IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

SPECIAL SECTION ON MOBILE EDGE COMPUTING

Received October 20, 2017, accepted November 18, 2017, date of publication November 29, 2017, date of current version March 9, 2018.

Digital Object Identifier 10.1109/ACCESS.2017.2778504

A Survey on the Edge Computing for the Internet of Things

WEI YU[1], FAN LIANG[1], XIAOFEI HE[1,2], WILLIAM GRANT HATCHER[1], CHAO LU[1], JIE LIN[2], AND XINYU YANG[2]

[1]Department of Computer and Information Sciences, Towson University, Towson, MD 21252, USA
[2]School of Electronic and Information Engineering, Xi'an Jiaotong University, Xi'an 710049, China

Corresponding authors: Wei Yu (wyu@towson.edu) and Xiaofei He (hexiaofei16@gmail.com)

This work was supported in part by the U.S. National Science Foundation under Grant CNS 1350145, in part by the University System of Maryland Wilson H. Elkins Professorship Fund, in part by the Natural Science Foundation of China under Grant 61572398, and in part by the Fundamental Research Funds for the Central Universities under Grant xjj2017149.

ABSTRACT The Internet of Things (IoT) now permeates our daily lives, providing important measurement and collection tools to inform our every decision. Millions of sensors and devices are continuously producing data and exchanging important messages via complex networks supporting machine-to-machine communications and monitoring and controlling critical smart-world infrastructures. As a strategy to mitigate the escalation in resource congestion, edge computing has emerged as a new paradigm to solve IoT and localized computing needs. Compared with the well-known cloud computing, edge computing will migrate data computation or storage to the network "edge," near the end users. Thus, a number of computation nodes distributed across the network can offload the computational stress away from the centralized data center, and can significantly reduce the latency in message exchange. In addition, the distributed structure can balance network traffic and avoid the traffic peaks in IoT networks, reducing the transmission latency between edge/cloudlet servers and end users, as well as reducing response times for real-time IoT applications in comparison with traditional cloud services. Furthermore, by transferring computation and communication overhead from nodes with limited battery supply to nodes with significant power resources, the system can extend the lifetime of the individual nodes. In this paper, we conduct a comprehensive survey, analyzing how edge computing improves the performance of IoT networks. We categorize edge computing into different groups based on architecture, and study their performance by comparing network latency, bandwidth occupation, energy consumption, and overhead. In addition, we consider security issues in edge computing, evaluating the availability, integrity, and the confidentiality of security strategies of each group, and propose a framework for security evaluation of IoT networks with edge computing. Finally, we compare the performance of various IoT applications (smart city, smart grid, smart transportation, and so on) in edge computing and traditional cloud computing architectures.

INDEX TERMS Edge computing, Internet of Things, survey.

I. INTRODUCTION

With the progressing development of information technology, the Internet of Things (IoT) has come to play an important role in our daily lives. Interconnected sensors/devices can collect and exchange different data amongst themselves through modern communication network infrastructure connected by millions of IoT nodes [1]–[4]. Then, a variety of IoT applications can provide more accurate and more fine-grained network services for users. In this case, more and more sensors and devices are being interconnected via IoT techniques, and these sensors and devices will generate massive data and demand further processing, providing intelligence to both service providers and users. In conventional cloud computing, all data must be uploaded to centralized servers, and after computation, the results need to be sent back to the sensors and devices. This process creates great pressure on the network, specifically in the data transmission costs of bandwidth and resources. In addition, the performance of the network will worsen with increasing data size.

A more critical situation arises for IoT applications that are time-sensitive, meaning that very short response times are non-negotiable (the smart transportation [5], smart

electricity grid [6], [7], smart city [8]–[10], etc.) and conventional cloud computing-based service definitively cannot satisfy the demand. This is because the computation processes need to be uploaded to the cloud, and the limited bandwidth and network resources are occupied by massive data transmissions, on top of the cloud already being far from the end users. Obviously, the result will be large latency in the networks, which is unacceptable for time-sensitive IoT applications. This is an important problem for IoT, as these applications will have an impact on safety and emergency response.

Furthermore, most IoT devices have limited power (smart sensors, etc.), and to extend the lifetime of devices, it is necessary to balance power consumption by scheduling computation to devices that have higher power and computational capabilities. In addition, processing data in computation nodes with the shortest distance to the user will reduce transmission time. In cloud computing-based service, the data transmission speed will be affected by the network traffic, and heavy traffic leads to long transmission times, increasing power consumption costs. Thus, scheduling and processing allocation is a critical issue that should be considered.

To address the aforementioned problems and issues, in this paper we summarize existing efforts and previous work [11]–[17], and present our view on edge computing for the IoT. Edge computing encompasses data computing and storage that is being performed at the network ''edge'' [18]–[25], nearby the user. Due to the locations of edge computing nodes being close to end users, the peak in traffic flows will be alleviated. In addition, it significantly mitigates the bandwidth requirements of the centralized network and reduces the transmission latency during data computing or storage in IoT. Thus, distributing computation nodes deployed at the edge can allow the offloading of traffic and computational pressure from the centralized cloud, and the response times of IoT applications can be faster than the corresponding cloud computing services. In addition, edge computing can migrate computational and communication overhead from nodes with limited battery or power supply to edge nodes with significant power resources. In doing so, the lifetime of the nodes with limited battery will be extended, such that the lifetime of the entire IoT network will be increased.

In this paper, our contributions are listed as follows:

- We review the advantages and disadvantages of edge computing, and categorize edge computing architectures into different groups. Also, we compare the performance of these categories in terms of response time, computation capacity, and storage space.
- We systematically investigate the essence of IoT, and review some typical IoT examples. Based on this investigation, we compare the performance of IoT devices in cloud computing and edge computing. Then, we list the benefits and challenges that edge computing pose on IoT networks.
- Based on thorough studies of both IoT and edge computing, we discuss the potential ability for integrating IoT and edge computing as edge computing-based IoT. Then, we introduce the problem space for edge computing-based IoT. From the designed problem space, we review architectures, performance, task scheduling, and security and privacy in edge computing.
- Furthermore, we illustrate the advantages and disadvantages of edge computing assisted IoT in transmission, storage, and computation. We discuss the new challenges from the perspectives of system integration, resource management, security and privacy, and advanced communication. We also present some IoT smart applications as examples to explain how the edge computing works with the IoT.

The remainder of this paper is organized as follows: In Section II, we briefly discuss the background and basic concepts of IoT, edge computing and cloud computing. In Section III, we list the characteristics of IoT and edge computing, and analyze the benefits of using edge computing to assist IoT, demonstrating the potential of integrating them together. Meanwhile, we introduce the architecture of the IoT and the structure of edge computing. In Section IV, we discuss the benefits that combine IoT and edge computing together. We identify the problem space and form transmission, storage, and computation perspectives to illustrate the details. In Section V, we discuss the challenges for edge computing-based IoT. Finally, we conclude the paper in Section VI.

II. REVIEW OF IoT AND EDGE COMPUTING

In this section, we will review the basic concepts of IoT and edge computing, and discuss the potential for integrating the two technologies.

A. INTERNET OF THINGS

The future direction of computing will exceed traditional computing based on stationary desktop [26]. Particularly, the IoT is merging into daily life rapidly, as a novel technology of the past few years. As a paradigm, IoT envisions that most physical devices, such as smart mobile phones, vehicles, sensors, actuators, and any other embedded devices will be connected and communicate with data centers, exchange information, and introduce the next massive jump in scale of data production.

Following various popularized technologies, such as smart transportation, smart city, smart grid and smart healthcare, people will not function without IoT suffusing their home and work existence. Thus, IoT will remarkably impact daily life of prospective users, and is the key to the future. IoT also takes an important role in the field of business. Indeed, IoT was reported as one of the most important technologies that will impact US interests in 2025 [27]. Likewise, the number of the interconnected psychical devices has transcended the human population of the world. In 2012, the number of interconnected physical devices increased to 9 billions [26],

and the estimated number of interconnected physical devices will be 75 billion around 2020 [28]. IoT devices will thus be one of the most important and eclipsing data sources for big data in future.

In the following, we will describe three different communication models for IoT.

1) MACHINE-TO-MACHINE COMMUNICATION

This communication model represents multiple devices, which can connect and exchange information between each other directly, without any intermediary hardware assistance [29]. These devices are able to connect with each other over different types of networks, including but not limited to Internet or IP networks. For example, Fig. 1 shows that a smart switch communicates with the smart light over Bluetooth 4.0.

FIGURE 1. An example of Machine-to-Machine communications.

These device-to-device networks allow devices to exchange information in hybrid communication protocols, which combine device-to-device and particular communication protocol to achieve the QoS requirements. This model is commonly used in numerous applications, such as smart home systems or automatic control in electrical systems, which communicate with each other via sending small data packets and have relatively low data rate requirements. The typical IoT devices of this type are smart door locks, smart switches, and smart lights, among others, which also typically only exchange small data packets.

From the users perspective, the problem of Machine-to-Machine communications is lack of compatibility, in which different devices from different manufacturers use different protocols. Using smart home devices as an example, Z-Wave protocol devices cannot communicate with the ZigBee protocol devices [30]. These compatibility issues limit the users choice and experience.

2) MACHINE-TO-CLOUD COMMUNICATION

In a device-to-cloud communication model, IoT devices demand service from a cloud application service provider, or store data into cloud storage disk [29], because of the limitations of the devices computational ability or storage space. This approach normally requires assistance from pre-existing communications strategies like conventional wired or Wi-Fi connections, shown in Fig. 2.

Though the Machine-to-Cloud communication solves the problems of the Machine-to-Machine model, this model is dependent to the traditional network, and the bandwidth and the network resources limit the performance of this communication model. To improve the performance of the

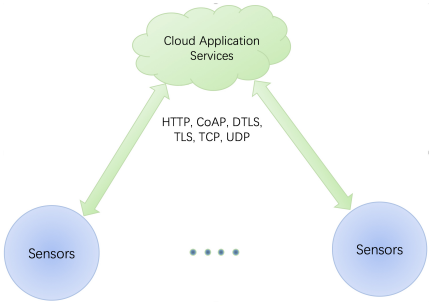


FIGURE 2. An example of Machine-to-Cloud communications.

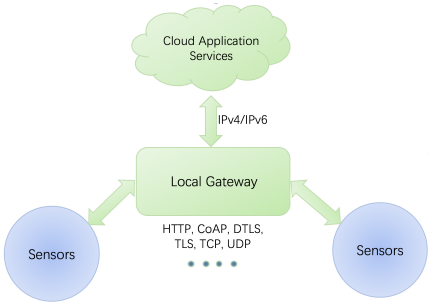


FIGURE 3. An example of Machine-to-Gateway communications.

Machine-to-Cloud communication model, it is necessary to optimize the network structure.

3) MACHINE-TO-GATEWAY COMMUNICATION

In the machine-to-gateway model, the device-to-application-layer gateway (ALG) model is considered as a proxy or middleware box [29]. In Fig. 3, we can see the structure of Machine-to-Gateway communications. In the application layer, some software-based security check schemes or other functionality like data or protocol translation algorithms run on a gateway or other network device, which acts an intermediary bridge between IoT devices and cloud application services. This improves the security and flexibility of the IoT network, migrates a part of the computation task to the application layer, and significantly reduces the power consumption of the IoT devices. For instance, the smart mobile phone acts as the gateway, running some applications to communicate with the IoT devices and the cloud. This appears in the personal health domain, such as when sensors generate data and connect with a personal smart phone, then the smart device will encrypt the data and upload to the cloud service providers.

B. CONVENTIONAL IoT COMPONENTS

Typically, there exist three types of components in an IoT network: sensors/devices, IoT gateways/local network, and backhaul network/cloud, representing the data source, data communication networks, and data processing, respectively.

1) SENSORS/DEVICES

In the IoT, millions of sensors are deployed in a wide area. These sensors are the key component of IoT, and they produce the majority of measurement data in the networks. These sensors can provide diverse types of data to help the IoT be aware of everything. In addition, the end devices of users generate most of the resource requirements. For end users, the devices can serve as human-computer interfaces to produce the requirements of users and forward them to the IoT. All these sensors and end devices will be interconnected so that they can exchange data with each other and provide additional services. Via the network that connects devices, each node can acquire its resource requirements for the IoT applications.

2) IoT GATEWAYS

The IoT gateways connect the network of the sensors and core networks to the cloud servers. When the end nodes generate resource requirements for IoT applications, they will send the data processing or storage tasks to the cloud servers. Although the sensors/devices can establish a network to transmit their generated data, it is necessary to carry out data pre-processing before forwarding them to the cloud servers. Thus, the IoT gateways will collect and aggregate the measurement data from the sensors/devices and forward them to the cloud servers. Generally speaking, the IoT gateways often carry out data pre-processing to reduce redundancy and unnecessary overhead. In addition, the IoT gateways will forward the results of the data processing from the cloud servers back to the end users.

3) CLOUD/CORE NETWORK

Via backhaul networks, cloud servers will receive the data and requirements from end users [31], [32]. To support IoT applications, the cloud servers have significant capacity for computation and storage. Thus, the cloud servers can satisfy the resource requirements of different applications. When the data processing is complete, the cloud servers will send the results back to the end users. Notice that for most IoT applications, the end users will ask for the cloud servers to accomplish the data processing tasks.

C. EDGE COMPUTING

Due to the rapid increase in the number of mobile devices, conventional centralized cloud computing is struggling to satisfy the QoS for many applications. With 5G network technology on the horizon [33]–[35], edge computing will become the key solution to solving this issue. One of major challenges associated with 5G technology is the Radio-Access Network (RAN) [35]. In RAN, mobile edge computing provides real-time RAN information. By using the real-time RAN information, the network providers can improve Quality-of-Experience (QoE) for end users, because real-time RAN will offer context-aware services [36].

As we mentioned before, the edge computing platform allows edge nodes to respond to service demands, reducing bandwidth consumption and network latency. Thus, the network operators can implement RAN into the edge to be handled by third-party co-operators, rapidly increasing the deployment of new applications. On the other hand, the computation nodes are operating under different third-party co-operators, making it difficult to deploy similar security schemes to ensure the same level of security.

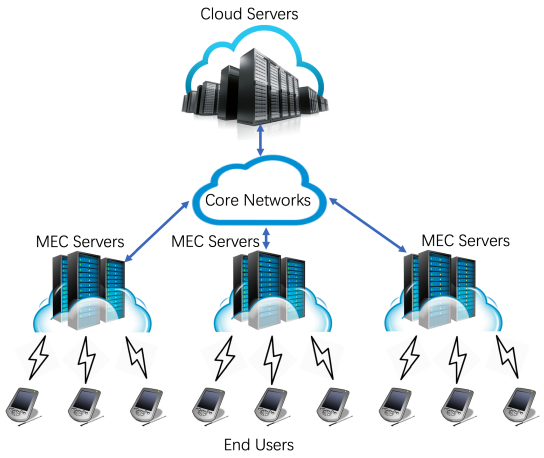


FIGURE 4. The basic edge computing architecture.

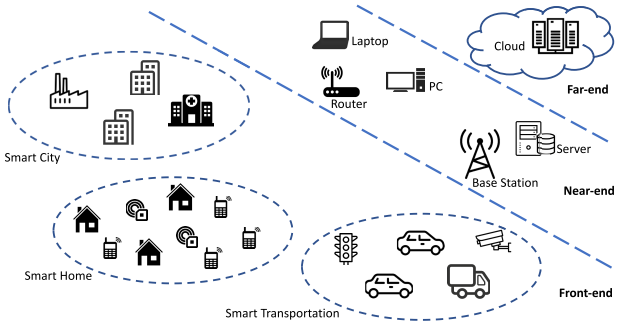


FIGURE 5. A typical architecture of edge computing networks.

D. EDGE COMPUTING ARCHITECTURE

Fig. 4 illustrates the basic architecture of edge computing. Notice that the edge computing servers are closer to the end user than cloud servers. Thus, even though the edge computing servers have less computation power than the cloud servers, they still provide better QoS (Quality of Service) and lower latency to the end users. To study the advantages and disadvantages of edge computing, we will focus on the architectures of both, and compare the two. Obviously, unlike cloud computing, edge computing incorporates edge computation nodes into the network. In this paper, the edge computation nodes are called edge/cloudlet servers. Generally speaking, the structure of edge computing can be divided into three aspects, the front-end, near-end, and far-end, as shown in Fig. 5. The differences among these areas are described below in detail.

1) FRONT-END

The end devices (e.g., sensors, actuators) are deployed at the front-end of the edge computing structure. The front-end environment can provide more interaction and better responsiveness for the end users. With the computing capacity provided by the plethora of nearby end devices, edge computing can provide real-time services for some applications. Nonetheless, due to the limited capacity of the end devices, most requirements cannot be satisfied at the front-end environment. Thus, in these cases, the end devices must forward the resource requirements to the servers.

2) NEAR-END

The gateways deployed in the near-end environment will support most of the traffic flows in the networks. The edge/cloudlet servers can have also numerous resource requirements, such as real-time data processing, data caching, and computation offloading. In edge computing, most of the data computation and storage will be migrated to this near-end environment. In doing so, the end users can achieve a much better performance on data computing and storage, with a small increase in the latency.

3) FAR-END

As the cloud servers are deployed farther away from the end devices, the transmission latency is significant in the networks. Nonetheless, the cloud servers in the far-end environment can provide more computing power and more data storage. For example, the cloud servers can provide massive parallel data processing, big data mining, big data management, machine learning, etc. [31], [32].

E. EDGE COMPUTING IMPLEMENTATION

To implement the aforementioned architecture of edge computing, some research efforts have already focused on the design of edge computing models. Typically, the following two models dominate: (i) Hierarchical model, and (ii) Software-defined model.

1) HIERARCHICAL MODEL

Considering that edge/cloudlet servers can be deployed at different distances from the end users, the edge architecture is divided into a hierarchy, defining functions based on distance and resources. Thus, a hierarchical model is suitable for describing the network structure of edge computing.

There have been a number of research efforts on hierarchical model. For example, Jararweh *et al.* in [37] proposed a hierarchical model, which integrates the Mobile Edge Computing (MEC) servers and cloudlet infrastructures. In this model, the mobile users can obtain their requested services as MEC provides the ability to meet their computing and storage needs. Tong *et al.* in [38] proposed a hierarchical edge cloud model, which can be used to serve peak loads demanded from mobile users. In this model, the cloudlet servers are deployed at the network edge and the regional edge cloud is established as a tree hierarchy, which consists of deployed edge servers. By leveraging this designed hierarchical structure, the computing abilities of edge servers can be further aggregated to meet the need of peak loads.

2) SOFTWARE-DEFINED MODEL

In addition, considering the hundreds of the applications and millions of end users and devices, the management of edge computing for IoT will be exceptionally complicated. Software Defined Networking (SDN) [39]–[42] can be a viable solution to deal with the complexity of edge computing management.

There have been a number of research efforts on SDN model. For example, Jaraweh *et al.* in [41] proposed a software defined model to integrate the Software Defined Systems capabilities and the MEC system. In this way, the management and the administration cost can be reduced. Du and Nakao in [42] proposed an application-specific MEC model. In their model, the paradigm of software-defined data plane is considered in a Mobile Virtual Network Operators (MVNOs) network. Authors designed mechanisms to carry out hop-count-based tethering detection and mobile-friendly optimization. Via the designed mechanisms, fairness among users can be realized by regulating the TCP concurrent connections. Manzalini and Crespi in [43] proposed an edge operating system, which leverages available open source software to achieve powerful network and service platforms. Salman *et al.* in [44] proposed an integration of three new concepts, including MEC, Software Defined Network (SDN), and Network Function Virtualization (NFV). In doing so, this solution is capable of achieving better MEC employment in mobile networks and can be further extended to enable IoT-wide deployment. Lin *et al.* in [45] proposed a Smart Applications on Virtual Infrastructure Software-Defined Infrastructure (SDI) Smart Edge architecture, which can be used to support the construction of various distributed network services and applications.

III. INTEGRATION OF IoT AND EDGE COMPUTING

In this section, we will discuss the potential to integrate IoT and edge computing. Based on our study of the characteristics of both IoT and Edge Computing, we compare the characteristics of IoT, edge computing, and cloud computing. Furthermore, we narrow our focus to the transmission, storage, and computation characteristics to illustrate how edge computing improves the performance of IoT.

A. OVERVIEW

Extending our previous discussion, IoT and edge computing are independently rapidly evolving. Despite their independence, the edge computing platform can help IoT to solve some critical issues and improve performance. Thus, in recent years, it has become clear that these should be integrated. From Fig. 3 and Fig. 4, we can see that IoT and edge computing have smiler characteristics, as further demonstrated

TABLE 1. Characteristics of IoT, edge and cloud computing.

	IoT	Edge	Cloud
Deployment	Distributed	Distributed	Centralized
Components	Physical devices	Edge nodes	Virtual resources
Computational	Limited	Limited	Unlimited
Storage	Small	Limited	Unlimited
Response Time	NA	Fast	Slow
Big data	Source	Process	Process

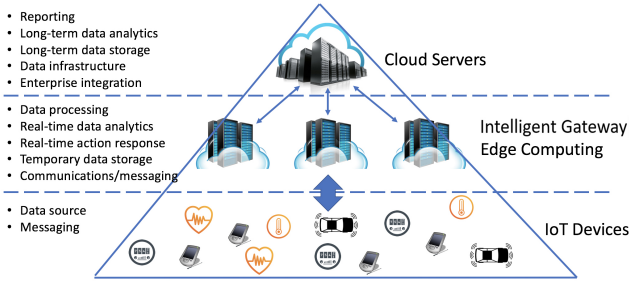


FIGURE 6. Layer architecture of edge computing-based IoT.

in Table 1. Notice that we also include cloud computing as a reference.

Fig. 6 illustrates the three-layer architecture of edge computing-based IoT. It has the same layers as the edge computing structure, and all IoT devices are end users for edge computing. In general, IoT can benefit from both Edge computing and Cloud computing, because of the characteristics of the two structures (i.e., high computational capacity and large storage). Nonetheless, edge computing has further advantages over cloud computing for IoT, even though it has more limited computational capacity and storage. Specifically, IoT requires fast response rather than high computational capacity and large storage. Edge computing offers a tolerable computational capacity, enough storage space, and fast response time to satisfy IoT application requirements.

On the other hand, edge computing can also benefit from IoT by extending the edge computing structure to deal with the edge computing nodes being distributed and dynamic. Either IoT devices or the devices that have residual computation power can be used as edge nodes to provide services. Significantly, a number of research efforts have sought to exploit cloud computing to assist IoT, but in many cases, edge computing can provide much more competitive performance. Due to the increasing number of IoT devices, IoT and edge computing are likely to become inseparable. As we discussed before, most IoT requirements fall into the three categories of transmission, storage, and computation. In the following, we will discuss each category in detail, presenting the advantages that they provide to Edge Computing-assisted IoT.

B. IoT PERFORMANCE DEMANDS

1) TRANSMISSION

The total response time can be computed as the sum of transmission time and processing time. In general, IoT devices create a voluminous amount of data, continuously, but have only limited computational requests [46]. Indeed, large network latency will be unacceptable, and cannot satisfy the QoS requirements. Specific examples include vehicle-to-vehicle communications and vehicle-to-infrastructure communications. Related to public safety concerns and the needs of first responders, response time must be very short too.

Unlike the traditional cloud, edge computing can provide numerous distributed computational nodes, which are close to the end users to supporting real-time information collection and analysis services [14]. Meanwhile, the edge computation nodes also provide acceptable computational capacity to handle the demands of IoT. Thus, the IoT application requirements do not need to undergo the delay in traditional cloud services, such as Amazon Cloud or Google Cloud, but instead can take advantage of the short transmission time of Edge computing.

2) STORAGE

As mentioned above, IoT is the source of prodigious data, and will become the most important part of big data generation, if it is not already. Thus, IoT needs to upload the massive data to edge or cloud based storage. The benefits of uploading to edge based storage is, of course, the short upload time. Nonetheless, the drawback to this is the concern of security in edge-based storage [47]. Because the edge nodes are running in different organizations, it is difficult to ensure the integrity, information protection, anonymity assessment, non-repudiation, and freshness of the original data [48], [49]. In addition, the storage space of edge nodes is limited, and there is no large-scale and long-lived storage to compare with the cloud computing data centers. Finally, when it is necessary to upload the data, different edge nodes will be employed and coordinated for storing the data, increasing the complexity of data management.

3) COMPUTATION

Most IoT devices have limited computation and energy resources, in which it is impossible to undertake on-site complex computational tasks. Generally speaking, IoT devices simply gather the data and transmit it to more powerful computing nodes, in which all the original data will be further processed and analyzed. Nonetheless, the computational capacity of individual edge nodes is limited, and thus the scalability of computational capacity for edge computing is a challenging problem. Still, IoT devices usually do not require much computational capacity, and the demands of IoT can be properly satisfied, especially for real-time services, by edge nodes. In addition, edge nodes mitigate the power consumption of the IoT devices through the offloading of computation tasks.

Based on the three categories above, we have constructed the problem space for Edge Computing-based IoT in Fig. 7.

In the following, we will discuss how Edge Computing-based IoT satisfies the requirements of transmission, storage,

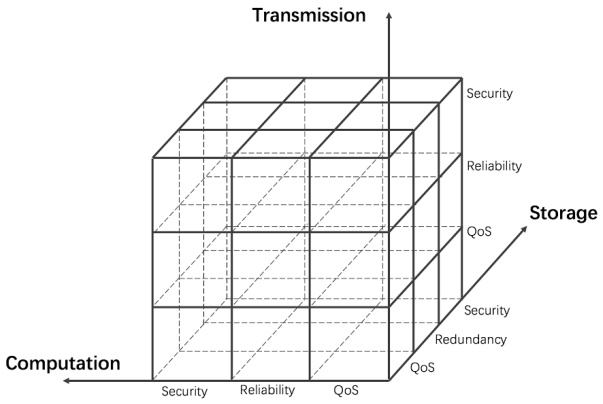


FIGURE 7. The problem space of Edge Computing-based IoT.

and computation, in detail. We will also provide some examples with which to analyze each characteristic.

IV. ADVANTAGES OF EDGE COMPUTING-BASED IoT

In this section, we assess the advantages of integrating IoT with edge computing.

A. TRANSMISSION

Network performance, which can be assessed by latency, bandwidth, and packet loss, among others, affects the transmission time. As discussed before, fast transmission time is the one of important benefits of edge computing, which can satisfy the QoS of time-sensitive applications, like the ''Live Video Analytics'' project from Microsoft [50]. The purpose of this project is build a real-time, low-cost system to analyze live videos, which are gathered from all the available cameras in a local open area. This system will work across a geo-distributed hierarchy of intelligent edges and large clouds [51]. One of the functions of this project is to predict vehicle traffic flow, which is obviously time-sensitive. The hierarchical architecture of edge computing guarantees a shorter transmission time than any other network [52].

Meanwhile, edge computing has also been developed to solve the bottleneck problem of network resources in IoT. By offloading the data computation and storage to end users, the response time and traffic flow will be significantly reduced. The hierarchical distributed edge nodes are able to satisfy the demands of time-sensitive applications such as ''Live Video Analytics'' [50], ''Human Action Classification'' [53], ''Motion Estimation'' [54], etc.

1) LATENCY/DELAY

Generally speaking, the latency of an application is the product of two components: computing latency and transmission latency. Computing latency indicates the time spent on data processing, which depends on the computing capacity of the system. It is clear that the sensors are often embedded devices with limited computing capacity, while the network servers will have a significant capacity to provide fast data processing. Nonetheless, the data transmission between the end devices and the cloud servers will cause a significant increase in the transmission latency.

Therefore, the challenge for edge computing is to determine ideal trade-off between computing latency and transmission latency, necessitating an optimal task offloading scheme to be developed to determine: whether a data processing task should be performed locally, be offloaded to the edge/cloudlet servers, or further offloaded to the remote cloud servers.

Quite recently, some mathematical methods have been designed to achieve this optimal resource allocation. For example, Liu *et al.* in [55] designed a delay-optimal computation task scheduling scheme. Via this scheme, a task can be decided to execute at the end device locally or be offloaded to the MEC server for cloud computing. The scheduling scheme considers a number of factors (the queuing state of the task buffer, the execution state of the local processing unit, etc.). Via the use of this scheduling scheme, the average delay for individual task and the average power consumption of the end devices can be reduced. Ketykó *et al.* in [56] proposed a model for multi-user computation offloading in 5G mobile edge computing. In this study, a multiple knapsack problem was formalized. With the solution to address the problem, the overall latency can be minimized. Liu *et al.* in [57] proposed a distributed computation offloading scheme for the multi-user computation offloading game problem in the mobile cloud computing environment. By solving this game problem, the total cost (i.e., energy consumption and time consumption) on mobile devices can be largely reduced.

In addition, the concept of opportunistic theory can be applied to solve the resource allocation challenge. There are various existing opportunistic schemes applied to different aspects of edge computing, and some of them show promising performance. For example, Tianze *et al.* in [58] proposed a consumption considering optimal scheme for task offloading in the mobile edge computing environment. With this scheme, the mobile devices can find a proper virtual machine to complete the task quickly, while saving energy. Rehman *et al.* in [59] proposed an opportunistic computation offloading scheme in a mobile edge cloud computing environment. Via analyzing the amount of unprocessed data, privacy configurations, contextual information, and other information, the proposed scheme was demonstrated to provide a suitable execution model for mobile devices. Via this scheme, the execution time and power consumption can be significantly reduced. Also, Gao in [60] proposed an analytical framework, which can exploit the potential of peer mobile devices at the tactical edge (opportunistically moving into communication range of each other). By considering the energy consumption and data transmission delay of computational task execution simultaneously, this framework is capable of improving the task completion ratio and completion time, as well as reducing the power consumption associated with task executions.

Obviously, computation offloading from the central cloud to the network edge can help reduce transmission delay via a proper offloading strategy. There are some efforts focused

on how to make optimal computing offloading decisions. For example, Wang *et al.* in [61] studied the partial computation offloading issue with DVS technology in the mobile edge computing environment. In their study, two optimization problems (energy consumption minimization (ECM) of smart mobile devices, and latency minimization (LM) of application execution) are formalized. With the use of the designed scheme, devices can achieve better performance with respect to energy consumption, latency, and admission probability.

In addition, Deng *et al.* in [62] proposed an adaptive sequential offloading game scheme for a multi-cell MEC scenario, and then designed a multi-user computation offloading algorithm. In their designed scheme, the mobile users make offloading decisions by considering the current interference as well as available computation resources. In this way, reduced latency and energy consumption can be realized by mobile users. Nam *et al.* in [63] proposed a clustered network service chaining scheme in the mobile edge computing environment. With the use of this scheme, the optimal number of clusters can be obtained so that the service time can be minimized. Fernando *et al.* in [64] proposed a work-sharing model for mobile edge-clouds to adapt the well-known work stealing mechanism known as Honeybee.

Despite the aforementioned schemes, there are other schemes based upon different goals, such as maximizing profit. For example, Sun and Ansari in [65] proposed a PRofIt Maximization Avatar pLacement (PRIMAL) scheme for mobile edge computing. With this scheme, the trade-off between the migration gain and the migration cost can be optimized. Lee and Flinn in [66] proposed a scheme, which selectively deploys redundancy to reduce the tail response time of vehicular applications. With passive measurement and historical data, both network latency and computing times for offloaded sensor processing can be estimated first, and then the cloud, roadside, or mobile phone platform with the fastest predicted response time will be selected. Rodrigues *et al.* in [67] proposed an analytic model for minimizing service delay in an edge cloud computing environment. Based on this model, the processing delay can be controlled by virtual machine migration and the transmission delay can be improved by adjusting transmission power. In this way, the lowest service delay can be achieved.

2) BANDWIDTH

As the IoT deploys a considerable number of sensors, the generated data is also extremely large. It is unacceptable for these data to be transmitted directly to cloud servers without any compression or processing. The massive data will consume immense network bandwidth and lead to a number of issues, such as transmission delay and packet loss. Thus, it is necessary for IoT gateways to perform data pre-processing and even aggreation before forwarding them to remote cloud servers. The challenge, then, is to control the traffic flow by optimally migrating data processing and aggregation tasks to reduce the bandwidth requirements of the end users while maintaining the quality of data.

There have been a number of research efforts devoted on this issue. For example, Abdelwahab *et al.* in [68] proposed an LTE-aware edge cloud architecture and an LTE-optimized memory replication protocol, called REPLISOM. The designed protocol can effectively schedule the memory replication operations. In this way, contentions among radio resources from devices accessing the resources simultaneously can be addressed. Sajjad *et al.* in [69] proposed a scheme to unify stream processing across the central and the near-the-edge data centers, which is called SpanEdge. With this scheme, the stream processing applications can be optimally deployed in a geo-distributed infrastructure so that bandwidth consumption and response latency can be significantly reduced.

In addition, Zhang *et al.* in [70] designed a mobile edge computing offloading framework in cloud-enabled vehicular networks. In this study, a contract-based computation resource allocation scheme is designed. With this scheme, the utility of MEC service providers can be maximized and the offloading requirements of the tasks can be satisfied, leading to the reduction of the latency and the transmission cost of the computation offloading. Nunna *et al.* in [71] proposed a real-time context-aware ad hoc collaboration system, which combines the novel communication architectures for 5G with the principles of mobile edge computing. Thus, it can be used in geographically bound low latency use cases. Papageorgiou *et al.* in [72] proposed a stream processing framework extension, which considers topology-external interactions (interactions with databases, users, critical actuators, and more). With this solution, the latency requirements violations can be eliminated and the cloud-to-edge bandwidth consumption can be reduced.

3) ENERGY

The end devices in the IoT may vary not only in network resources, but also in power resources and battery capacity. Thus, when an end device needs to perform data processing or data forwarding should be carefully considered with these factors in mind. It is important to maximize the lifetime of end devices, especially those with limited battery. To achieve this goal, edge computing can incorporate a flexible task offloading scheme which considers the power resources of each device.

A number of research efforts have been devoted on energy issue. For example, Gu *et al.* in [73] proposed the concept of fog computing-supported medical cyber-physical systems to host virtual medical device applications. With joint consideration for communication base station association, subcarrier allocation, computation base station association, virtual machine deployment, and task distribution, a low-complexity two-phase linear programming-based heuristic algorithm is proposed to solve the mixed-integer linear programming problem. With this scheme, total cost and better QoS can be realized for applications. Barcelo *et al.* in [74] proposed a comprehensive IoT-cloud service optimization framework. In this framework, the service distribution problem in the

investigated network is formalized as a min-cost mixed-cast flow problem. It is demonstrated that the smart IoT services can reduce power consumption by over 80 % after the proposed problem is resolved.

In addition, Zhang *et al.* in [75] proposed an energy-efficient computation offloading scheme, aiming to address the optimization problem. In this way, the energy consumption of the offloading system for MEC in 5G heterogeneous networks can be minimized. In this work, the energy cost of both task computing and file transmission is considered. Mao *et al.* in [76] proposed a Lyapunov optimization-based dynamic computation offloading (LODCO) scheme in a green MEC system, which consists of energy harvesting devices. With this low-complexity online algorithm, the execution cost and the reduction of computation failures is realized at the expense of only marginal execution delay degradation. Sardellitti *et al.* in [77] proposed a joint optimization scheme of the radio and computational resources for a multicell mobile-edge computing environment. With this scheme, the overall energy consumption users can be minimized under the latency constraints.

4) OVERHEAD

In network transmission, there exist header overhead and payload in each data packet. Due to the characteristics of data patterns in IoT, while most data packets are small, a massive number of IoT devices could introduce significant network overhead. Reducing the network overhead is another open challenge for edge computing. With the aid of edge/cloudlet servers, trivial packets can be aggregated and pre-processed in order to reduce the unnecessary overhead. Related to this issue, Plachy *et al.* in [78] proposed a cross-layer scheme, aiming to minimize overhead and improve transmission efficiency for 5G mobile networks.

B. STORAGE

Typically, cloud computing-based storage is centralized and implemented as complex, multi-layer systems, composed of groups of commodity servers and disk drives. It is built on top of the network, and is the convergence point of the network topology. Likewise, some edge nodes are responsible for servicing storage demands, but in contrast to the traditional cloud, edge computing-based storage is distributed at the edge of the network structure. It similarly combines clusters of disk drives, but also balances the storage demands to different edge nodes.

To satisfy QoS requirements, edge computing-based storage can leverage load balancing and failure recovery techniques to realize the requisite performance and availability. These load balancing techniques are capable of offloading the storage demands to different edge nodes, which mitigates the traffic in the network connection links. Furthermore, to distinguish the data failures (e.g., software, hardware, packet loss, noise, and power issues) in the massive data flow from multi-data sources, the failure recovery techniques are of key importance to edge computing storage.

1) STORAGE BALANCING

In IoT networks, devices usually have very limited storage space. All data that is collected or generated by the devices must be transmitted and stored in a storage server. Also, there are scores of IoT devices generating massive data simultaneously. If all the devices simultaneously store the data in cloud computing-based storage, the result will be significant obstruction in the network. For instance, the Microsoft ''Live Video Analytics'' project [50] generates massive data, which needs to be sent to storage within a very short time and needs to be incorporated into the analysis process in a timely manner. Based on these requirements, the sensors or cameras sending data to cloud computing-based storage will obviously not be satisfactory. Instead, based on the characteristics of edge computing storage, if the data is sent to the different edge storage nodes, long distance traffic in the network will be reduced.

To this end, storage balancing technologies are involved to realize edge computing-based storage for handling distributed IoT devices with different types of data streams, probabilities, and placements. There are a number of schemes related to storage balancing in [2] and [80]–[82]. For example, in [2], a resource allocation scheme and satisfaction function were proposed to handle the IoT storage issue. Here, the satisfaction function can be used to evaluate whether the allocated resources are sufficient to provide the requested service. Another scheme called the MMPacking balance scheme proposed in [80] can monitor different storage demand rates and use data stream replication to balance the traffic load and storage usage. The key feature of this scheme is the dropping of redundant data packets to save on storage space. Using storage balancing in edge computing-based storage can reduce the storage time by selecting the nearest edge storage nodes, or some storage processing rating and weighting schemes. Thus, with edge computing assistance, the ''Live Video Analytics'' [50] can upload data to the nearest edge storage nodes, satisfying the service requirements. Meanwhile, if a video packet is the same (e.g., the frame(s) are the same), the system will measure and drop some redundant packets to save storage space.

2) RECOVERY POLICY

As discussed above, the recovery policy is a key requirement in edge computing storage systems and reliability is clearly important in storing and retrieving accurate data representations. To increase the reliability, the system will check the availability of the storage nodes, duplicate the data, or use other nodes for redundancy.

a: AVAILABILITY

A storage service can become unavailable for a number of reasons. Typically, periodic pinging or heartbeat is conducted by monitoring systems to verify storage system health, and to identify the availability of edge nodes. Inevitably, storage services will at some point be unavailable. For example, a network device may be unavailable, the operating system on

edge storage node may crash or restart, the storage hardware may encounter an error, system automated repair process may remove or change the authority of the disks, or the entire system may shut down for maintenance. Based on empirical and statistical results, less than 10 % of failures last longer than 15 minutes [82]. In cloud computing-based systems, redundant storage servers are deployed to handle this problem. Nonetheless, in edge computing storage systems, the other available edge nodes will act as redundant storage. In IoT environments, massive numbers of devices constantly demand data storage. Thus, selection of the available storage service provider is important.

There are several available measurement schemes [82]–[84] proposed to handle this issue. All of them are able to select the available storage service providers. Furthermore, Ford *et al.* in [82] provide an algorithm to compute the mean time to failure (MTTF) and obtain the probability of the length of time that each edge node is available.

b: DATA REPLICATION

In IoT environments, the massive number of devices introduces constant demand for data storage. Obviously, the correctness of sensitive data is imperative, such as personal health data, energy consumption records, speed or traffic situations for smart vehicles, etc. Thus, the distributed storage systems must necessarily involve IoT environments for assistance to handle this massive demand and insure data accuracy.

Distributed storage systems can increase reliability and extend the MTTF by using replication [85]. In distributed storage systems, data is divided into many pieces, and each piece of data has fixed size and code blocks [86]. Also, the data pieces have fixed overlaps for each other. As a result, the data stored on each piece can be reconstructed from the other related pieces [82]. Edge computing-based storage is essentially a distributed storage system, and it is not only logically distributed, but physically distributed as well. Thus, with Edge computing-based storage assistance, sensitive IoT data can be replicated and the different pieces of data stored in different geological locations. This remarkably mitigates the risk of data loss.

C. COMPUTATION

In edge computing, each edge node has less computation power than what is available to cloud servers. Thus, the computation tasks need to be assigned to several edge nodes to meet the same demands. Recall that edge computing will satisfy the requirements of end users by offloading the computing and storage to the edge of the networks, and the task scheduling scheme becomes a key component for edge computing. In general, task scheduling schemes can be designed based on different objectives. In this section, we consider various methods to implement the task schedule in edge computing.

1) COMPUTATION OFFLOADING

To obtain greater efficiency in computation, edge computing must adjust the locations of different computation tasks.

a: LOCAL

In modern IoT systems, embedded chips have become cheaper and more widely adopted. Thus, the computing capacity of end devices has been significantly improved. Therefore, it is possible that the end users may perform some computing tasks in the Machine-to-Machine (M2M) network, which is formed by an array of IoT end devices. With a large number of the neighboring devices, the end users can obtain the shortest response time.

b: EDGE/CLOUDLET

Despite the M2M network of end devices providing some computing resources, M2M is not enough to satisfy all the resource requirements from all the end users. Thus, edge/cloudlet servers are required to provide the majority of network resources in the IoT. To adequately achieve this, the most critical issue is the task scheduling of the edge/cloudlet servers.

The objective of the task scheduling for edge/cloudlet servers is to find the optimal subset of servers under the given constraints to allocate. The optimal solution of this problem will obtain the minimum computing latency and transmission latency, minimum energy consumption on computing and communication, and the minimum bandwidth required by the IoT applications.

c: CLOUD

It is clear that some data processing or storage tasks require more resources than either M2M or Edge/Cloudlet can reasonably provide without taking up all of the available resources. In this case, the computation and storage must be accomplished in the traditional cloud servers. The cloud servers, having the largest computation capacity in the network, means that the tasks performed on the cloud servers will have the shortest computational latency. As a trade-off, the cloud servers also have the largest transmission latency, because of the long distance between the cloud servers and the end devices. Thus, there exists an important challenge of how to balance between the computational latency and the transmission latency.

2) PRICING POLICY

In the edge computing environment, the edge/cloudlet servers, or even other end users, can provide end users with the computation or communication resources requested for their computation tasks. Thus, resource allocation schemes can be derived through a proper pricing policy for the resources in the networks.

a: SINGLE SERVICE PROVIDER

Traditionally, the computation and communication resources in the edge/cloudlet servers are managed by a single service provider. That is to say, the service provider will

set the various prices for computation and communication resources of the edge/cloudlet servers deployed at different distances to the end devices. Then, the end users can minimize their financial cost by selecting the best available edge/cloudet servers and transferring the desired workload.

For example, Zhao *et al.* in [87] proposed a scheme to optimally allocate the edge computational resources based on the pricing policy in delay-aware mobile edge computing environment. With this scheme, the profit of the edge cloud can be maximized. Furthermore, this work demonstrates that the gain of the edge cloud can be impacted by the price of the remote cloud in some conditions. Kiani and Ansari in [88] proposed a hierarchical model in the form of field, shallow, and deep cloudlets. To realize the time-scale optimization for resource allocation, and address the convex optimization problem for bandwidth allocation, heuristic algorithms, as well as a centralized scheme, are studied.

b: MULTIPLE SERVICE PROVIDERS

Due to IoT connecting a diverse assortment of devices belonging to different parties, the computing or storage resources may not belong to a single service provider. This means that the users who require data processing tasks have to pay for the corresponding resources to different edge computing service providers. The proper pricing policy will encourage third parties to provide their computing or storage resources to IoT to ultimately gain the reward of service and payment from the end users. Furthermore, there will exist competition and cooperation among edge computing service providers. Thus, it is necessary for the emerging edge computing networks to make some efforts on the pricing policies between multiple service providers. In this direction, economics driven approaches such as auction [5], [89], [90] could be leveraged to manage resources.

3) PRIORITY

Priority is another important aspect of the computation task schedule in edge computing. With the concept of priority, the overall benefits of different IoT applications can be maximized. For example, real-time IoT applications, such as monitoring applications, will be assigned a higher priority, while other applications that consume more resources, such as multimedia peer-to-peer downloading, can be assigned a lower priority so that the total network performance can be improved. For example, Kamiyama *et al.* in [91] proposed a platform that can be used to measure the geographically deployed web objects from edge servers and reduce the latency to access web objects. You *et al.* in [92] proposed an offloading priority scheme, which considers both local computing energy and channel gains.

V. CHALLENGES OF EDGE COMPUTING-BASED IoT

As we discussed, there are numerous benefits for integrating edge computing to assist the IoT. In this section, we will discuss the challenges of Edge Computing-based IoT.

A. SYSTEM INTEGRATION

Supporting various kinds of IoT devices and different service demands in the edge computing environment is a significant challenge. Edge computing incorporates the combination of various platforms, network topologies, and servers. Essentially, it is a heterogeneous system. Thus, it will be difficult to program, and manage resources and data for diverse applications running on varying and heterogeneous platforms, in different locations.

From a programming perspective, in cloud computing, all applications and user programs are deployed and running on cloud servers. The cloud providers, such as Google and Amazon, have the responsibility to allocate those applications and programs in the suitable locations and hardware, and to make sure those applications and programs are running appropriately. Most users have no knowledge of how those applications run or allocate their resources and data. This is one of the benefits of cloud computing, because the cloud service is centralized and easy to manage. Also, developers need to use only one programming language to develop applications destined for a specific target platform, since the cloud application is only deployed on one particular cloud service provider.

In contrast, edge computing is quite different than cloud computing. Despite the benefits of the distributed topology, edge nodes are usually heterogeneous platforms. In this case, developers will face the serious difficulties in developing an application, which may be deployed and run in an edge computing platform. Some schemes have been devised to address the programmability challenges of edge computing, such as [36], [94], and [95], but none consider specific IoT purposes. In IoT, the first step is the discovery of edge nodes [93], meaning that, before the discovery process takes place, IoT devices do not know what kinds of platforms are deployed nearby. In addition, there is a huge number of server-side programs that need to be deployed on the edge nodes. Thus, how edge node providers deploy and manage those server-side programs is another challenging issue.

Regarding data management, various storage servers are running with various operating systems. This is a big challenge for file naming, resource allocation, file reliability management, etc. Because of the massive number of IoT devices generating and uploading data simultaneously, the naming of data resources becomes another big challenge. There are many traditional naming schemes, like DNS (Domain Name Service) and URI (Uniform Resource Identifier), and these satisfy cloud computing and most current networks. Nonetheless, these schemes are not fit for dynamic edge computing networks, and are not fit for IoT either. Furthermore, for multi-source and multi-task edge nodes, an IP-based naming scheme is not applicable, as IP-based naming schemes may be too costly for the edge nodes in multi-source and multi-task environments.

Several new naming schemes have been proposed, including Named Data Networking (NDN) [94] and MobilityFirst [95], which are designed for edge computing.

For example, the NDN naming scheme [94] provides a hierarchically structured name for the distributed network, and is friendly for edge node owners to manage. Nonetheless, it requires the addition of a proxy server to the network in order to integrate different kinds of communication protocols. Moreover, the NDN naming scheme needs source hardware information, raising the potential for information leakage. In the MobilityFirst naming scheme [95], the name is separated from the IP and MAC addresses to provide better mobility support. The problem of the MobilityFirst scheme is that it requires globally unique identification (GUID), which is not human friendly.

B. RESOURCE MANAGEMENT

The integration of IoT and edge computing necessitates complete and thorough understanding and optimization of resource management. IoT devices, often computation and resource deficient, will be drastically affected by network congestion and latency, utilizing more power to retransmit data in congested settings. Edge computing, as the nearest computing and storage resource, can provide an outlet to reduce latency of devices, and the decentralized resources will play an important role in motivating and sharing these assets.

The management of these resources can be conducted through a variety of means, so long as it is itself computationally cheap. Nonetheless, the significant heterogeneity of service providers, devices, and applications adds substantial complexity, and these interactions should not be overlooked. Specifically, the motivating factors in Edge/IoT resource management are concurrent with those of smart systems. In a system of multiple resource providers, and massively diverse applications and user needs, how to allocate, share, and price the direct service of these systems can be satisfied by maximizing/optimizing global welfare or some other metric, through competitive bidding, or other strategies [96].

1) AUCTION-BASED

Various economic-driven schemes can be used to manage network resources. For instance, auction schemes have been widely applied to many areas of computer science research, including mobile and cloud computing [97]–[99], and smart systems [5], [90], [100]–[102], as well as across various research spectrums. In application for edge resource management, auction schemes shall provide secure and privacy-preserving bidding on services by need and bid value, and shall satisfy the needs of users. In the context of edge computing and IoT, auction schemes shall be envisioned to hide users from service providers, and allocate service in a fair and unbiased way. For service providers, there is an incentive to maximize the use of their capacity to achieve the highest profit. This concept assumes a scenario where data center cloud and edge computing providers are different organizations, and where various edge nodes are hosted by different organizations as well. Assuming vast interconnected networks, subnetworks, ad hoc networks, etc., the targets, paths, and destinations of gargantuan network data needs to be handled efficiently, and must be appropriately distributed to satisfy QoS.

2) OPTIMIZATION

As formulated, the application of optimization could likewise handle resource allocation and division in edge computing. Like auction schemes, optimization can present beneficial properties to system participants, optimizing welfare or profit. Though organizations may intend for local edge systems to rely on subscription or patron services, as edge infrastructures provide a middle layer between users and cloud services, this notion may not be feasible. As applied to cloud and edge computing [77], [103], and various other areas of resource management, optimization has shown increasing promise, and is a contender that complements auction schemes.

C. SECURITY AND PRIVACY

As moving targets that span all domains, security and privacy are critical issues that demand careful consideration. In the adoption of Edge Computing-based IoT, these are, in fact, the most important issues. Edge computing is centered around the complex interweaving of multiple and varied technologies (peer-to-peer systems, wireless networks, virtualization, etc.), and requires the adoption of a comprehensive integrated system to safeguard and manage each technology platform, and the system as a whole.

Despite this lofty goal, the culmination of edge computing will raise some new and unforeseen security issues. Unique and unstudied scenarios, such as the interplay of heterogeneous edge nodes, and the migration of services across global and local scales, create the potential for original channels of malicious behavior. Furthermore, the inherent properties of edge computing may very well dictate what security and privacy measures are viable, and which cannot be realized. Similar to cloud computing, there are numerous distinct security issues and challenges in edge computing environments.

The distributed structure has numerous benefits for IoT. Nonetheless, the security and privacy of distributed structures is a significant challenge. With respect to privacy, edge computing could provide an effective computing platform to future IoT. As edge computing processes data at the edge, the privacy-sensitive information associated with end users could be exploited. Notice that sensing data from IoT systems is stored at edge nodes, which can be more vulnerable than cloud servers [2], [49]. Thus, privacy protection needs to be considered in edge computing and the effective privacy-preserving mechanisms, such as local differential privacy [104] and differential privacy with high utility [49], [105] need to be designed to protect the privacy of users in the edge computing-based IoT environment.

With respect to security, one of the typical security problems of edge computing is to authenticate gateways in different levels. An example is smart meters in residential homes, where each of the smart meters has its own IP address.

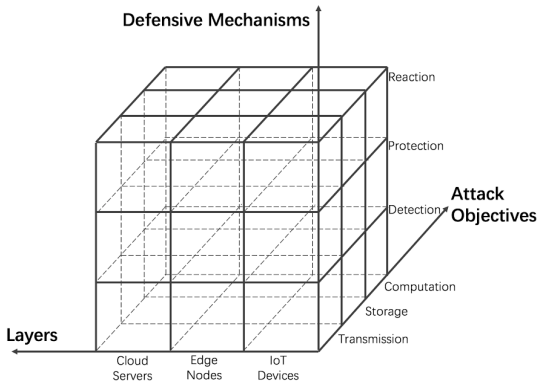


FIGURE 8. Problem space of Edge Computing-based IoT security.

In this context, an adversary could report false data, modify the other user data, tamper with their own smart meter, or spoof IP addresses, and further disrupt the effectiveness of energy management in IoT systems (smart grid, etc.) [2], [6], [7], [106]–[109]. From a management perspective, different edge nodes are managed by different owners, making it difficult to deploy an equivalent security strategy throughout. Based on the above concerns, we identify the problem space, shown in Fig. 8, and its dependence on the IoT structures in Fig. 6.

1) TRANSMISSION

Ensuring security in the data transmission process is one of the key challenges for Edge Computing-based IoT. During message transmission between end users and servers, some attacks (jamming attacks, sniffer attacks, worm propagation, resource-depletion denial-of-service, and others [110], [111]) could be launched to disable the links by congesting the network, or could monitor network data flow. Normally, in a traditional network, the configurations input by a network administrator need to be trustworthy and validated [112].

Nonetheless, Edge Computing-based IoT is deployed in the edge of the network structure, and various types of networks, such as Mobile Wireless Networks [113], Ultra-Dense Networks [33], [114], and Wi-Fi, will obviously be challenging to manage. Thus, in managing edge networks, significant management traffic will be necessarily generated, making it a further challenge to isolate regular data traffic. In this case, adversaries would be able to control the network easily [111]. To mitigate this problem, Software-Defined Networking (SDN) [40], [115] must be introduced. SDN can mitigate the aforementioned security risks from the following perspectives:

I. *Detection:* Deploying a Network Monitoring and Intrusion Detection System (IDS) provides the ability to monitor data traffic and scan data packets for applications to detect the malicious code. In SDN, it is easy to deploy an IDS system and improve the manageability of traffic flow in Edge Computing-based IoT.

II. *Protection:* To protect data in the transmission process, traffic isolation and prioritization is the most efficient method. Here, SDN is able to easily use VLAN ID to isolate different types of traffic into VLAN groups, and can be used to further segregate malicious traffic. Thus, traffic isolation and prioritization is usually used to prevent some types of attacks, including those that aim to congest the network or dominate shared resources and hardwares.

II. *Reactions:* Following from a long history of conventional countermeasures against network threats in cyber-physical systems [106], [107], [109], there are ongoing efforts to assess and prevent cyber-attacks in edge computing environments [116].

2) STORAGE

In Edge Computing-based IoT, massive data is generated by the innumerable sensors and devices, and all the storage is provided by different third party suppliers. User data is outsourced to those storage suppliers, whose storage devices are deployed in the edge of the network and located at many different physical addresses. There are numerous reasons why this clearly increases the risk of attacks. First, it is difficult to guarantee data integrity, since the data is separated into many parts and is stored across different storage locations, making it easy to lose data packets or store incorrect data. Second, the uploaded data in storage may be modified or abused by unauthorized users or adversaries, which will lead data leakage and other privacy issues.

To address these problems, various techniques can be used, such as homomorphic encryption [117], [118] so that integrity, confidentiality, and verifiability for edge storage systems can be realized. Furthermore, the technologies increase the security of users such that they are able to store their data to any untrusted servers [112]. For example, in [119], the authors proposed a privacy-preserving public auditing for protecting data stored in the cloud via involving a third-party auditor (TPA). The same technique can be used in edge storage as well. In [120], the authors proposed effective protocols to verify the file search results from the cloud. In this study, two protocols are designed. One is to enable verification of the file search result in the scenarios where users have the same security privilege. The other is to consider the scenario, in which users access files with different security privileges. In the edge computing environment, a user trusted TPA can improve the security of the storage system and reduce the management overhead. As for the security of TPA itself, it uses homomorphic encryption and the random mask technique to protect itself.

Another challenge for storage is ensuring data reliability. Traditional methods to detect and repair corrupted data in storage systems use erasure codes or network coding. Nonetheless, these require laborious program development and a great deal of storage overhead. For example, Anglano *et al.* [121] proposed a secure coding-based storage, which introduces Luby transform (LT) code into programs, reducing the storage space overhead and communication time, and increasing the data search speed.

From a management perspective, network Resource Access Control (RAC) is also an important method to protect the data in edge storage, and is the most efficient approach for data security. As a means for securing data resources, a secure network resource access system utilizes terminals to access network resources located behind enterprise firewalls. Specifically, a proxy server is located outside the firewall, and receives application data from a terminal, while a polling server is located inside the firewall. The polling server has several functions, these being the initialization of data transmission from proxy to polling server, the receipt of application and associated network resource data, and the direction of the application data to corresponding network resources based on the resource data.

3) COMPUTATION

Another important security challenge in Edge Computing-based IoT is to maintain security and privacy in uploading computational tasks to edge computation nodes.

To ensure computation security, Verifiable Computing [122] was introduced for Edge Computing-based IoT. Generally speaking, Verifiable Computing enables an untrusted computation node to offload the computation tasks. Meanwhile, this computational node maintains the verifiable results, and uses these results to compare them with the results calculated by some other trusted computation nodes as proof that the computing has been correctly completed. In the case of Edge Computing-based IoT, each IoT device should be able to verify the correctness of the results, which are computed by edge nodes. A system was built in [123], named Pinocchio, which allows the clients to verify computation results based solely on cryptographic assumptions. Using Pinocchio, a public evaluation key is created by clients, which describes the computation task, and servers will compare the value of the key and the computation result to prove correctness. This is similar to the verifiable computing protocol, which was proposed in [122]. This protocol allows the computation nodes to return a computationally-sound result, and the clients can check the result to verify the computational soundness of the computed task.

Due to the decentralized management of edge networks, which cannot supply adequate security and management features, it is a complex and difficult problem to manage and secure networks with such a large number of connected devices. For example, Hafeez *et al.* in [124] proposed a service-based solution to safeguard the network edge, called Securebox. By leveraging the security and network management features provided by the proposed system, the designed system can enable security services by detecting and reacting to malicious activities in the system.

Because computation tasks are migrated from the cloud to edge nodes, it is necessary to establish trust between edge servers and the end devices, especially without trusted third party security. Related to this issue, Clemens *et al.* in [125] proposed some solutions, which can extend integrity measurement and attestation systems to incorporate integrity evidence from edge devices under their restricted capabilities and constrained operating environments. Echeverría *et al.* in [126] proposed a trusted identity solution in disconnected environments based on identity-based cryptography and secure key exchange in the field, which operates without a trusted third party. With proper application-, OS-, network- and site-level controls, this solution can be resilient to most of the threats present in disconnected environments.

For other security issues, like software verification or malicious intrusion detection, there are several preliminary works designed to address these issues. For instance, Tan *et al.* in [127] proposed a bottom-up and foundational approach for verifying the security of the software stack in an IoT system, which is called BUFS. With this approach, the software of the end devices can be verified from the bottom-up. Mtibaa *et al.* in [128] proposed a defense technique for malicious device-to-device (D2D) communication called HoneyBot. With this method, the HoneyBot nodes are capable of identifying and isolating D2D insider attacks. Furthermore, it has been proven that the number and placement of HoneyBot nodes in the network can impact speed and accuracy measurements significantly.

In addition, to further improve the efficiency of threat analysis and detection and to reduce the performance impact of threat analysis and detection in edge computing-based IoT systems, edge computing infrastructures shall be leveraged to assist in threat analysis and detection [25], [31], [32]. For instance, the use of edge computing to improve the performance of detection of threats against IoT and smart systems must be studied (e.g. efficiently learning the profile of threats in parallel to speed up threat detection).

Designing an integrated defense system against cyber threats on edge servers, should span three generic and defense strategies: proactive defense, reactive defense, and predictive defense. In particular, for proactive defense, it is critical to develop techniques at both the data-level and the system-level. The detection should consider edge resource utilization and allocation, and require low overhead (time, code, memory, compute, I/O, storage, architecture heterogeneity, and others). At the data-level, mechanisms (data self-correction to detect and recover compromised computational data, and others) should be considered. At the system level, monitoring and detection tools in the edge system need to be to designed and integrated into the edge computing infrastructure to effectively and proactively discover exploitable vulnerabilities to make the system secure. For reactive detection techniques, effective techniques should be designed at both the data and the system levels. In addition, at the data level, techniques such as low-cost data attestation mechanisms [129], [130] shall be considered, which can confirm the integrity of data processing results and identify malicious nodes based on inconsistency of results. At the system level, effective anomaly detection techniques based on machine learning (such as deep learning) principles must be considered. Further, for predictive defense mechanisms, machine

learning-based techniques need to be designed to not only foresee impending system anomalies, but also to predict behaviours of new threats in the system.

D. ADVANCED COMMUNICATION

As a shift in the current paradigm of remote computation and storage, edge computing is removing the barriers to rapid, low-latency, high-computation applications. Likewise, the technologies of future 5G cellular networks, including Ultra-Dense Networks (UDNs), massive MIMO (Multiple-Input and Multiple-Output), and millimeter-wave, are improving daily, advancing to reduce latency, increase throughput, and support massively interconnected groups in dense networks [33], [34], [115]. With these advances in communication technologies, edge computing will further progress as integration of these technologies becomes inevitable.

5G Communication: 5G is known as the next generation communication technology. Its goal is provide ubiquitous network connectivity and access to information needed for users [33]–[35], [114]. Thus, the concepts of 5G, IoT, and edge computing can be integrated together to achieve flexible and efficient communication. In addition, 5G technology can help improve the efficiency of many IoT applications.

For example, Cau *et al.* in [131] proposed schemes for effective subscriber state management in 5G scenarios. Hung *et al.* in [132] conducted a comprehensive survey of fog network and cloud radio access network structures and discuss the need to integrate both for 5G. Chagh *et al.* in [133] proposed a VoWiFi solution with edge computing technology, which can help address its main drawback (i.e., the lack of user location). With the proposed scheme, location information related to VoWiFi users can be retrieved. Zeydan *et al.* in [134] proposed a big-data-enabled architecture for proactive content caching in 5G wireless networks. Ardi and Joshi [135] studied a cloud-based framework for accessing private medical records in the context of 5G networks. With this framework, the private records can be protected and access authorization can be enhanced.

E. SMART SYSTEM SUPPORT

Smart systems necessarily interweave network communication technologies with sensors and actuators to realize system awareness and subsequent remote control, and can be seen as an extension of IoT technologies [2], [3]. The integration of sensing devices provides untold opportunities for data collection, physical system management, and resource allocation and optimization. Key areas of smart systems include smart grid, smart city, smart transportation, smart health, and others. As more systems become smart, edge computing can provide the lowest latency computing and storage for computationally deficient devices. Similarly, data analysis at the edge can facilitate the highest resiliency to compromised systems.

1) SMART GRID

The Smart Grid is considered to be the next generation in power grid technology and implementation. To achieve the advantages afforded by the smart grid (e.g., safety, secure, self-healing), a large number of smart meters, sensors, and actuators are needed to collect and exchange measurement data in the smart grid [7], [107], [136]–[138]. Thus, edge computing has potential satisfy the requirements of smart grid deployment. Nonetheless, how to involve multiple edge servers to process the data streams from meters and sensors spanning large and varying areas and provide optimal and timely energy management decisions remain open issues.

Related to this direction, there are some existing research efforts. For example, Emfinger *et al.* in [136] proposed the RIAPS (Resilient Information Architecture Platform for the Smart Grid). With this architecture, some challenges can be solved, such as resource and network uncertainty. Kumar *et al.* in [137] leveraged the mobile edge computing paradigm and proposed a smart grid data management scheme based on a vehicular delay-tolerant network. In this study, the optimal charging for plug-in hybrid electric vehicles is designed.

2) SMART CITY

To effectively and efficiently use public resources in cities and increase the standard of living for the citizens, the concept of the Smart City has been proposed and realized [8], [9], [139], [140]. One of the most critical challenges is the non-interoperability of the heterogeneous technologies in cities. For example, Zanella *et al.* in [139] surveyed the relevant technologies, protocols, and architecture for urban IoT. Sapienza *et al.* in [140] investigated a scenario, which can exploit the MECs to recognize abnormal or critical events (terrorist threats, disasters, etc.). Although a large number of connected devices can affect network performance, they can be helpful in detecting the occurrence of anomalous events through user-generated content and appropriate algorithms.

Specifically, there are already some preliminary research studies in this direction, such as real-time video analysis with edge computing. For example, Zhang *et al.* in [141] proposed an Edge Video Analysis for Public Safety framework, named EVAPS. With this framework, the computing workload for real-time video analysis in both edge nodes and the cloud can be distributed in an optimized way. Then, unnecessary data transmissions can be eliminated and the energy of edge devices can be conserved. Chen *et al.* in [142] proposed a fog computing-based smart urban surveillance solution. With a case study of traffic monitoring, the proposed system can track speeding vehicles and obtain vehicle speed information in real-time.

3) SMART TRANSPORTATION

To achieve safe and effective autonomous driving, a cloud-based vehicle control system is needed, because it can collect information from the sensors via a vehicle-to-vehicle network [5], [106], [143], [144]. Thus, it can control and coordinate a large number of vehicles. It is obvious that a real-time management of vehicles necessitates strict

requirements, such as short latency, which can be provided by edge computing.

Related to this area, there are some existing research efforts. For example, Sasaki *et al.* in [145] proposed an infrastructure-based vehicle control system to support safe driving. In their designed system, states between edge and cloud servers are considered to enable resource sharing. With this proposed system, the latency can be significantly reduced and instability of the cloud control is mitigated. Lin *et al.* in [5] proposed a dynamic decision scheme for real-time route guidance by mitigating road congestion and improving transportation efficiency. It is worth noting that the real-time traffic information collected through vehicular networks can be processed via edge computing infrastructure, which can be further provided to drivers in real time.

VI. FINAL REMARKS

With the development of IoT, edge computing is becoming an emerging solution to the difficult and complex challenges of managing millions of sensors/devices, and the corresponding resources that they require. Compared with the cloud computing paradigm, edge computing will migrate data computation and storage to the ''edge'' of the network, nearby the end users. Thus, edge computing can reduce the traffic flows to diminish the bandwidth requirements in IoT. Furthermore, edge computing can reduce the transmission latency between the edge/cloudlet servers and the end users, resulting in shorter response time for the real-time IoT applications compared with the traditional cloud services. In addition, by reducing the transmission cost of the workload and migrating the computational and communication overhead from nodes with limited battery resources to nodes with significant power resources, the lifetime of nodes with limited battery can be extended, along with the lifetime of the entire IoT system. To summarize our work, we have investigated the architecture of edge computing for IoT, the performance objectives, task offloading schemes, and security and privacy threats and corresponding countermeasures of edge computing, and have highlighted typical IoT applications as examples.

REFERENCES

[1] D. Linthicum, ''Responsive data architecture for the Internet of Things,'' *Computer*, vol. 49, no. 10, pp. 72–75, 2016.

[2] J. Lin, W. Yu, N. Zhang, X. Yang, H. Zhang, and W. Zhao, ''A survey on Internet of Things: Architecture, enabling technologies, security and privacy, and applications,'' *IEEE Internet Things J.*, vol. 4, no. 5, pp. 1125–1142, Oct. 2017.

[3] J. A. Stankovic, ''Research directions for the Internet of Things,'' *IEEE Internet Things J.*, vol. 1, no. 1, pp. 3–9, Feb. 2014.

[4] J. Wu and W. Zhao, ''Design and realization of WInternet: From net of things to Internet of Things,'' *ACM Trans. Cyber-Phys. Syst.*, vol. 1, no. 1, pp. 2:1–2:12, Nov. 2016. [Online]. Available: http://doi.acm.org/10.1145/2872332

[5] J. Lin, W. Yu, X. Yang, Q. Yang, X. Fu, and W. Zhao, ''A real-time en-route route guidance decision scheme for transportation-based cyberphysical systems,'' *IEEE Trans. Veh. Technol.*, vol. 66, no. 3, pp. 2551–2566, Mar. 2017.

[6] Y. Yan, Y. Qian, H. Sharif, and D. Tipper, ''A survey on cyber security for smart grid communications,'' *IEEE Commun. Surveys Tuts.*, vol. 14, no. 4, pp. 998–1010, Oct. 2012.

[7] J. Lin, W. Yu, and X. Yang, ''Towards multistep electricity prices in smart grid electricity markets,'' *IEEE Trans. Parallel Distrib. Syst.*, vol. 27, no. 1, pp. 286–302, Jan. 2016.

[8] N. Mohamed, J. Al-Jaroodi, I. Jawhar, S. Lazarova-Molnar, and S. Mahmoud, ''SmartCityWare: A service-oriented middleware for cloud and fog enabled smart city services,'' *IEEE Access*, vol. 5, pp. 17576–17588, 2017.

[9] S. Mallapuram, N. Ngwum, F. Yuan, C. Lu, and W. Yu, ''Smart city: The state of the art, datasets, and evaluation platforms,'' in *Proc. IEEE/ACIS 16th Int. Conf. Comput. Inf. Sci. (ICIS)*, May 2017, pp. 447–452.

[10] M. D. Cia *et al.*, ''Using smart city data in 5G self-organizing networks,'' *IEEE Internet Things J.*, to be published.

[11] P. Corcoran and S. K. Datta, ''Mobile-edge computing and the Internet of Things for consumers: Extending cloud computing and services to the edge of the network,'' *IEEE Consum. Electron. Mag.*, vol. 5, no. 4, pp. 73–74, Oct. 2016.

[12] C. Vallati, A. Virdis, E. Mingozzi, and G. Stea, ''Mobile-edge computing come home connecting things in future smart homes using LTE device-to-device communications,'' *IEEE Consum. Electron. Mag.*, vol. 5, no. 4, pp. 77–83, Oct. 2016.

[13] A. V. Dastjerdi and R. Buyya, ''Fog computing: Helping the Internet of Things realize its potential,'' *Computer*, vol. 49, no. 8, pp. 112–116, 2016.

[14] D. Georgakopoulos, P. P. Jayaraman, M. Fazia, M. Villari, and R. Ranjan, ''Internet of Things and edge cloud computing roadmap for manufacturing,'' *IEEE Cloud Comput.*, vol. 3, no. 4, pp. 66–73, Jul./Aug. 2016.

[15] M. Jutila, ''An adaptive edge router enabling Internet of Things,'' *IEEE Internet Things J.*, vol. 3, no. 6, pp. 1061–1069, Dec. 2016.

[16] D. Sabella, A. Vaillant, P. Kuure, U. Rauschenbach, and F. Giust, ''Mobile-edge computing architecture: The role of MEC in the Internet of Things,'' *IEEE Consum. Electron. Mag.*, vol. 5, no. 4, pp. 84–91, Oct. 2016.

[17] 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.

[18] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, ''Edge computing: Vision and challenges,'' *IEEE Internet Things J.*, vol. 3, no. 5, pp. 637–646, Oct. 2016.

[19] B. Frankston, ''Mobile-edge computing versus the Internet?: Looking beyond the literal meaning of MEC,'' *IEEE Consum. Electron. Mag.*, vol. 5, no. 4, pp. 75–76, Oct. 2016.

[20] P. G. Lopez *et al.*, ''Edge-centric computing: Vision and challenges,'' *ACM SIGCOMM Comput. Commun. Rev.*, vol. 45, no. 5, pp. 37–42, 2015.

[21] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief. (Jan. 2017). ''A survey on mobile edge computing: The communication perspective.'' [Online]. Available: https://arxiv.org/abs/1701.01090

[22] W. Shi and S. Dustdar, ''The promise of edge computing,'' *Computer*, vol. 49, no. 5, pp. 78–81, 2016.

[23] H. Li, G. Shou, Y. Hu, and Z. Guo, ''Mobile edge computing: Progress and challenges,'' in *Proc. 4th IEEE Int. Conf. Mobile Cloud Comput., Services, Eng. (MobileCloud)*, Mar. 2016, pp. 83–84.

[24] 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.

[25] W. G. Hatcher, J. Booz, J. McGiff, C. Lu, and W. Yu, ''Edge computing based machine learning mobile malware detection,'' in *Proc. Nat. Cyber Summit (NCS)*, Jun. 2017.

[26] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, ''Internet of Things (IoT): A vision, architectural elements, and future directions,'' *Future Generat. Comput. Syst.*, vol. 29, no. 7, pp. 1645–1660, 2013.

[27] The national intelligence council sponsor workshop. (2008). *Intelligence, S. C. B., 2008. Disruptive Civil Technologies. Six Technologies With Potential Impacts on US Interests out to 2025*. [Online]. Available: https://fas.org/irp/nic/disruptive.pdf

[28] K. Rose, S. Eldridge, and L. Chapin, ''The Internet of Things: An overview,'' in *Proc. Internet Soc. (ISOC)*, 2015, pp. 1–53.

[29] F. Wortmann and K. Flüchter, ''Internet of Things,'' *Bus. Inf. Syst. Eng.*, vol. 57, no. 3, pp. 221–224, 2015.

[30] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, ''Internet of Things: A survey on enabling technologies, protocols, and applications,'' *IEEE Commun. Surveys Tuts.*, vol. 17, no. 4, pp. 2347–2376, 4th Quart., 2015.

[31] W. Yu, G. Xu, Z. Chen, and P. Moulema, ''A cloud computing based architecture for cyber security situation awareness,'' in *Proc. IEEE Conf. Commun. Netw. Secur. (CNS)*, Oct. 2013, pp. 488–492.

[32] Z. Chen *et al.*, "A cloud computing based network monitoring and threat detection system for critical infrastructures," *Big Data Res.*, vol. 3, pp. 10–23, Apr. 2016.
[33] W. Yu, H. Xu, H. Zhang, D. Griffith, and N. Golmie, "Ultra-dense networks: Survey of state of the art and future directions," in *Proc. 25th Int. Conf. Comput. Commun. Netw. (ICCCN)*, Aug. 2016, pp. 1–10.
[34] M. Agiwal, A. Roy, and N. Saxena, "Next generation 5G wireless networks: A comprehensive survey," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 3, pp. 1617–1655, 3rd Quart., 2016.
[35] P. Demestichas *et al.*, "5G on the horizon: Key challenges for the radio-access network," *IEEE Veh. Technol. Mag.*, vol. 8, no. 3, pp. 47–53, Sep. 2013.
[36] A. Ahmed and E. Ahmed, "A survey on mobile edge computing," in *Proc. 10th Int. Conf. Intell. Syst. Control (ISCO)*, Jan. 2016, pp. 1–8.
[37] Y. Jararweh, A. Doulat, O. AlQudah, E. Ahmed, M. Al-Ayyoub, and E. Benkhelifa, "The future of mobile cloud computing: Integrating cloudlets and mobile edge computing," in *Proc. 23rd Int. Conf. Telecommun. (ICT)*, May 2016, pp. 1–5.
[38] L. Tong, Y. Li, and W. Gao, "A hierarchical edge cloud architecture for mobile computing," in *Proc. 35th Annu. IEEE Int. Conf. Comput. Commun. (INFOCOM)*, Apr. 2016, pp. 1–9.
[39] G. Wang, Y. Zhao, J. Huang, and W. Wang, "The controller placement problem in software defined networking: A survey," *IEEE Netw.*, vol. 31, no. 5, pp. 21–27, Sep. 2017.
[40] D. Zhu, X. Yang, P. Zhao, and W. Yu, "Towards effective intra-flow network coding in software defined wireless mesh networks," in *Proc. 24th Int. Conf. Comput. Commun. Netw. (ICCCN)*, Aug. 2015, pp. 1–8.
[41] Y. Jararweh, A. Doulat, A. Darabseh, M. Alsmirat, M. Al-Ayyoub, and E. Benkhelifa, "SDMEC: Software defined system for mobile edge computing," in *Proc. IEEE Int. Conf. Cloud Eng. Workshop (IC2EW)*, Apr. 2016, pp. 88–93.
[42] P. Du and A. Nakao, "Application specific mobile edge computing through network softwarization," in *Proc. 5th IEEE Int. Conf. Cloud Netw. (Cloudnet)*, Oct. 2016, pp. 130–135.
[43] A. Manzalini and N. Crespi, "An edge operating system enabling anything-as—A-service," *IEEE Commun. Mag.*, vol. 54, no. 3, pp. 62–67, Mar. 2016.
[44] O. Salman, I. Elhajj, A. Kayssi, and A. Chehab, "Edge computing enabling the Internet of Things," in *Proc. IEEE 2nd World Forum Internet Things (WF-IoT)*, Dec. 2015, pp. 603–608.
[45] T. Lin, B. Park, H. Bannazadeh, and A. Leon-Garcia, "Demo abstract: End-to-end orchestration across SDI smart edges," in *Proc. IEEE/ACM Symp. Edge Comput. (SEC)*, Oct. 2016, pp. 127–128.
[46] 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* (Studies in Computational Intelligence), vol. 546. Cham, Switzerland: Springer, 2014, pp. 169–186.
[47] H. Jiang, F. Shen, S. Chen, K.-C. Li, and Y.-S. Jeong, "A secure and scalable storage system for aggregate data in IoT," *Future Generat. Comput. Syst.*, vol. 49, pp. 133–141, Aug. 2015.
[48] M. M. Hossain, M. Fotouhi, and R. Hasan, "Towards an analysis of security issues, challenges, and open problems in the Internet of Things," in *Proc. IEEE World Congr. Serv. (SERVICES)*, Jun. 2015, pp. 21–28.
[49] X. Yang, T. Wang, X. Ren, and W. Yu, "Survey on improving data utility in differentially private sequential data publishing," *IEEE Trans. Big Data*, to be published.
[50] G. Ananthanarayanan *et al.*, "Real-time video analytics: The killer app for edge computing," *Computer*, vol. 50, no. 10, pp. 58–67, 2017.
[51] G. Ananthanarayanan, V. Bahl, and P. Bodík. (2017). *Microsoft Live Video Analytics*. [Online]. Available: https://www.microsoft.com/en-us/research/project/live-video-analytics/
[52] J. R. Bergen, P. Anandan, K. J. Hanna, and R. Hingorani, "Hierarchical model-based motion estimation," in *Proc. Eur. Conf. Comput. Vis.*, 1992, pp. 237–252.
[53] J. C. Niebles and L. Fei-Fei, "A hierarchical model of shape and appearance for human action classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2007, pp. 1–8.
[54] G. Botella and C. García, "Real-time motion estimation for image and video processing applications," *J. Real-Time Image Process.*, vol. 11, no. 4, pp. 625–631, Apr. 2016. [Online]. Available: http://dx.doi.org/10.1007/s11554-014-0478-y
[55] J. Liu, Y. Mao, J. Zhang, and K. B. Letaief, "Delay-optimal computation task scheduling for mobile-edge computing systems," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jun. 2016, pp. 1451–1455.
[56] I. Ketykó, L. Kecskés, C. Nemes, and L. Farkas, "Multi-user computation offloading as multiple knapsack problem for 5G mobile edge computing," in *Proc. Eur. Conf. Netw. Commun. (EuCNC)*, 2016, pp. 225–229.
[57] Y. Liu, S. Wang, and F. Yang, "Poster abstract: A multi-user computation offloading algorithm based on game theory in mobile cloud computing," in *Proc. IEEE/ACM Symp. Edge Comput. (SEC)*, Oct. 2016, pp. 93–94.
[58] L. Tianze, W. Muqing, and Z. Min, "Consumption considered optimal scheme for task offloading in mobile edge computing," in *Proc. 23rd Int. Conf. Telecommun. (ICT)*, 2016, pp. 1–6.
[59] M. H. ur Rehman, C. Sun, T. Y. Wah, A. Iqbal, and P. P. Jayaraman, "Opportunistic computation offloading in mobile edge cloud computing environments," in *Proc. 17th IEEE Int. Conf. Mobile Data Manage. (MDM)*, vol. 1. Jun. 2016, pp. 208–213.
[60] W. Gao, "Opportunistic peer-to-peer mobile cloud computing at the tactical edge," in *Proc. IEEE Military Commun. Conf. (MILCOM)*, Oct. 2014, pp. 1614–1620.
[61] Y. Wang, M. Sheng, X. Wang, L. Wang, and J. Li, "Mobile-edge computing: Partial computation offloading using dynamic voltage scaling," *IEEE Trans. Commun.*, vol. 64, no. 10, pp. 4268–4282, Oct. 2016.
[62] M. Deng, H. Tian, and X. Lyu, "Adaptive sequential offloading game for multi-cell mobile edge computing," in *Proc. 23rd Int. Conf. Telecommun. (ICT)*, May 2016, pp. 1–5.
[63] Y. Nam, S. Song, and J.-M. Chung, "Clustered NFV service chaining optimization in mobile edge clouds," *IEEE Commun. Lett.*, vol. 21, no. 2, pp. 350–353, Feb. 2017.
[64] N. Fernando, S. W. Loke, and W. Rahayu, "Computing with nearby mobile devices: A work sharing algorithm for mobile edge-clouds," *IEEE Trans. Cloud Comput.*, to be published.
[65] X. Sun and N. Ansari, "PRIMAL: PRofit maximization avatar placement for mobile edge computing," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2016, pp. 1–6.
[66] H. Lee and J. Flinn, "Reducing tail response time of vehicular applications," in *Proc. IEEE/ACM Symp. Edge Comput. (SEC)*, Oct. 2016, pp. 103–104.
[67] T. G. Rodrigues, K. Suto, H. Nishiyama, and N. Kato, "Hybrid method for minimizing service delay in edge cloud computing through VM migration and transmission power control," *IEEE Trans. Comput.*, vol. 66, no. 5, pp. 810–819, May 2017.
[68] S. Abdelwahab, B. Hamdaoui, M. Guizani, and T. Znati, "REPLISOM: Disciplined tiny memory replication for massive IoT devices in LTE edge cloud," *IEEE Internet Things J.*, vol. 3, no. 3, pp. 327–338, Jan. 2016.
[69] H. P. Sajjad, K. Danniswara, A. Al-Shishtawy, and V. Vlassov, "SpanEdge: Towards unifying stream processing over central and near-the-edge data centers," in *Proc. IEEE/ACM Symp. Edge Comput. (SEC)*, Oct. 2016, pp. 168–178.
[70] K. Zhang, Y. Mao, S. Leng, A. Vinel, and Y. Zhang, "Delay constrained offloading for mobile edge computing in cloud-enabled vehicular networks," in *Proc. 8th Int. Workshop Resilient Netw. Design Modeling (RNDM)*, Sep. 2016, pp. 288–294.
[71] S. Nunna *et al.*, "Enabling real-time context-aware collaboration through 5G and mobile edge computing," in *Proc. 12th Int. Conf. Inf. Technol.-New Generat. (ITNG)*, Apr. 2015, pp. 601–605.
[72] A. Papageorgiou, E. Poormohammady, and B. Cheng, "Edge-computing-aware deployment of stream processing tasks based on topology-external information: Model, algorithms, and a storm-based prototype," in *Proc. IEEE Int. Congr. Big Data (BigData Congress)*, Jan. 2016, pp. 259–266.
[73] L. Gu, D. Zeng, S. Guo, A. Barnawi, and Y. Xiang, "Cost efficient resource management in fog computing supported medical cyber-physical system," *IEEE Trans. Emerg. Topics Comput.*, vol. 5, no. 1, pp. 108–119, Jan./Mar. 2015.
[74] M. Barcelo, A. Correa, J. Llorca, A. M. Tulino, J. L. Vicario, and A. Morell, "IoT-cloud service optimization in next generation smart environments," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 12, pp. 4077–4090, Dec. 2016.
[75] K. Zhang *et al.*, "Energy-efficient offloading for mobile edge computing in 5G heterogeneous networks," *IEEE Access*, vol. 4, pp. 5896–5907, 2016.
[76] Y. Mao, J. Zhang, and K. B. Letaief, "Dynamic computation offloading for mobile-edge computing with energy harvesting devices," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 12, pp. 3590–3605, Dec. 2016.
[77] S. Sardellitti, G. Scutari, and S. Barbarossa, "Joint optimization of radio and computational resources for multicell mobile-edge computing," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 1, no. 2, pp. 89–103, Jun. 2015.

[78] J. Plachy, Z. Becvar, and E. C. Strinati, "Cross-layer approach enabling communication of high number of devices in 5G mobile networks," in *Proc. IEEE 11th Int. Conf. Wireless Mobile Comput., Netw. Commun. (WiMob)*, Oct. 2015, pp. 809–816.

[79] J. Baliga, R. W. A. Ayre, K. Hinton, and R. S. Tucker, "Green cloud computing: Balancing energy in processing, storage, and transport," *Proc. IEEE*, vol. 99, no. 1, pp. 149–167, Jan. 2011.

[80] D. N. Serpanos, L. Georgiadis, and T. Bouloutas, "MMPacking: A load and storage balancing algorithm for distributed multimedia servers," in *Proc. IEEE Int. Conf. Comput. Design, VLSI Comput. Process. (ICCD)*, Jun. 1996, pp. 170–174.

[81] A. Singh, M. Korupolu, and D. Mohapatra, "Server-storage virtualization: Integration and load balancing in data centers," in *Proc. ACM/IEEE Conf. Supercomput.*, Nov. 2008, p. 53.

[82] D. Ford *et al.*, "Availability in globally distributed storage systems," in *Proc. Osdi*, vol. 10. 2010, pp. 1–7.

[83] F. Chang *et al.*, "Bigtable: A distributed storage system for structured data," *ACM Trans. Comput. Syst.*, vol. 26, no. 2, 2008, Art. no. 4.

[84] A. G. Dimakis, K. Ramchandran, Y. Wu, and C. Suh, "A survey on network codes for distributed storage," *Proc. IEEE*, vol. 99, no. 3, pp. 476–489, Mar. 2011.

[85] E. S. Andreas *et al.*, "Proactive replication for data durability," in *Proc. 5th Int. Workshop Peer-Peer Syst. (IPTPS)*, 2006, pp. 1–6.

[86] A. Van Kempen, E. Le Merrer, and N. Le Scouarnec, "Method of data replication in a distributed data storage system and corresponding device," U.S. Patent 8 812 801 B2, Aug. 19, 2014.

[87] T. Zhao, S. Zhou, X. Guo, Y. Zhao, and Z. Niu, "Pricing policy and computational resource provisioning for delay-aware mobile edge computing," in *Proc. IEEE/CIC Int. Conf. Commun. China (ICCC)*, Jul. 2016, pp. 1–6.

[88] A. Kiani and N. Ansari. (Dec. 2016). "Towards hierarchical mobile edge computing: An auction-based profit maximization approach." [Online]. Available: https://arxiv.org/abs/1612.00122

[89] Y. Zhang, C. Lee, D. Niyato, and P. Wang, "Auction approaches for resource allocation in wireless systems: A survey," *IEEE Commun. Surveys Tuts.*, vol. 15, no. 3, pp. 1020–1041, 3rd Quart., 2013.

[90] D. An, Q. Yang, W. Yu, X. Yang, X. Fu, and W. Zhao, "SODA: Strategy-proof online double auction scheme for multimicrogrids bidding," *IEEE Trans. Syst., Man, Cybern., Syst.*, to be published.

[91] N. Kamiyama, Y. Nakano, K. Shiomoto, G. Hasegawa, M. Murata, and H. Miyahara, "Priority control based on website categories in edge computing," in *Proc. IEEE Conf. Comput. Commun. Workshops (INFOCOM WKSHPS)*, Apr. 2016, pp. 776–781.

[92] C. You, K. Huang, H. Chae, and B.-H. Kim, "Energy-efficient resource allocation for mobile-edge computation offloading," *IEEE Trans. Wireless Commun.*, vol. 16, no. 3, pp. 1397–1411, Mar. 2017.

[93] B. Varghese, N. Wang, S. Barbhuiya, P. Kilpatrick, and D. S. Nikolopoulos, "Challenges and opportunities in edge computing," in *Proc. IEEE Int. Conf. Smart Cloud (SmartCloud)*, Nov. 2016, pp. 20–26.

[94] L. Zhang *et al.*, "Named data networking (NDN) project," Xerox Palo Alto Res. Center-PARC, Palo Alto, CA, USA, Tech. Rep. NDN-0001, 2010.

[95] D. Raychaudhuri, K. Nagaraja, and A. Venkataramani, "MobilityFirst: A robust and trustworthy mobility-centric architecture for the future Internet," *ACM SIGMOBILE Mobile Comput. Commun. Rev.*, vol. 16, no. 3, pp. 2–13, 2012.

[96] W. Yu, H. Zhang, Y. Wu, D. Griffith, and N. Golmie, "A framework to enable multiple coexisting Internet of Things applications," in *Proc. Int. Conf. Comput., Netw. Commun.*, Mar. 2018.

[97] X. Chen, L. Jiao, W. Li, and X. Fu, "Efficient multi-user computation offloading for mobile-edge cloud computing," *IEEE/ACM Trans. Netw.*, vol. 24, no. 5, pp. 2795–2808, Oct. 2016.

[98] A.-L. Jin, W. Song, P. Wang, D. Niyato, and P. Ju, "Auction mechanisms toward efficient resource sharing for cloudlets in mobile cloud computing," *IEEE Trans. Serv. Comput.*, vol. 9, no. 6, pp. 895–909, Nov. 2016.

[99] W. Shi, L. Zhang, C. Wu, Z. Li, and F. Lau, "An online auction framework for dynamic resource provisioning in cloud computing," *ACM SIGMETRICS Perform. Eval. Rev.*, vol. 42, no. 1, pp. 71–83, 2014.

[100] B. Ramachandran, S. K. Srivastava, C. S. Edrington, and D. A. Cartes, "An intelligent auction scheme for smart grid market using a hybrid immune algorithm," *IEEE Trans. Ind. Electron.*, vol. 58, no. 10, pp. 4603–4612, Oct. 2011.

[101] H. S. V. S. K. Nunna and D. Srinivasan, "Multiagent-based transactive energy framework for distribution systems with smart microgrids," *IEEE Trans. Ind. Informat.*, vol. 13, no. 5, pp. 2241–2250, Oct. 2017.

[102] Q. Yang, D. An, W. Yu, X. Yang, and X. Fu, "On stochastic optimal bidding strategy for microgrids," in *Proc. IEEE 34th Int. Perform. Comput. Commun. Conf. (IPCCC)*, Dec. 2015, pp. 1–8.

[103] P. Di Lorenzo, S. Barbarossa, and S. Sardellitti. (Jul. 2013). "Joint optimization of radio resources and code partitioning in mobile edge computing." [Online]. Available: https://arxiv.org/abs/1307.3835

[104] Z. Qin, Y. Yang, T. Yu, I. Khalil, X. Xiao, and K. Ren, "Heavy hitter estimation over set-valued data with local differential privacy," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur. (CCS)*, New York, NY, USA, 2016, pp. 192–203. [Online]. Available: http://doi.acm.org/10.1145/2976749.2978409

[105] X. Yang, X. Ren, J. Lin, and W. Yu, "On binary decomposition based privacy-preserving aggregation schemes in real-time monitoring systems," *IEEE Trans. Parallel Distrib. Syst.*, vol. 27, no. 10, pp. 2967–2983, Oct. 2016.

[106] J. Lin, W. Yu, N. Zhang, X. Yang, and L. Ge, "On data integrity attacks against route guidance in transportation-based cyber-physical systems," in *Proc. 14th IEEE Annu. Consum. Commun. Netw. Conf. (CCNC)*, Jan. 2017, pp. 313–318.

[107] Q. Yang, D. An, R. Min, W. Yu, X. Yang, and W. Zhao, "On optimal PMU placement-based defense against data integrity attacks in smart grid," *IEEE Trans. Inf. Forensics Security*, vol. 12, no. 7, pp. 1735–1750, Jul. 2017.

[108] Q. Yang, J. Yang, W. Yu, D. An, N. Zhang, and W. Zhao, "On false data-injection attacks against power system state estimation: Modeling and countermeasures," *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 3, pp. 717–729, Mar. 2014.

[109] W. Yu, D. Griffith, L. Ge, S. Bhattarai, and N. Golmie, "An integrated detection system against false data injection attacks in the smart grid," *Secur. Commun. Netw.*, vol. 8, no. 2, pp. 91–109, 2015. [Online]. Available: http://dx.doi.org/10.1002/sec.957

[110] S. Bhattarai, S. Wei, S. Rook, W. Yu, R. F. Erbacher, and H. Cam, "On simulation studies of jamming threats against LTE networks," in *Proc. Int. Conf. Comput., Netw. Commun. (ICNC)*, Feb. 2015, pp. 99–103.

[111] S. Bhattarai, S. Rook, L. Ge, S. Wei, W. Yu, and X. Fu, "On simulation studies of cyber attacks against LTE networks," in *Proc. 23rd Int. Conf. Comput. Commun. Netw. (ICCCN)*, Aug. 2014, pp. 1–8.

[112] S. Yi, Z. Qin, and Q. Li, "Security and privacy issues of fog computing: A survey," in *Proc. Int. Conf. Wireless Algorithms, Syst., Appl.*, 2015, pp. 685–695.

[113] Y. Huang, X. Yang, S. Yang, W. Yu, and X. Fu, "A cross-layer approach handling link asymmetry for wireless mesh access networks," *IEEE Trans. Veh. Technol.*, vol. 60, no. 3, pp. 1045–1058, Mar. 2011.

[114] W. Yu, H. Xu, A. Hematian, D. Griffith, and N. Golmie, "Towards energy efficiency in ultra dense networks," in *Proc. IEEE 35th Int. Perform. Comput. Commun. Conf. (IPCCC)*, Dec. 2016, pp. 1–8.

[115] C.-F. Lai, Y.-C. Chang, H.-C. Chao, M. S. Hossain, and A. Ghoneim, "A buffer-aware QoS streaming approach for SDN-enabled 5G vehicular networks," *IEEE Commun. Mag.*, vol. 55, no. 8, pp. 68–73, Aug. 2017.

[116] A. Mosenia and N. K. Jha, "A comprehensive study of security of Internet-of-Things," *IEEE Trans. Emerg. Topics Comput.*, vol. 5, no. 4, pp. 586–602, Oct./Dec. 2017.

[117] M. Van Dijk, C. Gentry, S. Halevi, and V. Vaikuntanathan, "Fully homomorphic encryption over the integers," in *Proc. Annu. Int. Conf. Theory Appl. Cryptograph. Techn.*, 2010, pp. 24–43.

[118] D. Li, Q. Yang, W. Yu, D. An, X. Yang, and W. Zhao, "A strategy-proof privacy-preserving double auction mechanism for electrical vehicles demand response in microgrids," in *Proc. IEEE Int. Perform. Comput. Commun. Conf. (IPCCC)*, Dec. 2017.

[119] B. Pavithra and C. S. Anita, "Privacy-preserving public auditing for data storage security in cloud computing," *Adv. Natural Appl. Sci.*, vol. 10, no. 14, pp. 118–122, 2016.

[120] F. Chen, T. Xiang, X. Fu, and W. Yu, "User differentiated verifiable file search on the cloud," *IEEE Trans. Serv. Comput.*, to be published.

[121] C. Anglano, R. Gaeta, and M. Grangetto, "Securing coding-based cloud storage against pollution attacks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 28, no. 5, pp. 1457–1469, May 2017.

[122] R. Gennaro, C. Gentry, and B. Parno, "Non-interactive verifiable computing: Outsourcing computation to untrusted workers," in *Advances in Cryptology—CRYPTO* (Lecture Notes in Computer Science), vol. 6223. Berlin, Germany: Springer, 2010, pp. 465–482.

[123] B. Parno, J. Howell, C. Gentry, and M. Raykova, "Pinocchio: Nearly practical verifiable computation," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2013, pp. 238–252.
[124] I. Hafeez, A. Y. Ding, L. Suomalainen, and S. Tarkoma, "Demo abstract: Securebox—A platform to safeguard network edge," in *Proc. IEEE/ACM Symp. Edge Comput. (SEC)*, Oct. 2016, pp. 117–118.
[125] J. Clemens, R. Pal, and P. Philip, "Poster abstract: Extending trust and attestation to the edge," in *Proc. IEEE/ACM Symp. Edge Comput. (SEC)*, Oct. 2016, pp. 101–102.
[126] S. Echeverría, D. Klinedinst, K. Williams, and G. A. Lewis, "Establishing trusted identities in disconnected edge environments," in *Proc. IEEE/ACM Symp. Edge Comput. (SEC)*, Oct. 2016, pp. 51–63.
[127] J. Tan, R. Gandhi, and P. Narasimhan, "Poster abstract: BUFS: Towards bottom-up foundational security for software in the Internet-of-Things," in *Proc. IEEE/ACM Symp. Edge Comput. (SEC)*, Oct. 2016, pp. 107–108.
[128] A. Mtibaa, K. Harras, and H. Alnuweiri, "Friend or foe? Detecting and isolating malicious nodes in mobile edge computing platforms," in *Proc. IEEE 7th Int. Conf. Cloud Comput. Technol. Sci. (CloudCom)*, Nov. 2015, pp. 42–49.
[129] S. Berger *et al.*, "TVDc: Managing security in the trusted virtual datacenter," *SIGOPS Oper. Syst. Rev.*, vol. 42, no. 1, pp. 40–47, Jan. 2008. [Online]. Available: http://doi.acm.org/10.1145/1341312.1341321
[130] X. Yang, X. He, W. Yu, J. Lin, R. Li, and Q. Yang, "Towards a low-cost remote memory attestation for the smart grid," *Sensors*, vol. 15, no. 8, pp. 20799–20824, 2015.
[131] E. Cau *et al.*, "Efficient exploitation of mobile edge computing for virtualized 5G in EPC architectures," in *Proc. 4th IEEE Int. Conf. Mobile Cloud Comput., Serv., Eng. (MobileCloud)*, Mar. 2016, pp. 100–109.
[132] S.-C. Hung, H. Hsu, S.-Y. Lien, and K.-C. Chen, "Architecture harmonization between cloud radio access networks and fog networks," *IEEE Access*, vol. 3, pp. 3019–3034, 2015.
[133] Y. Chagh, Z. Guennoun, and Y. Jouihri, "Voice service in 5G network: Towards an edge-computing enhancement of voice over Wi-Fi," in *Proc. 39th Int. Conf. Telecommun. Signal Process. (TSP)*, Jun. 2016, pp. 116–120.
[134] E. Zeydan *et al.*, "Big data caching for networking: Moving from cloud to edge," *IEEE Commun. Mag.*, vol. 54, no. 9, pp. 36–42, Sep. 2016.
[135] N. K. Ardi and N. Joshi, "Poster abstract: 5GHealthNet—A cloud based framework for faster and authorized access to private medical records through 5G wireless network," in *Proc. IEEE/ACM Symp. Edge Comput. (SEC)*, Oct. 2016, pp. 89–90.
[136] W. Emfinger, A. Dubey, P. Volgyesi, J. Sallai, and G. Karsai, "Demo abstract: RIAPS—A resilient information architecture platform for edge computing," in *Proc. IEEE/ACM Symp. Edge Comput. (SEC)*, Oct. 2016, pp. 119–120.
[137] N. Kumar, S. Zeadally, and J. J. P. C. Rodrigues, "Vehicular delay-tolerant networks for smart grid data management using mobile edge computing," *IEEE Commun. Mag.*, vol. 54, no. 10, pp. 60–66, Oct. 2016.
[138] G. Xu, W. Yu, D. Griffith, N. Golmie, and P. Moulema, "Toward integrating distributed energy resources and storage devices in smart grid," *IEEE Internet Things J.*, vol. 4, no. 1, pp. 192–204, Feb. 2017.
[139] A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi, "Internet of Things for smart cities," *IEEE Internet Things J.*, vol. 1, no. 1, pp. 22–32, Feb. 2014.
[140] M. Sapienza, E. Guardo, M. Cavallo, G. L. Torre, G. Leombruno, and O. Tomarchio, "Solving critical events through mobile edge computing: An approach for smart cities," in *Proc. IEEE Int. Conf. Smart Comput. (SMARTCOMP)*, May 2016, pp. 1–5.
[141] Q. Zhang, Z. Yu, W. Shi, and H. Zhong, "Demo abstract: EVAPS: Edge video analysis for public safety," in *Proc. IEEE/ACM Symp. Edge Comput. (SEC)*, Oct. 2016, pp. 121–122.
[142] N. Chen, Y. Chen, S. Song, C.-T. Huang, and X. Ye, "Poster abstract: Smart urban surveillance using fog computing," in *Proc. IEEE/ACM Symp. Edge Comput. (SEC)*, Oct. 2016, pp. 95–96.
[143] I. Kalamaras *et al.*, "An interactive visual analytics platform for smart intelligent transportation systems management," *IEEE Trans. Intell. Transp. Syst.*, to be published.
[144] H.-T. Wu and G.-J. Horng, "Establishing an intelligent transportation system with a network security mechanism in an Internet of vehicle environment," *IEEE Access*, vol. 5, pp. 19239–19247, 2017.
[145] K. Sasaki, N. Suzuki, S. Makido, and A. Nakao, "Vehicle control system coordinated between cloud and mobile edge computing," in *Proc. 55th Annu. Conf. Soc. Instrum. Control Eng. Jpn. (SICE)*, Sep. 2016, pp. 1122–1127.

WEI YU received the B.S. degree in electrical engineering from the Nanjing University of Technology, Nanjing, China, in 1992, the M.S. degree in electrical engineering from Tongji University, Shanghai, China, in 1995, and the Ph.D. degree in computer engineering from Texas A&M University in 2008. He is currently an Associate Professor with the Department of Computer and Information Sciences, Towson University, Towson, MD, USA. Before joining Towson University, he was with Cisco Systems Inc. for nine years. His research interests include cyberspace security and privacy, computer networks, cyber-physical systems, distributed computing, and big data analytics. He was a recipient of the 2014 NSF CAREER Award, the 2015 University System of Maryland (USM) Regents' Faculty Award for Excellence in Scholarship, Research, or Creative Activity, and the USM's Wilson H. Elkins Professorship Award in 2016. His paper has also received best paper awards from the IEEE ICC 2008, ICC 2013, the IEEE IPCCC 2016, and WASA 2017.

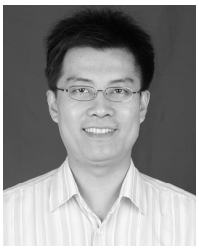
FAN LIANG received the Bachelor's degree in computer science from Northwestern Polytechnical University, Xi'an, China, in 2005, and the master's degree in computer engineering from the University of Massachusetts Dartmouth in 2015. He is currently pursuing the Ph.D. degree in computer science with Towson University. His research interests include wireless networks, big data, smart grid, and network security.

XIAOFEI HE received the B.S. degree from the Department of Computer Science and Technology, Xi'an Jiaotong University, China, in 2011, where he is currently pursuing the Ph.D. degree with the Department of Computer Science and Technology. His research interests include smart grid, Internet of Things, and wireless networks.

WILLIAM GRANT HATCHER received the B.Sc. degree in materials science and engineering from the University of Maryland. He is currently pursuing the master's degree in computer science with Towson University. His research interests include mobile computing and security, big data, and machine learning.

CHAO LU received the B.S. degree in engineering from Shandong University, China, in 1982, and the M.S. and Ph.D. degrees in engineering from The City University of New York. He has been with Towson University as a Professor of computer science since 1990. He has authored or co-authored over 80 research papers. His research interests include error-free computing, human motion classification, computer vision, parallel/distributed computing, and cyber security. He received the U.S. Federal Excellence in Technology Transfer Award and the Alan Berman Research Publications Award in 2001.

JIE LIN received the B.S. and Ph.D. degrees from the Department of Computer Science and Technology, Xi'an Jiaotong University, in 2009 and 2013, respectively. He is currently an Associate Professor with the Department of Computer Science and Technology, Xi'an Jiaotong University. His research interests include Internet-of-Things, cyberspace security, and computer networks.



XINYU YANG received the Bachelor's, master's, and Ph.D. degrees and the Diploma degree in computer science and technology from Xi'an Jiaotong University, China, in 1995, 1997, 2001, and 2001, respectively. He is currently a Professor with the Department of Computer Science and Technology, Xi'an Jiaotong University. His research interests include wireless communication, mobile ad hoc networks, and network security.