Contents

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.

Chapter 1

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. 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 , smart city ,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, and present our view on edge computing for the IoT.

Chapter 2

REVIEW OF IoT AND EDGE COMPUTING

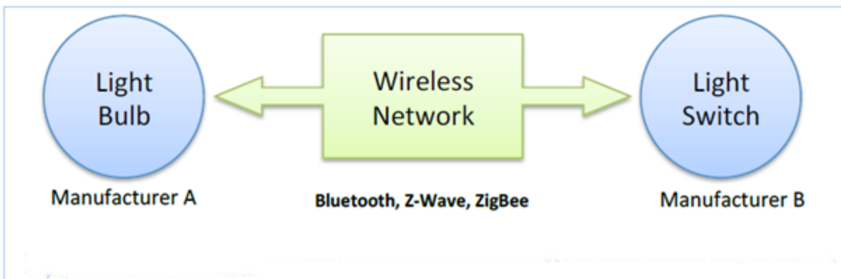
In this section, we will review the basic concepts of IoT and edge computing, and discuss the potential for integrating the two technologies.

2.1 INTERNET OF THINGS

The Internet of Things (IoT) refers to the use of intelligently connected devices and systems to leverage data gathered by embedded sensors and actuators in machines and other physical objects.IoT systems allow users to achieve deeper automation, analysis, and integration within a system. They improve the reach of these areas and their accuracy. IoT utilizes existing and emerging technology for sensing, networking, and robotics. IoT exploits recent advances in software, falling hardware prices, and modern attitudes towards technology. Its new and advanced elements bring major changes in the delivery of products, goods, and services; and the social, economic, and political impact of those changes.IoT describes a system where items in the physical world, and sensors within or attached to these items, are connected to the Internet via wireless and wired Internet connections. These sensors can use various types of local area connections such as RFID, NFC, Wi-Fi, Bluetooth, and Zigbee. Sensors can also have wide area connectivity such as GSM, GPRS, 3G, and LTE.
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 .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.



Example of device-to-device Communication model

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, because of the limitations of the devices computational ability or storage space. This approach normally requires assistance from preexisting 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 Machine-to-Cloud communication model, it is necessary to optimize the network structure.

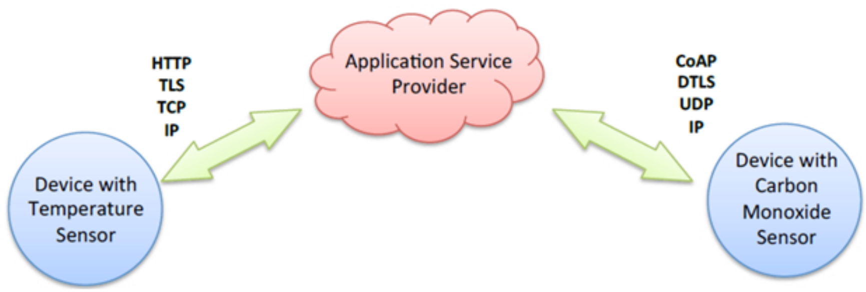
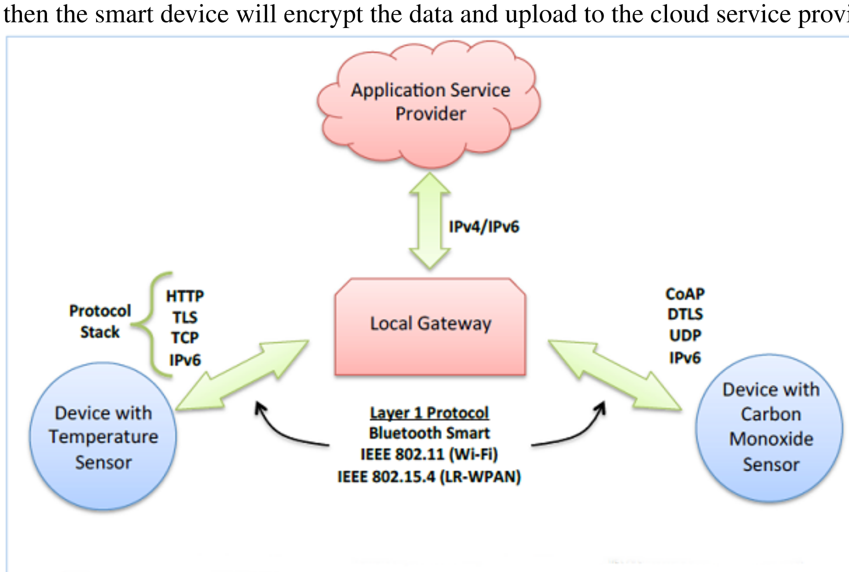


Fig.2

3) MACHINE-TO-GATEWAY COMMUNICATION

In the machine-to-gateway model, the device-to-applicationlayer gateway (ALG) model is considered as a proxy or middleware box . In Figure, 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.



2.2 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.

2.2.1 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.

2.2.2 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 preprocessing 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.

2.2.3 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.

2.3 EDGE COMPUTING

Edge computing allows data produced by internet of things (IoT) devices to be processed closer to where it is created instead of sending it across long routes to data centers or clouds.
Doing this computing closer to the edge of the network lets organizations analyze important data in near real-time a need of organizations across many industries, including manufacturing, health care, telecommunications and finance.
Edge computing is a mesh network of micro data centers that process or store critical data locally and push all received data to a central data center or cloud storage repository, in a footprint of less than 100 square feet, according to research firm IDC. It is typically referred to in IoT use cases, where edge devices would collect data sometimes massive amounts of it and send it all to a data center or cloud for processing. Edge computing triages the data locally so some of it is processed locally, reducing the backhaul traffic to the central repository.

2.4 EDGE COMPUTING ARCHITECTURE

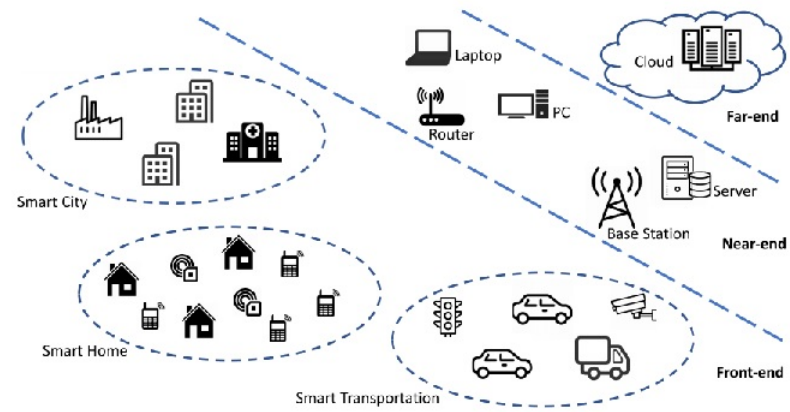


FIGURE 5. A typical architecture of edge computing networks.

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 frontend 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 frontend 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 nearend 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.

2.5 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.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 .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) 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.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 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 mobilefriendly optimization. Via the designed mechanisms, fairness among users can be realized by regulating the TCP concurrent connections. Manzalini and Crespi proposed an edge operating system, which leverages available open source software to

achieve powerful network and service platforms. Salman et al.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.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.

Chapter 3

INTEGRATION OF IoT AND EDGE COMPUTING

Here we 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.

3.1 OVERVIEW

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. we can see that IoT and edge computing have smiler characteristics, as further demonstrated in Table 1. Notice that we also include cloud computing as a reference.

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

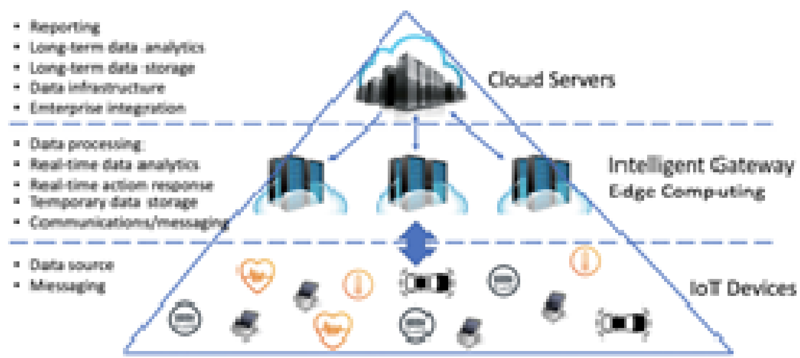


FIGURE 6. Layer architecture of edge computing-based IoT.

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.

3.2 IoT PERFORMANCE DEMANDS

3.2.1 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. Indeed, large network latency will be unacceptable, and cannot satisfy the QoS requirements.

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.

3.2.2 STORAGE

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.

3.2.3 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.

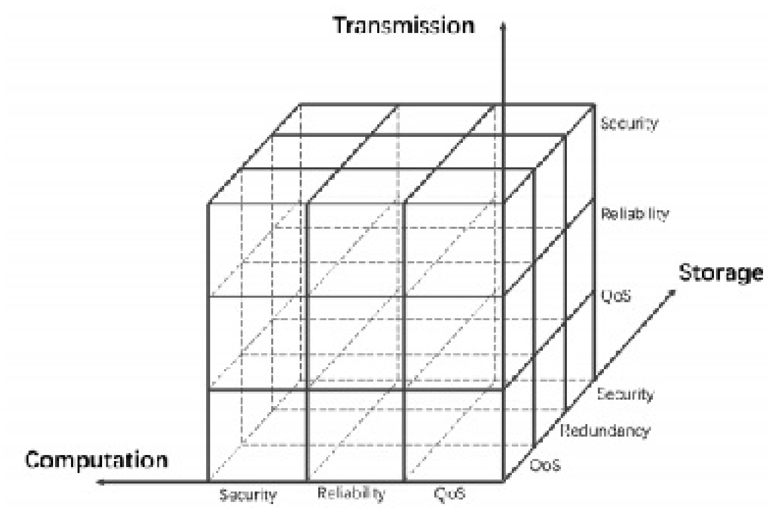


FIGURE 7. The problem space of Edge Computing-based IoT.

In the following, we will discuss how Edge Computing based IoT satisfies the requirements of transmission, storage,and computation,in detail.

Chapter 4

ADVANTAGES OF EDGE COMPUTING-BASED IoT

In this section, we assess the advantages of integrating IoT with edge computing in terms of Transmission,Storage and Computation.

4.1 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.
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, Human Action Classification, Motion Estimation, etc.

4.1.1 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.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.

4.1.2 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.
There have been a number of research efforts devoted on this issue. For example, Abdelwahab et al.proposed an LTE-aware edge cloud architecture and an LTEoptimized memory replication protocol, called REPLISOM. The designed protocol can effectively schedule the memory replication operations.

4.1.3 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.
Zhang et al.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.

4.1.4 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.proposed a cross-layer scheme, aiming to minimize overhead and improve transmission efficiency for 5G mobile networks.

4.2 B. STORAGE

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.

4.2.1 STORAGE BALANCING

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. For example, in, 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

4.2.2 RECOVERY POLICY

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.
b: DATA REPLICATION
In distributed storage systems, data is divided into many pieces, and each piece of data has fixed size and code blocks. 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. Edge computingbased 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.

4.3 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.Different 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.

4.3.1 COMPUTATION OFFLOADING

For 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
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
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.

4.3.2 PRICING POLICY

Resource allocation schemes can be derived through a proper pricing policy for the resources in the networks.
a: SINGLE SERVICE PROVIDER
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.
b: MULTIPLE SERVICE PROVIDERS
In this,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.

4.3.3 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.

Chapter 5

CHALLENGES OF EDGE COMPUTING-BASED IoT

In this section, we will discuss the challenges of Edge Computing-based IoT.

5.1 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.

5.2 RESOURCE MANAGEMENT

The integration of IoT and edge computing necessitates complete and thorough understanding and optimization of resource management.
The management of these resources can be conducted through a variety of means.

5.2.1 AUCTION-BASED

Various economic-driven schemes can be used to manage network resources.. 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.

5.2.2 OPTIMIZATION

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.

5.3 SECURITY AND PRIVACY

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.
Based on the concerns, we identify the problem space, shown in Fig. 8, and its dependence on the IoT structures in Fig. 6.

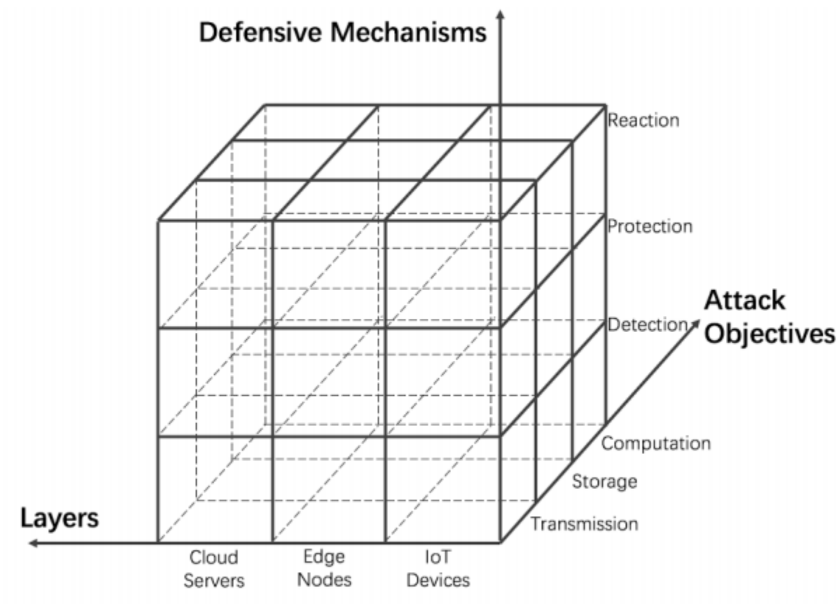


FIGURE 8. Problem space of Edge Computing-based IoT security.

Security and privacy can be concerned in terms of Transmission,Storage and Computation.

5.3.1 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 could be launched to disable the links by congesting the network, or could monitor network data flow.

To mitigate this problem, Software-Defined Networking (SDN) 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 mali-

cious traffic.
II. Reactions: Following from a long history of conventional countermeasures against network threats in cyberphysical systems, there are ongoing efforts to assess and prevent cyber-attacks in edge computing environments .

5.3.2 STORAGE

In edge computing IoT , There are numerous reasons which clearly states the increase in 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 , so that integrity, confidentiality, and verifiability for edge storage systems can be realized.
Another challenge for storage is ensuring data reliability.Anglano et al. 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.

5.3.3 COMPUTATION

Another important security challenge in Edge Computingbased IoT is to maintain security and privacy in uploading computational tasks to edge computation nodes.
To ensure computation security, Verifiable Computing 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.
To ensure computation security, Verifiable Computing 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.

5.4 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. With these advances in communication technologies, edge computing will further progress as integration of these technologies becomes inevitable.

5.5 SMART SYSTEM SUPPORT

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. 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. 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.
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.One of the most critical challenges is the non-interoperability of the heterogeneous technologies in cities.
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. 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.

3) SMART TRANSPORTATION

To achieve safe and effective autonomous driving, a cloudbased vehicle control system is needed, because it can collect information from the sensors via a vehicle-to-vehicle network. 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.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.

Chapter 6

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.

Chapter 7

REFERENCES

[1] D. Linthicum, Responsive data architecture for the Internet of Things, Computer, vol. 49, no. 10, pp. 7275, 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. 11251142, Oct. 2017.
[3] J. A. Stankovic, Research directions for the Internet of Things, IEEE Internet Things J., vol. 1, no. 1, pp. 39, 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. 2:12:12, 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. 25512566, 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. 9981010, Oct. 2012
[7] 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. 2079920824, 2015.
[8] 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. 100109.
[9] 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. 30193034, 2015.

[10] 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. 116120.
[11] E. Zeydan et al., Big data caching for networking: Moving from cloud to edge, IEEE Commun. Mag., vol. 54, no. 9, pp. 3642, Sep. 2016.
[12] N. K. Ardi and N. Joshi, Poster abstract: 5GHealthNetA 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. 8990.
[13] W. Emfinger, A. Dubey, P. Volgyesi, J. Sallai, and G. Karsai, Demo abstract: RIAPSA resilient information architecture platform for edge computing, in Proc. IEEE/ACM Symp. Edge Comput. (SEC), Oct. 2016, pp. 119120.
[14] 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. 6066, Oct. 2016.
[15] E. Zeydan et al., Big data caching for networking: Moving from cloud to edge, IEEE Commun. Mag., vol. 54, no. 9, pp. 3642, Sep. 2016.