This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2017.2702013, IEEE Access

Fog of Everything: energy-efficient networked computing architectures, research challenges, and a case study

Enzo Baccarelli, Paola G. Vinueza Naranjo, *Member, IEEE,* Michele Scarpiniti, *Senior Member, IEEE,* Mohammad Shojafar, *Member, IEEE,* and Jemal H. Abawajy, *Senior Member, IEEE*

***Abstract*—Fog Computing (FC) and Internet of Everything (IoE) are two emerging technological paradigms that, up to date, have been considered standing-alone. However, due to their complementary features, we expect that their integration can foster a number of computing and network-intensive pervasive applications under the incoming realm of the Future Internet. Motivated by this consideration, the goal of this position paper is fivefold. First, we review the technological attributes and platforms proposed in the current literature for the standing-alone FC and IoE paradigms. Second, by leveraging some use cases as illustrative examples, we point out that the integration of the FC and IoE paradigms may give rise to opportunities for new applications in the realms of the Internet of Energy, Smart City, Industry 4.0 and Big Data Streaming, while, at the same time, introducing new open issues. Third, we propose a novel technological paradigm, the Fog of Everything (FoE) paradigm, that integrates FC and IoE, and, then, we detail the main building blocks and services of the corresponding technological platform and protocol stack. Fourth, as a proof-of-concept, we present the simulated energy-delay performance of a small-scale FoE prototype, namely, the V-FoE prototype. Afterwards, we compare the obtained performance with the corresponding one of a benchmark technological platform, e.g., the V-D2D one. It exploits only Device-to-Device (D2D) links, in order to establish inter-thing "ad hoc" communication. Lastly, we point out the position of the proposed FoE paradigm over a spectrum of seemingly related recent research projects.**

***Keywords*—Fog of IoE, Virtualized networked computing platforms for IoE, Context-aware networking-plus-computing distributed resource management, Internet of Energy, Smart City, Industry 4.0, Big Data Streaming, Future Internet.**

Manuscript received Month Day, Year; revised Month Day, Year; accepted Month Day, Year.

Paola G. Vinueza Naranjo, Mohammad Shojafar, Michele Scarpiniti and Enzo Baccarelli are with the Department of Information Engineering, Electronics and Telecommunications (DIET), "Sapienza" University of Rome, Italy. (e-mail: {paola.vinueza, mohammad.shojafar, michele.scarpiniti, enzo.baccarelli}@uniroma1.it).

Jemal H. Abawajy is with the Faculty of Science, Engineering and Built Environment, Deakin University, Geelong, Australia. (Email: jemal@deakin.edu.au).

NOMENCLATURE

AP	Access Point
BD	Big Data
BDS	Big Data Streaming
$C2C$	Clone-to-Clone
CC	Cloud Computing
CDN	Content Delivery Network
CNT	Container
$D2D$	Device-to-Device
DCN	Data Center Network
EV	Electric Vehicle
$F2T$	Fog-to-Thing
FC	Fog Computing
FCL	Fog Clone
FDS	Fog Data Service
FN	Fog Node
FoE	Fog of Everything
FV	Fog Virtualization
$IaaS$	Infrastructure-as-a-Service
IoE	Internet of Everything
IoT	Internet of Things
MAN	Metropolitan Area Network
MEC	Mobile Edge Computing
MVP	Multi-core Virtual Processor
$P2P$	Peer-to-Peer
$PaaS$	Platform-as-a-Service
QoS	Quality-of-Service
$RFID$	Radio-frequency identification
$SaaS$	Software-as-a-Service
SC	Smart City
SG	Smart Grid
SOA	Service-Oriented Architecture
ST	Smart Transportation
$T2F$	Thing-to-Fog
$T2T$	Thing-to-Thing
$V2G$	Vehicular-to-Grid
$VLAN$	Virtual LAN
VM	Virtual Machine
VP	Virtual Processor
WAN	Wide Area Network
$WLAN$	Wireless Local Area Network
WSN	Wireless Sensor Network

I. INTRODUCTION

Consider a pervasive network of densely distributed energy and resource-limited wireless things (e.g., smart devices), all capable of gathering and transferring in real-time large volumes of heterogeneous environmental data. Due to the current energy-computing-bandwidth

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2017.2702013, IEEE Access

limitations of the wireless domain, up to date, a system of this complexity was unfeasible. However, the incoming Future Internet era is introducing two new paradigms, namely, the Fog Computing (FC) and the Internet of Everything (IoE), that, in principle, could open the doors to a new Fog of Everything (FoE) paradigm.

Fog Computing is a quite novel computing paradigm that aims at moving the Cloud Computing (CC) facilities and services to the access network, in order to reduce the delays induced by service deployments [1]. By fact, augmentation and virtualization of resource-poor wireless/mobile smart heterogeneous things are opening the doors to novel context-aware crowd-sensing applications, that allow spatially distributed human/machine users to capture, analyze and share environmental local data of social interest. This is, in turn, the realm of the so-called IoE paradigm [2], in which context-aware things autonomously setup and manage self-organizing networks. Interestingly, these networks are no longer human networks empowered by the presence of things. On the contrary, IoE networks are self-orchestrating eco-systems, that aim at providing services to humans by empowering the performance of the underlying Internet of Things (IoT) physical infrastructures [3]. This is attained by improving the functions of thing discovery and service composition, while suitably self-managing the limited computing-plus-communication resources of the involved things. The final goal of the IoE networks is to autonomously attain right energy consumption-vs.-attained performance tradeoffs without any human supervision [4], [5].

In order to implement the IoE vision, the technological paradigm of the FC is expected to provide the needed networking-plus-computing support by allowing the on-the-fly instantiation of software clones (e.g., virtual surrogates) of the physical things atop nearby resource-equipped cloudlets [6], [7], [8].

This is, indeed, the Fog-over-IoE scenario considered by this position paper. At this regard, we point out that the Fog paradigm is giving rise to stimulating discussions about its expected benefits and costs. Without doubt, servers' underutilization in Cloud-based large-scale remote data centers is a common phenomenon, mainly due to an over-provisioning of network and computing resources for handling workload peaks. As a consequence, electricity costs cover a large fraction of the overall operating costs of state-of-the-art Cloud-based data centers [9]. From this point of view, the Fog and IoE paradigms promise to reduce energy consumptions and related operating costs by leveraging their native self-organizing and self-scaling capabilities, as well as their pervasive spatial deployment [6].

A. Motivations, goals and organisation of the paper

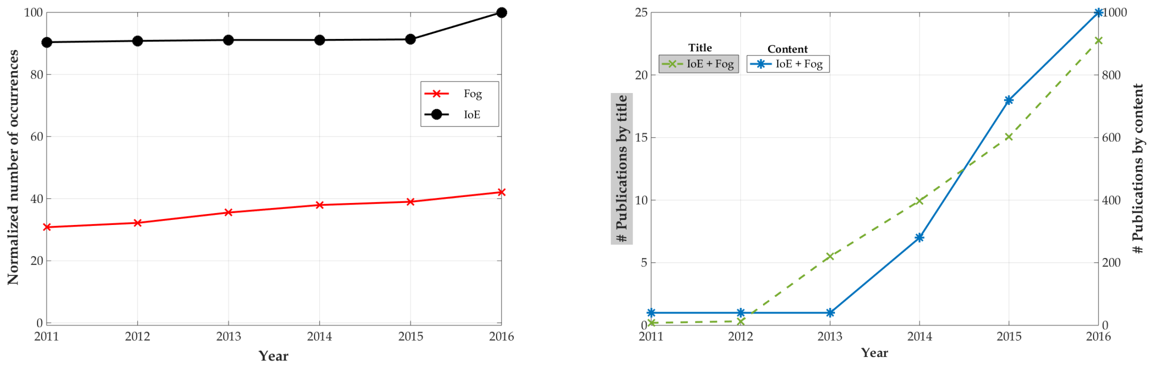
In order to corroborate these considerations, we carried out a statistical search over the Google for the occurrence of the IoE and Fog keywords in research contributions published during the last six years. Fig. 1(a) reports the (normalized) numbers of detected occurrences of the Fog and IoE keywords, while Fig. 1(b) plots the (absolute) numbers of occurrences of the joint Fog-plus-IoE keyword in the title and body of the searched papers.

An examination of plots of Fig. 1 leads to two main insights. First, the research literature is beginning to shift towards an integration of the Fog and IoE paradigms. Second, up to date, no research and/or technical contributions have been still published on the actual integration of the Fog and IoE pillar paradigms.

The aforementioned considerations motivate the current work. Formally speaking, it is a position paper that cascades a first part of survey type (see Sections II and III) to a second part in which some fresh technical contributions are provided (see Sections IV and V). Its main focus is on the networked computing architectures, offered services, IoE-based supported applications and energy-efficient resource management of Fog-based technological platforms.

The main goals and organization of this paper is as follows. First, in Section II, the paper provides a synoptic review of the standing-alone Fog and IoE paradigms, in order to unveil the complementary nature of their native attributes. Second, in Section III, the paper outlines the Quality of Service (QoS) requirements and related open challenges presented by some emerging applications that, in principle, could be promoted by a tight integration of the Fog and IoE pillar paradigms. For this purpose, the Internet of Energy, Smart City, Industry 4.0 and Big Data Streaming are considered as use cases. Third, motivated by these considerations, in Section IV, the paper proposes the FoE paradigm and details its technological platform and supporting protocol stack. Fourth, as a proof-of-concept, in Section V, a FoE-enabled case study is presented. Afterwards, the numerically evaluated energy-vs.-delay performance of a small-scale FoE prototype (e.g., the V-FoE prototype) is compared with the corresponding one of a benchmark technological platform (e.g., the V-D2D platform). This last relies only on "ad-hoc" Device-to-Device (D2D) WiFi-based links, in order to support inter-thing communication. Fifth, in Section VI, the paper discusses and poses under the right perspective the position of the proposed FoE paradigm over a spectrum of seemingly related research projects. Sixth, in the final Section VII, the paper summarizes the main presented results and outlines some directions for future research in the realm of the Future Internet.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2017.2702013, IEEE Access



(a) Normalized numbers of technical publications involving the Fog and IoE keywords.

(b) Absolute numbers of technical publications jointly involving the Fog-plus-IoE keywords.

Fig. 1: Research trends on the Fog, IoE and Fog-plus-IoE paradigms over the 2011-2016 time-window.

II. IoE and Fog Computing: a synoptic overview

The goal of this section is three-fold. First, we review and compare the native attributes of the standing-alone IoE, Fog and Cloud paradigms, in order to gain insight on their (possible) inter-play. Second, we perform a comparative review of two main technologies currently utilized for the virtualization of the computing and networking physical resources of data centers, namely, Virtual Machine (VM)-based and the CoNTainer (CNT)-based virtualization technologies. Third, we review and compare the service models done available by the resulting virtualized IoE-Fog-Cloud ecosystem. We anticipate that the comparative review of this Section II provides the motivation for the introduction of the proposed FoE paradigm of Section IV.

A. *Basic attributes and challenges of the IoE paradigm*

The recent past years have been characterized by two seemingly contrasting technological trends.

The first one regarded the surging of the Cloud model as ubiquitous computing paradigm and the resulting shift of computing, control and data storage capabilities towards remote and large-size data centers [9]. Since these data centers are far away from the network edge, end-users connect them through the Internet backbone.

The second trend concerned the surging of a number of heterogeneous user-oriented access and sensor devices, like tablets, smartphones, smart home appliances, access points, edge routers, roadside-placed cabinets for the smart control of the vehicular traffic, connected vehicles, smart meters for power grids, smart control systems for Industry 4.0 factories, just to name a few. Additional smart edge devices, like industrial and home robots, computers on a stick, RFID-based frequency tuners, are currently gaining momentum. The common feature of all these devices is that they are things that operate at the network edge. This is, indeed, the realm of the so-called IoE paradigm [2], [3], [4], [5].

Formally speaking, the *IoE model* refers to an ecosystem of (possibly heterogeneous) edge devices, that autonomously share and self-manage their limited resources, in order to attain a common system-wide goal [4], [5]. The peculiar attributes of the IoE ecosystem are (see Table I): (i) pervasive spatial deployment at the network edge; (ii) context-awareness; (iii) self-management; (iv) self-organization; and, (v) inter-thing social relationships. As sketched in Fig. 2, goal of the IoE model is to provide a spatially distributed technological platform for the pervasive support of Machine-to-Machine (M2M), People-to-Machine (P2M) and People-to-People (P2P) services [10], [11], [12], [13].

It therefore becomes of interest to pose the following question: "How this plethora of resource-limited edge devices may be organized and managed, in order to provide the aforementioned services by leveraging the Internet as the interconnecting technology?"

This question points to a right tradeoff between the two contrasting targets of "concentrated resources and centralized resource management", and "pervasive resource placement and distributed resource management". Under this perspective, the IoE paradigm is still searching for system-wide architectural solutions for many emerging application scenarios, such as, cyber-physical systems and embedded sensor networks. This paradigm still needs to address basic questions ranging from where to compute and where to store data along the Thing-to-Cloud path to how to allow the things to pool and share their limited computing-plus-networking resources [5].

From this point of view, the IoE paradigm introduces

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2017.2702013, IEEE Access

TABLE I: Attributes and main technological challenges of the IoE paradigm.

IoE Attributes
Pervasive spatial deployment
Context-awareness
Self-management
Self-organization
Inter-thing social relationships
Main IoE Challenges
Reduced communication latency
Wise usage of the Internet bandwidth
Resource limitation of the IoE devices
Self-organizing ecosystems
Intermittent network connectivity

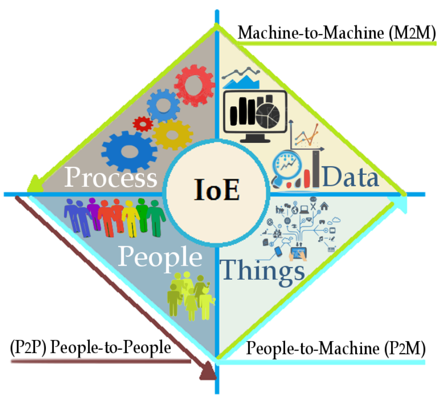


Fig. 2: Main services supported by the IoE paradigm.

a number of new challenges that cannot be adequately addressed by the Cloud and Host computing models alone. Some of such basic challenges are listed in Table I and discussed in the sequel.

Reduced communication delays – Industrial control systems (like manufacturing factories and smart grids) typically require sensor-to-controller communication delays of the order of milliseconds [14]. IoT applications (like vehicle-to-vehicle communication, virtual reality applications, online games) may require service-deployment latencies below a few tens of milliseconds [15], [16], [17], [18]. All these latency requirements cannot be supported by the remote Cloud alone.

Wise usage of the Internet bandwidth – The quickly growing number of connected things is generating big data flows at an exponential rate [4]. Routing all data generated by the edge things to the remote Cloud would congest the Internet backbone. This requires, in turn, that the processing of the data generated by the edge things is carried out as much as possible within the access network.

Resource limitations of the IoE devices – Many IoE devices (like sensors, actuators, controllers and embedded systems) are both resource and energy-limited. Therefore, they are not capable to rely solely on own capabilities, in order to fulfill their computing-communication tasks. However, due to the (aforementioned) constraints on the allowed delays and usage of the Internet bandwidth, offloading all tasks to the remote Cloud is not a feasible option [19], [20], [21], [22].

Self-organizing ecosystems – In principle, the self-cooperation and self-organization of the IoE devices would provide a first solution to the mentioned latency-bandwidth-resource limitations. However, how to enforce device cooperation through the dynamic setup and self-management of suitable inter-thing "social" networks is still a big challenge [3], [13], [23], [24].

Intermittent network connectivity – In order to support device mobility, both Device-to-Cloud and Device-to-Device reliable network connections should be guaranteed. However, due to the multi-hop nature of the Internet backbone and the short-range capability of the network technologies currently envisioned for the support of inter-device communication [25], [26], guaranteeing reliable network connections is a challenging task in the envisioned IoE realm [27], [28].

All these IoE challenges open, indeed, the doors to the FC paradigm of the next subsection.

B. Basic attributes of the Fog Computing model

A synoptic review of the body of papers recently appeared on the Fog topic [1], [6], [29], [30], [31], [32] points out that this model relies on the assumption that the computing tasks can be performed by nodes placed at the edge of the access network and in between the remote Cloud and the IoE devices. The final goal is to augment the computing-plus-bandwidth resources of the served devices without increasing too much the resulting service latencies. Under this perspective, the FC paradigm differs from the related Mobile Edge Computing (MEC) one [7], [8], [33], [34], in which only edge nodes are employed for the device augmentation. Both the FC and MEC paradigms rely,

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2017.2702013, IEEE Access

indeed, on edge nodes, that are one-hop away from the served IoE devices; however, as sketched in Fig. 3, the former paradigm integrates them with both the remote Cloud and the IoE devices, whereas the latter accounts only for the IoE devices.

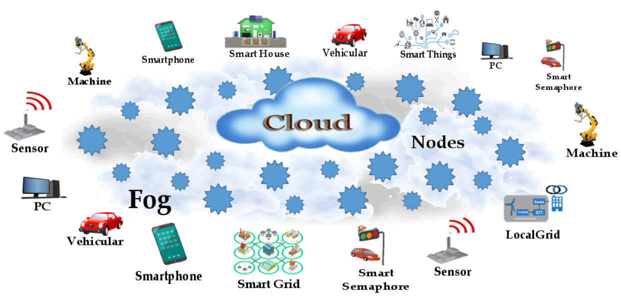


Fig. 3: A pictorial view of the three-tier FC model.

The model definition – From a formal point of view, we may see *Fog Computing* as a model to complement the Cloud through the distribution of the computing-plus-networking resources from remote data centers towards edge devices. The final goal is to save energy and bandwidth, while simultaneously increase the QoS level provided to the users.

As a consequence, Fog Nodes (FNs) are virtualized networked data centers, which run atop (typically, wireless) Access Points (APs) at the edge of the access network, in order to give rise to a three-tier IoE-Fog-Cloud hierarchical architecture [35], [36].

An examination of Fig. 4 points out that the main attributes of the Fog paradigm are the following ones [1], [6], [37], [38]:

- *Edge location and location awareness* – Being deployed in proximity of the served IoE devices, FNs may efficiently leverage the awareness of the states of the communication links (e.g., WiFi-based single-hop TCP/IP transport-layer connections) for the support of delay and delay-jitter sensitive applications, like video streaming [39], [40], [41];
- *Pervasive spatial deployment* – FNs support distributed applications, which demand for wide spatial deployments, like Wireless Sensor Network (WSN)-based applications [42];
- *Support for the mobility of the served devices* – FNs may exploit Fog-to-Thing (F2T) and Thing-to-Fog (T2F) single-hop WiFi links for data dissemination/aggregation [30], [43];
- *Low energy consumption through adaptive resource scaling* – Since Fog nodes are densely distributed over the spatial domain and connected to wireless access networks, they are typically equipped with capacity-limited batteries, that may be (hopefully) re-charged through renewable energy sources (like solar panels, wind turbines and/or micro-grids) [44], [45]. Hence, a main target of the Fog paradigm is the reduction of both the computing and networking energy consumptions through the adaptive horizontal (e.g., intra-Fog nodes) and vertical (e.g., inter-Fog nodes) scaling of the overall available resource pool;
- *Heterogeneity of the served devices* – According to the IoE paradigm, FNs must be capable to serve a large spectrum of heterogeneous devices, that ranges from simple RFID tags to complex user smartphones, tablets and multimedia mobile sensors [46], [47];
- *Dense virtualization* – IoE devices are resource-limited and densely deployed over the spatial domain. Hence, in order to provide device augmentation at minimum resource costs, Fog nodes must be capable to multiplex a large number of virtual clones with different resource demands onto a few number of physical servers [48];
- *Device isolation* – In order to guarantee trustworthiness to the served devices, the corresponding clones must run atop Fog servers as isolated virtual machines or containers [49].

The basic technological platform – According to the aforementioned attributes, Fig. 5 reports the main blocks that compose a virtualized Fog node [50].

These blocks operate at the Middleware layer of the underlying protocol stack and comprise (see Fig. 5):

- the *input and output buffers* – They smooth the peaks of the workload to/from the served devices;
- the *physical resources* – They comprise the physical servers, routers and physical channels that equip the Fog node;
- the *bank of Virtual Processors (VPs)* – They process the assigned workload on behalf of the served devices;
- the *Virtualization layer* – It acts at the Middleware layer and multiplexes in real-time the available physical resources among the running VPs;
- the *Virtual Switch* – It sustains and manages the inter-VP TCP/IP transport connections on an end-to-end basis; and,
- the *Adaptive Load Dispatcher* – It dispatches the input workload over the set of available VPs in a balanced way, in order to minimize the computing-plus-networking consumed energy, while meeting the QoS requirements of the served devices.

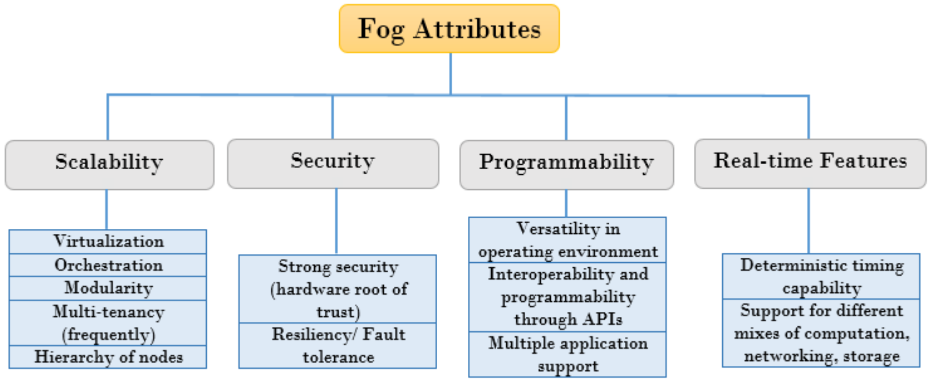


Fig. 4: Main native attributes of the Fog paradigm.

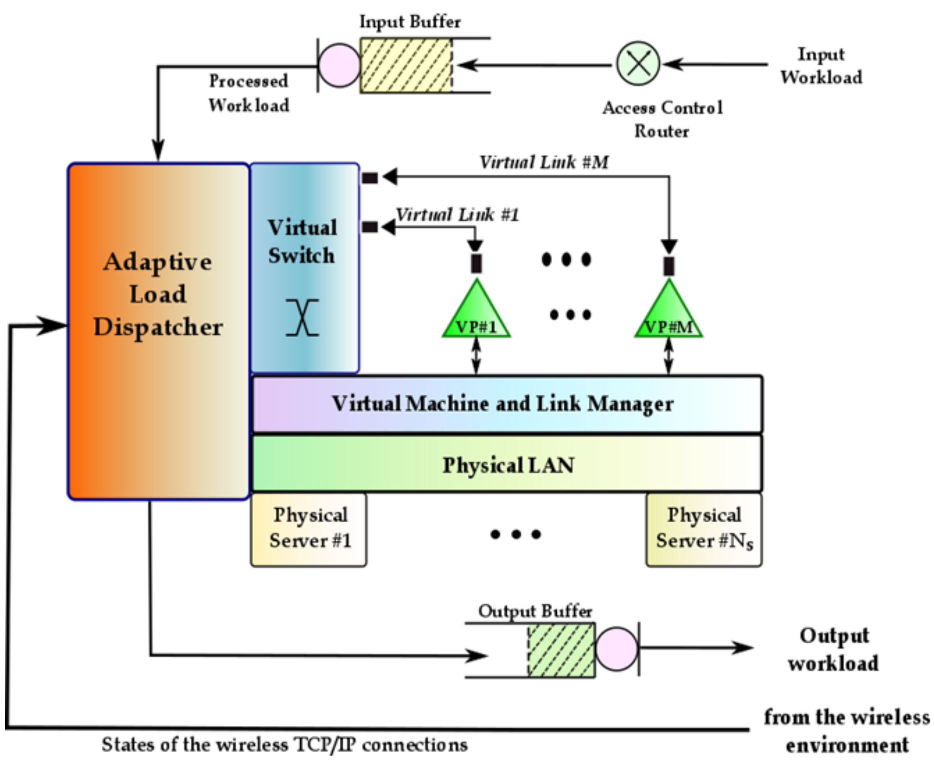


Fig. 5: A snapshot of the general architecture of a virtualized Fog node. It operates at the Middleware layer. Black boxes are Virtual Network Interface Cards (VNICs) supporting TCP/IP intra-Fog transport connections. VP:= Virtual Processor; M:= Number of served virtual clones.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2017.2702013, IEEE Access

The basic supported services – In principle, the Fog node of Fig. 5 may serve the connected devices according to three general models, namely, the *Infrastructure as a Service (IaaS)*, *Platform as a Service (PaaS)*, and *Software as a Service (SaaS)* models [9]. However, due to the stringent limitation of the hardware and software resources of the IoE devices, up to date, the SaaS model seems to be the most suitable one, in order to support IoE-based applications [3], [4], [5]. According to this consideration, the state-of-the-art Fog Data Service (FDS) by Cisco [51] is an IoE-compliant SaaS-oriented software, that aims at mapping raw sensor data into actionable information. This is done by providing to the requiring devices a basic set of primitive functions, that include [51]:

- content-based data filtering;
- intelligent encryption of plaintext sensor data;
- remote reconfiguration of sensor devices through REST-based APIs;
- time stamp-driven caching of sensor-acquired data;
- content correlation-based data fusion;
- dynamic management of intra-Fog databases.

C. *On the Fog-Cloud complementarity*

According to a commonly accepted definition provided by the National Institute for Standards and Technology (NIST) [9], Cloud Computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable networked computing resources (e.g., computing servers, network infrastructures, applications, and services), that can be rapidly provisioned and released with minimal management effort or service provider interaction [52], [53], [54]. Hence, since both the FC and CC models provide computing resources to the served devices on an on-demand basis, they share the following attributes [55], [56], [57]: (i) adaptive horizontal (e.g., intra-data center) scaling of the networking-plus-computing physical resources; and, (ii) maximization of the resource utilization through multiplexing of the physical resources and device virtualization.

However, passing to consider the Cloud-Fog complementarity, two main remarks are in order.

First, FNs are deployed in the access network, and the Device-Fog distance is typically limited up to one hop. Hence, the following attributes: (i) support for the device mobility; (ii) vertical (e.g., inter-data center) resource scaling; (iii) support for device heterogeneity; (iv) context-awareness; and, (v) communication latencies limited up to a few tens of milliseconds, are specific of the FC paradigm [31], [32]. However, these (positive) attributes are counterbalanced by the fact that: (i) FNs are small-size data centers with reduced fault-tolerance capability; and (ii) the Device-Fog communication typically relies on short-range WiFi/UWB/Bluetooth-based connections that are intermittent, interference-affected and failure-prone [58], [59], [60].

Second, Cloud nodes are large-size data centers, that concentrate a huge number of networked computing servers and storage devices [61], [62], [63]. Furthermore, they are connected to national-wide Service Providers by Internet Wide Area Networks (WANs) [30], so that Cloud-Device communication exploits ubiquitous 3G/4G cellular connections. Then, Cloud data centers offer: (i) measurable quality of the offered services for transparency and billing [64], [65]; and, (ii) high degree of fault tolerance. However, they are context-unaware, and, furthermore, suffer from high service deployment latencies, that may be of the order of hundreds of milliseconds [29], [61], [66], [62].

Table II recaps the reported considerations and presents a Cloud-vs.-Fog comparison under an IoE-oriented perspective.

Overall, the emerging conclusion is that the FC model co-exists with the (more traditional) CC one. In fact, FC makes the computing augmentation of resource-poor wireless devices feasible even in the IoE realm, where the number of served devices is expected to be very large and the latency constraints are stringent [67].

D. *A comparison of virtualization technologies in the IoE realm*

Virtualization is employed in Cloud and Fog-based data centers, in order to [68]: (i) dynamically multiplex the available physical computing, storage and networking resources over the spectrum of the served devices; (ii) provide homogeneous user interface atop (possibly) heterogeneous served devices; and, (iii) isolate the applications running atop the same physical servers, in order to provide trustworthiness. Roughly speaking, in virtualized data centers, each served physical device is mapped into a virtual clone that acts as a virtual processor and executes the programs on behalf of the cloned device [68]. In principle, two main virtualization technologies could be used to attain device virtualization, namely, the (more traditional) Virtual Machine (VM)-based technology [68], [69] and the (emerging) CoNTainer (CNT)-based technology [48], [70].

In a nutshell, their main architectural differences are that [70], [71]: (i) the VM technology relies on a Middleware software layer (e.g., the so called Hypervisor) that *statically* performs hardware virtualization, while the CNT technology uses an Execution Engine, in order to *dynamically* carry out resource scaling and multiplexing; and, (ii) a VM is equipped with an own (typically, heavy-weight) Guest Operating System (GOS), while a container comprises only application-related (typically, light-weight) libraries and shares with the other containers the Host Operating System (HOS) of the physical server.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2017.2702013, IEEE Access

TABLE II: Cloud-Fog complementarity and interplay. VM:= Virtual Machine; WAN:= Wide Area Network; WLAN:= Wireless Local Area Network.

Cloud attributes	Fog attributes
Vertical resource scaling	Vertical and horizontal resource scaling
Large-size and centralized	Small-size and spatially distributed
Multi-hop WAN-based access	Single-hop WLAN-based access
High communication latency and service deployment	Low communication latency and service deployment
Ubiquitous coverage and fault-resilient	Intermittent coverage and fault-sensitive
Context-unawareness	Context awareness
Limited support to device mobility	Full support to device mobility
Support to computing-intensive delay-tolerant analytics	Support to real-time streaming applications
Unlimited power supply (exploitation of electrical grids)	Limited power supply (exploitation of renewable energy)
Limited support to the device heterogeneity	Full support to the device heterogeneity
VM-based resource virtualization	Container-based resource virtualization
High inter-application isolation	Reduced inter-application isolation

The resulting pros and cons of these two virtualization technologies are summarized in Table III.

Shortly, the main pros of the CNT-based virtualization technology are that: (i) containers are light-weight and can be deployed significantly quicker than VMs; and, (ii) the physical resources required by a container can be scaled up/down in real-time by the corresponding Execution Engine, while, in general, physical resources are statically assigned to a VM during its bootstrapping.

However, since all containers running atop the same physical server share the same HOS, the main cons of the CNT-based virtualization technology are that: (i) the level of inter-application isolation (e.g., the level of trustworthiness) guaranteed by the container-based virtualization is typically below than the corresponding one offered by the VM-based technology; and, (ii) all the application libraries stored by the instantiated containers must be compliant with the HOS equipping the host physical server.

Overall, both virtualization technologies retain pros and cons. However, due to the expected large number of devices to be virtualized in IoE application environments, resorting to the CNT-based virtualization would allow to increase the number of virtual clones per physical server (e.g., the so-called virtualization density) [71]. We anticipate that this is the reason why the proposed FoE paradigm of Section IV relies on the container-based virtualization technology (see the Section IV.B for more details on this topic).

E. Feasible service models in the virtualized IoE-Fog-Cloud ecosystem

In computing systems that utilize only remote Cloud data centers, the IoE devices at the edge of the network may communicate with the Cloud servers by exploiting only Internet-based multi-hop WANs. In fact, under this scenario, all the computing and storage resources needed by the device augmentation are in the remote Clouds and IoE devices may access these remote resources by exploiting only the Server-Client model [9].

The picture changes radically under the three-tier IoE-Fog-Cloud ecosystem of Fig. 6. In this ecosystem, the physical resources needed by the device augmentation are no longer concentrated into the remote Cloud. In fact, FNs and inter-device cooperation allow to bring the computing and storage resources closer to the requiring devices. This leads, in turn, to two main benefits, namely, the contraction of the delays needed by service deployment and the reduction of the network traffic to be routed by the Internet WAN. These benefits arise from the fact that the ecosystem of Fig. 6 makes available *three* basic service models for the workload execution, namely, the Offloading, Aggregation and Peer-to-Peer models.

Under the *Offloading model*, FNs act as switches, in order to offload the traffic from the devices to the remote Cloud (e.g., Up-offloading) and from the remote Cloud to the devices (e.g., Down-offloading). In both cases, FNs perform a (possibly, partial) processing of the switched workload, in order to reduce the communication latency and forward to the remote Cloud only the most computing-intensive tasks.

TABLE III: VM-vs.-Container. HOS:= Host Operating System; GOS:= Guest Operating System.

VM attributes	Container attributes
Both the HOS and GOS are present	Only the HOS is present
Large delays for the VM deployment	Low delays for the container deployment
Static per-VM resource provisioning	Dynamic per-container resource provisioning
High level of VM isolation	Reduced level of container isolation
Low virtualization density	High virtualization density

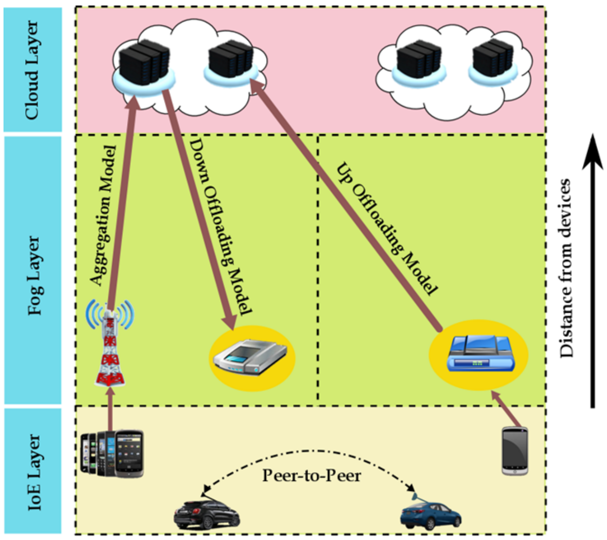


Fig. 6: Service models under the ecosystem composed by the IoE, Fog and Cloud layers.

Under the *Aggregation model*, data streams generated by multiple devices are gathered (and, possibly, suitably fused) by a multiplexing node at the edge of the network, and, then, are routed to the remote Cloud through the Internet WAN for further processing. The multiplexing node is a gateway router. It is connected to a 3G/4G base station, and may be also equipped with (limited) processing capability.

Finally, under the *Peer-to-Peer (P2P) model*, proximate devices make available their computing and storage capabilities, in order to share tasks and cooperate for workload execution. For this purpose, the cooperating devices build up D2D links, that typically rely on short-range wireless communication technology, like UWB, WiFi or Bluetooth [25], [26].

The choice of the right service model is application and environment-depending, and may also depend on the level of context-awareness of the involved devices [33]. At this regard, two main remarks are in order. First, the above three basic service models may be also composed into a number of *Hybrid models*, especially when heterogeneous tasks demand for different QoS requirements. Second, we anticipate that the proposed FoE paradigm of Section IV adopts a hybrid service model, that suitably combines the (aforementioned) Offloading and P2P basic ones of Fig. 6.

III. Recent work, emerging application areas and related open issues: an overview

An overview of the current literature points out that recent work on Fog-based applications is aligned along two main research lines, deal with resource management issues. However, the first line focuses on the energy and information management, while the second one regards the orchestration of application-induced QoS requirements. The main characteristics of the proposed management approaches are summarized in Tables IV, V and VI. Specifically, Table IV overviews some potential Fog-based applications for energy and information management. Roughly speaking, these applications aim at extending to the Fog realm some solving approaches previously deployed under the CC scenario. Table V reports some main findings on the energy efficiency of Fog and Cloud-based applications. They embrace Smart Grid (SG)-based applications that rely on the emerging smart-meter technology. Interestingly enough, since smart meters operate at the edge of the power grid, they constitute a real-world instance of "smart" IoE devices, that utilize the intermediate FNs as a bridge to the remote Cloud [72].

Overall, a synoptic examination of these tables points out that there are several open issues that must be still afforded in Fog-supported applications. They will be reviewed in the next subsections by leveraging four specific use cases as guidelines.

A. Emerging application areas and related QoS requirements

The quantitative analysis of the (typically heterogeneous) QoS demands of Fog-supported IoE services is a topic that, up to date, seems to be largely unexplored.

TABLE IV: Classification of the main application areas involving the IoE and Fog paradigms.

Technology	Applications	Fog Computing Applications
Energy management	– Micro-grid management [72], [73] – Dynamic demand response operated within the micro-grid [74] – Real-time monitoring on applications for SG [64]	– Data-metric communication with the implementation of private Fog for small size network – Fog applications dynamically increase bandwidth capacity to avoid congestion – Micro-grid to micro-grid interaction through Fog – Development of demand-response model for internal micro-grid operation
Information management	– Smart meter data streams in Cloud [74], [75], [76] – Dynamic data center operations [44], [65]	– Guaranteed work-flow latency and processing rates through Fog optimization – Dynamic pricing model for SG platforms – Suitable data transfer primitives from users to Fog, and vice-versa

TABLE V: Works involving the IoE and Fog paradigms for energy and information managements.

Work	Findings
Okay & Ozdemir, 2016 [77]	They proposed a FC based smart grid model. SG integrates green power resources into the energy distribution system, control power usage and balance energy load. SG uses smart meters which are responsible for management of the electricity consumption.
Jalali et al., 2016 [78]	They compared the energy consumption of applications by using centralized clouds with applications that use nano data centers.
Byers & Wetterwald, 2015 [79]	They pointed out that the large deployment of smart metering systems involves millions of end points, which make mandatory the use of intelligent concentrators.
Yan & Su, 2016 [80]	They proposed a FC-like solution for the existing smart meter infrastructure. This solution can be incorporated into the existing smart meter infrastructure.
Nazmudeen, Wan & Buhari, 2016 [81]	They showed through analyses that using a FC architecture for SG systems would be highly beneficial.

Motivated by this consideration, in Table VI, we report a comparative synoptic overview of (the limited number of) works that (at least, partially) address this issue. An examination of these works corroborates the conclusion that, in principle, computing and networking-intensive applications that require real-time processing of spatially distributed environmental data may gain benefit from the integration of the pillar IoE and FoE paradigms. As detailed in the following subsections, these characteristics are retained, indeed, by four broad application areas of growing practical interest, namely, Internet of Energy, Smart City, Industry 4.0 and Big Data Streaming. We believe that, in the next future, these application areas could provide “killer” use cases for the proposed FoE paradigm of Section IV.

1) Internet of Energy: The Internet of Energy represents the new frontier for the efficient management of the global and ever increasing electricity demand. By design, it relies on the integration of the power grid with actuators, smart meters and WSNs. The final goal is to to cope with the (traditional) unreliability suffered by the current electric grid [90], [91].

A first application field envisioned for the Internet of Energy is represented by the (aforementioned) Smart Grid [90], [92]. By design, it comprises power networks empowered by intelligent energy load balancers, that may run on edge devices (like smart meters) [93]. The SG model allows the usage of various renewable energy sources (like solar energy, wind power, hydroelectric, radiant energy, geothermal energy and biomass), in order to supply heterogeneous IoE devices, such as, home appliances, micro-grids, sub-stations and sensor nodes [94]. The underlying networking infrastructure acts as glue: it allows the SG to perform remote metering by (possibly) exploiting the support of smart Energy Storage Devices [95], [96].

A second application field envisioned for the Internet of Energy is provided by the integration of Electric

TABLE VI: Recent QoS-oriented works involving the IoE and Fog paradigms. Servers:=S, Network devices:=Nd, Cloudlets:=Cl, Vehicles:=V, Master-Slave:=M-S, Peer-to-Peer:=P2P, Cluster:=Clust, Network Management:=NM, Resource Management:=RM, Power Management:=PM, Application Management:=AM, Data Management:=DM, Latency Management:=LM, Cost Management:=CM, CO_2 Management:=CO_2M, Content Distribution Network:=CDN, Power-line communication:=PLC, Radio Access Network:=RAN, Vehicular Network:=VN, Mobile Network:=MN, Long-Range Passive Optical Network:=LRPON.

Work	Type of Fog Nodes					Form of Nodal collaboration			Considered performance metrics	Service Level Objective								Application Environment							Security			
	S	Nd	Cl	Bs	V	M–S	P2P	Clust		NM	RM	PM	AM	DM	LM	CM	CO_2M	IoT	DCN	PLC	RAN	VN	MN	LRPON	D.Energy	DoS attack	Authent.	Privacy
Lee et al., [82]	✗					✗			Data (flow)	✗								✗										
Aazam et al., [23]	✗					✗			Context (user)		✗							✗										
Jalali et al., [78]	✗						✗		Time (computing)			✗							✗									
Zhu et al., [13]	✗								Context (user)				✗						✗									
Zeng et al., [24]	✗						✗		Time (communication, computing)		✗							✗										
Hong et al., [83]		✗					✗		Data (size)																			
Nazmudeen et al., [81]		✗				✗			Data (size)					✗						✗								
Aazam et al., [84]		✗							Data (size)	✗								✗						✗				
Cirani et al., [46]		✗							Context (application)	✗								✗							✗			
Dsouza et al., [49]			✗				✗		Context (application)	✗								✗								✗		
Cardellini et al., [85]			✗					✗	Context (application)				✗					✗						✗				
Yan et al., [35]				✗				✗	Context (user)				✗								✗							
Gu et al., [86]				✗			✗		Cost (deployment, communication)		✗							✗										
Truong et al., [16]				✗		✗			Context (application)	✗												✗						
Oueis et al., [37]				✗				✗	Time (deadline)						✗								✗					
Hou et al., [17]					✗			✗	Context (user)		✗											✗						✗
Ye et al., [36]					✗				Time (deadline)				✗									✗						
Oueis et al., [87]				✗				✗	Data (flow)		✗		✗										✗					
Al Faruque et al., [88]		✗						✗	Energy consumption			✗						✗										
Shi et al., [18]		✗					✗		Context (application)				✗										✗					
Giang et al., [19]		✗					✗		Data (flow)				✗					✗										
Intharawijitr et al., [39]	✗								Time (communication, computation)						✗								✗					
Peng et al., [40]							✗		Data (size)	✗											✗							
Hassan et al., [41]		✗						✗	Cost (execution, communication)				✗										✗					✗
Zhang et al., [89]			✗				✗		Cost (deployment)							✗								✗				
Deng et al., [38]		✗					✗		Data (size)				✗										✗					
Do et al., [67]		✗							Energy consumption								✗		✗									
Aazam et al., [20]	✗								Context (user)		✗							✗										
Datta et al., [21]		✗					✗		Context (user)					✗								✗						
Aazam et al., [22]	✗								Context (user)		✗							✗										
Gazis et al., [47]		✗							Context (application)		✗							✗										

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2017.2702013, IEEE Access

Vehicles (EVs) with SGs [97]. This integration is motivated by the fact that EVs may also act as storage devices and feed power back to the grid. By doing so, they could be utilized for coping with the intermittent nature of the renewable energy sources by enabling the on-line matching of the energy generation times to the corresponding consuming times [45], [90]. It is expected that the resulting Vehicular-to-Grid (V2G) technological platform will provide new mechanisms for storing and supplying electric power, communicating with the grid and delivering electric energy to the grid [97]. Fig. 7 reports a reference architecture for the Internet of Energy.

According to the reported architecture, the Internet of Energy model relies on the vertical integration of four main sub-systems, namely, the Perception layer, the Network layer, the Fog layer and the Control layer. The final goal is to give rise to a customizable grid system, that is capable to perform a smart real-time scheduling of the energy demands. Specifically, according to Fig. 7, we have that [92]:

(i) the *Perception layer* performs data gathering. It is composed by (possibly, heterogeneous) sensors, sensor gateways and actuators, that are used for acquiring information and performing thing control and identification. These devices must be capable to support D2D communication by exploiting the available Network layer;

(ii) the *Network layer* comprises a set of (possibly, heterogeneous) network infrastructures, in order to support Device-to-SG communication. Since the IoE devices could be randomly distributed over the SG, it is expected that the Network layer relies on short/medium-range wireless transmission technologies, like WiFi, UWB and/or Bluetooth [26];

(iii) the *Fog layer* is responsible for the management of the massive volume of metered data. Since the FNs are equipped with both computing and networking capabilities and operate in the proximity of the edge devices, their role is to manage information and energy. They must guarantee a reliable flow of energy, in order to timely store the right amount of energy at the right locations. For this purpose, a final task of the FNs is to monitor in real-time the quality-of-energy, in order to quickly switch and route the energy flow [92];

(iv) the *Control layer* provides the interface between the services provided by the Internet of Energy platform and the end users. Since the data produced by Internet-of-Energy applications are large and heterogeneous, the Control layer may rely on the support of the remote Cloud, in order to perform offline data analytics.

2) Smart City: The keyword Smart City (SC) refers to the integration of the Information and Communication Technology (ICT) and the IoE platform into the urban environment. The final goal is the "smart" management of the overall city asset [98]. This aims at improving the human life through an optimized, balanced and integrated provision of utilities, vehicular transportation and smart lighting [99]. Therefore, a new paradigm shift is at the basis of the SC model. It demands for the integration of service-oriented infrastructures, innovation services and communication infrastructures [100]. By doing so, it is expected that the SC model may foster and improve the social relations between citizens and government.

A quite general system architecture for the Smart City model is reported in Fig. 8 [99].

By design, this architecture relies on the integration of four main layers, namely, the Physical Resource layer, the Network layer, the proximate Fog layer and the remote Supervision layer. According to Fig. 8, these layers play the following roles [100]:

(i) the *Physical Resource layer* performs data acquisition. For this purpose, it is composed by a number of heterogeneous IoE devices, that gather information from different city scenarios, like vehicular mobility, smart buildings and smart energy generators. All these devices must be capable to support D2D communication by exploiting the infrastructure provided by the Network layer;

(ii) the *Network layer* guarantees data transport between the edge devices and the overall SC infrastructure. Since edge devices may be randomly distributed over the SC area, it is expected that the Network layer relies on Metropolitan Area Network (MAN) communication technologies, like the 3G/4G cellular and/or WiMax ones [26];

(iii) task of the *Fog layer* is the real-time aggregation and orchestration of big volumes of gathered data, as well as their (possible) pre-processing through filtering and/or fusion. For this purpose, the Fog nodes may perform light-weight analytics;

(iv) the *Supervision layer* makes possible the user-friendly interface of services offered by the Smart City ecosystem with user-specific queries/requirements. Since Smart City applications may produce huge sets of (possibly, noisy, heterogeneous and incomplete) meta-data, the Supervision layer may also include a remote centralized Cloud, in order to perform complex data analytics and deep learning [101].

3) Industry 4.0: The keyword Industry 4.0 has been quite recently introduced to indicate the fourth industrial revolution. It refers to the integration of a pool of emerging technical advancements in the realm of the ICT (like IoE, WSNs, Big Data, Fog/Cloud Computing

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2017.2702013, IEEE Access

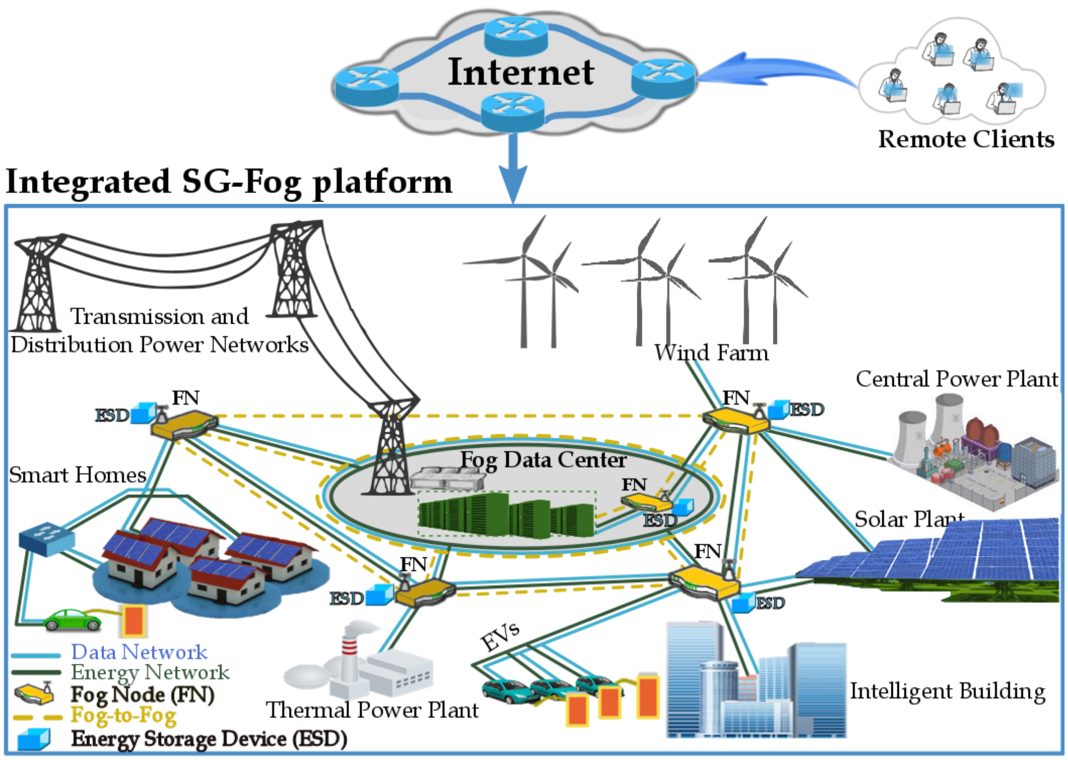


Fig. 7: Reference architecture for the Internet of Energy application scenario. EV:= Electrical Vehicle; SG:= Smart Grid.

and Mobile Internet) into manufacturing factories. For this purpose, in 2014, the strategic initiative "Industrie 4.0" has been proposed by the German government and, then, similar initiatives were launched by USA and China [102]. The core of the resulting Industry 4.0 paradigm is the "smart" factory, whose main building blocks are reported in Fig. 9 [14], [103].

By design, a "smart" factory requires a vertical integration of a four main subsystems, namely, the IoE-based Physical Resource layer, the Network layer, the proximate Fog layer and the remote Control layer. The final goal is to give rise to a reconfigurable manufacturing system, that may be used for the (possibly, simultaneous) production of heterogeneous products. Specifically, according to Fig. 9, we have that [104]:

(i) the *Physical Resource layer* is IoE-based and comprises smart things, like smart products, smart machines and smart conveyors. These things self-establish Thing-to-Thing (T2T) communication links by exploiting the corresponding Network layer. By doing so, they are capable to self-collaborate and self-organize, in order to attain a (pre-assigned) system-wide goal [105];

(ii) the *Network layer* provides, in turn, the communication services that are required by the underlying physical smart things, in order to implement the needed inter-thing negotiation mechanisms and communicate with the Fog layer. Due to the presence of mobile entities (like, automated guided vehicles and robots), the topology of a smart factory is expected to be highly time-varying. Hence, it is foreseen that the Network layer of Fig. 9 will rely on short/medium-range wireless networking technologies, like WiFi, UWB and/or Bluetooth [26];

(iii) thank to the virtualization of the underlying manufacturing things, it is expected that the *Fog layer* of Fig. 9 is capable to provide the scalable processing environment required by big data applications, in terms of computing, storage and networking resources. In fact, it is foreseen that, when operated, the smart manufacturing things at the Physical layer may produce massive streams of data, which require to be transferred to the Fog layer for further filtering and analytics;

(iv) finally, the *Control layer* allows remote people to access to the smart factory through Web-based portals and Internet gateways. In principle, it may also comprise a remote Cloud layer, in order to perform offline complex analytics on massive semi-

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2017.2702013, IEEE Access

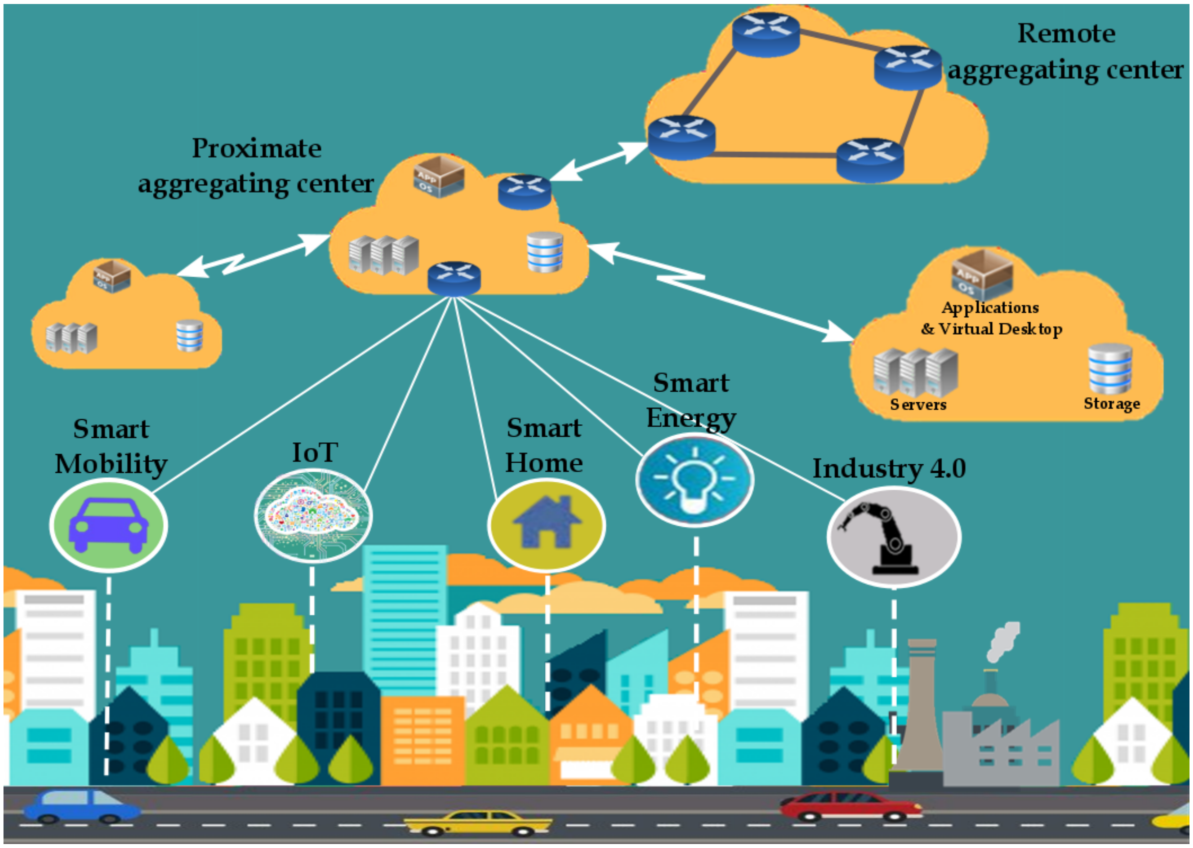


Fig. 8: Reference architecture for the Smart City application scenario. Source: www.impresamia.com.

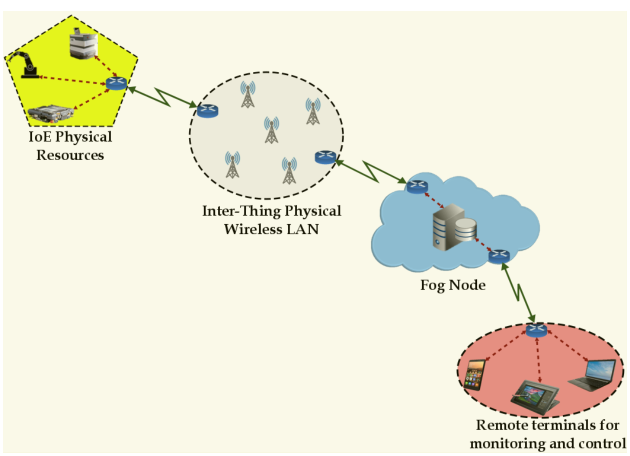


Fig. 9: The main building block of a "smart" factory.

permanent datasets. By doing so, people can access to the statistics provided by the Cloud and/or perform maintenance/diagnostic operations, even remotely through the Internet [106].

4) Big Data Streaming: Big Data Streaming (BDS) mobile computing has been recently proposed as a paradigm that exploits the integration of the Big Data, Cloud-based stream processing and broadband mobile Internet Networking paradigms [85], [107], [108]. Its specific target is the design and implementation of novel self-organizing spatially distributed networked computing platforms, in order to enable the real-time offloading and pervasive processing of environmental big data streams gathered by energy and bandwidth-limited wireless things (e.g., sensors, tablets, RFIDs, PDAs, smartphones). According to this target, the BDS paradigm relies on a "five Vs" formal characterization, that is, *Value* (huge value of gathered data that is scattered in massive data sets), *Volume* (huge amount of data to be processed), *Velocity* (fast rate of generation of new data), and *Volatility* (the gathered data must be communicated and processed in real time). The first four attributes are shared by all big data applications [74], [109], [110], while the last one (e.g., the Volatility) characterizes the big data streaming applications. This is due to the fact that the information conveyed by a stream of data deeply depends on both its time and space coordinates, so that, after gathering, the information value vanishes when the networking-plus-computing latency is beyond a suitable application-depending threshold [107].

Fig. 10 sketches the reference architecture of the BDS technological platform [108]. In principle, it is composed by five main blocks, namely, the *IoT layer*, the *radio access network*, the proximate *Fog layer*, the *Internet backbone* and the *remote Cloud layer*.

According to the reported architecture, big data streams are: (i) gathered by a number of spatially heterogeneous mobile/wireless devices scattered over the environment of interest; (ii) forwarded to proximate FNs over (typically, single-hop) WiFi/Cellular connections for local pre-processing (like, data compressing, fusion and filtering [111]); and, (iii) routed to Cloud-based remote data centers over (typically, multi-hop) Internet WANs for further post-processing.

By leveraging the reported architecture, it is expected that, in principle, the BDS paradigm may be capable to support three main classes of IoE-oriented applications, namely, Spatial Sensing (SS), Crowd-Sourcing (CS) and Data caching & Nomadic computing (DN) [108].

Goal of the SS-type applications is to provide Internet access to a (possibly, very large) number of heterogeneous sensors (like, RFIDs and biosensors), in order to enable the real-time exchange of data about the monitored environment. This requires, in turn, that a huge number of simultaneous transport connections is sustained without inducing time-consuming traffic congestion phenomena.

Under the CS realm, populations of non-professional users acquire environmental (typically, video/audio) data streams through own smartphones, and, then, share them in real-time by building up P2P transport connections. Hence, a key feature of this type of application is the collective real-time processing of locally gathered streams for spatial monitoring and/or infotainment services.

Finally, the DN applications rely on context-aware services for data caching and personal computing on-the-go. The goal is to allow users to share information in real-time by leveraging Fog-assisted social network platforms (like, Dropbox, iCloud, Facebook or YouTube). These applications require massive sets of inter-stream cross-correlation analytics, in order to quickly detect the occurrence of new social trends and/or anomalies [101].

To recap, since it is expected that the energy-saving support of the applications described in this section demands for an accurate characterization of the corresponding per-service resource usages, Table VII presents a synoptic (gross) indication of the networking QoS requirements that are expected to stem from the considered application fields.

B. Emerging open issues

The envisioned application areas and the related QoS requirements of Table VII open the doors to a set of open issues that should be suitably characterized, in order to allow the migration of the IoE and FoE paradigms from the theory to the practice. According to Fig. 11, we believe that the main foreseen open issues may be detailed as follows.

Main open issues on the Internet of Energy – The Internet of Energy paradigm of Fig. 7 introduces some novel peculiar attributes with respect to the traditional power grid, namely, reliability, heterogeneous topologies, real-time management and energy-sustainability. Therefore, in the Internet of Energy realm, main open issues regard [112]: (i) the composition, deployment and management of distributed services over pervasive SG-based infrastructures; (ii) the design of efficient and user-friendly mechanisms, that allow end-users to be capable to control the services provided by the Internet-of-Energy technological platform; (iii) the design of techniques for the dynamic allocation of the bandwidth, computing and storage resources available at the Physical layer of Fig. 7, in order to effectively cope with the fluctuations of the energy demand; and, (iv) the design of "ad hoc" deep learning-based algorithms and protocols for the management of massive data generated by millions of smart meters [101].

Main open issues on the Smart City – Smart cities will offer to the future citizens many non-traditional services, such as: (i) real-time traffic information; (ii) multi-agency coordination; (iii) generalized alerting services; (iv) quick responses to user's queries; (v) intelligent transportation and multi-modal ticketing services; and, (vi) dynamic interface with the public administration. As a consequence, we expect that, in the Smart City realm, main open issues may regard [99], [100]: (i) the design of shared government strategies, in order to plan and coordinate the overall city assets; (ii) the design of user-friendly representations of the recorded environmental data, that account for spatial, temporal and contextual aspects; and, (iii) the design of scalable optimization tools for distributed data analytics.

Main open issues on the Industry 4.0 – In a nutshell, the novel technical features introduced by the smart factory paradigm of Fig. 9 compared with the traditional factory model are [14]: (i) management of multiple heterogeneous small-lot products; (ii) dynamic routing of semi-worked products; (iii) convergence of the IoE, Fog and Cloud models; (iv) self-organization and self-managing of the working manufacturing machines; and, (v) generation, management and analysis of big data streams. As a consequence, in the Industry 4.0 realm, main open issues regard [104]: (i) the design

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2017.2702013, IEEE Access

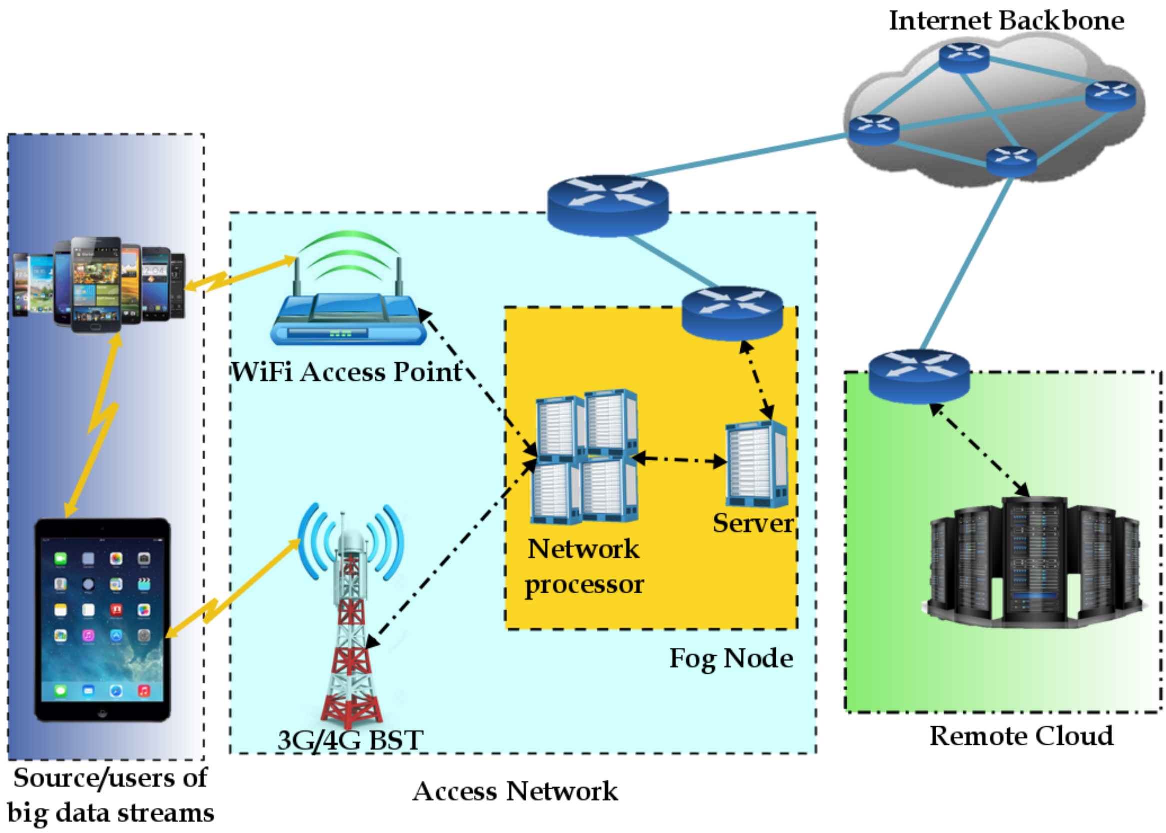


Fig. 10: Reference architecture for the Big Data Streaming scenario. Thick (resp., thin dashed-dotted) lines indicate TCP/IP (resp., Ethernet type) connections. BST:= Base StaTion.

TABLE VII: Expected networking QoS requirements for the application fields of Section III. A [3], [5], [6].

	Internet of Energy	**Smart City**	**Industry 4.0**	**Big Data Streaming**
Latency	≤ 200 (ms)	≤ 10 (ms)	≤ 5 (ms)	≤ 100 (ms)
Latency jitter	≤ 15 (ms)	≤ 3 (ms)	≤ 0.5 (ms)	≤ 10 (ms)
Packet loss rate	$\leq 10^{-2}$	$\leq 10^{-3}$	$\leq 10^{-4}$	$\leq 10^{-2}$
Bandwidth	≥ 50 (Kb/s)	≥ 2 (Mb/s)	≥ 200 (Kb/s)	≥ 10 (Mb/s)

of decision-making and negotiation mechanisms, in order to enforce inter-thing cooperation; (ii) the design of “ad hoc” broadband communication protocols for the localization and real-time transportation of massive streams of environmental data; (iii) the design of “ad hoc” analytic tools for filtering, fusion and analysis of the information conveyed by industrial big data streams; and, (iv) the design of adaptive factory-level controllers, that are capable to enforce self-organization and convergence to desired factory-wide targets.

Main open issues on the Big Data Streaming – The resource management of the technological infrastructures for the support of Big Data Streaming applications typically requires the real-time offloading of data and/or code to proximate and/or remote data centers through the available wireless access-plus-Internet networks, together with the corresponding real-time reconfiguration of the intra-data center computing-plus-networking resources. The ultimate goal of these actions would be the minimization of the overall inter/intra-data center computing-plus-networking energy consumptions under the QoS requirements of Table VII. This is the target of some recent management frameworks, such as S4 [113] and D-Streams [114]. However, these frameworks do not provide self-tuning of the employed networking-plus-computing resources to the time fluctuations of the input streams to be processed. Although the more recent

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2017.2702013, IEEE Access

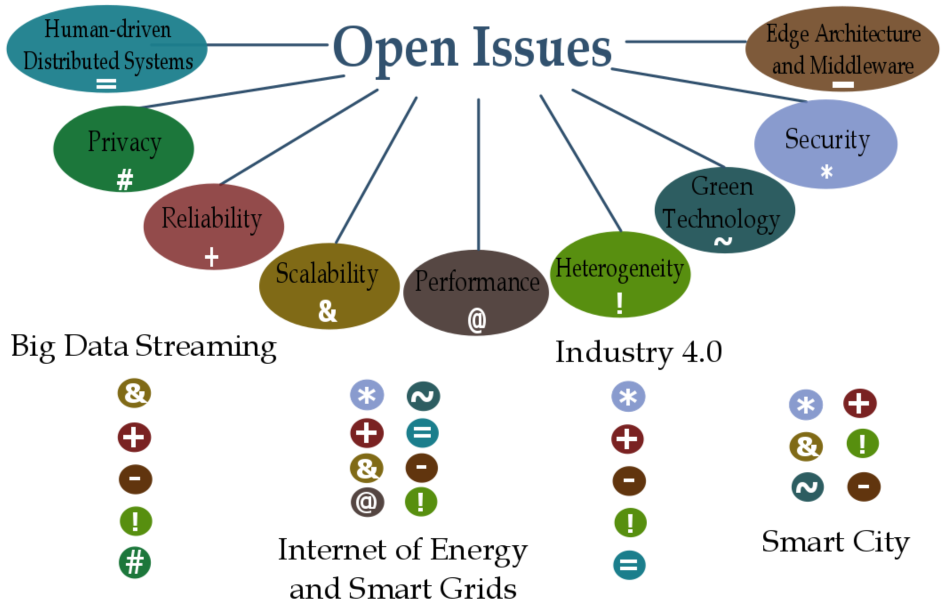


Fig. 11: Main application areas envisioned for the integration of the IoE and Fog paradigms and related open issues.

Time Stream [115] and PLAstiCC [116] management tools provide dynamic tuning of the intra-data center computing resources to the time-varying arrival rate of the input workload, they do not still consider the simultaneous scaling of the intra-data center network resources, and, furthermore, do not guarantee upper bounds on the resulting computing-plus-communication latencies. To recap, in the Big Data Streaming application scenario of Fig. 10, a still open issue is the design of integrated management tools that allow the adaptive and energy-efficient real-time reconfiguration of the computing-plus-networking virtual resources at the proximate/remote data centers and IoE devices.

Overall, all these reviewed open issues provide the motivation for the proposal of the FoE paradigm of Section IV.

IV. The proposed FoE paradigm

In principle, the challenges in Table I of the IoE model could be adequately addressed by the native attributes in Fig. 4 of the FC model. Table VIII stresses, indeed, the complementary features of these two pillar paradigms, and points out how the Fog could provide support to the IoE.

Motivated by this consideration, the rest of this Section IV details: (i) the main building blocks of the architecture of the proposed FoE technological platform (see Section IV-A); (ii) the role played by the virtual containers (see Section IV-B); and, (iii) the main functions of the corresponding FoE protocol stack (see Section IV-C).

A. The envisioned FoE technological platform and the supported service models

By design, the FoE paradigm aims at implementing the Fog-IoE integration fostered by Table VIII, in order to provide the technological support to the applications previously described in Sections III-A1, III-A2, III-A3 and III-A4. At this regard, we observe that all the (application-specific) technological platforms of Figs. 7, 8, 9, and 10 retain three common features. First, they rely on three-tier Device-Proximate Fog-Remote Cloud architectures. Second, they exploit single-hop WLANs and multi-hop WANs, in order to implement Device-Fog and Fog-Cloud connectivity, respectively. Third, in the case in which there are multiple proximate Fog nodes, these platforms are typically equipped with (possibly, wireless) backbones, in order to provide inter-Fog communication.

By accounting for these shared features, Fig. 12 reports the basic architecture of the virtualized technological platform for the support of the proposed FoE paradigm. Roughly speaking, the proposed FoE architecture is composed by the integration of the following six main building blocks:

- the *IoE layer*, where a number of (possibly, heterogeneous) things operate over multiple spatial

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2017.2702013, IEEE Access

TABLE VIII: Fog-IoE interplay: an opportunity for cooperation.

Fog attributes	Fog support to IoE
Deployment in the proximity of the devices	Operating at the network edge, Fog nodes may provide infrastructure-based support to IoE applications that demand for spatially distributed device deployment.
Context-awareness	Being located near to the served devices, Fog nodes may acquire context awareness in real-time, and exploit it for the support of latency and latency-jitter sensitive services, like interactive and/or monitoring services.
Support to the device mobility	Fog nodes may be arranged into spatial clusters, in order to serve mobile devices through single-hop links.
Reduced energy consumption	Inter-device communication may occur through (stable and energy-efficient) Device-Fog-Device up/down links in place of (intermittent and energy-hungry) D2D links.
Container-based dense virtualization	Virtual clones of the served devices may be dynamically packed into Fog servers as lightweight containers. The goal is to provide device augmentation on an on-demand basis, in order to dynamically support the resource-limited IoE devices.

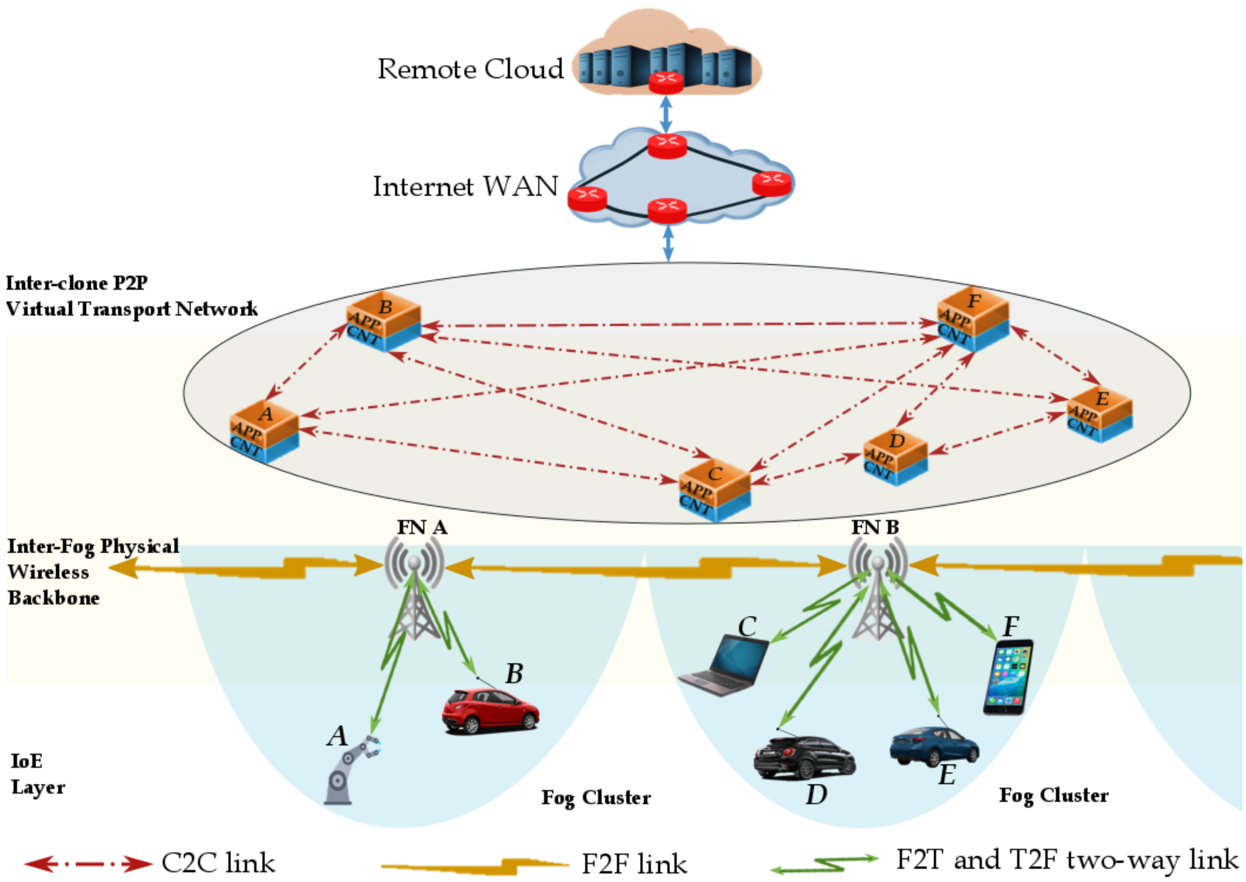


Fig. 12: The envisioned architecture for the FoE technological platform. Fog node A (resp., Fog node B) hosts FCL-A and FCL-B (resp., FCL-C, FCL-D, FCL-E and FCL-F) clones. FN:= Fog Node; FCL:= Fog CLone; P2P:= Peer-to-Peer; C2C:= Clone-to-Clone; F2F:= Fog-to-Fog; T2F:= Thing-to-Fog; F2T:= Fog-to-Thing; App:= Application code and libraries; CNT:= CoNTainer.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2017.2702013, IEEE Access

clusters. According to the IoE jargon, a thing is a resource-limited user device, that needs of resource augmentation, in order to execute its workload. A thing may be fixed, nomadic or even mobile (see Fig. 12);
- the *wireless (possibly, mobile) access network*, that supports Fog-to-Thing (F2T) and Thing-to-Fog (T2F) communication through TCP/IP connections running atop *IEEE802.11/15* single-hop links;
- a set of *inter-connected FNs*, that act as virtualized cluster headers;
- the (possibly, wireless) *inter-Fog backbone*, that provides inter-Fog connectivity and makes feasible inter-Fog resource pooling;
- the *Virtualization layer*, that allows each thing to augment its limited resources by exploiting the computing capability of a corresponding virtual clone. This last runs atop a physical server of the FN that currently serves the cloned thing;
- the resulting *overlay inter-clone virtual network*, that allows P2P inter-clone communication by relying on TCP/IP end-to-end transport connections.

Passing to describe the interplay of the aforementioned building blocks of Fig. 12, we observe that the remote Cloud is interconnected by a (multi-hop) Internet WAN to a set of virtualized FNs, that are spatially distributed over a wireless access network. Each FN is equipped with a (limited) number of virtualized physical servers, that are inter-connected by an intra-Fog (typically, Ethernet type) wired network. A FN covers a spatial area of diameter $Da\ (m)$ and serves a cluster of things. Being resource limited, each thing is augmented by a software clone, that runs in the serving FN and acts as a virtual server.

The role of the (typically, wireless and multi-antenna [117]) inter-Fog backbone of Fig. 12 is two-fold. First, it makes feasible the (aforementioned) horizontal dynamic scaling and pooling of the computing-plus-communication resources of the FNs (see the first row of Table II). Second, it allows each clone to migrate from a FN to another by tracking the spatial trajectory of the corresponding mobile thing.

The overall set of clones running atop all Fog servers constitutes an overlay P2P virtual network, that is composed by Clone-to-Clone (C2C) TCP/IP connections. For this purpose, intra-Fog wired Ethernet links and backbone-supported inter-Fog wireless links are used. Specifically, the former (resp., the latter) are used to sustain end-to-end transport connections among clones that run atop a same FN (resp., stop different FNs).

Passing to consider the service models supported by the FoE platform, two main remarks are in order. First, since the FNs of Fig. 12 may play the two-fold role of offloading and aggregating points for the traffic generated by the underlying things, the FoE paradigm is capable to support, by design, *all* the Up/Down Offloading, Aggregation and P2P service models of Section II-E. Second, we stress that a *main peculiar feature* of the proposed FoE paradigm is that the overlay network of Fig. 12 allows to move the implementation of the inter-thing links from the device-based physical bottom layer to the clone-based virtual upper layer of Fig. 12. This feature makes, in turn, feasible to replace unreliable, intermittent and mobility-affected D2D-based inter-thing physical links with reliable, static and TCP/IP-based inter-clone virtual transport connections. We anticipate that the numerical tests and performance comparisons of the next Section V support the actual effectiveness of this feature of the FoE platform.

B. The container-based virtualization of the IoE devices

Light-weight and fine-grain dynamic resource scaling is the key feature that makes appealing to resort to the container-based technology of Section II-D, in order to perform the virtualization of the FoE technological platform of Fig. 12.

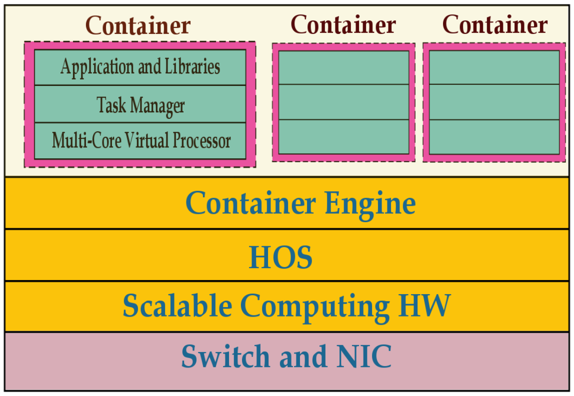
Motivated by this consideration, in Fig. 13(a), we report the main functional blocks of the virtualized architecture of the physical servers at the FNs [48], [70], [71].

At this regard, four main explicative remarks are in order.

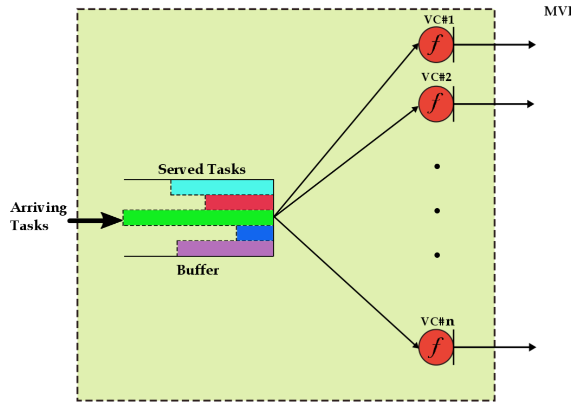
First, each server hosts a number: $N_{CNT} \geq 1$ of containers. All these containers share: (i) the server's Host Operating System (HOS); and, (ii) the pool of computing (e.g., CPU cycles) and networking (e.g., I/O bandwidth) physical resources done available by the CPU and Network Interface Card (NIC) that equip the host server. Task of the *Container Engine* of Fig. 13(a) is to dynamically allocate to the requiring containers the bandwidth and computing resources done available by the host server. For this purpose, the so-called *Weighted Processor Sharing (WPS)* scheduling discipline is typically implemented by the Container Engine [48], [71].

Second, each container plays the role of virtual clone for the associated physical thing. Hence, the container acts as a virtual processor and executes the tasks offloaded by the thing on behalf of it. For this purpose, each container is equipped with a *Multi-core Virtual Processor (MVP)*. This last comprises (see Fig. 13(b)): (i) a buffer, that stores the currently processed application tasks; and, (ii) a number $n \geq 1$ of (typically, homogeneous) *Virtual Cores (VCs)*, that run at the processing frequency f dictated by the Container Engine. Therefore, goal of the *Task Manager* of Fig. 13(a) is to allocate the pending application tasks over the set of virtual cores of Fig. 13(b) in a balanced and dynamic way. This is still done according to the aforementioned WPS scheduling discipline, so that the average frequency f_i at which is processed the i-th task equates [71]:

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2017.2702013, IEEE Access



(a)



(b)

Fig. 13: Container-based virtualization of a physical server equipping a Fog node. (a) Virtualized server architecture; (b) Architecture of a multi-core virtual processor. HW:= CPU HardWare; NIC:= Network Interface Card; HOS:= Host Operating System; MVP:= Multi-core Virtual Processor; VC:= Virtual Core; n:= Number of virtual cores; f:= Per-core processing frequency.

$$f_i = \frac{\varphi_i}{(\sum_{j \in \mathcal{T}} \varphi_j)} \times n \times f. \qquad (1)$$

In Eq. (1), we have that (see Fig. 13(b)): (i) n and f are the per-container number of virtual cores and the corresponding per-core processing frequency, respectively; (ii) $\mathcal{T}$ is the (typically, time-varying) set of tasks which are currently processed by the container; and, (iii) φ_i (resp., φ_j) is a positive coefficient, which fixes the relative priority of the i-th (resp., j-th) processed task. Hence, according to Eq. (1), the Task Manager may increase the processing frequency f_i of the i-th task by increasing the corresponding weight φ_i and/or by decreasing the number of the simultaneously served tasks (see Fig. 13(b)).

Finally, without loss of generality, we assume that the per-core and per-task processing frequencies f and f_i in Eq. (1) are measured in bit-per-second (e.g., (b/s)), while the task sizes are measured in bit. However, according to [118], the corresponding numbers s and s_i of CPU cycles per second may be directly computed as in:

$$s = \delta \times f, \quad \text{and} \quad s_i = \delta \times f_i. \qquad (2)$$

In Eq. (2), δ (measured in CPU cycles-per-bit, e.g., $(CPU\ cycles/b)$) is the so-called processing density of the running application. It fixes the (average) number of CPU cycles per processed bit, so that its value increases with the computing intensity of the considered application. For illustrative purpose, Table IX lists the (numerically evaluated) range of values of the processing density of some test applications of practical interest [118].

C. Management of the virtualized FoE technological platform: protocol stack and implemented QoS services

In order to suitably orchestrate the overall technological FoE platform of Fig. 12, we have designed and implemented in software the FoE protocol stack of Table X. It relies on the suitable integration of some QoS resource managers, that have been recently proposed in the literature for the distributed self-management of multi-tier virtualized networked computing platforms. Specifically, Table X overviews: (i) the layered architecture of the FoE protocol stack; (ii) the services offered by each protocol layer; and, (iii) the corresponding adopted solutions. These last have been implemented in software under the umbrella of the GAUChO research project [119].

Specifically, according to the FoE platform of Fig. 12, the corresponding protocol stack of Table X comprises the following four hierarchically organized layers:

- *IoE layer* – It provides both: *(a)* T2F access; and: *(b)* F2T broadcast services.
  (a) The T2F access service is implemented by resorting to the reservation-based access protocol developed in [124]. It exploits a Network Utility Maximization (NUM) approach, in order to provide collision-free access to the things served by a same FN. For this purpose, the implemented protocol dynamically allocates access time-windows and access rates to the requiring things on the basis of: (i) the volume of data to be uploaded, (ii) the per-thing available energy; and, (iii) the per-connection fading level.
  (b) The F2T broadcast service is implemented according to the solution presented in [50]. It

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2017.2702013, IEEE Access

TABLE IX: Numerically measured processing densities of some test applications [118].

Application type	Application density ($CPU cycles/b$)
Face recognition	2339 – 31680
400 frame video game	2640
Virus scanning	32946 – 36992
Video transcoding	200 – 1200

TABLE X: The FoE protocol stack: planned services and implemented QoS solutions.

Protocol Layers	Planned Services	Implemented QoS solutions
Cloud Layer	– Dynamic management of the running applications over the overall three-tier Cloud/Fog/IoE infrastructure.	– Dynamic orchestration of streaming applications of [120].
Overlay Layer	– Management of the migration bandwidth. – Clone migration.	– Energy-efficient bandwidth manager of [121]. – Follow-Me-Cloud dynamic solution of [122], [123].
Fog Layer	– Inter-Fog traffic management. – Management of the intra-Fog virtualized platforms.	– Cognitive solution of [124]. – Integrated resource management of [125].
IoE Layer	– Fog-to-Thing wireless broadcast. – Thing-to-Fog wireless access.	– Adaptive solution of [50]. – Dynamic reservation-based solution of [124].

periodically profiles the throughput sustained by the ongoing F2T TCP/IP connections and, then, dynamically adjusts the corresponding transmission parameters, in order to maximize both the energy and bandwidth efficiencies, under hard constraints on the per-connection minimum throughput and maximum tolerated delay-jitter.

- *Fog layer* – It performs both: *(a)* energy-efficient management of the networking and computing physical resources equipping each FN; and, *(b)* energy-efficient management of the inter-Fog traffic conveyed by the wireless backbone of Fig. 12.
  (a) In order to accomplish the first task, the integrated resource manager developed in [125] is implemented. It jointly performs traffic admission control, load balancing, flow control and dynamic CPU speed scaling. The pursued objective is the minimization of the overall energy consumed by each FN, under hard upper limits on the resulting per-task processing delays.
  (b) The context-aware scheduler developed in [124] is implemented, in order to control the bi-directional inter-Fog traffic over the wireless backbone of Fig. 12. The scheduler operates on a Time Division Duplex way and resorts to a cognitive data-fusion approach, in order to maximize the utilization of the backbone bandwidth under hard constraints on the per-connection packet collision rates.
- *Overlay layer* – It supports the overlay inter-clone P2P network of Fig. 12 by: *(a)* sustaining the inter-Fog clone migration; and, *(b)* dynamically managing the required migration bandwidth.
  (a) Clone migration is supported by the implementation of the so-called *Follow-Me-Cloud* framework [122], [123]. It comprises the signaling protocol and the associated logic, in order to allow "live" inter-Fog clone migration in response to the thing mobility.
  (b) The dynamic management of the required migration bandwidth is accomplished by implementing the bandwidth manager deployed in [121]. It minimizes the energy consumed by clone migrations under hard bounds on the corresponding migration times and service downtimes.
- *Cloud layer* – It orchestrates the overall Cloud-Fog-IoE platform of Fig. 12 on the basis of the specific features and QoS requirements of the

running applications. Hence, the solutions to be implemented at this layer must be "ad hoc" tailored on the expected attributes of the supported applications. The tested FoE prototype implements the *VTube* services detailed in [120]. They provide a set of *YouTube*-like service primitives for the real-time P2P sharing of streaming contents (like video clips, games and interactive multimedia books) over Fog-supported mobile content delivery networks [126].

The next Section V corroborates the actual effectiveness of the adopted solutions for the implementation of the protocol stack of Table X by presenting the tested performance of a FoE-based prototype.

V. A proof-of-concept case study: the V-FoE testbed

Roughly speaking, the overall FoE technological platform is composed by two distinct (although interconnected) sub-systems. The first one comprises the virtualized networked computing technological platform that equips each FN, e.g., the intra-Fog platform sketched in Fig. 13. The second sub-system covers the three-tier IoE-Fog-Cloud infrastructure, e.g., the inter-Fog platform sketched in Fig. 12. Besides the IoE devices, FNs and remote Cloud, this sub-system comprises also the underlying networking infrastructure, that is, the mobile access network, the inter-Fog wireless backbone, the overlay inter-clone virtual network and the Internet WAN.

The energy and delay-efficient management of the intra-Fog platform has been the specific focus of a number of quite recent contributions by the authors (see, for example, [32], [50], [69], [108], [125], [127], [128], [129], [130]). These contributions explore various solutions for the adaptive orchestration of the intra-Fog virtualized resources, under a number of computing and networking setups. Hence, in the sequel of this section, we focus on the performance tests and comparisons of the inter-Fog sub-system of Fig. 12, in order to check the actual effectiveness of the FC paradigm in supporting resource-limited wireless/mobile IoE devices.

For this purpose, the remaining part of this Section V: (i) discusses the motivations for the performed tests and presents the considered testing scenario; (ii) describes the main features of a small-scale FoE prototype (e.g., the V-FoE prototype), that has been implemented under the umbrella of the (currently on-going) GAUChO research project [119]; (iii) tests the energy and delay performance of the V-FoE prototype under various mobility scenarios; and, (iv) finally, compares the obtained V-FoE performances against the corresponding ones of a benchmark platform, that does not exploit Fog and/or Cloud infrastructures.

A. Motivations and goals of the performed tests and comparisons

In principle, inter-thing communication could be implemented by entirely relying on the P2P service model of Section II-E. For this purpose, D2D single-hop physical links among the communicating things may be built up at the IoE layer by exploiting short-range *IEEE802.11/15* transmission technologies, like WiFi, UWB and Bluetooth [25], [126]. These physical links operate in the "ad hoc" mode, and, then, do not require the support of Fog and/or Cloud infrastructures (see the bottom part of Fig. 6). However, we note that [25]: (i) due to fading and path-loss, the energy consumption of D2D links increases with the inter-thing distance in an (at least) cubic way; (ii) due to thing mobility, D2D mobile links are intermittent, and their average failure rates typically increase with the average thing speeds [131]; and, (iii) under the D2D model, the initiator thing needs, at first, to discover proximate things, and, then, must perform task distribution, thing synchronization and task retrieval. These operations may induce large service delays, especially when, due to the intermittent nature of the D2D connections, they abort several times before completing.

As anticipated at the end of Section IV-A, in principle, the inter-clone overlay network of Fig. 12 may be used, in order to cope with the aforementioned limitations of the D2D "ad hoc" communication model. This overlay network allows to move the implementation of the inter-thing links *from* the unreliable, D2D-based and energy-hungry IoE physical layer *up* to the reliable, TCP/IP-based and energy-efficient virtual overlay layer. By doing so, since the overlay C2C communication platform replaces the corresponding underlay D2D one, we expect that, in principle, the following two may benefits are attained:

- the mitigation of the aforementioned limitations of the "ad hoc" D2D communication platform through the utilization of stable (e.g., not intermittent) and energy-efficient (e.g., no mobility affected) intra-Fog Ethernet and inter-Fog backbone links; and,
- the reduction of the delays for the service discovery and setup.

B. Modeling the simulated framework: the V-FoE testbed

Being the Fog paradigm still in its infancy, large-scale real-world Fog infrastructures are not currently available for test purposes. Hence, in order to corroborate the aforementioned expectations, we have emulated in software a small-scale FoE prototype, namely, the *Vehicular FoE (V-FoE)* test-bed. It provides a proof-of-concept of the proposed FoE protocol stack by implementing (in software) the resource orchestration and management solutions in the last column of Table X.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2017.2702013, IEEE Access

Utilized simulation toolkit – For this purpose, we have adopted the (recently deployed) *iFogSim* toolkit [132]. As shortly described in the sequel, it natively retains three main features that allow a (quite direct) integration of the FoE protocol stack.

First, the *iFogSim* toolkit allows the simulation of FNs and IoE devices by tuning their computing, communication and storage capabilities, such as: (i) the number of computing cores and their CPU speed-vs.-computing power profiles; (ii) the bandwidths of their NICs and the corresponding transmission rate-vs.-communication power profiles; and, (iii) the available RAM for task storage.

Third, under the so-called *Edge-ward placement* mode [132], the *iFogSim* toolkit allows to implement and tune various resource orchestration policies, in order to attain the most energy-efficient allocation of the workload over the overall spectrum of the available IoE devices, FNs and remote Clouds.

Test scenario – The considered test scenario refers to a crowd-sourcing application, that involves end-users on board of vehicles. Specifically, as in [120], this scenario considers the real-time sharing of environmental video sequences. They are acquired on-the-fly by non-professional users, which are equipped with smartphones and move on board of vehicles over urban areas. The smartphones are assumed to be equipped with *VTube*-type APIs [120], so that they may launch P2P video streaming sessions when the vehicles come in contact. In the simulated scenario, by design, we have that: (i) two vehicles come in contact when they move over the same cluster (e.g., they are served by the same FN; see Fig. 12); (ii) after becoming in contact, two vehicles may establish a new P2P session with probability 0.5, provided that they are not already involved in other on-going P2P sessions; and, (iii) the time is slotted, and T_{SLT} (s) is the slot time.

After the launching, a session goes on, even while the involved vehicles move away to different clusters. The SEssion Duration T_{SED} and the Inter-Session time Interval T_{ISI} are randomly distributed over the time intervals: 600 $T_{SLT} - 1000$ T_{SLT} , and: 1100 $T_{SLT} -$ 1400 T_{SLT} , respectively.

The maximum vehicle speed is $v_{MAX} = 50$ (Km/h), and the considered average speeds are $\overline{v} = 5, 15, 25, 35$ and 45 (Km/h). The total number of simulated vehicles is $N_{VHC}^{(TOT)} = 260$. They are evenly distributed over $N_{CLS} = 13$ exagonal spatial clusters of diameter $Da = 650$ (m), which are arranged over concentric spatial rings.

Simulated mobility model – As in [133], vehicle mobility is simulated according to the so-called *Markovian random walk with random positioning.* According to this model, at the beginning of the time slots, each vehicle moves to a randomly selected neighborhood target cluster with probability α, or stays in the current cluster with probability $(1 - \alpha)$. After the selection of the target cluster, a point inside it is randomly chosen and the vehicle moves to it. By doing so, we have numerically ascertained that, in each time slot, one half of the simulated inter-thing (e.g., inter-vehicle) TCP/IP connections involves vehicles that are traveling over different clusters. Furthermore, according to [133], the inter-cluster transition probability α and the per-cluster average number of vehicle $\overline{N}_{VHC}$ may be accurately approximated by the following formulas:

$$\alpha = \overline{v}/v_{MAX}, \tag{3}$$

and

$$\overline{N}_{VHC} = \frac{1}{2} \times A_{JAM} \times (1 - \alpha) \times \mathcal{S}u, \tag{4}$$

where Su (m^2) is the cluster area, and A_{JAM} $(vehicle/m^2)$ is the per-cluster maximum spatial density of vehicles when vehicular congestion phenomena occur.

Power profiles of the simulated computing nodes – According to Fig. 12, each spatial cluster is served by a FN. This last comprises $N_{SER} = 7$ homogeneous quad-core Dell PowerEdge-type physical servers, which are equipped with 3.06 GHz Intel Xeon CPU and 8 GB of RAM. The per-server maximum and static (e.g., idle) power consumptions are [9]: $P_{SER}^{(MAX)} = 228$ (W), and $P_{SER}^{(STATIC)} = 118$ (W), respectively. A commodity wired Giga Ethernet switch provides intra-Fog connectivity. Each server may host up to $N_{CNT}^{(MAX)}$ Docker-type containers [48] of size: $S_{CNT} = 30$ (Mb). Each container clones a user smartphone (e.g., a thing) and, according to Fig. 13(b), it is equipped with a virtual processor with n_{COR} homogeneous virtual cores. Hence, according to the general model reported in [134] for the power consumption of (possibly, virtualized) multi-core processors, the average computing power $P_{CMP}^{(SER)}$ (W) wasted by a multi-core container may be modeled as follows:

$$P_{CMP}^{(SER)} = \frac{P_{SER}^{(STATIC)}}{N_{CNT}^{(MAX)}} + (1 - \rho_{COR}^{(SER)}) \times \frac{(P_{SER}^{(MAX)} - P_{SER}^{(STATIC)})}{N_{CNT}^{(MAX)}} \times n_{COR} \times \left(\frac{f_{SER}}{f_{SER}^{(MAX)}} \right)^{\gamma}. \tag{5}$$

In Eq. (5), we have that: (i) f_{SER} (bit/s) (resp., $f_{SER}^{(MAX)}$ (bit/s)) is the per-virtual core average (resp., maximum) processing frequency; (ii) $\gamma \cong 3$ is a dimension-less power exponent; and, (iii) $\rho_{COR}^{(SER)}$ is the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2017.2702013, IEEE Access

TABLE XI: Power reduction factors from [134].

Processor type	Power reduction factor $\rho_{COR}^{(SER)}$		
	2 Cores	3 Cores	4 Cores
Intel SpeedStep @ $2.0GHz$	6 %	7 %	8 %
Intel SpeedStep @ $2.5GHz$	30 %	30 %	30 %
AMD Cool 'n' Quiet @ $2.5GHz$	6 %	7 %	8 %

so-called power reduction factor. According to [134], it is formally defined as the fraction of the total consumed power that is shared by the processing cores for common target operations. As illustrated in Table XI, this fraction depends on both the power features of the considered multi-core processor and the number n_{COR} of processing cores. Its typical values fall into the range 6%−30%, and tend to somewhat increase with the number n_{COR} of utilized cores.

Before proceeding, we point out that, in principle, the same power model of Eq. (5) may be also used for the evaluation of the computing power: $P_{CMP}^{(MOB)}$ (W) consumed by each mobile user device. However, since the most part of the current IoE devices is still single-core and not virtualized, Eq. (5) simplifies to [134]:

$$P_{CMP}^{(MOB)} = P_{MOB}^{(STATIC)} + \\ + \left(P_{MOB}^{(MAX)} - P_{MOB}^{(STATIC)}\right) \times \left(\frac{f_{MOB}}{f_{MOB}^{(MAX)}}\right)^{\gamma}. \quad (6)$$

From a formal point of view, Eq. (6) is obtained by posing: $N_{CNT}^{(MAX)} = n_{COR} = 1$, and: $\rho_{COR}^{(SER)} = 0$ into Eq. (5).

Power profiles of the simulated TCP/IP connections – The simulated Vehicle-to-Fog, Fog-to-Vehicle and Fog-to-Fog wireless channels of Fig. 12 are assumed to be affected by frequency-flat block-type Rice fading and, according to [135], are assumed to be supported by the *IEEE802.11b WiFi* technology. The Rice factor of the mobile Vehicle-to-Fog and Fog-to-Vehicle channels is 7.4 *(dB)*, while the Rice factor of the static inter-Fog wireless backbone is 17 *(dB)*. Furthermore, we assume that all the resulting wireless/wired end-to-end transport-layer connections of Fig. 12 implement the TCP NewReno protocol, in order to guarantee reliability, even in the presence of fading/mobility/traffic congestion-induced connection failures [135]. Therefore, according to the results of the power analysis reported, for example, in [50] and [108], the average power P_{NET} (W) consumed by a TCP connection is related to the corresponding average transport throughput R_{NET} (b/s) as in:

$$P_{NET} = \Lambda \times (R_{NET} \times \overline{RTT})^{\eta} + P_{NET}^{(SETUP)}. \quad (7)$$

In Eq. (7), we have that: (i) η is a dimension-less positive exponent; (ii) $P_{NET}^{(SETUP)}$ (W) is the static power consumed by the connection setup; (iii) $\overline{RTT}$ (s) is the average round-trip-time of the considered connection; and, (iv) Λ (W/b) is the average dynamic power consumed by the connection on a per-bit basis. Table XII points out that the actual values of Λ, $\overline{RTT}$, η, and $P_{NET}^{(SETUP)}$ depend on the power-delay features of the utilized wireless/wired transmission technologies [135].

Energy wasted by the live migration of clones – By definition, the average energy $\mathcal{E}_{CLONE}^{(MIG)}$ (J) consumed by the inter-Fog migration of a clone over the wireless backbone of Fig. 12 equates the product: (network power) by (migration time). Therefore, by leveraging Eq. (7), we have that [69], [136]:

$$\mathcal{E}_{CLONE}^{(MIG)} \stackrel{\text{def}}{=} P_{NET} \times T_{MIG} = \\ = P_{NET} \times \left(\frac{S_{CNT} \times (1+OVH)}{R_{NET}}\right) \times (1+\overline{FN}_{CON}), \quad (8)$$

where: (i) the (dimension-less and positive) coefficient: OVH accounts for the migration-induced traffic overhead [69], [136]; and, (ii) the (non-negative) factor: $\overline{FN}_{CON}$ is the per-connection average number of failures, e.g., the average number of times that an on-going connection fails before completing. We anticipate that $\overline{FN}_{CON}$ depends on the power-delay profiles of the considered wireless/wired transmission technologies, as well as on the considered service and mobility models (see the next Section V-C).

C. Performance results and comparisons

The numerical results of the simulated V-FoE test-bed report the per-connection average consumed energies and the resulting round-trip-times of the P2P inter-clone overlay virtual network of Fig. 12. Specifically, the reported energy values account for: (i) the support of the instantiated Vehicle-to-Fog, Fog-to-Vehicle and Clone-to-Clone wireless/wired links; (ii) the processing of the workload by all involved mobile/fixed computing nodes; and, (iii) the support of the inter-Fog mobility-induced clone migrations.

Reference benchmark – For comparison purpose, we have also implemented in software and simulated a benchmark testbed, e.g., the *Vehicular D2D (V-D2D)* testbed. It operates under the same vehicular scenario previously described for the V-FoE test-bed, but, according to [25], [137], it utilizes only "ad-hoc" D2D *IEEE802.11b* single-hop links for the support of Vehicle-to-Vehicle TCP/IP transport connections.

At this regard, four main remarks are in order.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2017.2702013, IEEE Access

TABLE XII: Main default parameters of the simulated V-FoE testbed. The subscripts WD, BB and WL denote WireD (e.g., intra-Fog), BackBone-supported and WireLess (e.g., Vehicle-to-Fog, Fog-to-Vehicle and Vehicle-to-Vehicle) TCP/IP connections, respectively.

Parameter setting		
$T_{SLT} = 500\ (ms)$	$v_{MAX} = 50\ (Km/h)$	$N_{VHC}^{TOT} = 260$
$OVH = 0.2$	$N_{CLS} = 13$	$Da = 650\ (m)$
$\eta_{WD} = 1.1$	$\eta_{BB} = 2.1$	$\eta_{WL} = 3.5$
$\Lambda_{WL} = \Lambda_{BB} = 11.5\ (mW/Mb)$	$\Lambda_{WD} = 4.5\ (mW/Mb)$	$\overline{RTT}_{WD} = 0.7\ (ms)$
$R_{WD}^{(MAX)} = 4 \times R_{BB}^{(MAX)} = 500\ (Mb/s)$	$R_{WL}^{(MAX)} = 8.5\ (Mb/s)$	$\overline{RTT}_{BB} = 12\ (ms)$
$P_{SER}^{(STATIC)} = 118\ (W)$	$P_{SER}^{(MAX)} = 228\ (W)$	$P_{NET,WD}^{(SETUP)} = 18\ (mW)$
$f_{SER}^{(MAX)} = 9.5\ (Mb/s)$	$f_{SER} = 2.4\ (Mb/s)$	$P_{NET,WL}^{(SETUP)} = P_{NET,BB}^{(SETUP)} = 525\ (mW)$
$\rho_{COR}^{(SER)} = 0.06$	$\delta = 500\ (CPU cycles/b)$	$S_{CNT} = 30\ (Mb)$
$N_{SER} = 7$	$N_{CLS} = 13$	$\overline{v} = 5, 15, 25, 35, 45\ (Km/h)$
$P_{MOB}^{(MAX)} = 0.2\ (W)$	$P_{MOB}^{(STATIC)} = 0.12\ (W)$	$\gamma = 3$
$f_{MOB}^{(MAX)} = 0.9\ (Mb/s)$	$f_{MOB} = 0.25\ (Mb/s)$	$n_{COR} = 2$

First, the (general) power-vs.-rate model of Eq. (7) applies also to WiFi-supported D2D transport connections. However, in this case, the resulting per-connection round-trip-time $\overline{RTT}$ becomes quite sensitive on the corresponding (generally, time-varying) inter-vehicle distance $d\ (m)$ and tends to scale up/down proportionally to it, that is [25], [137], [135]:

$$\overline{RTT} \propto d. \tag{9}$$

Second, the CPU power consumed by a device engaged into a D2D connection may be still evaluated through Eq. (6).

Third, in order to carry out fair performance comparisons, the traffic flows conveyed by all the simulated TCP/IP connections are randomly scaled and cyclically delayed versions of the mother traffic trace in Fig. 14. It reports the normalized I/O traffic flow actually measured from four RAID volumes of an enterprise data center in Microsoft [138]. In the carried out tests, the actual peak traffic values are set to 80% of the maximum throughput $R_{NET}^{(MAX)}$ of the corresponding TCP/IP connections.

Fourth, in the FoE paradigm, T2F access is managed by the context-aware reservation-based protocol proposed in [124], that, by design, guarantees collision-free access (see Table X and the related text). Hence, in order to perform fair comparisons, the simulation of the WiFi-supported V-D2D benchmark testbed has been carried out under the assumption that the utilized Carrier Sensing Multiple Access with Collision Avoidance (CSMA/CA) protocol guarantees collision-free (e.g., multiple access interference-free) communication. Although this assumption may be somewhat optimistic and may tend, indeed, to over-estimate the actual performance of the benchmark V-D2D testbed, we anticipate that the numerical plots of Figs. $15-18$ corroborate the performance superiority of the proposed V-FoE platform.

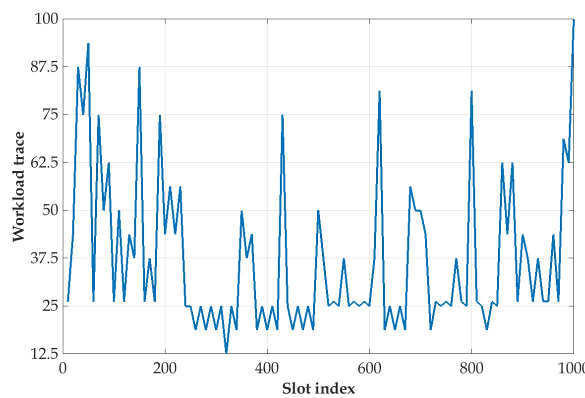


Fig. 14: Normalized sample trace of an I/O traffic flow from an enterprise data center in Microsoft [138].

Obtained numerical results – An examination of the V-FoE energy curves of Fig. 15 gives rise to two remarks. First, since the per-Fog number of the turned ON physical servers decreases for growing values of the per-server virtualization capacity $N_{CNT}^{(MAX)}$ (see Eq. (5)), all the V-FoE energy plots of Fig. 15 decrease with $N_{CNT}^{(MAX)}$ at fixed average vehicle speed $\overline{v}$. Therefore, since the virtualization density of the container-based virtualization technology is higher than the corresponding one of the VM-based technology (see Section II-D), a first conclusion is that the former technology is more energy-efficient than the latter one. Second, we have numerically ascertained that, in the simulated scenario, the average number of performed clone migrations increases of about 4.2 times by passing from $\overline{v} = 5$ (Km/h) to $\overline{v} = 45$ (Km/h). This is the reason why the V-FoE energy curves of Fig. 15 scale up of about 21% for increasing values of the average speed of the simulated vehicles.

Passing to consider the V-FoE - vs.- V-D2D performance comparison, Fig. 16 points out that: (i) the V-D2D test-bed exhibits values of the average number of the per-connection failures that are higher than the corresponding ones of the V-FoE platform; and, (ii) a similar conclusion holds for the rates of the increment of the average numbers of the per-connection failures with the vehicle speed. Specifically, an examination of the bar plots of Fig. 16 shows that the average number of the per-connection failures of the V-D2D (resp., V-FoE) testbed increases of about 3 times (resp., 1.5 times) by passing from $\overline{v} = 5$ (Km/h) to $\overline{v} = 45$ (Km/h). Therefore, the carried out tests lead to two main conclusions. First, due to the presence of the Fog infrastructure, the connection failures suffered by the V-FoE testbed are mainly due to (sporadic) traffic congestion phenomena in the access network. Second, due to its "ad hoc" (e.g., infrastructure-free) nature, the benchmark V-D2D testbed is very sensitive on: (i) the fading and path-loss impairments; and, (ii) the mobility-induced increments of the average distances of the sustained D2D connections.

This performance trend is confirmed by the bar plots of Figs. 17 and 18, that open, in turn, the doors to three additional considerations.

First, the V-FoE testbed is more energy efficient of the benchmark V-D2D one, and the measured per-connection average energy gaps are around 20%, 24%, 26.7%, 29.9%, and 33% at $\overline{v} = 5,\ 15,\ 25,\ 35,$ and 45 (Km/h), respectively (see Fig. 17).

Second, the increment of the energy consumed by the V-FoE connections is almost entirely induced by the corresponding increment of the average number of clone migrations. This last, in turn, is induced by the increment of the average vehicle speed.

Third, since live migrations of clones entail, by design, very limited service interruptions [69], [136], the corresponding average-round-trip times of the V-FoE connections are almost insensitive on the vehicle speed and remain around 22 $(ms) - 23$ (ms) (see Fig. 18). On the contrary, due to the increasing propagation delays and failure-induced TCP re-transmissions, Fig. 18 shows that the corresponding average round-trip-time of the V-D2D connections quickly scales up with the values of the average vehicle speed, and it passes from 26.5 (ms) at $\overline{v} = 5$ (Km/h) to 78.5 (ms) at $\overline{v} = 45$ (Km/h).

Overall, the reported comparative performance results confirm the (aforementioned) expectation about the improved delay and energy efficiencies of the of proposed FoE technological platform of Fig. 12.

VI. ONGOING RESEARCH PROJECTS ON FOE-RELATED TOPICS

FoE is a new paradigm, so that, at the best of the authors' knowledge, the number of research projects specifically tailored on this model is still vanishing. Hence, in this section, we describe some ongoing projects and research initiatives that, in our opinion, are mainly related to some main topics featured by the FoE paradigm.

The PRIN2015 project: "A Green Adaptive Fog Computing and Networking Architecture (GAUChO)" [119] aims at designing a novel distributed and heterogeneous architecture, that is capable to functionally integrate and jointly optimize FC and network functions onto a same platform. In addition, the development of suitable analytic methods and the definition of appropriate techniques enable extra relevant characteristics of the GAUChO platform, including ubiquity, decentralized management, cooperation, proximity to the users, dense geographical distribution, efficient support for mobility and real-time applications.

The "Vehicular Fog energy-efficient QoS mining and dissemination of multimedia Big Data streams (V-Fog)" project by Sapienza University of Rome [139] is a Fog-related project that aims at defining, designing and validating integrated resource-management and data-mining distributed adaptive algorithms for vehicular networks. The V-Fog final goal is the energy-efficient support of real-time BDS applications, such as multimedia human activity recognition and infotainment interactive services. Moreover, the actual (still unexplored) transport capabilities promised by the novel paradigm of multipath-TCP is investigated, while a cognitive approach is pursued for the integrated design of the V-Fog architecture.

The project: "TROPIC: Distributed computing, storage and radio resource allocation over cooperative femtocells" [140] proposes a communication paradigm for the ever-increasing ubiquitous wireless access traffic demands. CC services queried by smarphones could be moved from remote server farms to FNs, so to improve user experience, latency, and download/upload speed.

As previously remarked, a goal of the FoE paradigm is to develop Fog-aided IoE architectures that enable

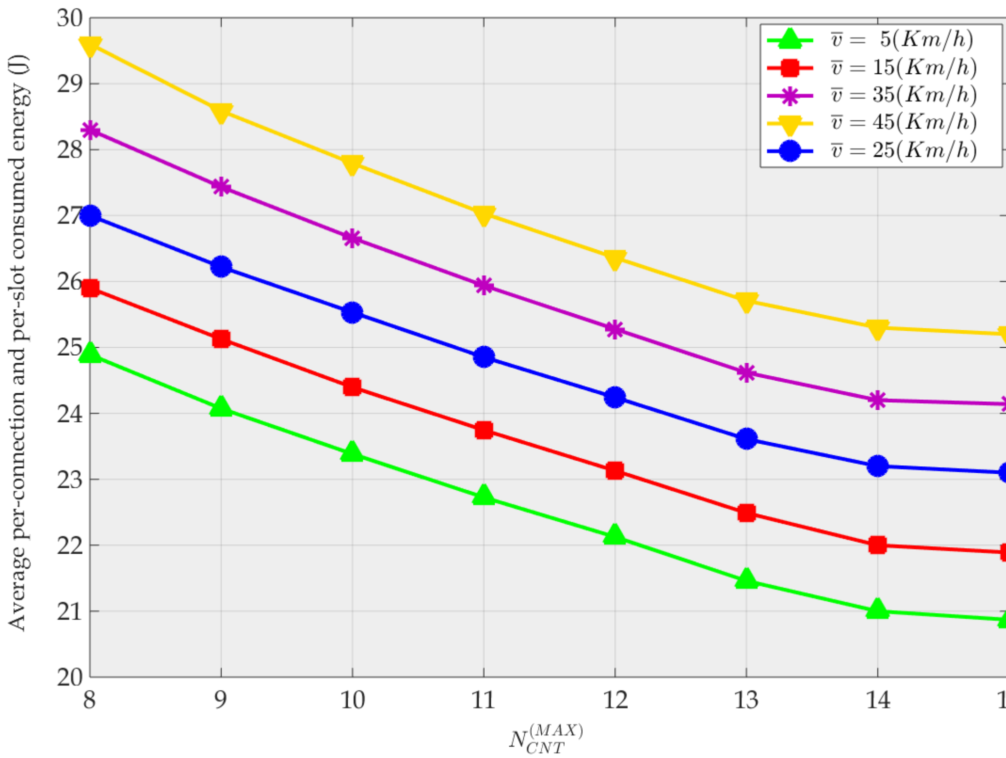


Fig. 15: Per-connection and per-slot average energies consumed by the $V\text{-}FoE$ testbed at $\overline{v} = 5,\ 15,\ 25,\ 35,\ 45\ (Km/h)$.

interoperability between applications and different sensor technologies. This is also a target of some recent (in some cases, still ongoing) research projects, namely, EBBITIS [141], IoT.est [142], NEBULA [143], BETaaS [144], iCORE [145], BUTLER [146], MobilityFirst [147], and COMPOSE [148]. They address the interoperability issues under the more traditional realm of IoT. Specifically, the common main target of EBBITIS [141], IoT.est [142], NEBULA [143] and BETaaS [144] is to develop business services and Service Oriented Architecture (SOA)-based Cloud-aided architectures, in order to semantically integrate WEB 2.0 services into IoT. Furthermore, the main goal of BUTLER [146] and MobilityFirst [147] is to design distributed networked computing architectures for the support of real-time context-aware services, while a main common goal of iCORE [145] and COMPOSE [148] is to deploy unified networked computing architectures that exploit things' virtualization, in order to encompass the technological heterogeneity of current IoT sensors.

A set of ongoing IoT-inspired projects (namely, GAMBAS [149], IoTService [150], CALIPSO [151]) focuses on the design of cross-layer networked computing architectures. Their common goal is to improve the energy efficiency of the overall resulting Cloud-aided IoT system, which is also a target of the FoE paradigm. Toward this end, GAMBAS [149] and IoT-Service [150] operate at the Middleware layer. In particular, GAMBAS focuses on an adaptive Middleware for enabling dissemination of context-aware services, while IoTService aims at deploying a Middleware platform for the self-composition of smart IoT services. Conversely, CALIPSO [151] adopts a cross-layer approach that embraces the Application, Middleware and Network layers. It aims at increasing the lifetime of the underlying IoT infrastructure by improving the cross-layer energy efficiency.

The Mobile & Cloud Computing Laboratory (Mobile & Cloud Lab) of the University of Tartu (Estonia) [152] is running another project, named "Service-Oriented Fog Computing with the Interconnected Mobile Edge of Things". It aims at addressing the integration-related challenges involved in applying the FC model at the edge network of the IoT systems, and addresses numerous challenges, such as: discoverability, limited computational power and storage of IoT devices, management,

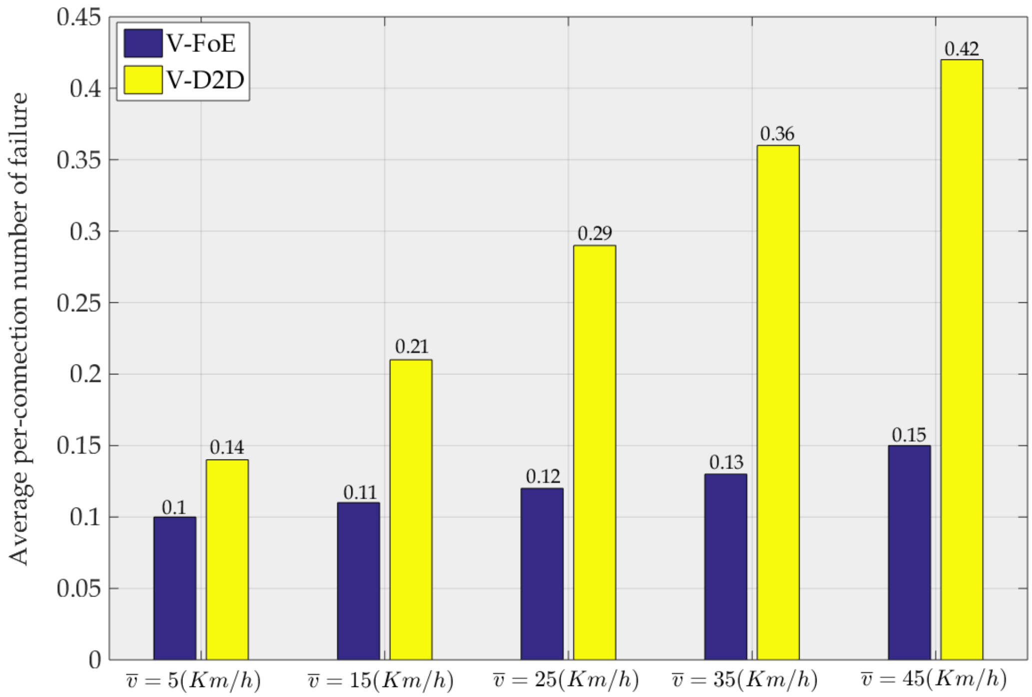


Fig. 16: Per-connection average numbers of failures: $\overline{FN}_{CON}$ of the V-FoE and V-$D2D$ testbed at $\overline{v}$ $(Km/h) = 5,\ 15,\ 25,\ 35,\ 45$, and $N_{CNT}^{(MAX)} = 13$.

and privacy/reliability.

The Eclipse ioFog [153] project focuses on the deployment of an open platform comprised by extensible frameworks, tools and runtimes. The Eclipse ioFog set of technologies is a FC layer, that can be installed on any hardware running Linux. Once installed, it provides a common runtime for microservices to be run at the edge network. In addition to this common runtime, ioFog also provides a set of useful services, including a message bus, a dynamic configuration of the micro-services and a remote debugging. An automated interconnection of ioFog instances is, then, provided by the ComSat component. All system components are available in distributed format by January, 2017. This includes ioFog, ComSat, ioAuthoring and the Fabric Controller.

Goal of the EU-funded OpenIoT project [154] is to develop a Middleware platform for gathering and pruning (e.g., filtering) messages, while guaranteeing that suitable events are generated for the interested user applications. Interestingly, as in the FoE paradigm, a major focus of OpenIoT is on the energy-efficient gathering and timely transmission of data streams generated by mobile things to proximate FNs. The ultimate goal is to design and develop a SW platform, which is capable to acquire and manage data streams generated from heterogeneous sensors, in order to provide pay-as-you-go based IoT services.

Like the FoE paradigm, a target of the IoTCloud project [155] is the integration of smart things (like tablets, smart phones and automata) with the surrounding environment, in order to properly manage sensor-generated messages and provide API to the interested user applications. It is foreseen that the resulting IoT-Cloud technological platform is equipped with IP cameras and integrated atop the FutureGrid Cloud testbed.

IoTToolkit is a USA project [156], that aims at developing SW toolkits, in order to actually enable the interaction between IoT and Fog through the integration of different protocols already available for the management of the IoT and Fog infrastructures.

ClouT [157] is a research project jointly deployed by industry, academia and city administrations from Europe and Japan. Like the FoE paradigm, ClouT aims at developing services and applications for municipalities and their users, in order to build up user-centric applications that rely on the symbiotic integration of the IoT and Cloud paradigms. Running under the um-

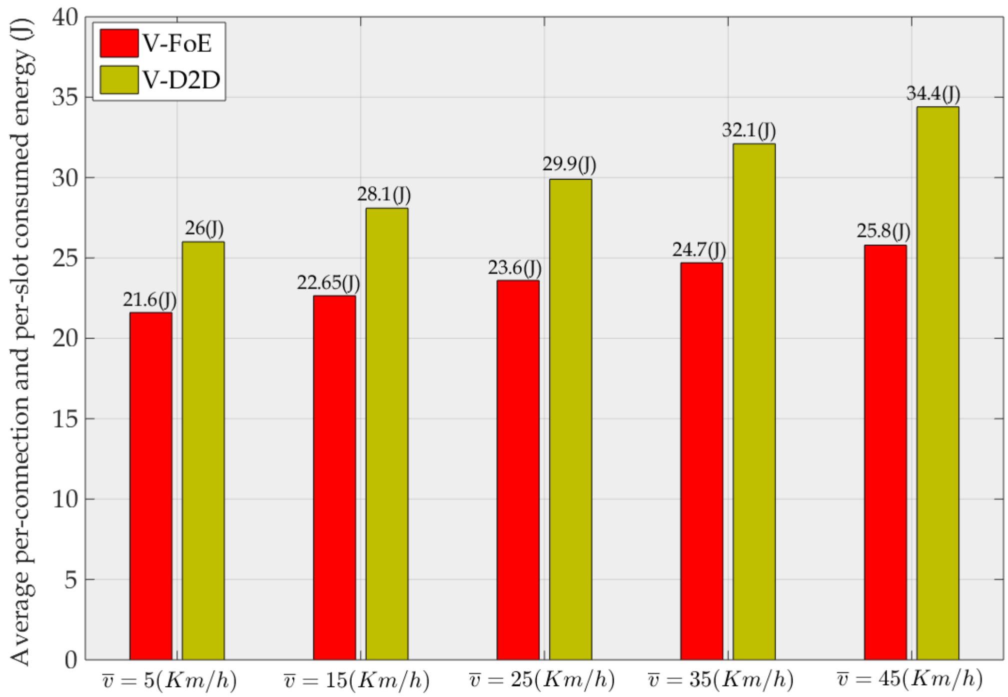


Fig. 17: Per-connection and per-slot average energies consumed by the $V\text{-}FoE$ and $V\text{-}D2D$ test-beds at $\overline{v}\ (Km/h) = 5,\ 15,\ 25,\ 35,\ 45,$ and $N_{CNT}^{(MAX)} = 13$.

brella of "Smart Cities", the applications targeted by the ClouT project cover green public transportation, safety and emergency management, as well as city event monitoring.

Finally, IoT6 [158] is an EU-funded project, that aims at exploiting the new capabilities offered by IPv6 under the umbrella of the Future Internet of Things. As the FoE paradigm, a main objective of the IoT6 project is the design and deployment of a scalable IPv6-based SOA, which is capable to dispatch the computing power offered by FNs among nearby heterogeneous smart things.

VII. Conclusion and Future Works

A main lesson stemming from the results reported in this paper is that a key challenge for coping with the unpredictable large volume of data generated by IoE-based applications is the design of a spectrum of hierarchically-organized networked computing nodes, namely, proximate Fog and remote Cloud data centers. The final goal is the adaptive energy-efficient reconfiguration and orchestration of the virtualized computing-plus-communication resources available at the computing nodes and thing devices under real-time constraints on the allowed computing-plus-communication delay and service latency.

In order to attain this goal, the performance results of the Section V suggest that three main research directions could be further pursued under the FoE realm. The first one stems from the consideration that, in the next years, IoE devices will be equipped with multiple (possibly, heterogeneous) wireless network interface cards. This opens the doors to the design of energy-efficient transport protocols, that rely on the emerging Multipath TCP paradigm [159]. The target should be the increment of the per-connection throughput, while limiting the energy overhead induced by the parallel utilization of multiple radio interfaces. A second research direction is motivated by the consideration that the native self-organizing feature of the IoE model induces hierarchical relationships among the involved things [4]. This should require the design of new Network-layer communication primitives for IoE-based ecosystems, in order to implement suitable forms of selective multicast that account for the relative roles of the involved IoE devices [4]. Finally, a third research direction relies on the consideration that the proposed FoE architecture is inherently multi-tier and distributed (see Section IV-A), and exploits the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2017.2702013, IEEE Access

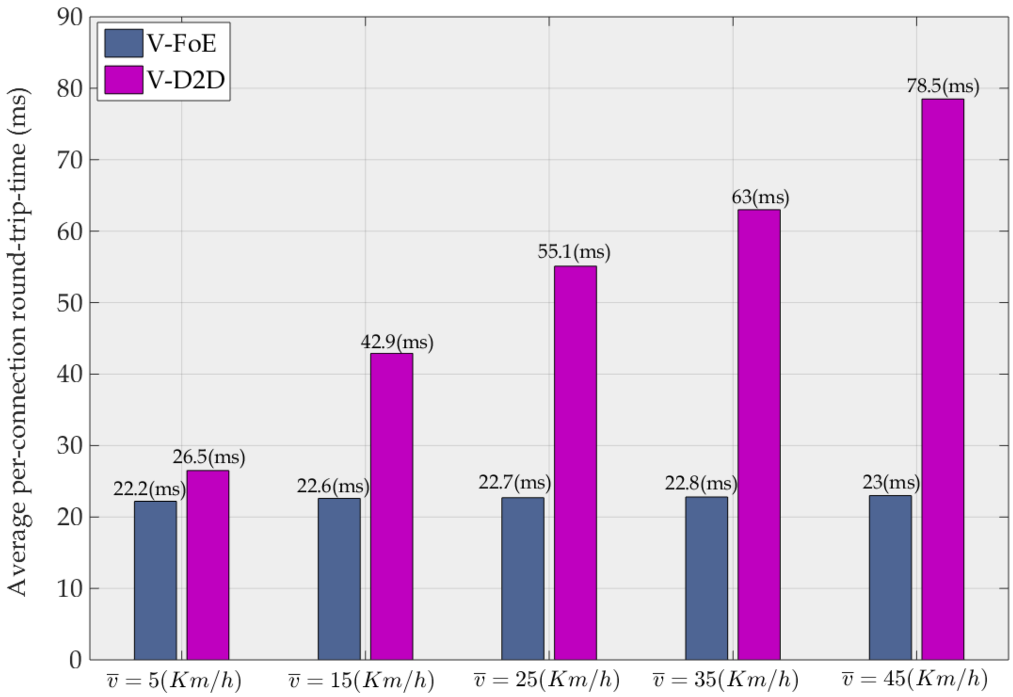


Fig. 18: Per-connection average round-trip-times of the $V\text{-}FoE$ and $V\text{-}D2D$ testbed at $\overline{v}$ (Km/h) $=$ $5,\ 15,\ 25,\ 35,\ 45$, and $N_{CNT}^{(MAX)} = 13$.

inter-networking of local (e.g., IoE devices), proximate (e.g., FNs) and remote (e.g., Cloud nodes) computing entities. On the basis of this consideration, the design of distributed and adaptive resource orchestrators that jointly perform the energy and delay-efficient allocation and scheduling of the offered workload over the full spatial spectrum of the available computing nodes is a still challenging research issue [33], [34], [35].

Finally, we point out that, since the FoE model stems from two emerging paradigms, e.g., FC and IoE, it is in the infancy and, then, is continuously evolving. In this position paper, we have provided an outlook on some main research areas and challenges that are mainly related to the required computing networked architectures and energy efficiency. However, several other related and/or tangential research fields, that have been not covered by this paper, can be identified. For example, since FoE relies on distributed networked computing architectures by design, it is expected that innovative solutions tackling distributed security, trust worthy and thing authentication will be needed, in order to allow the migration of the FoE paradigm from the theory to the practice. This opens the doors to further work.

ACKNOWLEDGEMENTS

This work has been supported by the project: “GAUChO-A Green Adaptive Fog Computing and networking Architectures” funded by the MIUR Progetti di Ricerca di Rilevante Interesse Nazionale (PRIN) Bando 2015- grant 2015YPXH4W_004, and by the project: “V-FOG: Vehicular Fog energy-efficient QoS mining and dissemination of multimedia Big Data streams”, funded by Sapienza University of Rome Bando 2016.

REFERENCES

[1] F. Bonomi, “Connected vehicles, the internet of things, and fog computing,” in *in Proceedings of the Eighth ACM International Workshop on VehiculAr Internetworking*. Las Vegas, 2011.

[2] J. Bradley, J. Loucks, J. Macaulay, and A. Noronha, “Internet of everything (IoE) value index,” *White Paper CISCO and/or its affiliates*, 2013.

[3] E. Borgia, “The Internet of Things vision: Key features, applications and open issues,” *Computer Communications*, vol. 54, pp. 1–31, 2014.

[4] F. Da Costa, *Rethinking the Internet of Things: a scalable approach to connecting everything*. Apress, 2013.

[5] S. V. Vandebroek, "Three pillars enabling the Internet of Everything: Smart everyday objects, information-centric networks, and automated real-time insights," in *2016 IEEE International Solid-State Circuits Conference (ISSCC)*. IEEE, 2016, pp. 14–20.

[6] 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*. ACM, 2012, pp. 13–16.

[7] M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies, "The case for VM-based cloudlets in mobile computing," *Pervasive Computing, IEEE*, vol. 8, no. 4, pp. 14–23, 2009.

[8] M. Satyanarayanan, Z. Chen, K. Ha, W. Hu, W. Richter, and P. Pillai, "Cloudlets: at the leading edge of mobile-cloud convergence," in *Mobile Computing, Applications and Services (MobiCASE), 2014 6th International Conference on*. IEEE, 2014, pp. 1–9.

[9] C. Wu and R. Buyya, *Cloud Data Centers and Cost Modeling: A complete guide to planning, designing and building a cloud data center*. Morgan Kaufmann, 2015.

[10] Y. Zhang and Y. Zhou, "Transparent computing: a new paradigm for pervasive computing," *Ubiquitous Intelligence and Computing*, pp. 1–11, 2006.

[11] G. Peng, "CDN: Content distribution network," *arXiv preprint cs/0411069*, 2004.

[12] D. Niyato, L. Xiao, and P. Wang, "Machine-to-machine communications for home energy management system in smart grid," *Communications Magazine, IEEE*, vol. 49, no. 4, pp. 53–59, 2011.

[13] J. Zhu, D. S. Chan, M. S. Prabhu, P. Natarajan, H. Hu, and F. Bonomi, "Improving web sites performance using edge servers in fog computing architecture," in *Service Oriented System Engineering (SOSE), 2013 IEEE 7th International Symposium on*. IEEE, 2013, pp. 320–323.

[14] S. Wang, J. Wan, D. Zhang, D. Li, and C. Zhang, "Towards smart factory for Industry 4.0: A self-organized multi-agent system with big data based feedback and coordination," *Computer Networks*, vol. 101, pp. 158–168, 2016.

[15] R. Suryawanshi and G. Mandlik, "Focusing on mobile users at the edge of internet of things using fog computing," *International Journal of Scientific Engineering and Technology Research*, vol. 4, no. 17, pp. 3225–3231, 2015.

[16] N. B. Truong, G. M. Lee, and Y. Ghamri-Doudane, "Software defined networking-based vehicular Ad hoc Network with Fog Computing," in *2015 IFIP/IEEE International Symposium on Integrated Network Management (IM)*. IEEE, 2015, pp. 1202–1207.

[17] X. Hou, Y. Li, M. Chen, D. Wu, D. Jin, and S. Chen, "Vehicular fog computing: A viewpoint of vehicles as the infrastructures," *IEEE Transactions on Vehicular Technology*, 2016.

[18] H. Shi, N. Chen, and R. Deters, "Combining Mobile and Fog Computing: Using CoAP to Link Mobile Device Clouds with Fog Computing," in *2015 IEEE International Conference on Data Science and Data Intensive Systems*. IEEE, 2015, pp. 564–571.

[19] N. K. Giang, M. Blackstock, R. Lea, and V. C. Leung, "Developing iot applications in the fog: a distributed dataflow approach," in *Internet of Things (IOT), 2015 5th International Conference on the*. IEEE, 2015, pp. 155–162.

[20] M. Aazam, M. St-Hilaire, C.-H. Lung, and I. Lambadaris, "PRE-Fog: IoT trace based probabilistic resource estimation at fog," in *2016 13th IEEE Annual Consumer Communications & Networking Conference (CCNC)*. IEEE, 2016, pp. 12–17.

[21] S. K. Datta, C. Bonnet, and J. Haerri, "Fog Computing architecture to enable consumer centric Internet of Things services," in *2015 International Symposium on Consumer Electronics (ISCE)*. IEEE, 2015, pp. 1–2.

[22] M. Aazam, M. St-Hilaire, C.-H. Lung, and I. Lambadaris, "MeFoRE: QoE based resource estimation at Fog to enhance QoS in IoT," in *Telecommunications (ICT), 2016 23rd International Conference on*. IEEE, 2016, pp. 1–5.

[23] M. Aazam and E.-N. Huh, "Fog computing micro datacenter based dynamic resource estimation and pricing model for iot," in *2015 IEEE 29th International Conference on Advanced Information Networking and Applications*. IEEE, 2015, pp. 687–694.

[24] D. Zeng, L. Gu, S. Guo, Z. Cheng, and S. Yu, "Joint Optimization of Task Scheduling and Image Placement in Fog Computing Supported Software-Defined Embedded System," *IEEE Transactions on Computers*, 2016.

[25] A. Asadi, Q. Wang, and V. Mancuso, "A survey on device-to-device communication in cellular networks," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 4, pp. 1801–1819, 2014.

[26] E. Baccarelli, M. Biagi, R. Bruno, M. Conti, and E. Gregori, *"Broadband wireless access networks: a roadmap on emerging trends and standards", in: Broadband services: Business models and technologies for community networks*. Wiley, pp.2515-240, 2005.

[27] M. Firdhous, O. Ghazali, and S. Hassan, "Fog computing: Will it be the future of cloud computing?" in *The Third International Conference on Informatics & Applications (ICIA2014)*, 2014.

[28] K. Saharan and A. Kumar, "Fog in comparison to cloud: A survey," *International Journal of Computer Applications*, vol. 122, no. 3, 2015.

[29] "Fog Computing and the Internet of Things," 2015. [Online]. Available: http://www.cisco.com/c/dam/en_us/solutions/trends/iot/docs/

[30] K. Kai, W. Cong, and L. Tao, "Fog computing for vehicular Ad-hoc networks: paradigms, scenarios, and issues," *The Journal of China Universities of Posts and Telecommunications*, vol. 23, no. 2, pp. 56–96, 2016.

[31] "IoT, from Cloud to Fog Computing," 2015. [Online]. Available: http://blogs.cisco.com/perspectives/iot-from-cloud-to-fog-computing

[32] P. G. Vinueza Naranjo, M. Shojafar, L. Vaca, C. Canali, R. Lancellotti, and E. Baccarelli, "Big Data Over SmartGrid – A Fog Computing Perspective," *24rd International Conference on Software, Telecommunications and Computer Networks-Softcom*, 2016.

[33] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "Mobile Edge Computing: Survey and Research Outlook," *arXiv preprint arXiv:1701.01090*, 2017.

[34] A. V. Dastjerdi, H. Gupta, R. N. Calheiros, S. K. Ghosh, and R. Buyya, "Fog Computing: Principals, Architectures, and Applications," *arXiv preprint arXiv:1601.02752*, 2016.

[35] S. Yan, M. Peng, and W. Wang, "User access mode selection in fog computing based radio access networks," *arXiv preprint arXiv:1602.00766*, 2016.

[36] D. Ye, M. Wu, S. Tang, and R. Yu, "Scalable Fog Computing with Service Offloading in Bus Networks," in *Cyber Security and Cloud Computing (CSCloud), 2016 IEEE 3rd International Conference on*. IEEE, 2016, pp. 247–251.

[37] J. Oueis, E. C. Strinati, and S. Barbarossa, "The Fog Balancing: Load Distribution for Small Cell Cloud Computing,"

in *2015 IEEE 81st Vehicular Technology Conference (VTC Spring)*. IEEE, 2015, pp. 1–6.

[38] R. Deng, R. Lu, C. Lai, T. H. Luan, and H. Liang, "Optimal Workload Allocation in Fog-Cloud Computing Towards Balanced Delay and Power Consumption," *IEEE Internet of Things Journal*, 2012.

[39] K. Intharawijitr, K. Iida, and H. Koga, "Analysis of fog model considering computing and communication latency in 5G cellular networks," in *2016 IEEE International Conference on Pervasive Computing and Communication Workshops (PerCom Workshops)*. IEEE, 2016, pp. 1–4.

[40] M. Peng, S. Yan, K. Zhang, and C. Wang, "Fog computing based radio access networks: Issues and challenges," *arXiv preprint arXiv:1506.04233*, 2015.

[41] M. A. Hassan, M. Xiao, Q. Wei, and S. Chen, "Help your mobile applications with fog computing," in *Sensing, Communication, and Networking-Workshops (SECON Workshops), 2015 12th Annual IEEE International Conference on*. IEEE, 2015, pp. 1–6.

[42] S. S. Iyengar and R. R. Brooks, *Distributed Sensor Networks: Sensor Networking and Applications*. CRC press, 2012.

[43] F. Bonomi, "Connected vehicles, the internet of things, and fog computing," in *The Eighth ACM International Workshop on Vehicular Inter-Networking (VANET), Las Vegas, USA*, 2011, pp. 13–15.

[44] M. Ghamkhari and H. Mohsenian-Rad, "Energy and performance management of green data centers: A profit maximization approach," *Smart Grid, IEEE Transactions on*, vol. 4, no. 2, pp. 1017–1025, 2013.

[45] W. Kempton and J. Tomić, "Vehicle-to-grid power implementation: From stabilizing the grid to supporting large-scale renewable energy," *Journal of power sources*, vol. 144, no. 1, pp. 280–294, 2005.

[46] S. Cirani, G. Ferrari, N. Iotti, and M. Picone, "The IoT hub: a fog node for seamless management of heterogeneous connected smart objects," in *Sensing, Communication, and Networking-Workshops (SECON Workshops), 2015 12th Annual IEEE International Conference on*. IEEE, 2015, pp. 1–6.

[47] V. Gazis, A. Leonardi, K. Mathioudakis, K. Sasloglou, P. Kikiras, and R. Sudhaakar, "Components of fog computing in an industrial internet of things context," in *Sensing, Communication, and Networking-Workshops (SECON Workshops), 2015 12th Annual IEEE International Conference on*. IEEE, 2015, pp. 1–6.

[48] D. Bernstein, "Containers and cloud: From lxc to docker to kubernetes," *IEEE Cloud Computing*, vol. 1, no. 3, pp. 81–84, 2014.

[49] C. Dsouza, G.-J. Ahn, and M. Taguinod, "Policy-driven security management for fog computing: Preliminary framework and a case study," in *Information Reuse and Integration (IRI), 2014 IEEE 15th International Conference on*. IEEE, 2014, pp. 16–23.

[50] M. Shojafar, N. Cordeschi, and E. Baccarelli, "Energy-efficient Adaptive Resource Management for Real-time Vehicular Cloud Services," *IEEE Transactions on Cloud Computing*, vol. PP, pp. 1–14, 2016.

[51] "Cisco solutions," (Date last accessed 2016). [Online]. Available: http://www.cisco.com/c/dam/en/us/solutions/collateral/trends/at-aglancec45734964.pdf

[52] T. Lu, M. Chen, and L. L. Andrew, "Simple and effective dynamic provisioning for power-proportional data centers," *IEEE Trans. on Parallel and Distributed Systems*, vol. 24, no. 6, pp. 1161–1171, 2013.

[53] A. Beloglazov, J. Abawajy, and R. Buyya, "Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing," *Future generation computer systems*, vol. 28, no. 5, pp. 755–768, 2012.

[54] G. Lovász, F. Niedermeier, and H. de Meer, "Performance tradeoffs of energy-aware virtual machine consolidation," *Cluster Computing*, vol. 16, no. 3, pp. 481–496, 2013.

[55] S. Azodolmolky, P. Wieder, and R. Yahyapour, "Cloud computing networking: challenges and opportunities for innovations," *IEEE Communications Magazine*, vol. 51, no. 7, pp. 54–62, 2013.

[56] V. Mathew, R. K. Sitaraman, and P. Shenoy, "Energy-aware load balancing in content delivery networks," in *Proc. of IEEE INFOCOM*, 2012, pp. 954–962.

[57] T. Guérout, T. Monteil, G. Da Costa, R. N. Calheiros, R. Buyya, and M. Alexandru, "Energy-aware simulation with DVFS," *Simulation Modelling Practice and Theory*, vol. 39, pp. 76–91, 2013.

[58] E. Baccarelli, N. Cordeschi, and V. Polli, "Optimal self-adaptive QoS resource management in interference-affected multicast wireless networks," *IEEE/ACM Transactions on Networking (TON)*, vol. 21, no. 6, pp. 1750–1759, 2013.

[59] E. Baccarelli and M. Biagi, "Error resistant space-time coding for emerging 4G-WLANs," in *Wireless Communications and Networking, 2003. WCNC 2003. 2003 IEEE*, vol. 1. IEEE, 2003, pp. 72–77.

[60] N. Cordeschi, T. Patriarca, and E. Baccarelli, "Stochastic traffic engineering for real-time applications over wireless networks," *Journal of Network and Computer Applications*, vol. 35, no. 2, pp. 681–694, 2012.

[61] R. Kaur and M. Mahajan, "Fault Tolerance in Cloud Computing," *International journal of Science Technology & Management (IJSTM)*, 2015.

[62] R. Guerraoui and M. Yabandeh, "Independent faults in the cloud," in *Proceedings of the 4th International Workshop on Large Scale Distributed Systems and Middleware*. ACM, 2010, pp. 12–17.

[63] R. Buyya, C. S. Yeo, and S. Venugopal, "Market-oriented cloud computing: Vision, hype, and reality for delivering it services as computing utilities," in *High Performance Computing and Communications, 2008. HPCC'08. 10th IEEE International Conference on*. Ieee, 2008, pp. 5–13.

[64] C.-T. Yang, W.-S. Chen, K.-L. Huang, J.-C. Liu, W.-H. Hsu, and C.-H. Hsu, "Implementation of smart power management and service system on cloud computing," in *Ubiquitous Intelligence & Computing and 9th International Conference on Autonomic & Trusted Computing (UIC/ATC), 2012 9th International Conference on*. IEEE, 2012, pp. 924–929.

[65] Z. Liu, A. Wierman, Y. Chen, B. Razon, and N. Chen, "Data center demand response: Avoiding the coincident peak via workload shifting and local generation," *Performance Evaluation*, vol. 70, no. 10, pp. 770–791, 2013.

[66] R. Jhawar, V. Piuri, and M. Santambrogio, "Fault tolerance management in cloud computing: A system-level perspective," *IEEE Systems Journal*, vol. 7, no. 2, pp. 288–297, 2013.

[67] C. T. Do, N. H. Tran, C. Pham, M. G. R. Alam, J. H. Son, and C. S. Hong, "A proximal algorithm for joint resource allocation and minimizing carbon footprint in geo-distributed fog computing," in *2015 International Conference on Information Networking (ICOIN)*. IEEE, 2015, pp. 324–329.

[68] M. Portnoy, *Virtualization essentials*. John Wiley & Sons, 2012.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2017.2702013, IEEE Access

[69] E. Baccarelli, D. Amendola, and N. Cordeschi, “Minimum-energy bandwidth management for QoS live migration of virtual machines,” *Computer Networks*, vol. 93, pp. 1–22, 2015.

[70] S. Soltesz, H. Pötzl, M. E. Fiuczynski, A. Bavier, and L. Peterson, “Container-based operating system virtualization: a scalable, high-performance alternative to hypervisors,” in *ACM SIGOPS Operating Systems Review*, vol. 41, no. 3. ACM, 2007, pp. 275–287.

[71] X. Xu, H. Yu, and X. Pei, “A novel resource scheduling approach in container based clouds,” in *Computational Science and Engineering (CSE), 2014 IEEE 17th International Conference on*. IEEE, 2014, pp. 257–264.

[72] T. Rajeev and S. Ashok, “A cloud computing approach for power management of microgrids,” in *Innovative Smart Grid Technologies-India (ISGT India), 2011 IEEE PES*. IEEE, 2011, pp. 49–52.

[73] C. Wei, Z. M. Fadlullah, N. Kato, and I. Stojmenovic, “On optimally reducing power loss in micro-grids with power storage devices,” *Selected Areas in Communications, IEEE Journal on*, vol. 32, no. 7, pp. 1361–1370, 2014.

[74] Y. Simmhan, S. Aman, A. Kumbhare, R. Liu, S. Stevens, Q. Zhou, and V. Prasanna, “Cloud-based software platform for big data analytics in smart grids,” *Computing in Science & Engineering*, vol. 15, no. 4, pp. 38–47, 2013.

[75] B. Lohrmann and O. Kao, “Processing smart meter data streams in the cloud,” in *Innovative Smart Grid Technologies (ISGT Europe), 2011 2nd IEEE PES International Conference and Exhibition on*. IEEE, 2011, pp. 1–8.

[76] S. Rusitschka, K. Eger, and C. Gerdes, “Smart grid data cloud: A model for utilizing cloud computing in the smart grid domain,” in *Smart Grid Communications (SmartGridComm), 2010 First IEEE International Conference on*. IEEE, 2010, pp. 483–488.

[77] F. Y. Okay and S. Ozdemir, “A fog computing based smart grid model,” in *Networks, Computers and Communications (ISNCC), 2016 International Symposium on*. IEEE, 2016, pp. 1–6.

[78] F. Jalali, K. Hinton, R. Ayre, T. Alpcan, and R. S. Tucker, “Fog Computing May Help to Save Energy in Cloud Computing,” *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 5, pp. 1728–1739, 2016.

[79] C. C. Byers and P. Wetterwald, “Fog Computing Distributing Data and Intelligence for Resiliency and Scale Necessary for IoT: The Internet of Things (ubiquity symposium),” *Ubiquity*, vol. 2015, no. November, p. 4, 2015.

[80] Y. Yan and W. Su, “A fog computing solution for advanced metering infrastructure,” in *Transmission and Distribution Conference and Exposition (T&D), 2016 IEEE/PES*. IEEE, 2016, pp. 1–4.

[81] M. S. H. Nazmudeen, A. T. Wan, and S. M. Buhari, “Improved throughput for Power Line Communication (PLC) for smart meters using fog computing based data aggregation approach,” in *Smart Cities Conference (ISC2), 2016 IEEE International*. IEEE, 2016, pp. 1–4.

[82] W. Lee, K. Nam, H.-G. Roh, and S.-H. Kim, “A gateway based fog computing architecture for wireless sensors and actuator networks,” in *2016 18th International Conference on Advanced Communication Technology (ICACT)*. IEEE, 2016, pp. 210–213.

[83] K. Hong, D. Lillethun, U. Ramachandran, B. Ottenwälder, and B. Koldehofe, “Mobile fog: A programming model for large-scale applications on the internet of things,” in *Proceedings of the second ACM SIGCOMM workshop on Mobile cloud computing*. ACM, 2013, pp. 15–20.

[84] M. Aazam and E.-N. Huh, “Fog computing and smart gateway based communication for cloud of things,” in *Future Internet of Things and Cloud (FiCloud), 2014 International Conference on*. IEEE, 2014, pp. 464–470.

[85] V. Cardellini, V. Grassi, F. L. Presti, and M. Nardelli, “On QoS-aware scheduling of data stream applications over fog computing infrastructures,” in *2015 IEEE Symposium on Computers and Communication (ISCC)*. IEEE, 2015, pp. 271–276.

[86] L. Gu, D. Zeng, S. Guo, A. Barnawi, and Y. Xiang, “Cost-Efficient Resource Management in Fog Computing Supported Medical CPS,” *IEEE Transactions on Emerging Topics in Computing*, 2015.

[87] J. Oueis, E. C. Strinati, S. Sardellitti, and S. Barbarossa, “Small Cell Clustering for Efficient Distributed Fog Computing: A Multi-User Case,” in *Vehicular Technology Conference (VTC Fall), 2015 IEEE 82nd*. IEEE, 2015, pp. 1–5.

[88] M. A. Al Faruque and K. Vatanparvar, “Energy Management-as-a-Service Over Fog Computing Platform,” *IEEE Internet of Things Journal*, vol. 3, no. 2, pp. 161–169, 2016.

[89] W. Zhang, B. Lin, Q. Yin, and T. Zhao, “Infrastructure deployment and optimization of fog network based on microDC and LRPON integration,” *Peer-to-Peer Networking and Applications*, pp. 1–13, 2016.

[90] M. Jaradat, M. Jarrah, A. Bousselham, Y. Jararweh, and M. Al-Ayyoub, “The internet of energy: Smart sensor networks and big data management for smart grid,” *Procedia Computer Science*, vol. 56, pp. 592–597, 2015.

[91] Y. Ma, X. Wang, X. Zhou, Z. Gao, Y. Wu, J. Yin, and X. Xu, “An overview of energy internet,” in *Control and Decision Conference (CCDC), 2016 Chinese*. IEEE, 2016, pp. 6212–6215.

[92] V. C. Güngör, D. Sahin, T. Kocak, S. Ergüt, C. Buccella, C. Cecati, and G. P. Hancke, “Smart grid technologies: communication technologies and standards,” *Industrial informatics, IEEE transactions on*, vol. 7, no. 4, pp. 529–539, 2011.

[93] S. M. Amin and B. F. Wollenberg, “Toward a smart grid: power delivery for the 21st century,” *Power and energy Magazine, IEEE*, vol. 3, no. 5, pp. 34–41, 2005.

[94] M. Hashmi, S. Hänninen, and K. Mäki, “Survey of smart grid concepts, architectures, and technological demonstrations worldwide,” in *Innovative Smart Grid Technologies (ISGT Latin America), 2011 IEEE PES Conference on*. IEEE, 2011, pp. 1–7.

[95] J. Gao, Y. Xiao, J. Liu, W. Liang, and C. P. Chen, “A survey of communication/networking in Smart Grids,” *Future Generation Computer Systems*, vol. 28, no. 2, pp. 391–404, 2012.

[96] W. Xu and W. Wang, “Power Electronic Signaling Technology – A New Class of Power Electronics Applications,” *IEEE Transactions on Smart Grid*, vol. 1, no. 3, pp. 332–339, 2010.

[97] F. Mwasilu, J. J. Justo, E.-K. Kim, T. D. Do, and J.-W. Jung, “Electric vehicles and smart grid interaction: A review on vehicle to grid and renewable energy sources integration,” *Renewable and Sustainable Energy Reviews*, vol. 34, pp. 501–516, 2014.

[98] C. Kyriazopoulou, “Smart city technologies and architectures: A literature review,” in *Smart Cities and Green ICT Systems (SMARTGREENS), 2015 International Conference on*. IEEE, 2015, pp. 1–12.

[99] S. E. Bibri and J. Krogstie, “Smart sustainable cities of

the future: An extensive interdisciplinary literature review," *Sustainable Cities and Society*, 2017.

[100] G. S. Yovanof and G. N. Hazapis, "An architectural framework and enabling wireless technologies for digital cities & intelligent urban environments," *Wireless personal communications*, vol. 49, no. 3, pp. 445–463, 2009.

[101] E. Baccarelli and R. Cusani, "Recursive Kalman-type optimal estimation and detection of hidden Markov chains," *Signal Processing*, vol. 51, no. 1, pp. 55–64, 1996.

[102] "Recommendations for implementing the strategic initiative INDUSTRIE 4.0," (Date last accessed 2016). [Online]. Available: http://www.acatech.de/fileadmin/user.

[103] S. Wang, J. Wan, D. Li, and C. Zhang, "Implementing smart factory of industrie 4.0: an outlook," *International Journal of Distributed Sensor Networks*, 2016.

[104] F. Tao, Y. Zuo, L. Da Xu, and L. Zhang, "IoT-based intelligent perception and access of manufacturing resource toward cloud manufacturing," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 2, pp. 1547–1557, 2014.

[105] R. Drath and A. Horch, "Industrie 4.0: Hit or hype?[industry forum]," *IEEE industrial electronics magazine*, vol. 8, no. 2, pp. 56–58, 2014.

[106] Q. Liu, J. Wan, and K. Zhou, "Cloud manufacturing service system for industrial-cluster-oriented application," *Internet Technology*, vol. 15, no. 3, pp. 373–380, 2014.

[107] S. Schneider, M. Hirzel, and B. Gedik, "Tutorial: stream processing optimizations," in *Proceedings of the 7th ACM international conference on Distributed event-based systems*. ACM, 2013, pp. 249–258.

[108] E. Baccarelli, N. Cordeschi, A. Mei, M. Panella, M. Shojafar, and J. Stefa, "Energy-efficient dynamic traffic offloading and reconfiguration of networked data centers for big data stream mobile computing: review, challenges, and a case study," *IEEE Network*, vol. 30, no. 2, pp. 54–61, 2016.

[109] M. Chen, S. Mao, and Y. Liu, "Big Data: A survey," *Mobile Networks and Applications*, vol. 19, no. 2, pp. 171–209, 2014.

[110] D. M. West, "Big data for education: Data mining, data analytics, and web dashboards," *Governance Studies at Brookings*, pp. 1–10, 2012.

[111] S. Li, L. Da Xu, and X. Wang, "Compressed sensing signal and data acquisition in wireless sensor networks and internet of things," *Industrial Informatics, IEEE Transactions on*, vol. 9, no. 4, pp. 2177–2186, 2013.

[112] A. J. Conejo, J. M. Morales, and L. Baringo, "Real-time demand response model," *Smart Grid, IEEE Transactions on*, vol. 1, no. 3, pp. 236–242, 2010.

[113] L. Neumeyer, B. Robbins, A. Nair, and A. Kesari, "S4: Distributed stream computing platform," in *Data Mining Workshops (ICDMW), 2010 IEEE International Conference on*. IEEE, 2010, pp. 170–177.

[114] M. Zaharia, T. Das, H. Li, S. Shenker, and I. Stoica, "Discretized Streams: An Efficient and fault-tolerant model for stream processing on large clusters." *HotCloud*, vol. 12, pp. 10–10, 2012.

[115] Z. Qian, Y. He, C. Su, Z. Wu, H. Zhu, T. Zhang, L. Zhou, Y. Yu, and Z. Zhang, "Timestream: Reliable stream computation in the cloud," in *Proc. of 8th ACM European Conf. on Computer Systems*, 2013, pp. 1–14.

[116] A. G. Kumbhare, Y. Simmhan, and V. K. Prasanna, "Plasticc: Predictive look-ahead scheduling for continuous dataflows on clouds," in *Cluster, Cloud and Grid Computing (CCGrid), 2014 14th IEEE/ACM International Symposium on*. IEEE, 2014, pp. 344–353.

[117] E. Baccarelli and M. Biagi, "Performance and optimized design of space-time codes for MIMO wireless systems with imperfect channel estimates," *IEEE Transactions on Signal Processing*, vol. 52, no. 10, pp. 2911–2923, 2004.

[118] J. Kwak, Y. Kim, J. Lee, and S. Chong, "DREAM: Dynamic resource and task allocation for energy minimization in mobile cloud systems," *IEEE Journal on Selected Areas in Communications*, vol. 33, no. 12, pp. 2510–2523, 2015.

[119] "http://www.gaucho.unifi.it."

[120] T. H. Luan, L. X. Cai, J. Chen, X. Shen, and F. Bai, "VTube: Towards the media rich city life with autonomous vehicular content distribution," in *Sensor, Mesh and Ad Hoc Communications and Networks (SECON), 2011 8th Annual IEEE Communications Society Conference on*. IEEE, 2011, pp. 359–367.

[121] D. Amendola, N. Cordeschi, and E. Baccarelli, "Bandwidth management VMs Live Migration in Wireless Fog Computing for 5G Networks," in *Cloud Networking (Cloudnet), 2016 5th IEEE International Conference on*. IEEE, 2016, pp. 21–26.

[122] T. Taleb and A. Ksentini, "An analytical model for follow me cloud," in *Global Communications Conference (GLOBECOM), 2013 IEEE*. IEEE, 2013, pp. 1291–1296.

[123] ——, "Follow me cloud: interworking federated clouds and distributed mobile networks," *IEEE Network*, vol. 27, no. 5, pp. 12–19, 2013.

[124] N. Cordeschi, D. Amendola, M. Shojafar, and E. Baccarelli, "Distributed and adaptive resource management in cloud-assisted cognitive radio vehicular networks with hard reliability guarantees," *Vehicular Communications*, vol. 2, no. 1, pp. 1–12, 2015.

[125] E. Baccarelli, P. G. V. Naranjo, M. Shojafar, and M. Scarpiniti, "Q*: Energy and delay-efficient dynamic queue management in TCP/IP virtualized data centers," *Computer Communications*, vol. 102, pp. 89–106, 2017.

[126] X. Zhu, D. S. Chan, H. Hu, M. S. Prabhu, E. Ganesan, and F. Bonomi, "Improving video performance with edge servers in the Fog computing architecture." *Intel Technology Journal*, vol. 19, no. 1, 2015.

[127] P. G. V. Naranjo, M. Shojafar, H. Mostafaei, Z. Pooranian, and E. Baccarelli, "P-SEP: a prolong stable election routing algorithm for energy-limited heterogeneous fog-supported wireless sensor networks," *The Journal of Supercomputing*, vol. 73, no. 2, pp. 733–755, 2017.

[128] M. Shojafar, N. Cordeschi, D. Amendola, and E. Baccarelli, "Energy-saving adaptive computing and traffic engineering for real-time-service data centers," in *Communication Workshop (ICCW), 2015 IEEE International Conference on*. IEEE, 2015, pp. 1800–1806.

[129] N. Cordeschi, M. Shojafar, D. Amendola, and E. Baccarelli, "Energy-efficient adaptive networked datacenters for the QoS support of real-time applications," *The Journal of Supercomputing*, vol. 71, no. 2, pp. 448–478, 2015.

[130] N. Cordeschi, M. Shojafar, and E. Baccarelli, "Energy-saving self-configuring networked data centers," *Computer Networks*, vol. 57, no. 17, pp. 3479–3491, 2013.

[131] E. Baccarelli, R. Cusani, and S. Galli, "A novel adaptive receiver with enhanced channel tracking capability for TDMA-based mobile radio communications," *IEEE Journal on Selected Areas in Communications*, vol. 16, no. 9, pp. 1630–1639, 1998.

[132] H. Gupta, A. V. Dastjerdi, S. K. Ghosh, and R. Buyya, "iFogSim: A Toolkit for Modeling and Simulation of Resource Management Techniques in Internet of Things,

Edge and Fog Computing Environments," *arXiv preprint arXiv:1606.02007*, 2016.

[133] T. H. Luan, X. Ling, and X. S. Shen, "Provisioning QoS controlled media access in vehicular to infrastructure communications," *Ad Hoc Networks*, vol. 10, no. 2, pp. 231–242, 2012.

[134] R. Basmadjian and H. de Meer, "Evaluating and modeling power consumption of multi-core processors," in *Proceedings of the 3rd International Conference on Future Energy Systems: Where Energy, Computing and Communication Meet*, 2012, p. 12.

[135] N. Cheng, N. Lu, N. Zhang, X. S. Shen, and J. W. Mark, "Vehicular WiFi offloading: Challenges and solutions," *Vehicular Communications*, vol. 1, no. 1, pp. 13–21, 2014.

[136] F. Xu, F. Liu, H. Jin, and A. V. Vasilakos, "Managing performance overhead of virtual machines in cloud computing: A survey, state of the art, and future directions," *Proceedings of the IEEE*, vol. 102, no. 1, pp. 11–31, 2014.

[137] Y. Li, L. Sun, and W. Wang, "Exploring device-to-device communication for mobile cloud computing," in *Communications (ICC), 2014 IEEE International Conference on*. IEEE, 2014, pp. 2239–2244.

[138] Z. Zhou, F. Liu, Y. Xu, R. Zou, H. Xu, J. C. Lui, and H. Jin, "Carbon-aware load balancing for geo-distributed cloud services," in *2013 IEEE 21st International Symposium on Modelling, Analysis and Simulation of Computer and Telecommunication Systems*. IEEE, 2013, pp. 232–241.

[139] "Vehicular Fog for energy-efficient QoS mining and dissemination of multimedia Big Data streams (V-Fog)," (Date last accessed 2016). [Online]. Available: http://www.uniroma1.it/ricerca/finanziamenti/bandi-di-ateneo

[140] "ICT TROPIC: Distributed computing, storage and radio resource allocation over cooperative femtocells, EU Project, STREP FP7," (Date last accessed 2015). [Online]. Available: http://www.ict-tropic.eu/

[141] "The EBBITIS Project," (Date last accessed Jan. 2017). [Online]. Available: https://datatracker.ietf.org/

[142] "The IoT.est Project: Internet of things environment for service creation and testing," (Date last accessed Jan. 2017). [Online]. Available: http://ics-iot.weebly.com/iotest.html

[143] "The Nebula Project: Future internet architecture," (Date last accessed Jan. 2017). [Online]. Available: http://nebula-fia.org

[144] "The BETaaS Consortium Project," (Date last accessed Jan. 2017). [Online]. Available: http://www.betaas.eu

[145] "The iCore Project," (Date last accessed Jan. 2017). [Online]. Available: http://www.iot-icore.eu

[146] "The BUTLER Project," (Date last accessed Jan. 2017). [Online]. Available: http://www.iot-butler.eu/

[147] "NSF Project: The MobilityFirst," (Date last accessed Jan. 2017). [Online]. Available: http://www.nets-fia.net/

[148] "The COMPOSE Project: Collaborative open market to place objects at your service," (Date last accessed Jan. 2017). [Online]. Available: http://www.compose-project.eu/

[149] "The GAMBAS Project: Generic adaptive middleware for behavior-driven autonomous services," (Date last accessed Jan. 2017). [Online]. Available: http://www.gambas-ict.eu/

[150] "The iotservice project," (Date last accessed Jan. 2017). [Online]. Available: http://cds.kaist.ac.kr/iotservice/?pageid=2

[151] "The CALIPSO Project," (Date last accessed Jan. 2017). [Online]. Available: http://www.calipso.wayforlight.eu/

[152] "University of Tartu (Estonia)," (Date last accessed 2016). [Online]. Available: http://mc.cs.ut.ee/mcsite/projects/fog-computing-in-the-interconnected-mobile-edge-of-things

[153] "Eclipse ioFog," (Date last accessed 2016). [Online]. Available: https://projects.eclipse.org/projects/iot.iofog/reviews/creation-review

[154] "The OpenIOT Project," (Date last accessed Jan. 2017). [Online]. Available: http://www.openiot.eu/

[155] "The IoT Cloud Project," (Date last accessed Jan. 2017). [Online]. Available: https://sites.google.com/site/opensourceiotcloud/

[156] "The IOTToolkit: Reference implementation of the smart object API," (Date last accessed Jan. 2017). [Online]. Available: http://iot-toolkit.com/

[157] "The ClouT Project: a joint European-Japanese project," (Date last accessed Jan. 2017). [Online]. Available: http://clout-project.eu/home2/

[158] "The IoT6 Project: IPv6 for IoT," (Date last accessed Jan. 2017). [Online]. Available: http://iot6.eu

[159] Y.-C. Chen, Y.-s. Lim, R. J. Gibbens, E. M. Nahum, R. Khalili, and D. Towsley, "A measurement-based study of multipath tcp performance over wireless networks," in *Proceedings of the 2013 conference on Internet measurement conference*. ACM, 2013, pp. 455–468.

Enzo Baccarelli is a full professor in Information and Communication Engineering at the DIET Dept. of Sapienza University of Rome. He received the Laurea degree in Electronic Engineering and the Ph.D. degree in Communication Theory and Systems, both from Sapienza University of Rome. In 1995, he received the Post-Doctorate degree in Information Theory and Applications from the INFOCOM Dept. of Sapienza University of Rome. His current research focuses on data networks, distributed computing and adaptive optimization.

Paola G. Vinueza Naranjo is currently a third year PhD student in Information Communication and Telecommunication (ICT) at the Sapienza University of Rome. She received Bsc. and Msc. degrees Engineering in Computer Science at Universidad Nacional de Chimborazo, Riobamba, Ecuador (2005–2010), respectively. Her research interests include cloud computing, energy-saving, and Fog-based wireless systems. She is a member of IEEE Computer Society.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2017.2702013, IEEE Access

Michele Scarpiniti was born in Leonberg, Germany, in 1978. He received the "Laurea" degree in Electrical Engineering with honors from the Sapienza University of Rome, Italy, in 2005 and the Ph.D. in Information and Communication Engineering in 2009. Since March 2008 he is an Assistant Professor of Circuit Theory and Multimedia Signal Processing with the Department of Information Engineering, Electronics and Telecommunications, at Sapienza University of Rome. His present research interests include nonlinear adaptive filters, audio processing and neural networks for signal processing, ICA and blind signal processing.

Mohammad Shojafar is currently is a CNIT senior researcher at the University of Rome Tor Vergata. Recently, he completed an Italian project named "SAMMClouds", supported by the University of Modena and Reggio Emilia, Modena, Italy. He completed his Ph.D. in ICT at the Sapienza University of Rome in May 2016. He received his Msc in software engineering in Qazvin Islamic Azad University, Qazvin, Iran in 2010. Also, he received his Bsc. in computer engineering-software major by Iran University of science and technology, Tehran, Iran in 2006. His current research focuses on distributed computing, wireless communications, and AI optimization.

Jemal H. Abawajy is a full professor at Faculty of Science, Engineering and Built Environment, Deakin University, Australia. He is a Senior Member of IEEE Society. Professor Abawajy is currently the Director of the Distributing System Security (DSS). Professor Abawajy has served on the editorial-board of numerous international journals and currently serving as associate editor of the IEEE Transaction on Cloud Computing, International Journal of Big Data Intelligence and International Journal of Parallel, Emergent and Distributed Systems. He has also guest edited many special issue journals.