developing custom applications. This principle, if extended to the fog environment, allows to implement a model where a node (i.e., the server) provides a generic fog service through a specific library/platform. Within this scenario a fog node can send an application to be executed if the receiving fog node has installed a compatible library/platform. In this case the exchanged data include also the code/application to be executed.

Finally, in a cloud computing implementation, a IaaS model allows to use virtualized resources for installing and running a custom system, including the whole processing stack. This principle, if extended to the fog environment, allows to implement a model where a node (i.e., the server) provides a generic fog service on a generic platform trough a programmable infrastructure (e.g., the virtualization environment). In this case a complete instance of a virtual machine implementing a specific service is executed and only the virtualization system (i.e., hypervisor) is needed on the server node. Due to the limited resources available in a fog computing node, lightweight virtualization paradigms must be considered, e.g., containers and unikernels. In this scenario the process becomes more complex since a direct offload of the whole virtualized environment from the requesting node to the server node cannot be considered as feasible. To this aim a repository should be foreseen for storing the available virtualized OS implementations to be used upon the request of a fog node; indeed, in this case, the exchanged data include the whole virtualized instance image.

It is quite clear that the implementation of the cloud service models on a fog environment follows the same behavior when going from the SaaS to the IaaS through the PaaS, from the lowest to the highest grade of flexibility and from the fastest to the slowest processing speed.

TABLE I
BOLOGNA TESTBED COMPONENTS

Device type	Number	Characteristics	Usage
Intel Edison	4	i386 32 bit; 2 Atom 500MHz; 1 GB RAM; 4GB Flash; Wi-Fi 802.11n	Fog Node
Intel Joule 570x	3	i386 64 bit; 4 Atom 1.7Ghz; 4 GB RAM; 16GB Flash; Wi-Fi 802.11 (2.4&5 GHz); Bluetooth LE (4.2); Graphic processor (Gen9LP)	Fog Node
Rasberry Pi3	2	CPU 4×ARM Cortex-A53 1.2GHz; RAM 1 GB; 10/100 Ethernet, Wi-Fi 2.4GHz 802.11n, Bluetooth 4.1 & LE; microSD; GPIO 40-pin.	Fog Access Point
Android Devices	4	Various characteristics	Fog Node
DELL PowerEdge T630	1	2 CPUs Intel Xeon E5-2640 v4 2.4GHz 25M Cache HT (20 core, 40 vCPU), 64 GB RAM, 1.3 TB HDD RAID 1, 4 NICs Gbit Ethernet	Virtualized Environment; Repository; Centralized Cloud

The testbed has been implemented by using different device types, each one with different characteristics that make them more suitable to be used for a specific task. The devices used in the testbed are listed in Table I.

By leveraging on the three previous models we have for the moment focused on an Android based testbed implementation. The testbed should cope with three main phases able to provide and execute fog services:

- Network Discovery: when a node is turned on should start a discovery phase for searching other nodes in the

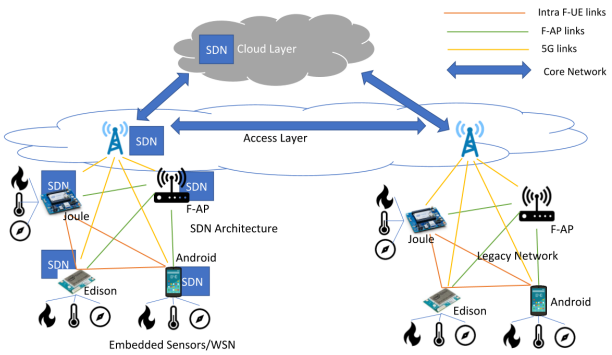

Fig. 4. The Gaucho Network Architecture

nearby capable of building any of the fog service models;

- Service Discovery: after having discovered any node, the incumbent node should be able of discovering which services are implemented by the nodes in the area;
- Service Delivery: the fog service implementation is performed by following one of the previously introduced models depending on the requested service.

The network architecture that has been foreseen to be used, represented in Fig. 4, consider to use both legacy and Software Defined Network (SDN) based solutions. While the legacy solution requires that each node should have already implemented the network and service discovery, in an SDN environment, thanks to the flows reprogrammability and resource virtualization, the incumbent fog node can transparently request a given service and, by gaining from the SDN implementation, the deployed node are able to help it in finding the nodes fulfilling its requirements. This can be done thanks to a logically centralized control plane, that as a counter-back may introduce an additional overhead and latency; this aspect in particular is now under study.

The components were used to set up a testing environment for application sharing in a fog environment. To this aim four different types of application were tested by leveraging on a suitable Android implementation of the Fog Network primitives, and based on SaaS and PaaS models.

Among others we will focus here on the iterative execution of an hashing operation through the HMAC algorithm, to discover the key given a string and its Hash. This operation has been executed locally and remotely by exploiting both SaaS and PaaS models. The SaaS model has been implemented through the exchange of the string between two different Android nodes. Differently, the PaaS model has been implemented by exploiting the BeanShell library installed on Android devices. Beanshell is a Java based library that allows to remotely execute a Java code. In this case the requesting node is sending the entire Java code that is executed remotely. The results are also compared with a local execution.

In Table II the delay values for different operations have been reported. The values are including the whole remote execution, including both transmission/reception and processing

TABLE II
ITERATIVE OPERATION DELAYS

Device	Local Delay [s]	SaaS Delay [s]	PaaS Delay [s]
Nexus 5	0.068	-	-
Samsung S4	-	0.265	9.479
Nomu S30	-	0.262	19.682
Asus K01A	-	0.257	26.353

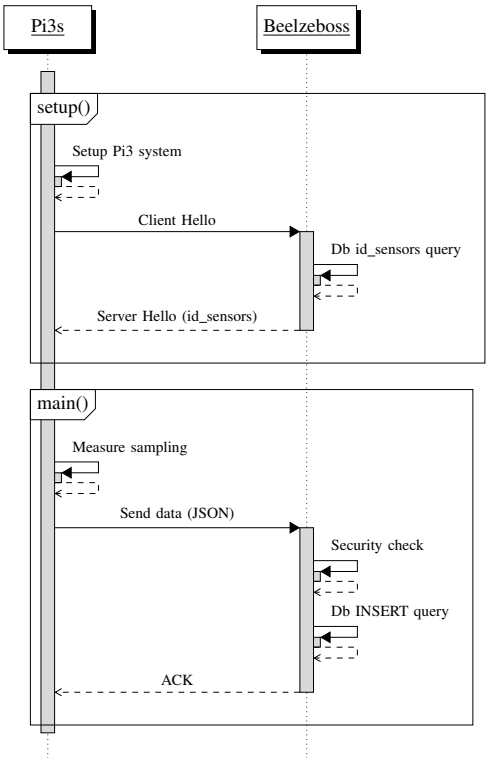

Fig. 5. GAUChO system

phases. It is possible to notice that the SaaS execution delay is higher than the local execution but the increase is not dramatic, while the PaaS execution requires much more time. This is due to the fact that the Android devices should execute a given code on a external library requiring much more processing effort. It is also possible to notice that the device processing capabilities do not affect the SaaS execution time, while PaaS is strongly affected by the device capabilities.

B. University of Florence Testbed

The testbed built at the University of Florence is made by 5 Raspberry Pi3 Model B (Adam, Lilith, Zeruel, Ireul, and Sachiel), 3 Raspberry Pi3 Model B+ (Arael, Ramiel, and Israfel), and a server (Beelzeboss). All the GAUChO scripts are written in Python 3.x following a client/server model, with a custom application using TLS/TCP and DTLS/UDP transport protocols, a simple webserver (which allows, after authentication, to download the data, and to see in real time the trend of the system), and a MySQL database.

In figure 5 the sequence diagram of the implemented system is shown. The goal of the testbed is to mimic a wireless sensor

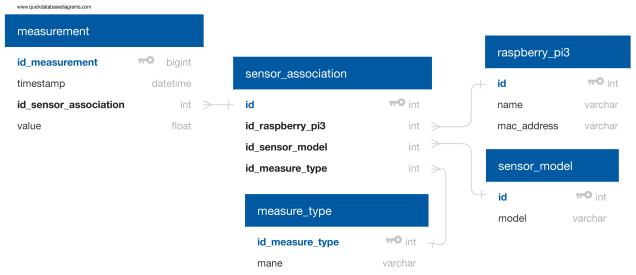

Fig. 6. GAUChO database

network where nodes can have multiple interfaces (wired and wireless), and computationally intensive tasks can be executed on the sensor nodes and/or on the a fog node.

At the system startup, the Pi3s (client in the client-server paradigm), wait until the network interface is enabled (i.e., they have a valid IP address). Afterwards, they contact the server to obtain a sensor unique ID, which is stored in the database (shown in figure 6). The server, after verifying the client identity, sends the appropriate ID. This phase is also used by the server to check which Pi3 have successfully completed the startup process.

After the setup phase, the Pi3 starts sampling sensor every 5 minutes. Currently all the Pi3s are equipped with 3 different sensors: BMP-280, SHT-31D and DHT-22. When all the samples are gathered, the client sends a JSON string to the server including an header, the time stamp and the measurements (identified by the previous ID). When the server receives the message, first perform a security check (authentication, authorization and confidentially) and, if it does not detect any anomalies, a database query is perform to insert the new measures. After receiving an ACK, the clients enter in a sleep mode until a new sampling is triggered.

Error checks are performed, and they are triggered by nodes becoming unresponsive or security violations.

At the moment the data analysis is performed by the fog node. However, the testbed will soon be extended to include SDN support through Open vSwitch. The goal is to use both wireless and wired network interfaces to provide FN security [18], L3 routing, and seamless offload of computation units to nodes.

IV. Conclusions

This paper introduced a novel architectural view particularly suited for distributed and heterogeneous systems, able to integrate both Fog Computing and Networking approaches, with the aim of supporting low-latency and energy-efficiency applications as well as security, self-adaptation, and spectrum efficiency by means of a strict collaboration among devices.

In particular, (ii) the design of an integrated protocol architecture supporting mobile Fog-oriented services, and (ii) a couple of Fog computing testbeds have been addressed to point out the capability and effectiveness of the proposed approach.

Acknowledgment

This work has been supported by the project “GAUChO - A Green Adaptive Fog Computing and Networking Architecture” funded by the MIUR Progetti di Ricerca di Rilevante Interesse Nazionale (PRIN) Bando 2015 - grant 2015YPXH4W-004.

References

[1] S. Murugesan and I. Bojanova, “Mobile cloud computing,” in *Encyclopedia of Cloud Computing*. Wiley-IEEE Press, 2016.

[2] S. Abolfazli, Z. Sanaei, E. Ahmed, A. Gani, and R. Buyya, “Cloud-based augmentation for mobile devices: Motivation, taxonomies, and open challenges,” *IEEE Communications Surveys Tutorials*, vol. 16, no. 1, pp. 337–368, First 2014.

[3] Q. F. Hassan, “Cloud and fog computing in the internet of things,” in *Internet of Things A to Z: Technologies and Applications*. Wiley-IEEE Press, 2018.

[4] A. Aljumah and T. A. Ahanger, “Fog computing and security issues: A review,” in *2018 7th International Conference on Computers Communications and Control (ICCCC)*, May 2018, pp. 237–239.

[5] M. Mukherjee, L. Shu, and D. Wang, “Survey of fog computing: Fundamental, network applications, and research challenges,” *IEEE Communications Surveys Tutorials*, 2018.

[6] C. Krupitzer, F. M. Roth, S. VanSyckel, G. Schiele, and C. Becker, “A survey on engineering approaches for self-adaptive systems,” *Pervasive and Mobile Computing*, vol. 17, pp. 184–206, 2015, 10 years of Pervasive Computing’ In Honor of Chatschik Bisdikian.

[7] M. Ahmed, Y. Li, M. Waqas, M. Sheraz, D. Jin, and Z. Han, “A survey on socially-aware device-to-device communications,” *IEEE Communications Surveys Tutorials*, 2018.

[8] S. Y. Lien, C. C. Chien, F. M. Tseng, and T. C. Ho, “3GPP device-to-device communications for beyond 4G cellular networks,” *IEEE Communications Magazine*, vol. 54, no. 3, pp. 29–35, March 2016.

[9] G. Rigazzi, F. Chiti, R. Fantacci, and C. Carlini, “Multi-hop d2d networking and resource management scheme for m2m communications over lte-a systems,” in *2014 International Wireless Communications and Mobile Computing Conference (IWCMC)*, Aug 2014, pp. 973–978.

[10] Y. Niu, Y. Li, D. Jin, L. Su, and A. V. Vasilakos, “A survey of millimeter wave communications (mmwave) for 5g: opportunities and challenges,” *Wireless Networks*, vol. 21, no. 8, pp. 2657–2676, Nov 2015.

[11] L. Lei, Z. Zhong, K. Zheng, J. Chen, and H. Meng, “Challenges on wireless heterogeneous networks for mobile cloud computing,” *IEEE Wireless Communications*, vol. 20, no. 3, pp. 34–44, June 2013.

[12] Y. Toor, P. Muhlethaler, A. Laouiti, and A. D. L. Fortelle, “Vehicle ad hoc networks: applications and related technical issues,” *IEEE Communications Surveys Tutorials*, vol. 10, no. 3, pp. 74–88, Third 2008.

[13] R. A. Uzcategui, A. J. D. Sucre, and G. Acosta-Marum, “Wave: A tutorial,” *IEEE Communications Magazine*, vol. 47, no. 5, pp. 126–133, May 2009.

[14] X. Duan, Y. Liu, and X. Wang, “Sdn enabled 5g-vanet: Adaptive vehicle clustering and beamformed transmission for aggregated traffic,” *IEEE Communications Magazine*, vol. 55, no. 7, pp. 120–127, 2017.

[15] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, “Fog computing and its role in the internet of things,” in *Proceedings of the First Edition of the MCC Workshop on Mobile Cloud Computing*, ser. MCC ’12. New York, NY, USA: ACM, 2012, pp. 13–16.

[16] K. Yeow, A. Gani, R. W. Ahmad, J. J. P. C. Rodrigues, and K. Ko, “Decentralized consensus for edge-centric internet of things: A review, taxonomy, and research issues,” *IEEE Access*, vol. 6, pp. 1513–1524, 2018.

[17] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan, “Chord: A scalable peer-to-peer lookup service for internet applications,” *SIGCOMM Comput. Commun. Rev.*, vol. 31, no. 4, pp. 149–160, Aug. 2001.

[18] T. Pecorella, L. Brilli, and L. Mucchi, “The Role of Physical Layer Security in IoT: A Novel Perspective,” *Information*, vol. 7, no. 3, p. 49, Aug. 2016. [Online]. Available: http://www.mdpi.com/2078-2489/7/3/49