Edge Computing for the Internet of Things: A Case Study

Gopika Premsankar, Mario Di Francesco, and Tarik Taleb

***Abstract*—The amount of data generated by sensors, actuators, and other devices in the Internet of Things (IoT) has substantially increased in the last few years. IoT data are currently processed in the cloud, mostly through computing resources located in distant data centers. As a consequence, network bandwidth and communication latency become serious bottlenecks. This paper advocates edge computing for emerging IoT applications that leverage sensor streams to augment interactive applications. First, we classify and survey current edge computing architectures and platforms, then describe key IoT application scenarios that benefit from edge computing. Second, we carry out an experimental evaluation of edge computing and its enabling technologies in a selected use case represented by mobile gaming. To this end, we consider a resource-intensive 3-D application as a paradigmatic example and evaluate the response delay in different deployment scenarios. Our experimental results show that edge computing is necessary to meet the latency requirements of applications involving virtual and augmented reality. We conclude by discussing what can be achieved with current edge computing platforms and how emerging technologies will impact on the deployment of future IoT applications.**

***Index Terms*—Edge computing, fog computing, Internet of Things (IoT), mobile gaming.**

I. Introduction

THERE has been a substantial growth in the data generated by mobile and Internet of Things (IoT) devices. These devices (including sensors, smartphones, and wearables) are characterized by limited computational and energy resources. Such limitations are currently addressed by offloading processing and storage from resource-constrained devices to the cloud [1]. Indeed, the cloud is an ideal solution for computation offloading due to its on-demand and scalable nature. However, cloud computing resources are hosted in large data centers built in locations far away from most end-users. This results in a high communication latency between end-users and the cloud. Moreover, the increasing amount of data exchanged adds substantial stress on the network links to the cloud.

Manuscript received December 2, 2017; accepted January 25, 2018. Date of publication February 12, 2018; date of current version April 10, 2018. This work was supported by the Academy of Finland under Grant 299222 and Grant 305507. *(Corresponding author: Tarik Taleb.)*

G. Premsankar and M. Di Francesco are with the Department of Computer Science, School of Science, Aalto University, 00076 Espoo, Finland (e-mail: gopika.premsankar@aalto.fi; mario.di.francesco@aalto.fi).

T. Taleb is with the Department of Computer and Information Security, Sejong University, Seoul, South Korea, and also with the School of Electrical Engineering, Aalto University, 00076 Espoo, Finland (e-mail: tarik.taleb@aalto.fi).

Digital Object Identifier 10.1109/JIOT.2018.2805263

To help address these issues, the concept of *edge* or *fog* computing has been proposed [2], [3]. According to this paradigm, computing resources are made available at the edge of the network, close to (or even co-located with) end-devices. Placing computing resources in close proximity to the devices generating the data reduces communication latency. Furthermore, network-intensive data can be processed and analyzed just one hop away from end-devices, thereby reducing the bandwidth demands on network links to distant data centers. The ease of processing and storing data close to the devices generating them will enable new services [4]–[7]. Finally, edge computing platforms support mobility of devices and geographically distributed applications [3]. Mobility and geographical distribution are indeed the key characteristics of IoT deployments that can particularly benefit from edge computing. A few representative applications include content delivery to vehicles, real-time analytics of data collected by mobile devices and environmental monitoring through geographically distributed wireless sensor networks.

The concept of bringing content closer to end-users is not new. Content delivery or distribution networks (CDNs) [8] deploy resources that replicate content from a source location onto servers close to the end-users. Information-centric networking (ICN) [9] is a similar approach for enhancing the Internet infrastructure to explicitly support content-based routing and forwarding. However, the CDN and ICN paradigms are limited to noninteractive content [10]; for instance, IoT data can be cached at the edge of the network [11], [12]. On the other hand, edge computing servers also provide computational capabilities and can host interactive applications that support user mobility. Furthermore, an edge computing platform can relieve privacy concerns as the data generated from IoT devices are stored and processed within nodes in the edge network. This means that data can be preprocessed to remove private information before being sent to the cloud [13]. Besides, offloading computation to resources closer to the users (and data centers that are not far away) can help reduce the energy consumption at the end-devices [14].

The contributions of this paper are twofold. First, we classify and survey current edge computing architectures and platforms, then describe key IoT application scenarios that benefit from edge computing. Second, we carry out an experimental evaluation of edge computing and its enabling technologies in a selected use case represented by mobile gaming. Specifically, we demonstrate that edge computing is necessary to achieve

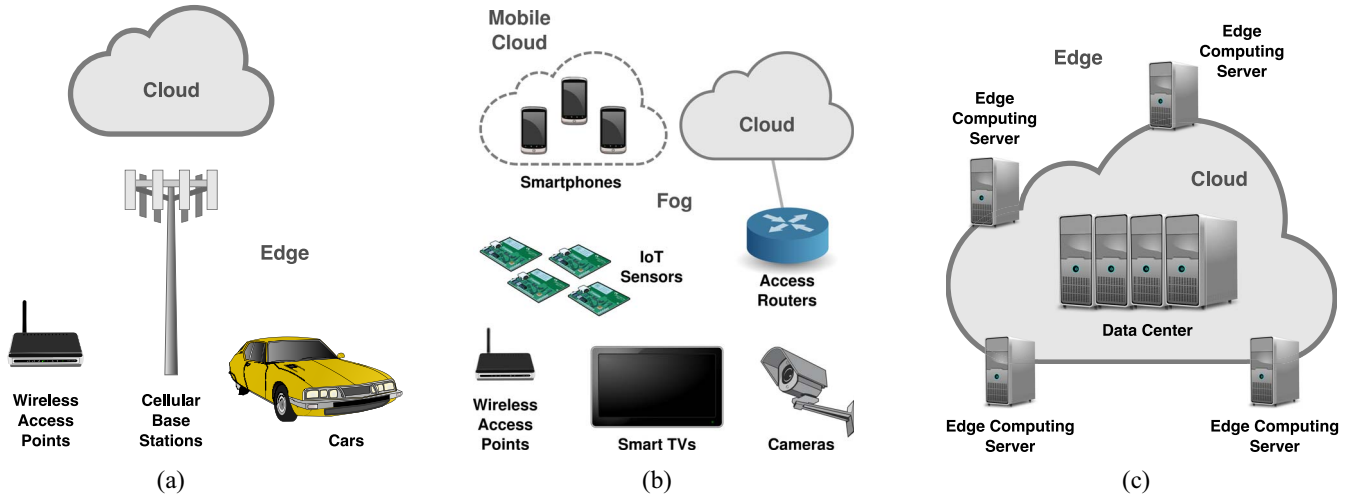

Fig. 1. Classification of edge computing platforms. (a) Resource-rich servers deployed at the edge. (b) Heterogeneous edge nodes. (c) Edge-cloud federation. All devices not depicted within a cloud with a solid outline have edge computing capabilities.

satisfactory quality of experience in the considered use case. Indeed, mobile gaming relies on low latency and reliable communications as well as sensor data from mobile devices to create an immersive end-user experience [15], [16]. Pokémon Go[1] and Ingress[2] are examples of successful games that combine augmented reality and sensor information such as user location [17], [18]. Accordingly, we evaluate the benefits of edge computing for mobile gaming scenarios in this paper. In particular, we consider gaming as a paradigmatic example of a larger class of applications that rely on rendering complex 3-D environments, including virtual and augmented reality.

The rest of this paper is organized as follows. Section II reviews and classifies existing edge computing platforms. Section III introduces the key enabling technologies behind edge computing. Section IV discusses the requirements of IoT applications and the benefits of edge computing for such applications. Section V presents the results of the performance evaluation of edge computing in mobile gaming. Section VI discusses our findings in relation to the currently available technologies. Finally, Section VII provides some concluding remarks and outlines possible directions for future research.

II. Edge Computing: Classes and Architectures

Different architectures have been proposed to realize edge computing platforms. A review of these architectures reveals that the edge of the network is not clearly defined and the nodes expected to participate at the edge can vary. Besides, the terminology used to describe the edge differs greatly, with the same term being used to define different architectures and functionality [22]–[24]. Thus, we begin by classifying the proposed architectures into three categories (Fig. 1). The categorization is based on common features of deployments. However, in practice, features from one category can be used in combination with others. One category relies on *resource-rich servers deployed close to the end-devices*. Another group leverages resources from *heterogeneous nodes at the edge*, including the end-devices themselves. A third category is based on the *federation of resources at the edge and centralized data centers*. We detail the features of each category next. Table I summarizes our classification.

A. Resource-Rich Servers Deployed at the Edge

One option to realize an edge computing platform is to deploy resource-rich servers in the network to which end-users connect. Satyanarayanan *et al.* [2] presented virtual machine (VM)-based *cloudlets* deployed on WiFi access points, one hop away from end-devices. A cloudlet is described as a "data center in a box" that offers a cluster of multicore computing capacity, storage and wireless LAN connectivity toward the edge. Ha *et al.* [14] proposed a multitiered system using cloudlets to provide cognitive assistance for users. Video and sensor data collected from users through Google Glass are processed on the cloudlet to provide real-time assistance. Simoens *et al.* [25] presented a scalable three-tier system using cloudlets for analytics and automated tagging of crowd-sourced video from user devices. Since the introduction of cloudlets, further research has proposed integrating cloudlets with femtocells, LTE base stations or even cars [13]. Greenberg *et al.* [26] described the design of micro data centers consisting of thousands of servers and capable of hosting interactive applications for end-users. While these data centers have been used for deploying CDNs and email applications [26], they can be repurposed to host cloudlets [25]. Similar to cloudlets, Wang *et al.* [27] proposed deploying a small set of servers on WiFi access points or a base station in the radio access network. The authors refer to this deployment as a micro cloud.

In the telecommunications ecosystem, multiaccess edge computing (MEC) follows a similar approach of deploying resource-rich resources at the edge. In this paradigm,

[1][Online]. Available: http://www.pokemongo.com/
[2][Online]. Available: http://www.ingress.com/

TABLE I
SUMMARY OF EDGE COMPUTING PLATFORMS AND THEIR FEATURES

Approach	Edge nodes	Edge network
Cloudlets [2]	Compact-size data centers deployed on WiFi access points, femtocells or LTE base stations	WiFi, 3G or LTE
Mobile cloudlets [13]	Compact-size data centers on cars	3G or LTE
Multi-access edge computing [19]	Servers deployed in the radio access network	3G, LTE, WiFi or other access technologies
Fog computing [3]	Heterogeneous nodes including high-end servers, routers, access points and set-top boxes	Multiple wireless access technologies including WiFi, 3G and LTE
Mobile cloud [20]	Neighboring mobile nodes form a cloud with one device chosen as resource coordinator	Local networking through WiFi or Bluetooth; Internet connectivity with WiFi, 3G and LTE
Edge cloud [21]	Compute or storage nodes deployed in the edge network and federated to cloud data centers	Home/enterprise networks and WiFi hotspots
FUSION [10]	Service nodes deployed on access points, local data centers and centralized data centers	Not defined

cloud computing resources, storage, and IT services are deployed in the radio access component of mobile networks. Such a platform consists of MEC servers integrated onto base stations or radio network controllers, while applications run on these servers through VMs. One of the key benefits of this architecture is the possibility to expose real-time radio link information to applications deployed at the edge. Since the initial white paper when MEC was known as mobile-edge computing [19], ETSI has expanded the scope of MEC to include access technologies other than mobile. An MEC platform can also be deployed as a gateway in indoor environments and provide services including augmented reality, building management and social network applications [24].

B. Heterogeneous Edge Nodes

In contrast to the solutions described above, edge computing platforms can leverage a diverse set of computing resources. Bonomi *et al.* [3] proposed a *fog platform* characterized by a highly virtualized system of heterogeneous nodes, ranging from resource-rich servers to more constrained edge routers, access points, set-top boxes, and even end-devices (including smartphones and connected vehicles). The authors also recognize the heterogeneity of wireless connectivity as a key aspect of end-devices. Thus, different wireless access technologies are supported by their solution. A similar concept is presented in [28], with edge-devices (including smartphones and connected vehicles), routers and on-demand dedicated compute instances employed for processing data in the fog platform. Chiang and Zhang [29] described a system which leverages computing resources on end-devices (including smartphones, Google Glass, and home storage devices) and the cloud to carry out real-time data stream mining [30]. Nishio *et al.* [20] defined fog computing as a cooperation-based *mobile cloud*, wherein heterogeneous mobile devices opportunistically share their resources to deliver services and applications. The proposed architecture consists of a local cloud formed by mobile devices in a neighboring area. These nodes can then share resources with other nodes in the same local cloud. One of the nodes is elected as a local resource coordinator and manages the allocation of tasks for services. These tasks can run on either devices in the local cloud or on the back-end cloud. Elkhatib *et al.* [31] proposed the use of small and low-power computers such as Raspberry Pis to host fog services. The Raspberry Pis can be clustered together as independent and portable mini-clouds, which can be deployed in indoor or outdoor environments.

C. Edge-Cloud Federation

Another option to realize edge platforms is based on the federation of computing resources at the edge and centralized data centers. Chang *et al.* [21] described this concept as an *edge cloud*. In their system, *edge apps* are used to deliver services at the edge as well as in distant cloud centers. The authors describe the use of edge apps to deploy indoor 3-D localization and video monitoring applications. Similarly, Farris *et al.* [32] proposed the federation of private and public clouds to enable integrated IoT applications. In this architecture, the edge node dynamically orchestrates the federation to maximize the number of executed tasks. Federation of clouds is also a key aspect of the FUSION architecture proposed by Griffin *et al.* [10]. In this paradigm, services are deployed on a cloud infrastructure distributed throughout the Internet. Application developers can deploy services in geographically distributed *execution zones* that can be located on IP routers, access points, base stations in the radio access network, and so on. Elias *et al.* [33] leveraged an edge cloud (that mirrors public cloud services) along with a federated cloud to perform image classification with very low time and bandwidth requirements. The federated architecture and mirroring of the public cloud enables the use of existing open source repositories for machine learning and image classification at the edge.

III. Enabling Technologies

The edge computing platforms described earlier are made possible by a few key enabling technologies, which are also crucial in the evolution of current mobile networks to their fifth generation (5G). In particular, 5G encompasses many new technologies addressing low latency and reliable communications, radio spectrum scarcity, energy-efficient operations, and an increasing amount of data from heterogeneous devices. Besides, 5G networks are expected to support programmable and flexible deployments of services and core network functions through network function virtualization (NFV) and software-defined networking (SDN) [34]. These technologies are expected to play a key role in the development of edge computing platforms as well. A detailed survey of the enabling technologies is provided in [24]; we summarize the most important ones next and describe how they can be used at the edge.

A. Virtualization

Virtualization enables cloud computing providers to run multiple independent software instances on a single physical server. These instances can access the underlying physical resources while being isolated from each other. This isolation enables instances to run without interfering with (or even being aware of) other instances running on the same server. Currently, VMs are the dominant means of deploying virtualized instances in cloud computing environments [35]. A software abstraction layer (hypervisor), lying between the VMs and the physical hardware, allows VMs to use the underlying CPU, storage and networking resources. Each VM runs its own guest operating system (OS) on top of the host server OS. Although hypervisor-based virtualization allows for excellent isolation of workloads and multitenancy, the hypervisor layer incurs in non-negligible overhead. Container-based virtualization has been proposed as a light-weight alternative to hypervisor-based virtualization. In this case, virtualized instances do not need to run a separate OS and can share the resources of the underlying host OS. Modifications are made to the OS to ensure isolation between containers. This form of virtualization allows reducing instance start times and generally results in better performance [34].

Apart from sharing physical resources, virtualization also allows the migration of VMs or containers. More specifically, migration consists in moving computing resources from one physical server to another. This is very useful for several scenarios, including the consolidation of virtualized instances to reduce data center energy consumption or to adapt to user mobility. *Live migration* is a technique that reduces the time during which a virtual instance is not accessible as it is being moved from one server to another. While live migration of VMs has existed for a long time and is being used extensively by cloud providers, migration of containers is still relatively new. Virtualization and live migration techniques are particularly important for edge computing platforms. These virtualization technologies are also the basis of NFV and SDN as described next.

B. Network Function Virtualization and Software-Defined Networking

NFV [36] involves the implementation of network functions as software modules that can run on general-purpose hardware. It decouples the software from the underlying hardware by leveraging the previously described virtualization technologies. With this approach, different network functions and services no longer need to run on dedicated hardware. Instead, they can be executed on general-purpose nodes. NFV also offers improved flexibility, as virtualized network functions can be deployed in the location most suited for efficient delivery of mobile applications and services.

SDN [37] complements NFV by decoupling the management or control plane from the data plane over which data packets are forwarded. SDN enables easier and more flexible management of networks through abstractions and a logically centralized controller that handles policy and forwarding decisions. Moreover, the softwarization of the controller allows for faster deployment of new services. SDN, together with NFV, enables flexible and programmable deployment of software-based modules, thereby simplifying network configuration and management. Besides, these technologies are extremely important for network operators to quickly deploy new software functions with a limited cost. For instance, NFV enables automated deployment of virtual resources to meet a sudden increase in the traffic generated by an IoT application at a certain location. Jointly with the availability of edge computing platforms, NFV can bring the needed virtual resources close to end-users, for instance, on equipment within their premises. On top of this, SDN enables automated orchestration of virtualized instances as well as flexible policy control and routing of the increased traffic to the newly deployed resources.

C. Computation Offloading

Typically, computation and storage are offloaded from resource-constrained mobile devices to the cloud [1], i.e., processing-heavy tasks are sent for execution in the cloud which, in turn, sends the results back to the devices [38]. Clearly, offloading can involve edge computing platforms instead of (or in addition to) the centralized cloud. In either case, end-devices access cloud resources as a thin client or through a Web browser [39]. Offloading computation from devices has several benefits. For instance, the battery life of resource constrained end-devices can be extended by avoiding complex local processing. Offloading to the edge instead of the cloud results in even lower energy consumption at the end-device [24], [25]. Moreover, offloading computation enables several types of applications to run on resource-constrained devices, including mobile gaming, mobile learning, natural language processing and mobile healthcare.

IV. Edge Computing for IoT Applications

The IoT is characterized by resource-constrained devices such as sensors, smartphones, wearable devices and machines connected to the Internet. The IoT lays out a foundation for

the digitalization of the physical world that can be described in terms of machine-friendly data. Once sampled or generated, these data can be automatically processed and interpreted to provide innovative services in diverse areas ranging from mobile healthcare, to smart power generation and intelligent transportation systems. Several distinctive features of the IoT make it well suited for deployments based on edge computing platforms [3]. The characteristics and some of the representative use cases enabled by edge computing are as follows.

1) *Low-latency communication* is critical for several IoT applications, including connected vehicles, mobile gaming, remote health monitoring, warehouse logistics, and industrial control systems. These scenarios are characterized by real-time actions or responses based on processing data generated by end-devices.
2) An increasing amount of data generated by IoT deployments today are *bandwidth-intensive*, including video from surveillance cameras, police patrol cars and user devices. Placing computational resources one-hop away from high-bandwidth data sources implies that less data need to be sent to the distant cloud data centers [13]. For instance, videos and sensor data from hazardous locations can be processed locally to provide real-time information to responders in public safety applications [40].
3) *Geographical distribution* is a key characteristic of IoT applications based on sensor networks. Indeed, the related use cases highly benefit from processing data locally through edge computing platforms. One example is given by collision avoidance systems deployed at the edge of vehicular networks, e.g., at the roadside units deployed for communication purposes. These systems depend on sensor data (such as location, velocity, acceleration, and so on) generated by both vehicles and pedestrians. Processing the data locally replaces sending data to distant cloud data centers, thus achieving low-latency communications.
4) *Device mobility* places additional demands on low-latency processing of device data. Indeed, edge computing platforms support migration of virtualized resources based on the mobility of end-devices, thus allowing the data generated by these devices to be processed locally and with a satisfactory quality of experience.

The current state of the art in the IoT largely involves device-driven (i.e., machine-type) communications, and does not require explicit human intervention. The next step forward is the tight integration of digital data and physical environments through real-time wireless communications. An emerging class of applications involve the integration of sensor- or user-generated inputs and artificially created 3-D scenarios. For instance, samples generated by sensor nodes may be used to replicate the movements and gestures of a person in a specific physical setting onto a virtual environment.

The real-time nature of interactive applications is deeply connected with perception and actuation. The response time of human beings for different types of stimuli varies depending on the specific circumstances. However, it can be as low as a few milliseconds for situations where rapid actions are taken or expected. The response time of an artificial system significantly affects the user experience of human-computer interactions too. For instance, there are very stringent latency and reliability constraints for immersive applications, such as virtual and augmented reality [41], [42]. Indeed, virtual reality builds an immersive environment that adapts to the movements and the interactions of users. Augmented reality extracts information about the physical world and overlays it on the field of view of a person, i.e., on a head-mounted display. The recent success of Pokémon Go highlights the popularity of games that incorporate the physical location of the user and augmented reality into mobile games. Future games are expected to combine additional wearable sensor data to enable context-aware and more immersive gaming experiences. The response times of such games significantly affect the user experience [43], [44]. The response time becomes especially important in multiuser games whose gameplay can be significantly affected by delays in processing user- or sensor-generated input.

In the cases outlined above, the application needs to create a virtual environment and adapt it according to the user inputs. Such an environment needs to be realistic, thereby involving complex 3-D modeling and rendering operations that are resource-intensive. The cloud computing paradigm offers abundant computation and storage capabilities that can be leveraged to realize close to photorealistic rendering in a very short time. However, cloud resources can be located far away from the user, thus introducing significant delays. While they would be acceptable for noninteractive applications, these delays cannot be tolerated in the considered use cases, as they would result in a poor user experience.

The edge computing paradigm offers the same key features of the cloud in a location close to the user, thus resulting in a much shorter latency. An important question is: *what can edge computing achieve with the current state of the art*? To address this question, we focus on the specific use case of mobile gaming next.

V. Use Case: Mobile Gaming

We carry out an experimental evaluation of mobile gaming using a prototype edge computing platform. This use case is particularly meaningful given the currently available technologies. For our evaluation, we select Neverball,[3] an open source 3-D arcade game wherein the player controls a ball by tilting the floor so as to collect coins and reach a designated exit point through several levels. Neverball is representative of a larger class of applications that rely on rendering complex 3-D environments, including virtual and augmented reality. Moreover, due its peculiar gameplay, it can be used as a fitness game, for instance, by employing balance boards (including Nintendo's Wii Fit) as input devices [45]. In such scenarios, the end-to-end latency is extremely important as the user needs very quick

[3][Online]. Available: http://neverball.org

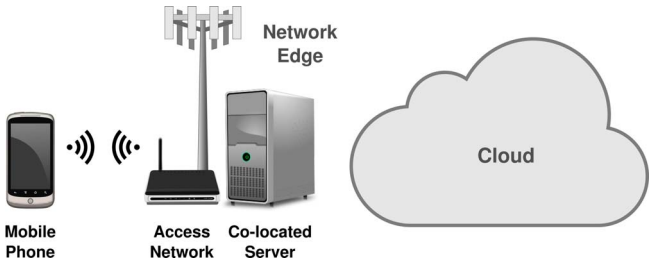

Fig. 2. Testbed setup used for the network edge scenario.

responses from the server to have a good quality of experience. In our scenario, we use a mobile phone (end-device) that sends the relevant game input to the gaming server, which in turn renders the content and streams the video back to the end-device.

A. Testbed Setup

In our evaluation, we carry out experiments by using the open-source GamingAnywhere cloud gaming platform [46]. We focus on the *response delay* and use the same methodology described by Huang *et al.* [46]. Specifically, we define the response delay as the time elapsed between an action performed by the user and the occurrence of the corresponding outcome at the client device. The response delay includes three main components.

1) *Processing Delay (PD):* The time taken by the server to process the user input and render the corresponding frame.
2) *Playout Delay (OD):* The time taken by the client to decode and display the frame on its own screen.
3) *Network Delay (ND):* The round trip time (RTT) between the client and the server, which accounts for the time taken to send data from the client to the server and back.

In our experiments, we consider two access technologies, i.e., WiFi and LTE. In both cases, we use our university network, in particular, the NetLeap 4G network provided by Nokia Solutions and Networks for LTE. The client device is a Google Nexus 5 mobile phone running Android 5.1.1. The gaming server is deployed on a workstation with a 4-core Intel Xeon E3-1230 CPU, 16 GB of RAM, and two NVIDIA Quadro 2000 GPUs. We consider three different server deployment scenarios: 1) a local deployment at the network edge; 2) a special-purpose cloud computing infrastructure; and 3) a commercial public cloud provider.

Specifically, the network edge scenario is represented by a server co-located with an LTE base station and a workstation deployed in the same wireless network as the client device for WiFi, as illustrated in Fig. 2. This setup corresponds to the architecture presented in Fig. 1(a), wherein computing resources are deployed in the edge network. The second considered scenario, instead, consists of a special-purpose cloud infrastructure, namely, the cPouta service offered by CSC (the Finnish IT Center for Science), running OpenStack and located in Kajaani, Finland. Finally, the last scenario uses a commercial public cloud provider, i.e., Amazon; in this case, we consider the two geographically closest EC2 data center locations, i.e., Frankfurt and Ireland.

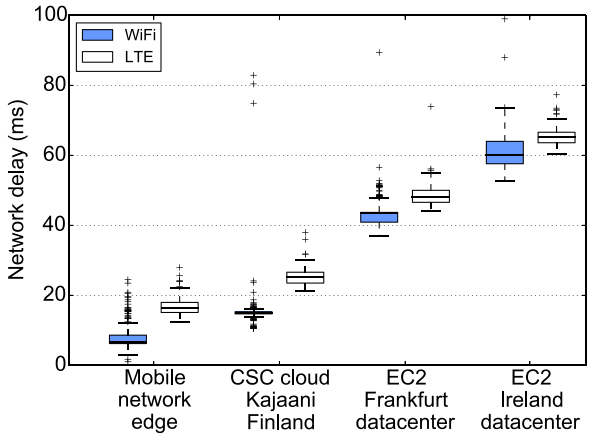

Fig. 3. Impact of network access technologies and server deployment on the ND.

The Neverball game runs on the Ubuntu 14.04.4 Linux OS under three different configurations: 1) bare metal, i.e., directly on the host OS without any virtualization technology; 2) within a Linux container; and 3) within a virtualized instance. We employ Docker (version 1.10.3) to run a container and QEMU (version 2.5.0) to run a VM on the host. In each case, we assign one of the GPUs of the host to the container or VM. For the server deployment on Amazon EC2, we use two types of GPU instances, both featuring Intel Xeon E5-2670 processors and NVIDIA GRID K520 GPUs: g2.2xlarge (with one GPU and eight vCPUs) and g2.8xlarge (with four GPUs and 32 vCPUs). We use dedicated instances, running on reserved physical servers and isolated from others, to reduce the impact of the data-center load on the performance.

We ran four iterations of each experiment to characterize the statistical significance of the obtained results. We replayed gaming sessions that were previously recorded to achieve consistent outcomes that are not considerably affected by the timing of the input events. Individual sessions lasted from 1 to 3 min. We configured GamingAnywhere to stream videos at 30 frames/s with a bitrate of 4.5 Mb/s for all experiments.

B. Experimental Results

We first study the impact of server deployment on the ND. Then, we examine the overhead of different virtualization technologies and how the screen resolution affects the response delay. Finally, we quantify the impact of additional computational resources offered by the cloud on the PD.

Fig. 3 shows the ND for the considered access technologies (i.e., WiFi and LTE) as a function of the location of the server deployment as a box plot. To obtain these results, we measured the ND in terms of the RTT between the client and the server obtained through ICMP ping messages. The ping measurements were carried out at regular intervals over a duration of at least 6 h.

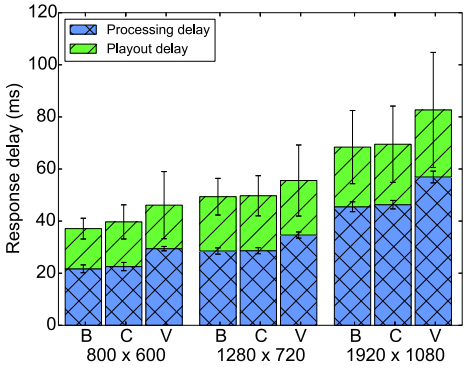

Fig. 4. Impact of the resolution on the response delay for different configurations: bare metal (B), containers (C), and virtual machines (V).

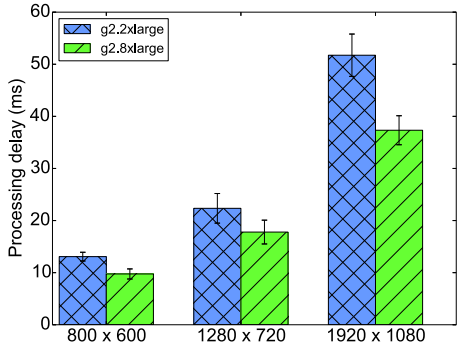

Fig. 5. Average PD as a function of the resolution for different Amazon GPU instances.

The figure clearly shows the impact of the data center on the ND: the two scenarios with servers located in Finland obtain significantly lower values (i.e., consistently below 25 ms) than those of the data centers located in other countries, with a geographical distance[4] of about 1500–2000 km. The edge network scenario allows an ND of less than 20 ms over LTE, which can be considered the state of the art of currently available wireless communication technologies. A public cloud incurs in a delay that is at least twice as much, i.e., 50 ms in the best case. To a certain extent, the ND of the CSC cloud is quite low, but this is also related by the high-speed network connecting our university with the data-center in Kajaani—similar results would be difficult to achieve with standard cloud providers. Besides, WiFi obtains shorter delays than LTE on the average, even though the difference becomes small when the data center is far. Moreover, the variance of the results obtained with WiFi is significantly higher than that with LTE. This implies that the jitter is lower over LTE, making it actually a better choice for our use case, as it involves video streaming.

Fig. 4 shows the components of the response delay that are not affected by the network (i.e., the processing and the OD) for different virtualization technologies as a function of the screen resolution. Specifically, the bare metal, container, and VM configurations are indicated as B, C, and V (respectively) in the figure. The figure demonstrates that the performance of containers is almost the same as the bare metal configuration, irrespective of the video resolution. Hypervisor-based virtualization, instead, incurs in about 30% more PD. This additional delay is significant for the considered scenario: for instance, a resolution of 1280×720 pixels can be actually streamed at the target value of 30 frames/s with containers, but not with VMs. With the highest resolution of 1920×1080 pixels, none of the considered configurations allows to match the desired frame rate.

The figure also shows that the OD at the client is significant: it is actually comparable to the PD when the resolution is not full HD. The OD increases with the screen resolution too, even though it never exceeds 25 ms on the average. The variance of the OD is much higher than that of the PD, increases with the resolution and is higher when VMs are used. This can be related to the higher variance in the content of the source video content before compression, and on the additional jitter affecting communications when using hypervisor-based virtualization.

Next, we evaluate whether the use of more powerful computing resources in the cloud (thereby reducing the PD) can compensate for the ND in accessing these resources. Fig. 5 shows the average PD on two different types of EC2 instances as a function of the screen resolution (we do not present the OD as it is similar to the results obtained earlier). The g2.2xlarge instance achieves shorter PDs than those for the bare metal configuration for the resolutions of 800×600 pixels and 1280×720 pixels. However, at the highest resolution of 1920×1080 pixels, the g2.2xlarge instance takes on average 5 ms longer than the bare metal workstation. With the more powerful g2.8xlarge instance, the PD is shorter than the bare metal configuration for all resolutions. These results indicate that the additional computational resources offered by the cloud are not effective for the full HD resolution. In fact, most of the PD is due to the encoding of the source video content, as opposed to rendering. At 800×600 pixels, the g2.2xlarge is on average 12 ms faster than the bare metal workstation and at 1920×1080 pixels, 8 ms faster. Even though the reduction in the PD is significant as a percentage (i.e., both GPU instances are twice as fast as the bare metal configuration), the actual gain is rather limited in terms of the sheer numbers.

VI. Discussion

Although we have focused on the use case of gaming, the results of our experimental evaluation offer insightful considerations on using edge computing for next-generation IoT applications. It is clear that hosting computing resources very close to the end-users, possibly at the access network edge, is the only viable option to achieve a satisfactory quality of experience. While a response delay below 150 ms is generally considered acceptable for interactive applications, fast-paced interactions cannot tolerate delays beyond 70 ms [43]. By combining the components of the response delay shown in Figs. 3 and 4, it is apparent how fast-paced interactions cannot be satisfactory when using any of the data centers in the considered public cloud. Indeed, the mobile network edge configuration allows to play the game at an

[4]In contrast, Kajaani is less than 500 km away from Helsinki.

HD resolution with processing times below 70 ms; not even using the CSC cloud allows this result. As demonstrated by Fig. 5, the availability of much higher computational resources in the cloud (rather than at the edge) does not help, as the gain in PD is not enough to overcome the additional ND.

Our experiments involved only one end-user. However, the key observations from our experimental evaluation remain the same even for scenarios involving multiple users. In fact, the considered use cases relies on one VM (or container) per user, as it usually happens in mobile (cloud) gaming [46]. In the case of online multiplayer gaming, a single virtualized instance (container or VM) is deployed for rendering per user at the edge [47]. Thus, the fundamental observations from our experiments still hold. Moreover, traffic flow control within the data center networks (at the edge) can be used to reduce the service latency between different components for a large number of users [48]. Indeed, issues related to network access would become more critical in a scenario involving a large number of users. In this case, edge computing platforms can leverage virtualization, NFV and SDN to scale out resources when the number of end-users increases. Specifically, NFV can be employed to deploy virtualized gaming modules at the edge and exploit real-time information on the access network to appropriately tune application parameters (for instance, video encoding parameters in the cloud gaming use case). Edge orchestrators can also launch VMs (or containers) with higher processing capabilities based on the requirements of the specific games. Finally, SDN can be employed to manage networking at the edge and flexibly control network flows. Furthermore, mobility of users can be handled through live migration of edge computing resources (VMs or containers).

Our evaluation has considered rather limited options concerning the availability and location of devices with suitable computing capabilities. However, the number of these devices is rapidly increasing in the infrastructure of mobile network operators; a similar trend is expected for content and service providers as well. As a result, selection and management of edge nodes will become a crucial aspect for successful deployment of next-generation IoT applications. That is exactly where NFV and SDN come into play [49]. Complex real-time resource allocation and optimization problems will emerge, beyond the current state of the art [50].

The delays incurred by current technologies are the main limiting factor of the performance that can be achieved today. Indeed, it is impossible to lower the response time to 10 ms or even less without fundamental advances in both wireless communication and computing technologies. While one of the objectives in the 5G initiative is to reduce latency to 1 ms, it is still unclear whether a significant reduction of the factors affecting delay beyond the access network can be achieved without a fundamental breakthrough. What we can certainly confirm is that edge computing is definitely needed to move forward in achieving such ambitious goals.

VII. Conclusion

This paper has investigated the suitability of edge computing for emerging IoT applications. Specifically, we evaluated the performance of edge computing for mobile gaming as a representative scenario of new applications incorporating physical sensory inputs in addition to those explicitly generated by the user. The obtained results have shown that edge computing is necessary to enable fast-paced interactive games. Although regional data centers allow to significantly reduce network latency, only hosting resources at the edge enables a satisfactory quality of experience for gaming. Furthermore, increasing the computational capabilities of the servers in the cloud does not compensate for the increase in network latency. Therefore, deploying even limited computing resources at the edge helps improve the quality of experience in the considered use case. An interesting future research direction is represented by a large-scale evaluation of mobile gaming based on the edge computing paradigm. Such a study could consider, for instance, online multiplayer games. It would also be interesting to compare the performance of different edge computing architectures for specific application scenarios. We hope that this paper will encourage further research in this direction.

Acknowledgment

The authors would like to thank T. Käämäräinen, J. Kotimäki, J. Savikko, and L. Tirkkonen for their help in the setup of the experiments.

References

[1] H. T. Dinh, C. Lee, D. Niyato, and P. Wang, "A survey of mobile cloud computing: Architecture, applications, and approaches," *Wireless Commun. Mobile Comput.*, vol. 13, no. 18, pp. 1587–1611, Dec. 2013.

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

[3] F. Bonomi, R. Milito, P. Natarajan, and J. Zhu, "Fog computing: A platform for Internet of Things and analytics," in *Big Data and Internet of Things: A Roadmap for Smart Environments*. Cham, Switzerland: Springer, Mar 2014, pp. 169–186.

[4] R. Atat *et al.*, "Enabling cyber-physical communication in 5G cellular networks: Challenges, spatial spectrum sensing, and cyber-security," *IET Cyber Phys. Syst. Theory Appl.*, vol. 2, no. 1, pp. 49–54, Apr. 2017.

[5] K. Wang, Y. Wang, Y. Sun, S. Guo, and J. Wu, "Green industrial Internet of Things architecture: An energy-efficient perspective," *IEEE Commun. Mag.*, vol. 54, no. 12, pp. 48–54, Dec. 2016.

[6] J. Wu, S. Guo, J. Li, and D. Zeng, "Big data meet green challenges: Big data toward green applications," *IEEE Syst. J.*, vol. 10, no. 3, pp. 888–900, Sep. 2016.

[7] J. Wu, S. Guo, J. Li, and D. Zeng, "Big data meet green challenges: Greening big data," *IEEE Syst. J.*, vol. 10, no. 3, pp. 873–887, Sep. 2016.

[8] P. A. Frangoudis, L. Yala, A. Ksentini, and T. Taleb, "An architecture for on-demand service deployment over a telco CDN," in *Proc. IEEE ICC*, Kuala Lumpur, Malaysia, May 2016, pp. 1–6.

[9] B. Ahlgren, C. Dannewitz, C. Imbrenda, D. Kutscher, and B. Ohlman, "A survey of information-centric networking," *IEEE Commun. Mag.*, vol. 50, no. 7, pp. 26–36, Jul. 2012.

[10] D. Griffin *et al.*, "Service oriented networking," in *Proc. EuCNC*, Bologna, Italy, Jun. 2014, pp. 1–5.

[11] R. Li, H. Harai, and H. Asaeda, "An aggregatable name-based routing for energy-efficient data sharing in big data era," *IEEE Access*, vol. 3, pp. 955–966, 2015.

[12] R. Li, H. Asaeda, and J. Li, "A distributed publisher-driven secure data sharing scheme for information-centric IoT," *IEEE Internet Things J.*, vol. 4, no. 3, pp. 791–803, Jun. 2017.
[13] M. Satyanarayanan *et al.*, "Edge analytics in the Internet of Things," *IEEE Pervasive Comput.*, vol. 14, no. 2, pp. 24–31, Apr./Jun. 2015.
[14] K. Ha *et al.*, "Towards wearable cognitive assistance," in *Proc. 12th Int. Conf. Mobile Syst. Appl. Services*, Bretton Woods, NH, USA, Jun. 2014, pp. 68–81.
[15] S. Liszio and M. Masuch, "Designing shared virtual reality gaming experiences in local multi-platform games," in *Proc. Int. Conf. Entertainment Comput.*, Vienna, Austria, 2016, pp. 235–240.
[16] J. Farman, "Locative life: Geocaching, mobile gaming, and embodiment," in *Proc. Digit. Arts Culture*, 2009. [online]. Available: https://escholarship.org/uc/item/507938rr
[17] S. Chess, "Augmented regionalism: Ingress as geomediated gaming narrative," *Inf. Commun. Soc.*, vol. 17, no. 9, pp. 1105–1117, 2014.
[18] E. Graells-Garrido, L. Ferres, and L. Bravo, "The effect of Pokémon go on the pulse of the city: A natural experiment," *EPJ Data Sci.*, vol. 6, no. 1, Sep. 2017, doi: 10.1140/epjds/s13688-017-0119-3.
[19] *Mobile-Edge Computing Introductory Technical White Paper*, ETSI, Sophia Antipolis, France, Sep. 2014.
[20] T. Nishio, R. Shinkuma, T. Takahashi, and N. B. Mandayam, "Service-oriented heterogeneous resource sharing for optimizing service latency in mobile cloud," in *Proc. ACM 1st Int. Workshop Mobile Cloud Comput. Netw. (MobileCloud)*, Bengaluru, India, Jul. 2013, pp. 19–26.
[21] H. Chang, A. Hari, S. Mukherjee, and T. V. Lakshman, "Bringing the cloud to the edge," in *Proc. IEEE Conf. Comput. Commun. Workshops (INFOCOM WKSHPS)*, Toronto, ON, Canada, Apr. 2014, pp. 346–351.
[22] P. Mach and Z. Becvar, "Mobile edge computing: A survey on architecture and computation offloading," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 3, pp. 1628–1656, 3rd Quart., 2017.
[23] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A survey on mobile edge computing: The communication perspective," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 4, pp. 2322–2358, 4th Quart., 2017.
[24] T. Taleb *et al.*, "On multi-access edge computing: A survey of the emerging 5G network edge cloud architecture and orchestration," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 3, pp. 1657–1681, 3rd Quart., 2017.
[25] P. Simoens *et al.*, "Scalable crowd-sourcing of video from mobile devices," in *Proc. 11th Annu. Int. Conf. Mobile Syst. Appl. Services*, Taipei, Taiwan, 2013, pp. 139–152.
[26] A. Greenberg, J. Hamilton, D. A. Maltz, and P. Patel, "The cost of a cloud: Research problems in data center networks," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 39, no. 1, pp. 68–73, 2008.
[27] S. Wang *et al.*, "Mobility-induced service migration in mobile micro-clouds," in *Proc. IEEE Mil. Commun. Conf. (MILCOM)*, Baltimore, MD, USA, 2014, pp. 835–840.
[28] 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 *Proc. 2nd ACM SIGCOMM Workshop Mobile Cloud Comput.*, Hong Kong, 2013, pp. 15–20.
[29] M. Chiang and T. Zhang, "Fog and IoT: An overview of research opportunities," *IEEE Internet Things J.*, vol. 3, no. 6, pp. 854–864, Dec. 2016.
[30] L. Canzian and M. Van Der Schaar, "Real-time stream mining: Online knowledge extraction using classifier networks," *IEEE Netw.*, vol. 29, no. 5, pp. 10–16, Sep./Oct. 2015.
[31] Y. Elkhatib *et al.*, "On using micro-clouds to deliver the fog," *IEEE Internet Comput.*, vol. 21, no. 2, pp. 8–15, Mar./Apr. 2017.
[32] I. Farris, L. Militano, M. Nitti, L. Atzori, and A. Iera, "Federated edge-assisted mobile clouds for service provisioning in heterogeneous IoT environments," in *Proc. IEEE 2nd World Forum Internet Things (WF-IoT)*, Milan, Italy, 2015, pp. 591–596.
[33] A. R. Elias, N. Golubovic, C. Krintz, and R. Wolski, "Where's the bear?—Automating wildlife image processing using IoT and edge cloud systems," in *Proc. IEEE/ACM 2nd Int. Conf. Internet Things Design Implement. (IoTDI)*, Pittsburgh, PA, USA, 2017, pp. 247–258.
[34] T. Taleb, A. Ksentini, and R. Jantti, "'Anything as a service' for 5G mobile systems," *IEEE Netw.*, vol. 30, no. 6, pp. 84–91, Nov./Dec. 2016.
[35] W. Felter, A. Ferreira, R. Rajamony, and J. Rubio, "An updated performance comparison of virtual machines and Linux containers," IBM Res., Austin, TX, USA, Rep. RC25482 (AUS1407-001), Jul. 2014.
[36] "Network functions virtualisation? Introductory white paper," Darmstadt, Germany, SDN OpenFlow World Congr., White Paper, Oct. 2012. [Online]. Available: http://portal.etsi.org/NFV/NFV_White_Paper.pdf
[37] "Software-defined networking: The new norm for networks," Palo Alto, CA, USA, ONF, White Paper, 2012.
[38] K. Kumar, J. Liu, Y.-H. Lu, and B. Bhargava, "A survey of computation offloading for mobile systems," *Mobile Netw. Appl.*, vol. 18, no. 1, pp. 129–140, 2013.
[39] Y. Lin, T. Kämäräinen, M. Di Francesco, and A. Ylä-Jääski, "Performance evaluation of remote display access for mobile cloud computing," *Comput. Commun.*, vol. 72, pp. 17–25, Dec. 2015.
[40] B. Kantarci and H. T. Mouftah, "Trustworthy sensing for public safety in cloud-centric Internet of Things," *IEEE Internet Things J.*, vol. 1, no. 4, pp. 360–368, Aug. 2014.
[41] S. R. Ellis, K. Mania, B. D. Adelstein, and M. I. Hill, "Generalizeability of latency detection in a variety of virtual environments," in *Proc. Human Factors Ergonomics Soc. Annu. Meeting*, vol. 48, 2004, pp. 2632–2636.
[42] R. T. Azuma, "A survey of augmented reality," *Presence Teleoperators Virtual Environ.*, vol. 6, no. 4, pp. 355–385, 1997.
[43] M. Jarschel, D. Schlosser, S. Scheuring, and T. Hoßfeld, "An evaluation of QoE in cloud gaming based on subjective tests," in *Proc. IEEE 5th Int. Conf. Innov. Mobile Internet Services Ubiquitous Comput. (IMIS)*, Seoul, South Korea, 2011, pp. 330–335.
[44] T. Kämäräinen, M. Siekkinen, Y. Xiao, and A. Ylä-Jääski, "Towards pervasive and mobile gaming with distributed cloud infrastructure," in *Proc. IEEE 13th Annu. Workshop Netw. Syst. Support Games*, Nagoya, Japan, 2014, pp. 1–6.
[45] D. Fitzgerald, N. Trakarnratanakul, L. Dunne, B. Smyth, and B. Caulfield, "Development and user evaluation of a virtual rehabilitation system for wobble board balance training," in *Proc. 30th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. (EMBS)*, Vancouver, BC, Canada, 2008, pp. 4194–4198.
[46] C.-Y. Huang, K.-T. Chen, D.-Y. Chen, H.-J. Hsu, and C.-H. Hsu, "GamingAnywhere: The first open source cloud gaming system," *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 10, no. 1s, 2014, Art. no. 10.
[47] Y. Deng, Y. Li, R. Seet, X. Tang, and W. Cai, "The server allocation problem for session-based multiplayer cloud gaming," *IEEE Trans. Multimedia*, to be published, doi: 10.1109/TMM.2017.2760621.
[48] W. Xia, P. Zhao, Y. Wen, and H. Xie, "A survey on data center networking (DCN): Infrastructure and operations," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 1, pp. 640–656, 1st Quart., 2017.
[49] S. Dutta, T. Taleb, and A. Ksentini, "QoE-aware elasticity support in cloud-native 5G systems," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Kuala Lumpur, Malaysia, 2016, pp. 1–6.
[50] H.-J. Hong, D.-Y. Chen, C.-Y. Huang, K.-T. Chen, and C.-H. Hsu, "Placing virtual machines to optimize cloud gaming experience," *IEEE Trans. Cloud Comput.*, vol. 3, no. 1, pp. 42–53, Jan./Mar. 2015.

Gopika Premsankar received the M.Sc. (Tech.) degree from Aalto University, Espoo, Finland, in 2015, where she is currently pursuing the Ph.D. degree at the Department of Computer Science.

Her current research interests include IoT, edge computing, and the application of operations research to these areas.

Mario Di Francesco received the Ph.D. degree in information engineering from the University of Pisa, Pisa, Italy, in 2009.

He is an Associate Professor with the Department of Computer Science, Aalto University, Espoo, Finland. His current research interests include IoT, pervasive computing, mobile networking, and performance evaluation.

Dr. Di Francesco was a recipient of the Best Paper Award at the UbiComp Conference in 2014, as well as at the International Conference on the IoT in 2015. He is an Area Editor of the *Pervasive and Mobile Computing Journal*. He was the Co-Chair of the Third IEEE Workshop on the IoT: Smart Objects and Services in 2014.

Tarik Taleb received the B.E. degree (with distinction) in information engineering and M.Sc. and Ph.D. degrees in information sciences from Tohoku University, Sendai, Japan, in 2001, 2003, and 2005, respectively.

He is currently a Professor with the School of Electrical Engineering, Aalto University, Espoo, Finland, where he is the Founder and the Director of the MOSA!C Laboratory. Prior to his current academic position, he was a Senior Researcher and a 3GPP Standards Expert with NEC Europe Ltd., Heidelberg, Germany. He was then leading the NEC Europe Laboratories Team researching on research and development projects on carrier cloud platforms, an important vision of 5G systems. Before joining NEC and till 2009, he was an Assistant Professor with the Graduate School of Information Sciences, Tohoku University, in a laboratory fully funded by KDDI. From 2005 to 2006, he was a Research Fellow with the Intelligent Cosmos Research Institute, Sendai. His current research interests include architectural enhancements to mobile core networks (particularly 3GPP's), network softwarization and slicing, mobile cloud networking, network function virtualization, software defined networking, mobile multimedia streaming, intervehicular communications, and social media networking.

Prof. Taleb was a recipient of the 2017 IEEE ComSoc Communications Software Technical Achievement Award for his outstanding contributions to network softwarization, the Best Paper Award at IEEE-flagged conferences for his research work, and the Best Workshop Award by the IEEE Communication Society for the IEEE Workshop on Telecommunications Standards: From Research to Standards, and a co-recipient of the 2017 IEEE Communications Society Fred W. Ellersick Prize, the 2009 IEEE ComSoc Asia–Pacific Best Young Researcher Award, the 2008 TELECOM System Technology Award from the Telecommunications Advancement Foundation, the 2007 Funai Foundation Science Promotion Award, the 2006 IEEE Computer Society Japan Chapter Young Author Award, the Niwa Yasujirou Memorial Award, and the Young Researcher's Encouragement Award from the Japan Chapter of the 2003 IEEE Vehicular Technology Society. He has also founded and has been the Steering Committee Chair of the IEEE Conference on Standards for Communications and Networking. He is the General Chair of the 2019 edition of the IEEE Wireless Communications and Networking Conference to be held in Marrakech, Morocco. He is the Guest Editor-in-Chief of the IEEE JSAC Series on Network Softwarization and Enablers. He is/was on the Editorial Board of the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, *IEEE Wireless Communications Magazine*, the IEEE JOURNAL ON INTERNET OF THINGS, the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, IEEE COMMUNICATIONS SURVEYS AND TUTORIALS, and a number of Wiley journals. Upon until 2016, he served as the Chair of the Wireless Communications Technical Committee, the largest of the IEEE ComSoC. He also served as the Vice Chair of the Satellite and Space Communications Technical Committee of the IEEE ComSoc from 2006 to 2010. He has been on the Technical Program Committee of different IEEE conferences, including Globecom, ICC, and WCNC, and chaired some of their symposia. He has also been directly engaged in the development and standardization of the evolved packet system as a member of 3GPP's System Architecture Working Group. He is a member of the IEEE Communications Society Standardization Program Development Board.