EdgeIoT: Mobile Edge Computing for the Internet of Things

Citation:
Xiang Sun and Nirwan Ansari, "EdgeIoT: Mobile Edge Computing for the Internet of Things," *IEEE Communications Magazine, Special Issue on Internet of Things,* DOI: 10.1109/MCOM.2016.1600492CM, vol.54, no.12, pp. 22-29, December 2016.

URL:
http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=7786106

EdgeIoT: Mobile Edge Computing for Internet of Things

Xiang Sun, *Student Member, IEEE,* and Nirwan Ansari, *Fellow, IEEE*

***Abstract*—In order to overcome the scalability problem of the traditional Internet of Things (IoT) architecture (i.e., data streams generated from the distributed IoT devices are transmitted to the remote cloud via the Internet for further analysis), this article proposes a novel mobile edge computing for the IoT architecture, i.e., edgeIoT, to handle the data streams at the mobile edge. Specifically, each BS is connected to a fog node, which provides computing resources locally. On the top of the fog nodes, the Software Defined Networking (SDN) based cellular core is designed to facilitate the packet forwarding among fog nodes. Meanwhile, we propose a hierarchical fog computing architecture in each fog node to provide flexible IoT services while maintaining user privacy, i.e., each user's IoT devices are associated with a proxy Virtual Machine (VM) (located in a fog node), which collects, classifies, and analyzes the devices' raw data streams, converts them into metadata, and transmits the metadata to the corresponding application VMs (which are owned by IoT service providers). Each application VM receives the corresponding metadata from different proxy VMs and provides its service to users. In addition, a novel proxy VMs migration scheme is proposed to minimize the traffic in the SDN based core.**

***Index Terms*—Internet of things, edge computing, fog computing, virtual machine, VM migration**

I. Introduction

Today, a tremendous number of smart devices and objects are embedded with sensors, enabling them to sense real-time information from the environment. This phenomenon has culminated to the intriguing concept of the Internet of Things (IoT) in which all the smart things, such as smart cars, wearable devices, laptops, sensors, and industrial and utility components, are connected via a network of networks and empowered with data analytics that are forever changing the way we work, live and play. In the past few years, many startups are embracing and actualizing the concept of IoT in areas such as smart homes/buildings, smart cities, intelligent healthy cares, smart traffic, smart environments, etc. Although IoT can potentially benefit the whole society, many technical issues remain to be addressed.

First, the data streams generated by the IoT devices are in high volume and fast velocity (European Commission predicted that there will be 50 to 100 billion smart devices connected to the Internet by 2020 [1]. Also, Cisco predicted that the devices that are connected to the Internet will generate 507.5 ZB per year by 2019 [2]). Meanwhile, owing to the flexible and efficient resource provisioning in the cloud [3], the big IoT data generated from the distributed IoT devices are transmitted to the remote cloud, a smart "brain" for processing the big data, via the Internet in the traditional IoT architecture [4], [5], as shown in Fig. 1. However, the Internet is not scalable and efficient enough to handle the big IoT data. Meanwhile, transferring the big data is expensive, consuming a huge amount of bandwidth, energy and time. Second, since the big IoT data streams are transmitted to the cloud in high volume and fast velocity, it is necessary to design an efficient data processing architecture to explore the valuable information in real-time. Third, user privacy remains a challenging unsolved issue, i.e., in order to obtain services and benefits, the users should share their sensed data with IoT service providers and these sensed data may contain the user personal information. Thus, it is critical to design a data sharing framework so that users can acquire IoT services while their privacy is guaranteed. In this article, we propose an efficient and flexible IoT architecture, i.e., edgeIoT, by leveraging fog computing and Software Defined Networking (SDN) to collect, classify, and analyze the IoT data streams at the mobile edge. The article makes the following contributions:

- We propose edgeIoT by bringing the computing resources close to IoT devices so that the traffic in the core network can be alleviated and the end-to-end (E2E) delay between computing resources and IoT devices is minimized;
- We design a hierarchical fog computing architecture to provide flexible and scalable computing resource provisioning for each user as well as each IoT service provider;
- We propose and evaluate a novel proxy VM migration scheme to minimize the traffic in the core network.

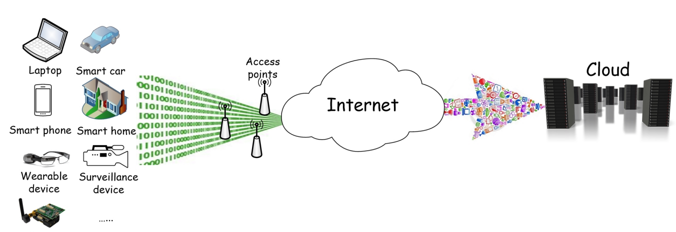


Fig. 1: The traditional IoT architecture.

The rest of the article is structured as follows: in Section II, we introduce a new mobile edge computing for IoT architecture, i.e., edgeIoT, and explain its efficiency and flexibility; in Section III, we unveil the challenges in designing the edgeIoT architecture and propose some possible solutions; we conclude the article in Section IV.

X. Sun and N. Ansari are with Advanced Networking Lab., Helen and John C. Hartmann Department of Electrical & Computer Engineering, New Jersey Institute of Technology, Newark, NJ 07102, USA. E-mail:{xs47, nirwan.ansari}@njit.edu.

II. Mobile edge computing for IoTs

Fog computing [6] (which is defined as a distributed computing infrastructure containing a bunch of high performance Physical Machines (PMs) that are well connected with each other) is an emerging computing paradigm by bringing the computing capabilities close to the distributed IoT devices. Thus, deploying a number of fog nodes in the network can locally collect, classify, and analyze the raw IoT data streams, rather than transmitting them to the cloud; this can significantly alleviate the traffic in the core network and potentially speedup the big IoT data process. Yet, where to deploy the fog nodes to facilitate the communications between the IoT devices and the fog nodes is still an open issue. The optimal fog computing deployment ensures each IoT device with access to computing capabilities everywhere with low E2E delay and without significantly increasing the traffic of the core network. It is difficult to optimize the deployment of fog nodes owing to the mobility and heterogeneity features of the IoT devices, e.g., wearable devices and mobile phones is moving over time, and different IoT devices have different data transmission requirements, i.e., some energy nonsensitive devices (such as mobile phones and surveillance devices) need high-speed data rate and some energy sensitive devices (such as sensor nodes) require low-speed and low-energy data transmission. The heterogeneous data transmission requirements among the IoT devices result in different devices adopting different wireless access technologies.

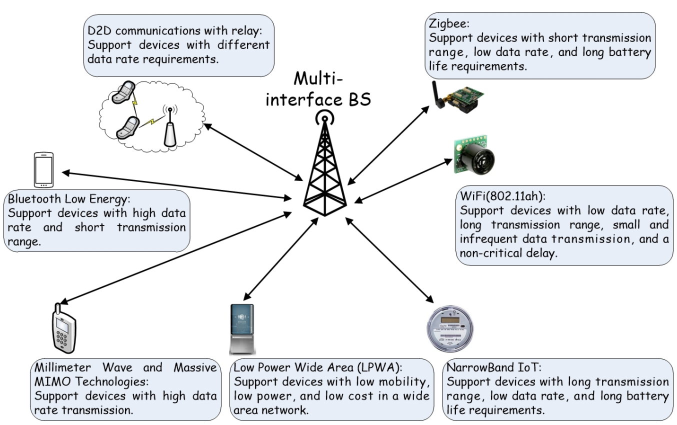


Fig. 2: The illustration of a multi-interface BS.

A. Multi-interface base stations in the cellular network

A huge number of Base Stations (BSs), which have already been deployed in the mobile network, provide high radio coverage. Thus, the distributed BSs have the potential to connect all the IoT devices whether they are moving or static. In order to support different data transmission requirements of the IoT devices, each BS may equip with multiple wireless interfaces, as shown in Fig. 2, to facilitate emerging IoT based wireless communications technologies such as Zigbee, D2D communications with relay, Bluetooth Low Energy, millimeter wave and massive MIMO communications, Low Power Wide Area technologies, and NarrowBand IoT communications. Thus, the multi-interface BS can be considered as a wireless gateway to aggregate all the raw data streams from the local IoT devices. Therefore, a potential deployment is to connect each BS with a fog node to process the aggregated raw data streams.

B. The edgeIoT architecture

Fig. 3 shows the proposed edgeIoT. The locations of the fog nodes are flexible, i.e., a fog node can be directly connected to a BS via high speed fibers to transmit the local data streams with the minimum E2E delay, or can be deployed at the edge of the cellular core network so that different BSs can share the same fog node to process their local data streams. Instead of applying the traditional cellular core network, which leads to the inefficient, inflexible and unscalable packet forwarding and QoS management, the SDN based cellular core is introduced [7], [8]. OpenFlow switches are adopted in the SDN cellular core to separate out all control functions from the data forwarding function. All the switches as well as BSs are controlled by the OpenFlow controller via the OpenFlow protocol [9]. The OpenFlow controller manages the forwarding plane of BSs and OpenFlow switches, monitors the traffic at the data plane and establishes user sessions. Also, it provides Application Programming Interfaces (APIs) to network management operators so that different network functionalities, such as mobility management, user authentication, authorization and accounting, network visualization and QoS control, can be added, removed, and modified flexibly.

Note that each fog node has the ability to access the cloud through the Internet to provision computation availability and flexible application service deployment. That is, when the fog nodes do not have enough computing resources to process their local data streams, they can offload their computing workloads to the cloud at the expense of consuming more network resources and higher communications latency. Furthermore, IoT applications can be deployed in the local fog nodes or in the remote cloud to offer services to users. The flexible application service deployment will be detailed in Sec. II-C.

C. Hierarchical fog computing architecture

Most of the data generated by the users' devices contain personal information, such as the photos/videos taken by mobile phones and smart cars, GPS information, health information sensed by wearable devices, and smart home status sensed by the sensors deployed in the smart home. Analyzing these humongous data can benefit not only the user itself but also the whole society. For instance, analyzing the photos/videos taken by the devices can identify and track a terrorist. Specifically, the application provider sends a photo of the terrorist to each fog node, and each fog node locally performs face matching to compare the terrorist's photo with the photos/videos taken by the local devices. If matched, the fog node will upload the corresponding photos/videos to the cloud for further processing. Thus, it seems that users have to share their personal data in order to provision such services. The main challenge is to maintain user privacy in provisioning such services.

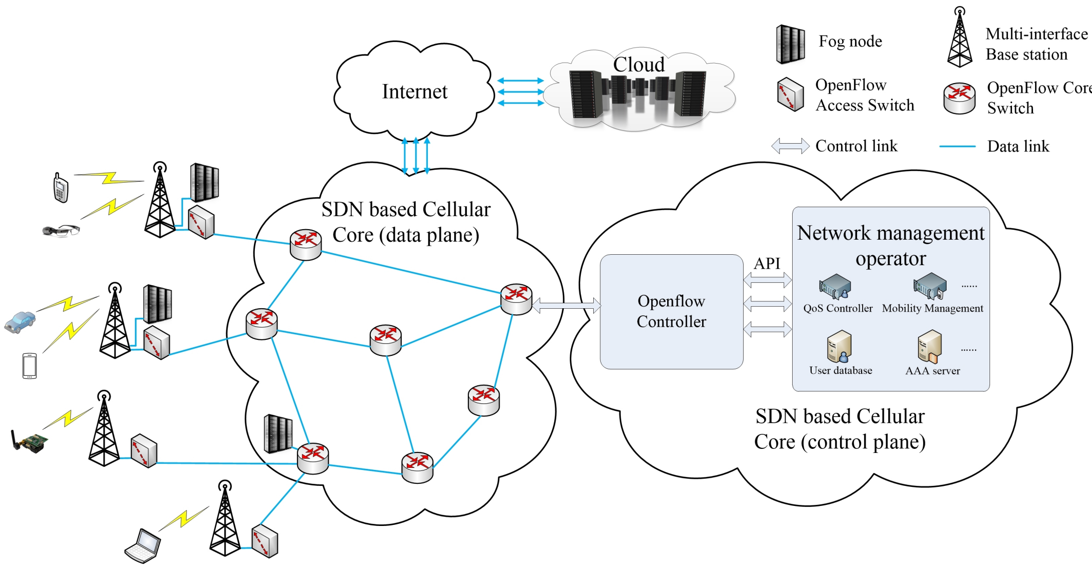


Fig. 3: The edgeIoT architecture.

To tackle this challenge, we propose a hierarchical fog computing architecture. As shown in Fig. 4, each user[1] is associated to a proxy VM, which is considered as the user's private VM (located in a nearby fog node) that provides flexible computing and storage resources. IoT devices belonging to the user is registered to the user's proxy VM, which collects the raw data streams generated from its registered devices via the multi-interface BS, classifies them into different groups based on the type of data (i.e., structurize the raw data steams), generates the metadata by analyzing the corresponding data streams, and sends the metadata to the corresponding application VM. Note that the metadata contains valuable information generated from the raw data streams without violating user privacy. For instance, in the terrorist detection application, only the locations and the time stamps of the matched photos/videos, rather than the original photos/videos, are uploaded to the application VM. The application VM, which is owned by the IoT service provider, offers the semantic model for generating the metadata by each proxy VM (such as the face matching algorithm in the terrorist detection application), receives the metadata from different Proxy VMs, and provide services to users. For instance, all the terrorists will be identified, tracked and arrested by analyzing the metadata from different Proxy VMs, thus safeguarding our society.

The locations of proxy VMs can be dynamic, i.e., if the registered devices are statically deployed (such as the sensors in the smart home), the proxy VM can also remain static in the nearby fog node; if some of the registered devices are mobile (such as the user's mobile phone and wearable devices move from home to workplace), as shown in Fig. 5, the user's proxy VM can be decomposed into two proxy VMs: one proxy VM continues to serve the static IoT devices (in the home) and the other proxy VM migrates to the other fog nodes as the mobile IoT devices roam away. The purpose for doing proxy VM migration is to minimize the traffic (i.e., uploading the raw data streams from the mobile devices to its proxy VM in the fog node) of the cellular core network as well as the E2E delay between the user's mobile IoT devices and its proxy VM.

Proxy VM decomposition refers to the deconsolidation of the original proxy VM into two separate proxy VMs, each of which serves a subset of the registered IoT devices from the original proxy VM (i.e., each proxy VM contains profiles and semantic models of its served IoT devices); conversely, proxy VM composition refers to the consolidation of two proxy VMs (which belong to the same user) into one proxy VM, which serves all the registered IoT devices from the original two proxy VMs. In addition, proxy VM migration involves moving the whole proxy VM (containing profiles, semantic models and recent sensed data of the registered IoT devices) from a source PM to a destination PM. The proxy VM composition/decomposition process always invokes the proxy VM migration process.

The locations of application VMs are also dynamic and flexible, i.e., each application VM can be deployed in the local mode, remote mode or add-on mode.

- **Local application VM deployment** refers to the deployment of an application VM in the fog node to analyze the metadata generated by the local proxy VMs[2]; for instance, in the ParkNet application [10], which helps users find available parking spots in the urban area, each local proxy VM collects the sensed data streams from its smart cars (note that each smart car is equipped with a GPS receiver and a passenger-side-facing ultrasonic rangefinder to generate the location and parking spot occupancy information) and generates the metadata, which

[1]A user can be a person (who owns various private IoT devices), an entity/company (which deploys a set of IoT devices in the area, such as the surveillance cameras), or a group of users who trust each other and share the same proxy VM.

[2]The local proxy VMs refer to the proxy VMs and the application VMs located in the same fog node.

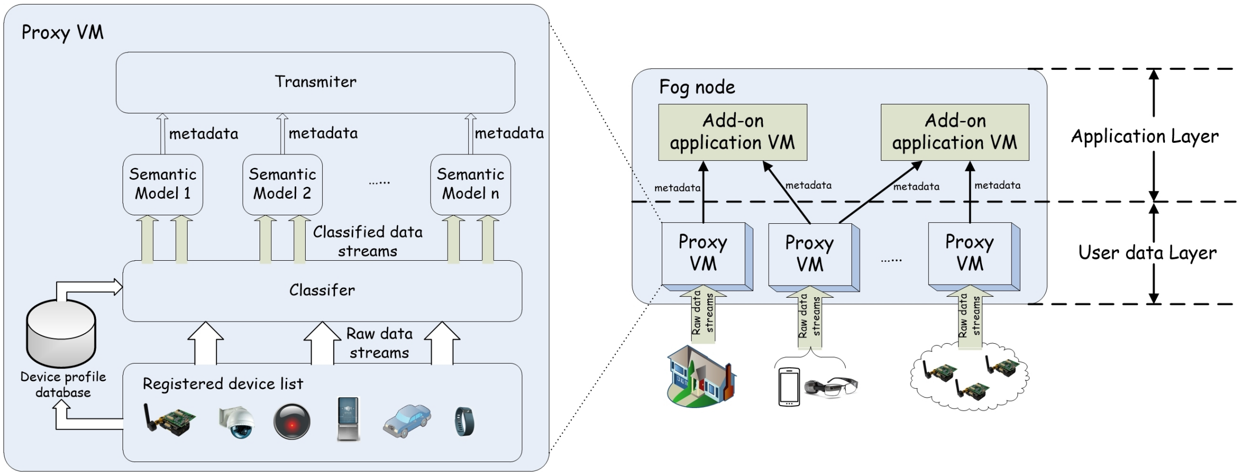


Fig. 4: The hierarchical fog computing architecture.

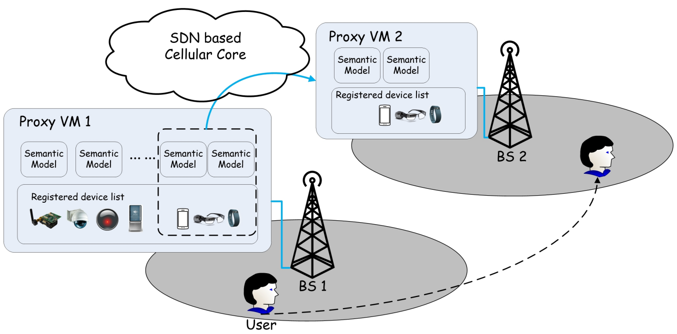


Fig. 5: The illustration of the proxy VM decomposition and migration process.

identify the available parking spots, to the application VM. The application VM will inform and assign the available parking spots to the local smart cars.

- **Remote application VM deployment** refers to the deployment of an application VM in the remote cloud to analyze the metadata generated by the proxy VMs from different fog nodes. This deployment is necessary if an application VM needs information from a large area, such as traffic rerouting applications. Specifically, the goal of the application is to detect the traffic hotspots and select the best routing (i.e., the least time to reach the destination) for users. In order to detect the traffic hotspots, each smart car is equipped with sensors to measure the location and speed of the car. The sensed data streams are transmitted to the proxy VMs, which locally analyze the data streams and generate the metadata indicating the traffic congestion degree of the location. The central server in the remote cloud receives the metadata from the proxy VMs and select the best route for each user.
- **Add-on application VM deployment**, i.e., event-triggered application VM deployment, implies that an application VM can be locally created by some events, such as the terrorist detection application and the find-missing-children application [11]. The events, like lost children and terrorist activities detection, are reported in a specific area and the applications need to identify and track the lost children/terrorists. Then, the applications will be created in each fog node in the area and request each proxy VM in the fog node to run the face matching algorithm in order to compare the recent photos/videos captured by the proxy VMs' registered devices to the photos of lost children/terrorists, and return the locations and time stamps of the photos/videos if found.

D. How to implement edgeIoT applications

If a user is interested in one IoT application, for instance, the ParkNet application, she can download and install this app in her smart car/mobile phone. Accordingly, the user's proxy VM will install the semantic model (which calculates the available parking spots based on the sensed data) provided by the ParkNet application, and the semantic model in the user's proxy VM would have the permission to access the sensed data generated by the GPS receiver and the passenger-side-facing ultrasonic rangefinder equipped in the user's smart car. As a reward, the user can request to find and reserve an available park spot via the ParkNet application.

III. Challenges in implementing edgeIoT

In this section, we will point out some challenges in implementing the proposed edgeIoT architecture and the corresponding solutions.

A. Identifications between IoT devices and their proxy VMs

Initially, each user's IoT devices should be identified/registered by its proxy VM. The proxy VM should know the IDs[3] of all the user's devices and their corresponding characteristics (i.e., static or mobile devices, smart sensors sensing data or actuators responding actions, the types of sensed data, etc). On the other hand, the user's IoT devices should also be informed of the ID of the proxy VM so that

[3] Recently, many methods have been proposed to identify IoT devices, such as electronic product codes, ubiquitous codes, the IPv6 addressing method, etc.

they (i.e., sensor devices) can transmit the private information to the correct proxy VM or they (i.e., actuator devices) can receive commands from the correct proxy VM.

Each mobile IoT device's proxy VM may vary over time owing to the decomposition/composition processes. Thus, the proxy VM need to inform its registered mobile IoT devices when the decomposition/composition processes are triggered.

B. Proxy VMs mobility management

When a mobile IoT device roams from one BS into another BS, it should report its new location (i.e., the mobile IoT device is within the BS's coverage) to the Mobility Management Entity (MME), which is a network management operator in the OpenFlow control layer, through the location update procedure. Mobility management is critical in edgeIoT because proxy VM decomposition/composition processes and proxy VM migration processes are determined by the locations of the proxy VMs' registered IoT devices. The proxy VM should be aware of the locations of its registered IoT devices so that it can communicate with the corresponding IoT devices via the IoT device's associated BS.

Adopting the existing mobility management in the existing LTE network is one solution; however, it requires each mobile IoT device to be equipped with a SIM card for the identification and supporting the location update protocol involved in the LTE network, which is not scalable and economical. Since most of the IoT devices are attached to their users, one alternative is to establish a local cluster network (such as the body area network) consisting mobile IoT devices. The user's mobile phone or other wearable device acts as a cluster head, which can be considered as a gateway to report the locations of the IoT devices in the network, aggregate the IoT devices' sensed data streams, and upload the data streams to the corresponding proxy VM. Note that the cluster head should have the localization capability to identify its geographic location or its associated BS's ID by applying existing wireless localization technologies, such as WiFi based localization, LTE mobility management, and Bluetooth Low Energy (BLE) beacons based localization. The location of the cluster head represents the locations of all the members in the local cluster network.

C. IoT devices migration management

As mentioned earlier, the IoT device's proxy VM can be decomposed and migrated among the fog nodes in order to minimize the latency for uploading the sensed data streams from the IoT devices as well as reduce the traffic load of the SDN based cellular core. It is not necessary to migrate the IoT device's proxy VM whenever the IoT device roams into a new BS's coverage area, i.e., some proxy VM migrations cannot reduce the latency but increase the traffic load of the core network. For instance, as shown in Fig. 5, a user's mobile IoT devices roam from BS 1 into BS 2, and thus their proxy VM (denoted as proxy VM 2) is decomposed from the original proxy VM (denoted as proxy VM 1) and migrates to fog node 2. If migrating proxy VM 2 from fog node 1 to fog node 2 takes T units of time (note that before the migration process is completed, the mobile IoT devices still need to upload their raw data streams to proxy VM 1 via the SDN based cellular core) and the mobile IoT devices move out of the coverage area of BS 2 before the migration process is completed, such migration is obviously inappropriate because it increases the traffic load of the SDN based cellular core (i.e., all the raw data streams generated from the user's mobile IoT devices should still traverse the SDN based cellular core; in addition, extra traffic is introduced for doing migration) without improving the E2E delay between user's mobile IoT devices and their proxy VM.

It is thus necessary to estimate the profit for migrating the proxy VM among the fog nodes whenever the user's mobile IoT devices roam into a new BS. The migration profit, denoted as p, is defined as the total SDN based core network traffic reduction between migrating the proxy VM and without migrating the proxy VM whenever the user's mobile IoT devices roam into a new BS, i.e., $p = L^{static} - L^{mig}$, where L^{mig} and L^{static} are the total traffics generated in the SDN based core network for doing migration and without doing migration, respectively. L^{mig} comprises two parts: the migration traffic and the total data streams transmitted between the proxy VM and its registered IoT devices during the migration process[4], i.e., $L^{mig} = T^{mig}\left(r^{mig} + r^{data}\right)$, where T^{mig} is the total migration time, r^{mig} is the average bandwidth provisioning for doing migration, r^{data} is the average data rate for transmitting the data streams between the user's mobile IoT devices and their proxy VM. Meanwhile, L^{static} is contributed to the total data streams transmitted between the proxy VM and its registered mobile IoT devices when the mobile IoT devices remain in the new BS, i.e., $L^{static} = T^{BS} r^{data}$, where T^{BS} is the retention time of the mobile IoT devices remained in the new BS.

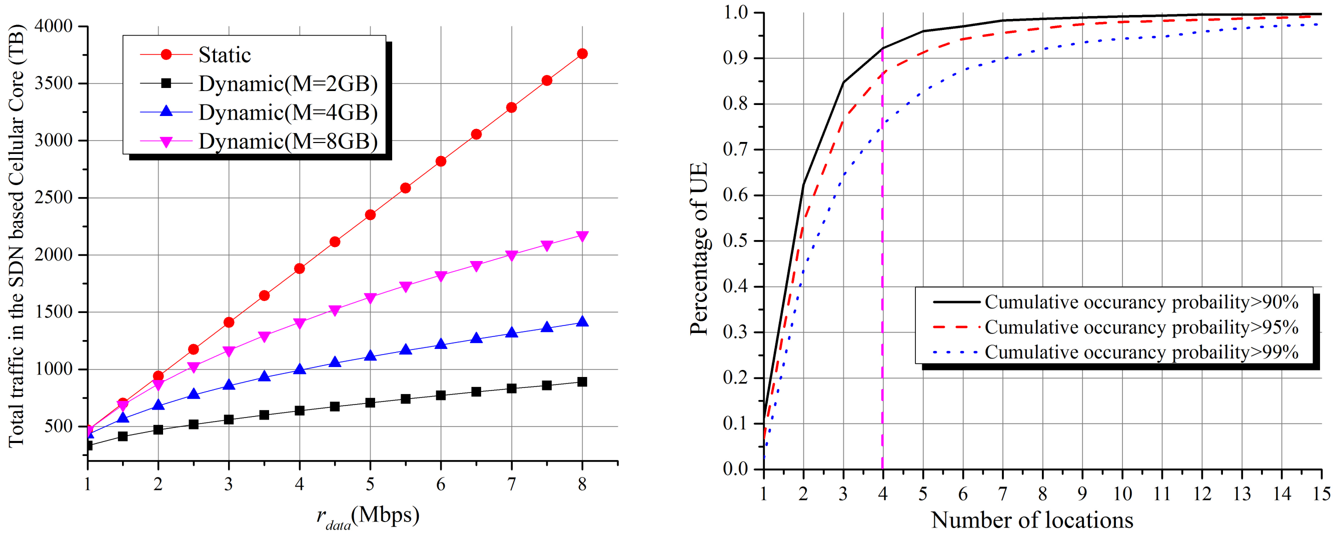
Apparently, an appropriate proxy VM migration implies that the estimated migration profit is larger than a predefined value ε, i.e., $L^{static} - L^{mig} > \varepsilon$, where $\varepsilon \geq 0$. Thus, we can derive:

$$\begin{aligned} & T^{BS} r^{data} - T^{mig}\left(r^{mig} + r^{data}\right) > \varepsilon \\ & \Rightarrow T^{mig} < \tfrac{T^{BS} r^{data} - \varepsilon}{r^{mig} + r^{data}} \end{aligned} \qquad (1)$$

Eq. 1 indicates that the migration can benefit the network only if the migration time T^{mig} is less than $\frac{T^{BS} r^{data} - \varepsilon}{r^{mig} + r^{data}}$. Owing to the fact that about 10% to 30% of all human movements are attributed to their social relationship, while 50% to 70% to periodic behaviors [12], we believe that the dynamics of future human movements can be reliably predicted based on mathematical models. Mobile IoT devices are usually attached to their users, and thus the value of T^{BS} is predictable. Meanwhile, the values of r^{mig} and r^{data} can also be estimated based on their historical traces. Therefore, the value of $\frac{T^{BS} r^{data} - \varepsilon}{r^{mig} + r^{data}}$ can be reliably estimated. In order to evaluate the migration according to Eq. 1, the migration time T^{mig} should also be predicted.

Normally, the proxy VM migration process comprises many iterations. In the first iteration, all the memory of the source

[4] After the migration is completed, the proxy VM is placed in the fog node, whose connected BS is serving the mobile IoT devices, and so the data streams generated by the mobile IoT devices no longer traverse the SDN based core network to reach the proxy VM.



(a) Total traffic in the SDN based cellular core versus the average data rate of mobile IoT devices (given $\varsigma = 500Kbits$).

(b) The statistical results of the user mobility trace.

Fig. 6: Simulation results.

proxy VM is migrated to the destination. Since source proxy VM is still serving the user's IoT devices, the content of the memory may change during the first iteration. Thus, in the second iteration, the dirty memory pages, which are generated in the first iteration, will be transmitted to the destination. The iteration is repeated until the dirty memory pages, which are generated in the previous iteration, are less than the predefined threshold, denoted as ς. Then, the source proxy VM stops serving its IoT devices and transmits the rest of the dirty memory pages to the destination; finally, the destination proxy VM resumes to serve its IoT devices. Thus, the migration time should be a function of the average data rate for doing the migration r^{mig}, the average dirty memory pages generation rate r^{dir}, the initial proxy VM memory size M and the threshold value ς, i.e., $T^{mig} = f(r^{mig}, r^{dir}, M, \varsigma)$. Based on the model proposed by [13], the migration time can be reliably estimated given the average transmission data rate for doing migration.

In order to investigate how the proxy VM migration affects the total traffic in the core network, we evaluate the total traffic in the cellular core network during the day by applying the dynamic proxy VM migration as compared to the static proxy VM deployment (i.e., each proxy VM does not migrate among fog nodes after its initial deployment). In order to emulate each user's behavior, we have obtained data traces of more than 13,000 users and extracted their mobility in one day in Heilongjiang province in China. The whole area contains 5,962 BSs (each BS is connected with a fog node) and each user's location (i.e., the user within the BS's coverage area) is monitored for every minute during the day. Meanwhile, the SDN based cellular core network can guarantee the average transmission rate for doing migration to be $20Mbps$, i.e., $r^{mig} = 20Mpbs$. Each user's mobile IoT devices are attached to their own user and generates the data streams over time with the same average data rate r_{data}. Fig. 6(a) shows the total traffic in the cellular core network during the day by varying the average data rate for transmitting the data streams between the user's mobile IoT devices and their proxy VM. Clearly, applying dynamic proxy VM migration can reduce more traffic in the SDN based cellular core as compared to the static proxy VM deployment when r_{data} increases. However, as the total amount of memory of each proxy VM (i.e., M) increases, the total traffic in the core network significantly increases accordingly. This is because as the value of M increases, the migration time becomes longer (i.e., more traffic would be generated for doing migration), and thus more proxy VMs are preferred to stay in their original fog nodes in order to avoid the huge volume of migrate traffic.

One solution to alleviate the traffic load of the core network (when the value of M is large) is to pre-allocate replicas of the users' proxy VMs in the fog nodes. Specifically, the major part of the memory is the semantic models and device profiles (which are not dynamically changed after the initial installation) in the proxy VM. Thus, the replicas of the mobile IoT's semantic models can be pre-allocated to the corresponding fog nodes, whose connected BSs are commonly visited by the user (such as the user's home and workplace). Note that we further analyze the mentioned user's mobility trace and find out that each user mainly settles in some areas covered by a few number of BSs, i.e., as shown in Fig. 6(b), 92.22%, 86.93% and 75.65% of the users spend 90%, 95% and 99% of the time during the day (viz., 21.6, 22.8 and 23.76 hours) to stay at only four locations, respectively. This observation helps us determine the proper number and locations of replicas for each user's IoT devices. Thus, if a proxy VM tries to migrate to another fog node (which contains one of the proxy VM's replicas), rather than transmitting the whole memory of the proxy VM, only the differences (between the proxy VM migration and its replicas) need to be transferred, thus dramatically reducing the migration time as well as the migration traffic.

D. Energy consumption consideration

Deploying fog nodes at the network edge may increase the operational cost for processing the IoT data streams as compared to processing the IoT data streams in the centralized cloud (which provisions efficient and flexible resource and power management to minimize the energy consumption of the cloud). However, introducing green energy in the proposed edgeIoT architecture can substantially reduce the operational cost (i.e., reduce on-grid energy consumption) for the edgeIoT providers [14]. Specifically, each fog node can be powered by both green energy and on-grid energy. The fog node would first consume green energy and then on-grid energy if green energy is not enough to satisfy the energy demands of the hosting proxy VMs in the fog node. Some fog nodes, which have less energy demand and more green energy generated, would have excess of green energy while some, which have more energy demands and less green energy generated, would consume on-grid energy. Thus, proxy VMs can be migrated from the fog nodes (which consume on-grid energy) to the fog nodes (which have excessive green energy) in order to further reduce on-grid energy consumption.

IV. Conclusion

This article proposes a new architecture, edgeIoT, in order to efficiently handle the raw data streams generated from the massive distributed IoT devices at the mobile edge. The proposed edgeIoT architecture can substantially reduce the traffic load in the core network and the E2E delay between IoT devices and computing resources as compared to the traditional IoT architecture, and thus facilitate the IoT services provisioning. Moreover, this article had raised three challenges in implementing the proposed edgeIoT architecture and provided the potential solutions.

References

[1] H. Sundmaeker, P. Guillemin, P. Friess, and S. Woelffle, "Vision and Challenges for Realising the Internet of Things," European Commission Information Society and Media, Tech. Rep., March, 2010.

[2] Networking, Cisco Visual, "Cisco Global Cloud Index: Forecast and Methodology, 2014-2019," White paper, 2015.

[3] X. Sun and N. Ansari, "Optimizing Resource Utilization of a Data Center," *IEEE Communications Surveys & Tutorials*, DOI: 10.1109/COMST.2016.25582032016, early access, 2016.

[4] H. L. Truong and S. Dustdar, "Principles for Engineering IoT Cloud Systems," in *IEEE Cloud Computing*, vol. 2, no. 2, pp. 68–76, Mar.–Apr. 2015.

[5] L. Wang and R. Ranjan, "Processing Distributed Internet of Things Data in Clouds," in *IEEE Cloud Computing*, vol. 2, no. 1, pp. 76–80, Jan.–Feb. 2015.

[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*, Aug. 13–17, 2012, Helsinki, Finland, pp. 13–16.

[7] X. Jin, L. E. Li, L. Vanbever, and J. Rexford, "Softcell: Scalable and flexible cellular core network architecture," in *Proceedings of the 9th ACM conference on Emerging networking experiments and technologies*, Santa Barbara, CA, Dec. 09–12, 2013, pp. 163–174.

[8] X. Sun, N. Ansari, and Q. Fan, "Green Energy Aware Avatar Migration Strategy in Green Cloudlet Networks," *2015 IEEE 7th International Conference on Cloud Computing Technology and Science (CloudCom)*, Vancouver, BC, Nov. 30–Dec. 3, 2015, pp. 139–146.

[9] A. Lara, A. Kolasani, and B. Ramamurthy, "Network Innovation using OpenFlow: A Survey," *IEEE Commun. Surv. Tut.*, vol.16, no.1, 2014, pp. 483–512.

[10] S. Mathur, *et al.*, "Parknet: drive-by sensing of road-side parking statistics," *Proceedings of the 8th international conference on Mobile systems, applications, and services*, San Francisco, CA, Jun. 15–18, 2010, pp. 123–136.

[11] M. A. Khan, *et al.*, "Moitree: A Middleware for Cloud-Assisted Mobile Distributed Apps," in *the 4th IEEE International Conference on Mobile Cloud Computing, Services, and Engineering (MobileCloud16)*, Oxford, UK, Mar.29–Apr.1, 2016, pp. 21-30.

[12] E. Cho, S.A. Myers, and J. Leskovec, "Friendship and mobility: user movement in location-based social networks," in *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, San Diego, CA, Aug. 21–24, 2011, pp. 1082–1090.

[13] X. Sun and N. Ansari,"PRIMAL: PRofIt Maximization Avatar pLacement for Mobile Edge Computing," in *Proceedings of IEEE International Conference on Communications (ICC)*, Kuala Lumpur, Malaysia, May 23–27, 2016.

[14] X. Sun and N. Ansari, "Green Cloudlet Network: A Distributed Green Mobile Cloud Network," *IEEE Network Magazine*, to appear; also available in Computing Research Repository (CoRR), arXiv:1605.07512, 2016.

Xiang Sun [S'13] received the B.E. degree in electronic and information engineering and the M.E. degree in technology of computer applications from Hebei University of Engineering, Hebei, China. He is currently working towards the Ph.D. degree in Electrical Engineering at the New Jersey Institute of Technology (NJIT), Newark, New Jersey. His research interests include mobile edge computing, big data networking, green edge computing and communications, cloud computing, and Internet of Things.

Nirwan Ansari [S'78,M'83,SM'94,F'09] received BSEE (summa cum laude with a perfect GPA) from the New Jersey Institute of Technology (NJIT), MSEE from the University of Michigan, Ann Arbor, and PhD from Purdue University, West Lafayette, IN. He is Distinguished Professor of Electrical and Computer Engineering at NJIT, where he joined in 1988. He has also assumed various administrative positions at NJIT and has been Visiting (Chair) Professor at several universities. He has also (co-)authored more than 500 technical papers, over 200 in widely cited refereed journals/magazines. He has guest-edited a number of special issues, covering various emerging topics in communications and networking. His current research focuses on green communications and networking, cloud computing, and various aspects of broadband networks. Prof. Ansari has served on the Editorial Board and Advisory Board of over ten journals, including as a Senior Technical Editor of IEEE Communications Magazine (2006–2009). He was elected to serve in the IEEE Communications Society (ComSoc) Board of Governors as a member-at-large (2013–2015). He has chaired ComSoc technical committees, and has been actively organizing numerous IEEE International Conferences/Symposia/Workshops, assuming various leadership roles. He frequently delivers keynote addresses, distinguished lectures, tutorials, and invited talks. Some of his recognitions include some Best Paper awards, several Excellence in Teaching Awards, the Thomas Alva Edison Patent Award (2010), New Jerseys Inventors Hall of Fame Inventor of the Year Award (2012), NCE Excellence in Research Award (2014), Purdue University Outstanding Electrical and Computer Engineer Award (2015), and designation as an IEEE Communications Society Distinguished Lecturer (2006-2009). He has also been granted over thirty US patents.