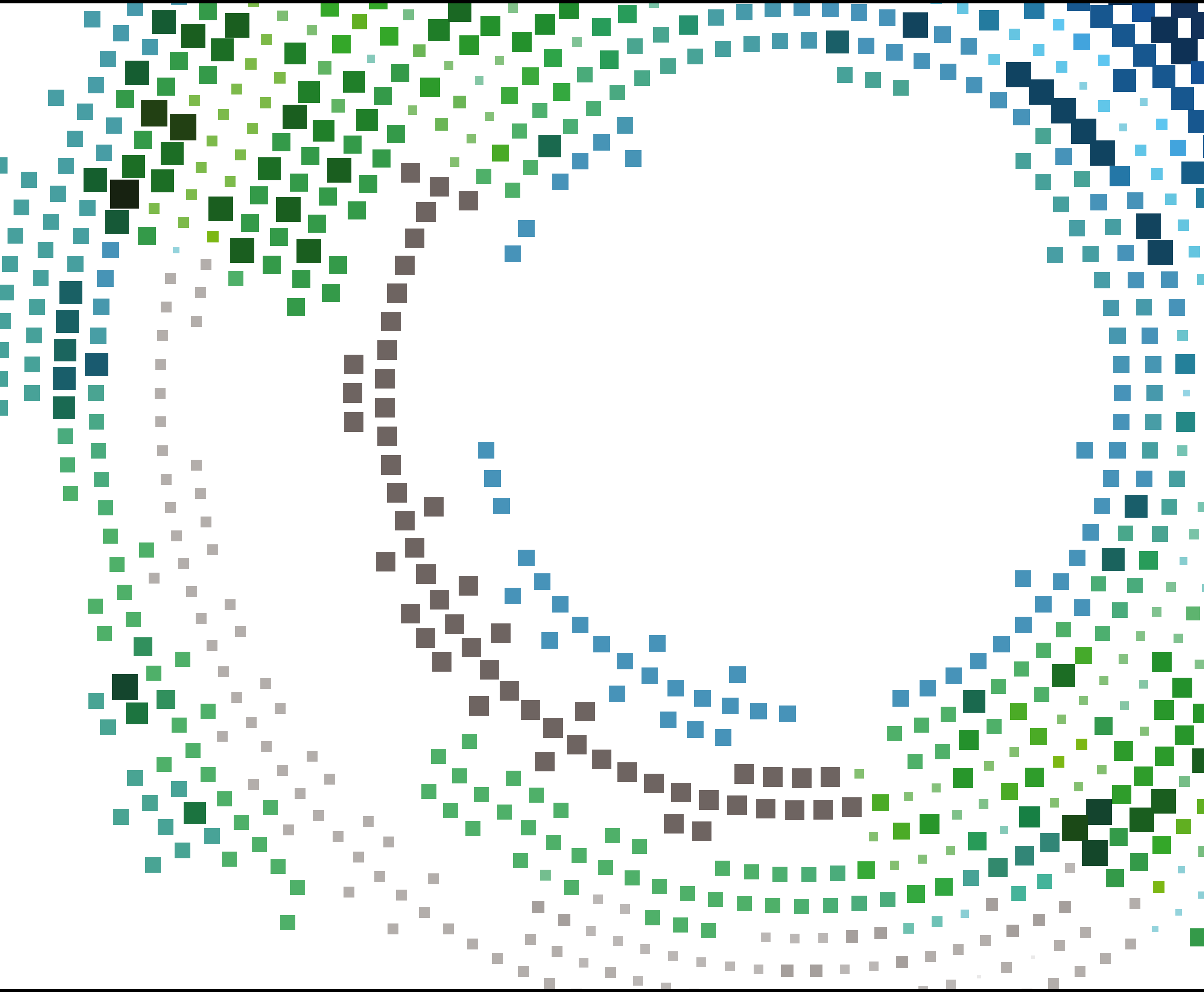
Mobile Information Systems

Enabling Technologies towards Next Generation Mobile Systems and Networks

Guest Editors: Yeong M. Jang, Juan C. Cano, Kun Yang, and Young-June Choi



Enabling Technologies towards Next Generation Mobile Systems and Networks

Mobile Information Systems

Enabling Technologies towards Next Generation Mobile Systems and Networks

Guest Editors: Yeong M. Jang, Juan C. Cano, Kun Yang, and Young-June Choi

Editor-in-Chief

Editorial Board

Contents

Hindawi Publishing Corporation
Mobile Information Systems
Volume 2016, Article ID 9805636, 2 pages
http://dx.doi.org/10.1155/2016/9805636

Editorial

Enabling Technologies towards Next Generation Mobile Systems and Networks

Yeong M. Jang,[1] Juan C. Cano,[2] Kun Yang,[3] and Young-June Choi[4]

[1] *Kookmin University, Seoul, Republic of Korea*
[2] *Universitat Politècnica de Valencia, Valencia, Spain*
[3] *University of Essex, Colchester, UK*
[4] *Ajou University, Suwon, Republic of Korea*

Correspondence should be addressed to Yeong M. Jang; yjang@kookmin.ac.kr

Received 4 December 2016; Accepted 5 December 2016

Mobile services have become an essential part in the era of 5G networks. The mobile system will be based on cloud computing, IoT, user-centric services, and mobile communication. Cloud service is necessary to support mobility and real-time operation, reliable content delivery such as content-centric network, and content delivery networks together with mobile cloud systems. Next generation mobile systems and networks are also facing a huge challenge to handle a large number of IoT devices because in the near future all the devices will be connected with each other. So effective networks and systems are necessary to manage and handle the increasing numbers of devices such as cloud computing, automated network management, new service platforms, and new network architectures (e.g., Software Defined Network (SDN) and Network Function Virtualization (NFV)) towards the promising 5G mobile networks and services.

In this special issue, we have invited a few papers that address such issues. Among them, ten selected papers are addressed to researchers and engineers practicing in the scientific areas for the next generation mobile system and networks.

The paper entitled "An Architecture of IoT Service Delegation and Resource Allocation Based on Collaboration between Fog and Cloud Computing" by A. A. Alsaffar et al. presents an architecture and new algorithm for smart IoT service based on three conditions for managing and delegating user request. They also propose a new technique to take care of fog and cloud environment by allocating resources to ensure QoS and service level agreement (SLA). Their proposed scheme shows improved management, better service delegation, efficient resource allocation, and big data distribution compared to the existing methods.

The paper "Multivariate Multiple Regression Models for a Big Data-Empowered SON Framework in Mobile Wireless Networks" by Y. Shin et al. outlines the background of big data, big data self-organizing networks (BSON) framework, and multiple regression models. The authors propose multivariate multiple regression models for the BSON framework with the implementation using MapReduce.

The paper titled "mCSQAM: Service Quality Assessment Model in Mobile Cloud Services Environment" by Y.-R. Shin and E.-N. Huh proposes an architecture named mCSQAM to determine the quality metrics for limiting the problems of cloud computing. The authors propose an Analytic Hierarchy Process (AHP) method to access the mobile cloud services based on different requirements from the service consumers.

The paper named "Securing SDN Southbound and Data Plane Communication with IBC" by J. Lam et al. presents a distributed SDN secured communication with a multidomain capable Identity-Based Cryptography (IBC) protocol, particularly for the southbound and wireless data plane communication. They also analyzed the TLS-secured Message Queuing Telemetry Transport (MQTT) message exchange protocol to find out the possible bandwidth saved with IBC. The authors argue that this system is easier to use because it ensures higher network performance and lower power consumption of the IoT devices as well as supporting more IoT devices without upgrading the infrastructure.

The paper "Data-Driven Handover Optimization in Next Generation Mobile Communication Networks" by P.-C. Lin et al. addresses network densification problems (mobility problems) for next generation mobile communication networks due to the increasing network capacity. The authors propose a data-driven handover optimization (DHO) method to mitigate the problems. The DHO approach collects data from mobile communication to form a model through a key performance indicator (KPI). The authors think that the results using the proposed approach could successfully relieve the mobility problems.

The paper named "Hierarchical Brokering with Feedback Control Framework in Mobile Device-Centric Clouds" by C.-L. Chen et al. presents a hierarchical brokering architecture (HiBA) and Mobile Multicloud Networking (MMCN) feedback control framework for next generation device-centric cloud computing with the mathematical analysis for availability and network latency. The authors perform an experiment with HiBA federates heterogeneous mobile and fixed devices in three tiers using different network interfaces. From the results, it shows that the approach is an amended platform for mobile cloud computing and ensures a sensible solution to various services.

The paper titled "SDN Based User-Centric Framework for Heterogeneous Wireless Networks" by Z. Lu et al. designs a new framework for heterogeneous wireless networks with the support of user-centricity to fulfill users' preferences and requirements. Away from the conventional framework SDN based framework provides better performance. As SDN has decoupled data and control plane virtually and logically centralized structure, it is easy to manage the HetNets in an efficient and flexible way. In this paper, the authors also analyze the possible overheads of the user-centric framework such as signaling overhead and control delay.

The paper "Performance Evaluation of Moving Small-Cell Network with Proactive Cache" by Y. M. Kwon et al. presents a moving small-cells (mSCs) network, its architecture, and the proposed proactive caching mechanism. The results confirm that the QoS of moving cell can be improved by using mSCs together with proactive caching which also reduces the wireless backhaul load and increases the overall network capacity. The authors also argue that the overall network performance is highly dependent on the number of mSCs deployed, cache size, and content popularity.

The paper entitled "SmartCop: Enabling Smart Traffic Violations Ticketing in Vehicular Named Data Networks" by S. H. Ahmed et al. addresses a problem of smart traffic violation ticketing (TVT) in Vehicular Ad hoc Networks (VANETs). The existing technologies are not suitable for VANETs as its dependency on named contents instead of host contents. In this paper, the authors propose a smart TVT system for vehicular named data networking termed as SmartCop which helps a cop vehicle (CV) to issue a TVT to the offenders autonomously. They also provide simulated comparison results for messaging delay, ticket issuing delay, and percentage of detection for different vehicles, CVs, and also the vehicles speeds which are being assessed.

The paper entitled "Survey of Promising Technologies for 5G Networks" by N. T. Le et al. provides a comprehensive survey of the promising technologies for 5G networks such as Software Defined Networking (SDN), cloud computing, IoT, and other wireless technologies. They outline the surveys and provide the future research direction of these technologies for 5G networks.

Acknowledgments

We would like to thank the authors who submitted their articles to this issue and the reviewers for providing constructive and timely feedback.

Yeong M. Jang
Juan C. Cano
Kun Yang
Young-June Choi

Hindawi Publishing Corporation
Mobile Information Systems
Volume 2016, Article ID 2676589, 25 pages
http://dx.doi.org/10.1155/2016/2676589

Review Article

Survey of Promising Technologies for 5G Networks

Nam Tuan Le,[1] Mohammad Arif Hossain,[1] Amirul Islam,[1] Do-yun Kim,[2] Young-June Choi,[2] and Yeong Min Jang[1]

[1] *Department of Electronics Engineering, Kookmin University, Seoul, Republic of Korea*
[2] *Department of Computer Engineering, Ajou University, Suwon, Republic of Korea*

Correspondence should be addressed to Yeong Min Jang; yjang@kookmin.ac.kr

Received 1 April 2016; Revised 18 October 2016; Accepted 25 October 2016

Academic Editor: Yuh-Shyan Chen

As an enhancement of cellular networks, the future-generation 5G network can be considered an ultra-high-speed technology. The proposed 5G network might include all types of advanced dominant technologies to provide remarkable services. Consequently, new architectures and service management schemes for different applications of the emerging technologies need to be recommended to solve issues related to data traffic capacity, high data rate, and reliability for ensuring QoS. Cloud computing, Internet of things (IoT), and software-defined networking (SDN) have become some of the core technologies for the 5G network. Cloud-based services provide flexible and efficient solutions for information and communications technology by reducing the cost of investing in and managing information technology infrastructure. In terms of functionality, SDN is a promising architecture that decouples control planes and data planes to support programmability, adaptability, and flexibility in ever-changing network architectures. However, IoT combines cloud computing and SDN to achieve greater productivity for evolving technologies in 5G by facilitating interaction between the physical and human world. The major objective of this study provides a lawless vision on comprehensive works related to enabling technologies for the next generation of mobile systems and networks, mainly focusing on 5G mobile communications.

1. Introduction

Mobile communication and wireless networks have advanced phenomenally during the last decade. The ever-growing increase in the demand for resources, especially for multimedia data, with high quality of service (QoS) requirements, has promoted the development of 3G and 4G wireless networks. Nevertheless, the achievements of the development in technology cannot fulfill the proper satisfaction. Therefore, the idea of 5G networks that represent networks beyond 4G has become the need of the hour. 5G networks have come into existence owing to the numerous challenges facing 4G networks, such as need for higher data rate and capacity, lower cost, lower end-to-end latency, and massive inter-device connectivity. However, a comprehensive analysis of future networks or next generation networks of information systems that discusses in related forums and standardization is really challenging. The enabling technologies for next generation mobile systems and networking have been surveyed in this paper, which provides readers a clear vision of the current status.

The planning for future network architecture seems to be the definition of Next Generation Networks (NGN). NGN, a great issue for the internet protocol- (IP-) based future of mobile network infrastructure, is considered as a convergence of communication networks which tries to reduce cost and offers integrated services via a core backbone network. It inherits three different advantages of various networking technologies, namely, layered structure, standard interfaces and multiple services, and functions that can be implemented in several layers ranging from MAC to application. With the increase in the number of Internet users and QoS requirements, NGN has become a moving trend for deployment. It established convergence of user access and integrated communication network services with IP technology. The motivation behind the migration of networking systems from the traditional telecommunication network to NGN has been developed based on the advantages of backbone cost reduction, possibility of fast and new service deployment, controllable QoS, compatibility between fixed and wireless networks, network management centralization, and so on. Existing network services based multimedia application such as voice, data, and video transmission at high speeds will be offered as an important outcome of NGN deployment for the fixed and mobile service integration topology. Furthermore, NGN provides low-cost service at high data rates.

The concept of the future network can also be the fifth-generation mobile system, 5G. Over the course of the long history of mobile communication systems from the first generation to 4G LTE-A (Long Term Evolution Advanced), the mobile communications industry has achieved enormous advances in data communication. The next generation can be a revolution in mobile networks that will achieve the best performance in terms of coverage capability, energy consumption, data speeds of 1 Gbps, and better security and energy efficiency over spectral compared to previous networking systems. However, the next generation wireless communication network has not been defined and characterized exactly. Research on 5G has been initiated by many projects, organizations, and standardization forums. Such research on 5G might be directed by the limitations of current technologies. The key requirements of 5G are real wireless communication with no limitation of coverage edge, access policy, and density zone. Secondly, the network should be able to support high-resolution multimedia (HD) broadcasting service. Thirdly, it should have faster data speeds than the previous generations. Finally, it should support new services based on wearable devices. In addition, the NGN is expected to have massive interdevice connections, which can be termed as Connection of Things. The research on 5G is different from that on previous-generation networks because of the limitations of resources in the RF band. The 5G wireless network will mainly focus on new spectrum, multiple-input-multiple-output (MIMO) diversity, transmission access, and new architecture for capacity and connection time [1].

It is a very challenging issue to meet the QoS requirement at a selection service for any network architecture. The convergence of networking and cloud computing are under the consideration in NGNs to cope with the QoS demand. The controllability, management, and optimization of computing resources are the main factors affecting networking performance in the case of cloud computing. One of the advantages of cloud computing is that it is encapsulation-free, which means that users can access services from any location irrespective of host or end device. User can use services without understanding how they operate or deliver data. However, large numbers of vendors are getting interested in information support, storage, and resource computation using cloud-hosting services. With traditional web technology-based services, the relative positions of client and server strongly affect the system QoS and the quality of experience (QoE) [2]. Therefore, future-generation wireless networks are faced with multiple emerging challenges. The Internet of things (IoT) has emerged as one of the leading technologies for future-generation technologies because it is based on the concept of device interconnection, which can be a step toward achieving the QoS and QoE requirements. It is a conceptualization of a cyber-physical system (CPS), a way for using embedded technologies in the future-generation network. Physical systems are unified with the networking and computation system. The scalability of the future-generation network depends on the IoT system because it is a method of assisting connections among a large number of devices in a whole system. IoT has evolved as a system of uninterrupted communication between any device with another device at any place and time. However, the IoT architecture has come under question lately because it is very difficult to support all devices in an inflexible architecture with the traditional networking system. Consequently, several organizations, companies, and committees are working on the standardization issues of IoT to create a unique platform for future-generation networks.

The development of networking depends on the flexibility and mobility of users, and server visualization, which plays an important role in responding effectively and in a timely manner to the dynamic requirements of applications or users. The traditional network infrastructure is continuously becoming obsolete because of the lack of these features. Moreover, manual changes in network configurations increase the complexity of network management, making it nearly impossible at times. The existing infrastructure cannot support priority-based packet-forwarding or dynamic resource allocation to users. Hence, network management, at its root level, has become a challenging issue owing to the limitations of traditional hardware-based networking, such as complex and costly network configuration, and lack of policy changes and fault management. As networking technologies evolve, the network should be able to support the ever-changing networking functionalities of future network infrastructure, such as integration with new services, dynamic network control, better QoS, and efficient packet-forwarding. However, traditional networks cannot support the ever-changing demand of networking technologies. Therefore, software-defined networking (SDN), an emerging technology, can be employed to overcome the limitations of the current networks with the separation of network control from the underlying data planes or switching devices. By breaking the virtual integration between the data plane and the control plane and by using a centralized SDN controller, SDN provides flexibility in changing network policies, easy hardware implementation, and facilitates network innovation and evolution [3, 4]. By integrating SDN with network function virtualization (NFV), one can gain a global view of the entire network by using an open interface such as OpenFlow and the centralized network controller. SDN can support new services and programs at any level of user requirement or need. Furthermore, SDN has attracted considerable interest from both academia and industry over the past few years. It is, after all, an important step in the evolution and development of future network infrastructures.

In this paper, we provide a comprehensive overview of the ongoing research on the enabling technologies for the 5G network. We present the status of work on the important technologies and service models for the next generation of mobile systems and networks. The remainder of this paper is organized as follows. A new model for network control, SDN, and NFV is described in Section 2, while Section 3 presents a survey of the cloud computing model from the viewpoints of network operation and management. The current standardization status, architectures, and applications of IoT for 5G networks are discussed in Section 4. An overview of mobile access networks is presented in Section 5. Our concluding remarks are given in Section 6.

2. Software-Defined Networking (SDN) for 5G

2.1. SDN and NFV

2.1.1. Software-Defined Networking (SDN). Software-defined Networking (SDN) has been introduced for data networks and next generation Internet [5–8]. It has been defined in several ways. The most unambiguous and established definition is provided by the Open Networking Foundation (ONF) [9, 10], a public association dealing with the standardization, development, and commercialization of SDN. The definition is as follows:

> *"Software-Defined Networking (SDN) is an emerging architecture that is dynamic, manageable, cost effective, and adaptable, where control is decoupled from data forwarding and the underlying infrastructure, and directly programmable for network services and applications".*

According to this definition, SDN has the following characteristics: (i) it decouples network control from the underlying data plane (i.e., switches and routers); (ii) it allows the control plane to be programmed directly through an open interface, for instance, OpenFlow [11, 12]; and (iii) it uses a network controller, (i.e., SDN controller) to define the behavior and operation of the networking infrastructure. SDN can be an ideal prospective for the high-bandwidth, dynamic nature of network management. SDN provides the flexibility to change the network configuration at the software level, thus reducing the necessity of modification at the hardware level. SDN makes it easier to introduce and deploy new applications and services than the traditional hardware-operated networking architectures. It also ensures the QoS at any level of user requirement. Consequently, it will be an attracting architecture from the viewpoint of reconfiguring and redirecting complex networks for real-time management.

2.1.2. Network Function Virtualization (NFV). An important observation of SDN is NFV [13]. SDN and NFV are mutually beneficial, but they are not fully dependent on each other. In fact, network functions can be employed and virtualized without using an SDN and vice versa. As it is complementary to SDN, NFV can effectively decouple network functionalities and implement them in software. Thus, it can decouple network functions, for instance, routing decisions, from the underlying hardware devices such as routers and switches, and centralize them at remote network servers or in the cloud through an open interface such as OpenFlow. Hence, the overall network architecture can be highly flexible for fast and adaptive reconfiguration.

The combined functionalities of SDN and NFV [14] make SDNs more advantageous than traditional hardware-based networks. The main advantages can be listed as follows: cost minimization, reduced power consumption through equipment consolidation, reduced processing time by minimizing the typical network operator cycle of innovation, centralized network provisioning by decoupling the data plane from network control plane, extension of capabilities, hardware savings, cloud abstraction, guaranteed content delivery, physical versus virtual networking management, and so on. The advantages of SDN are well explained in [15]. Figure 1 shows a comparison between conventional hardware-based networks and SDN. Furthermore, the differences between SDN and conventional hardware-based networks are summarized in Table 1.

Table 1: Differences between SDN and conventional networking.

Software-defined networking	Conventional hardware-based networking
Data and control plane are decoupled by API or OpenFlow	Data and control plane are mounted on same plane, new protocol for every service
Automatic reconfigurable and repolicing logically centralized configuration	Static or manual configuration and reconfiguration takes time
SDN can prioritize or block specific packets	Conventional network leads all packets the same way
Provides global or comprehensive network views leading to consistent and effective policies	Provides limited information about networks
Easy to program according to application and user needs and can be developed quick via software upgrades	Difficult to replace the existing program with new ideas and works according to packet-forwarding tables

2.2. SDN Functionalities. SDN can support multiple functionalities because of its centralized controller and separated data and control plane. The SDN's functionalities, along with its layers and planes, are shown in Figure 2. The general functionalities of SDN are as follows.

Programmability. Network control is directly programmable as the control plane is decoupled from the forwarding or data plane. SDN allows the control plane to be programmed using different software development tools along with the function of customization of the control network according to user requirements.

Centrally Managed. In an SDN, the controller network is logically centralized, thus providing a comprehensive view of the network that appears to the applications or users as a logical device.

Flexibility. SDN provides flexibility to network managers. Network managers can manage, configure, secure, and optimize network parameters very rapidly through dynamic, automated SDN programs. This helps the controllers respond to traffic variations. As controllers run in software, SDN affords the flexibility of synchronization through the network

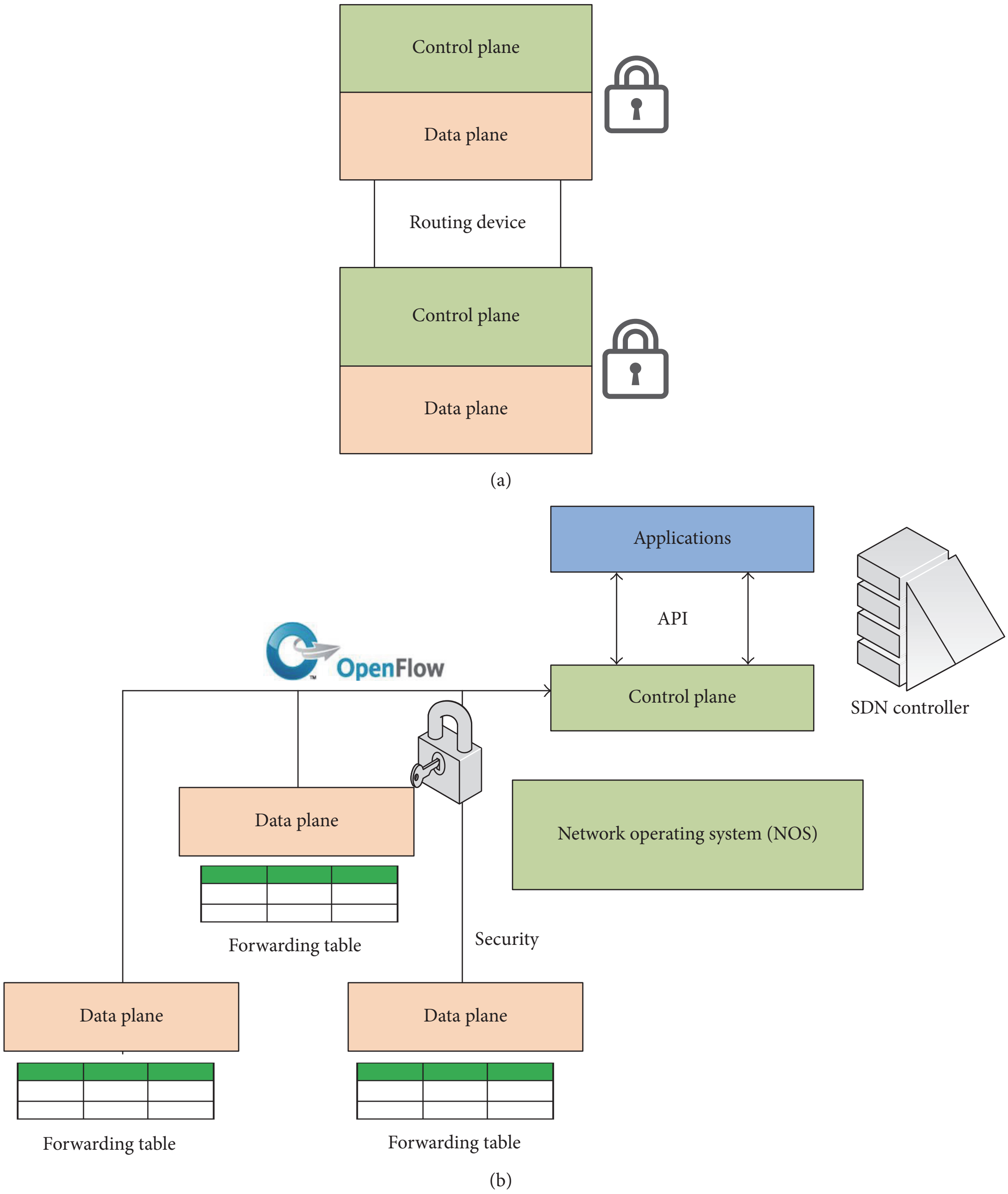


Figure 1: Comparison between (a) traditional hardware-based network and (b) SDN.

operating system (NOS) approach on different physical or virtual hosts.

Granularity. Since networking is spreading across different protocol layers and the level of data flow is aggregating as well, SDN has the features to control the traffic flow with different granularity on the protocol layers and at the aggregate level. These can vary from the core networks to a single connection in a home LAN.

Protocol Independence. SDN has a key feature called protocol independence. It helps run or control a variety of networking protocols and technologies on different SDN network layers. It also enables one to change policies from old to new technologies and supports different protocols for different applications.

Open Standard-Based. Instead of multiple vendor devices and protocols, SDN controllers simplify network operation and design based on controller instructions applied through an open standard.

Ability of Dynamic Control. SDN has the ability to modify the network traffic flow dynamically. Dynamic reconfiguration

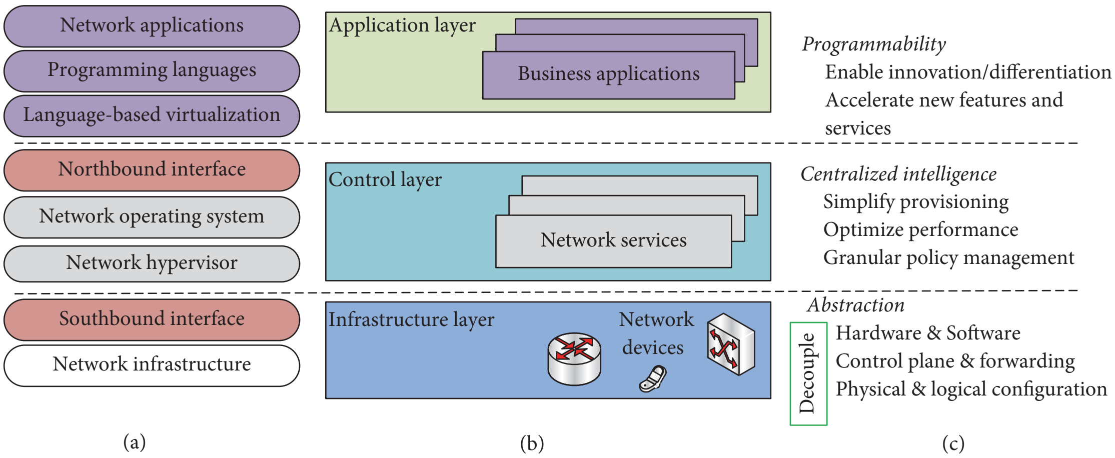


Figure 2: SDN (a) layers, (b) planes, and (c) functionalities.

covers wide-area networks, and in data center networks, where constant or continuous transportation of real or virtual machines and their network control schemes need to change in minutes or even seconds.

2.3. SDN Architecture for 5G. ONF proposed a simple high-level architecture for SDN. This model can be separated into three layers, namely, an infrastructure layer, a control layer, and an application layer, assembled over each other, as shown in Figure 2(b) [9]. These three layers are described below.

The *infrastructure layer* mainly consists of forwarding elements (e.g., physical and virtual switches, routers, wireless access points) that comprise the data plane. These devices are mainly responsible for (i) collecting network status, storing them temporally in local network devices and sending the stored data to the network controllers and (ii) for managing packets based on the rules provided by the network controllers or administrators. They allow the SDN architecture to perform packet switching and forwarding via an open interface.

The *control layer*, also known as control plane, maintains the link between the application layer and the infrastructure layer through open interfaces. Three communication interfaces allow the controller to interact with other layers, namely, the southbound interface for interacting with the infrastructure layer, northbound interface for interacting with the application layer, and east/westbound interfaces for communicating with groups of controllers. Their functions may include reporting network status and importing packet-forwarding rules and providing various service access points in various forms.

The *application layer* is designed mainly to fulfill user requirements. It consists of the end-user business applications that consume network services. SDN applications are able to control and access switching devices at the data layer through the control plane interfaces. SDN applications include network visualization, dynamic access control, security, mobility and migration, cloud computing, and load balancing (LB). Figure 3 shows the overall architecture of SDN for the 5G mobile system. The details of SDN layers are explained below.

2.3.1. Infrastructure Layer. The underlying infrastructure layer in SDN consists of switching devices that are interconnected to communicate in a single physical network. In SDN, these forwarding devices are generally represented as basic forwarding hardware or device. These devices are connected wirelessly, using optical fibers, optical wires, cloud networks, and so forth. They maintain connection with the controller through an open interface known as the southbound interface. In most SDNs, OpenFlow is used as the open southbound interface. OpenFlow is a flow-oriented protocol and has switches and port abstraction for flow control.

OpenFlow. The OpenFlow protocol maintained by ONF [19] is a fundamental element for developing SDN solutions and can be treated as an encouraging consideration of any networking abstraction. OpenFlow, the first leading authorized communications interface linking the forwarding and controls layers of the SDN architecture, allows manipulation and control of the forwarding plane of network devices (e.g., switches and routers) both physically and virtually. OpenFlow helps SDN architecture to adapt to the high-bandwidth, dynamic nature of user applications, adjust the network to different business needs, and interestingly reduce management and maintenance complexity. Figure 4(a) shows the model of the OpenFlow protocol whereas the algorithm is shown in Figure 4(b). When a new flow or packet reaches, some lookup manner originates in the primary lookup table and concludes either with a match in the flow tables or with an error depending on the rules specified by the controller. When the packets do not acknowledge what to do with a distinct incoming packet, default information to forward the packet to the controller is “send to controller” in the case of any unmatched entry. If a link or port change is triggered, event-based messages are sent by forwarding devices to the controller.

Once the rules are matched with the flow rules, the rule’s counter is incremented and actions based on the set rules start getting executed. This could lead to forwarding of a packet, after modifying some of its header fields to a specific port or (i) dropping of the packet and (ii) reporting of the packet back

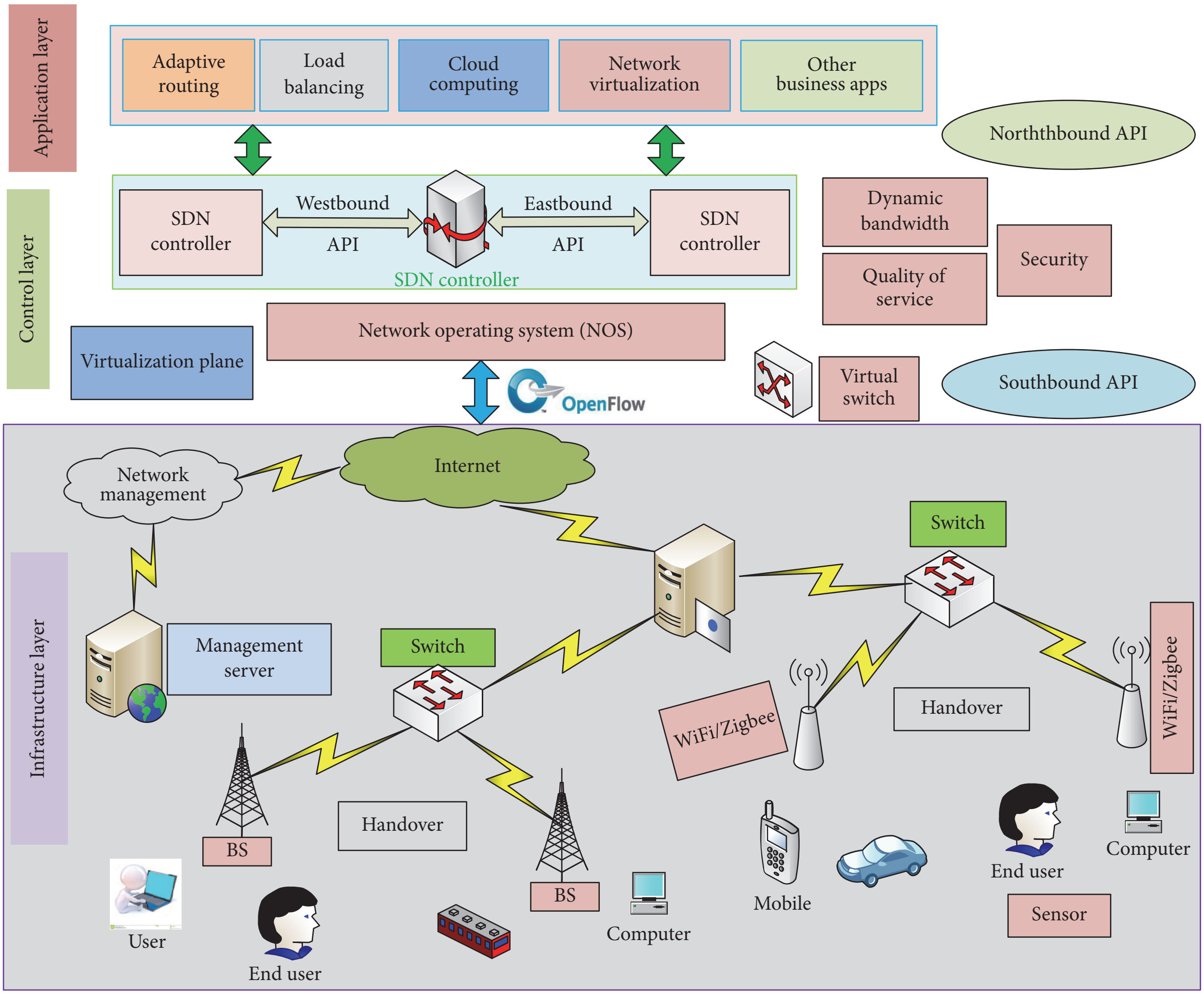


Figure 3: SDN architecture for 5G.

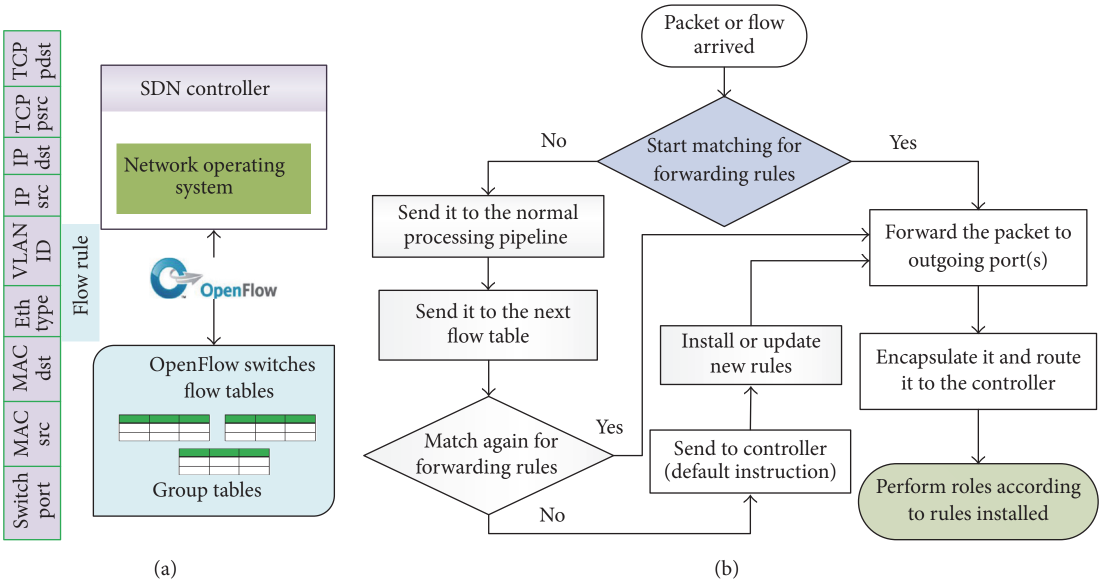

TCP pdst
TCP psrc
IP dst
IP src
VLAN ID
Eth type
MAC dst
MAC src
Switch port
Flow rule
SDN controller
Network operating system
OpenFlow
OpenFlow switches flow tables
Group tables
(a)
Packet or flow arrived
Start matching for forwarding rules
No
Yes
Send it to the normal processing pipeline
Send it to the next flow table
Match again for forwarding rules
Yes
No
Forward the packet to outgoing port(s)
Install or update new rules
Send to controller (default instruction)
Encapsulate it and route it to the controller
Perform roles according to rules installed
(b)

Figure 4: (a) OpenFlow model and (b) detailed process of OpenFlow protocol.

to the controller. The summary some of the most significant characteristics of the data plane [20]. However, OpenFlow is not the only available southbound interface for SDNs. There are other API proposals such as Forwarding and Control Element Separation (ForCES) [21]; Open vSwitch Database (OVSDB) [22]; Protocol-oblivious Forwarding (POF) [23, 24]; OpFlex [25]; OpenState [26]; Revised OpenFlow Library (ROFL) [27]; Hardware Abstraction Layer (HAL) [28, 29]; and Programmable Abstraction of Data path (PAD) [30].

2.3.2. Network Controller or Network Operating System (NOS). The network controller, SDN controller or NOS, is the heart of SDN architecture. It lies between network devices and applications. It is based on operating systems in computing. In [31], the controller is described as software abstraction that controls all functionalities of any networking system. It maintains control over the network through three interfaces, namely, southbound interface (e.g., OpenFlow), northbound interface (e.g., API), and east/westbound interfaces. The southbound interface abstracts the functionalities of programmable switches and connects them to the controller. The northbound interface [32] allows high-level policies or network applications to be deployed easily and transmits them to the NOS, while the east/westbound interfaces maintain communications between groups of SDN controllers. Thus far, many SDN controllers have been proposed by researchers to facilitate controller functionalities. For example, NOX [33] is the first, publicly available OpenFlow controller implementation that can run in Windows, Linux, Mac OS, and other platforms; an extension of NOX has been implemented in POX [34], which is a real Python-based controller; a Java-based controller implementation is called Beacon [35], while Floodlight controller is an extension of Beacon [36], and so on.

The *functionalities* of an SDN controller can be classified into four categories: (i) a high-level language for SDN applications to define their network operation policies; (ii) a rule update process to install rules generated from those policies; (iii) a network status collection process to gather network infrastructure information; and (iv) a network status synchronization process to build a global network view using the network statuses collected by each individual controller.

(1) One of the fundamental functions of the SDN controller is to translate application specifications into packet-forwarding rules. This function advances a protocol to address communication between its application layer and control layer. Therefore, it is imperative to realize some high-level languages (e.g., C++, Java, and Python) for the development of applications between the interface and the controllers.

(2) An SDN controller is accountable for generating packet-forwarding rules as well as describing the policies perfectly and installing the rules into relevant devices. Meanwhile, the forwarding rules should be updated with policy changes. Furthermore, the controller should maintain consistency for packet-forwarding by using either the original rule set/updated rule set or by using the updated rules after the update process is completed.

(3) SDN controllers accumulate network status to provide a global view of the entire network to the application layer. The network status includes duration time, packet number, data size, and flow bandwidth. A helpful and commonly employed method for network statistics data collection is Traffic Matrix I [37]. TM controls the volume of all traffic data that passes through all sources and destinations in any network.

(4) Unauthorized control of the centralized controller can degrade controller performance. Generally, this can be overcome by maintaining a consistent global view of all controllers. Moreover, SDN applications play a significant part in ensuring application simplicity and guaranteeing network consistency.

2.3.3. Application Layer. As shown in Figure 3, the application layer is located at the top layer of the SDN architecture. SDN application interacts with the controller through the northbound interface to achieve an unambiguous network function in order to fulfill the network operators' requirements. They request network services or user requirements and then manipulate these services. Although there is a well-defined standardized southbound interface such as OpenFlow, there is no standard northbound interface for interactions between controllers and SDN applications. Therefore, we can say that the northbound interface is a set of software-defined APIs, not a protocol. SDN applications can provide a global network view with instantaneous status through northbound APIs. We can categorize SDN applications according to their related basic network functionality or domain including QoS, security, traffic engineering (TE), and network management. However, several SDN applications can be developed for specific use cases in a given environment.

2.4. Applications. SDN can modify the network configuration according to user requirements. To justify the advantages of the SDN architecture, in this survey, we present a few SDN applications.

Wireless and Mobile. In a wireless sensor network, SDN provides benefits such as flexibility, optimized resource allocation, and easier management. The SDN controller permits sensor nodes to support multiple applications as they have the flexibility to set any new policies or rules. In [38], SDN in ad hoc networks has been deployed to apply the concepts of abstraction to wireless ad hoc networks for smartphones. This SDN-enabled mobile infrastructure has been implemented in the Android operating system that is more secure and easier for modification and extension.

Load Balancing (LB). LB is an important technique for online resources management to control the data flows from different applications in order to keep the link utilization at its lowest level. Moreover, the choice of an appropriate link is very important to enhance service functionality, increase throughput, avoid network overloading, minimize cost, and reduce response time. In [39], an OpenFlow-based load balancing solution is presented. When using SDN technologies, load balancing can be integrated using the OpenFlow switch, thus avoiding the need for a separate device. Moreover, SDN allows load balancing to operate on any flow granularity.

Network Management. It is reported that more than 60% of network failure happens due to human configuration [40] errors and failure in order to provide an automated and comprehensive network management system. SDN provides an abstract view of the entire network, which makes network management more flexible and automated. In SDN, a network is managed from a centralized controller based on controller flow tables and flow rules that are distributed throughout the network through its interfaces, which ensures a more flexible, granular management [41].

Network Security. In traditional networks, firewalls or proxy servers are used to protect the physical network. SDN uses a centralized architecture to deal with network security issues. SDN supervision of flows across the entire network and monitoring of user behavior allows SDN architecture to detect and prevent damage. If attacks are detected, the SDN controller can install packet-forwarding rules in the underlying switching devices to successfully prevent the attack from entering and propagating in the network [42]. One of the problems of SDN for attack detection in the case of high network traffic is that the flow tables are not sufficient to support the high-traffic flow. Therefore, in [43], a solution in the form of a real-time security system has been proposed.

Multimedia and QoS. Existing network architecture is based on end-to-end data transmission, but not supported for multimedia traffic (e.g., video streaming, video conferencing, and video on demand) though in case of real-time transmission, it requires high levels of efficiency and quality with tolerable delay and error rate. According to studies by CISCO, IP video traffic will increase from 67 percent in 2014 to 80 percent by 2019 [44]. SDN provides greater QoS by effectively selecting the optimized path among all available paths. In [45], authors proposed enhancement or optimization methods for improving end-to-end multimedia QoS over SDN.

Monitoring and Measurement. The control application needs to monitor the link constantly in terms of latency and bandwidth to optimize data flow provisioning. SDN allows a network to perform certain monitoring operations without any additional hardware or other overheads because an SDN inherently collects information about the entire network to maintain a global network view through a logically centralized controller.

2.5. Challenges and Future Direction in 5G.

Cellular network technologies have experienced explosive evolution during the last decade. Moreover, the number of mobile devices and the data traffic are increasing exponentially because network applications are extending from the traditional hardware-based to real-time communication in social networks, e-commerce, and entertainment. However, hardware-based cellular systems depend on insecure and inflexible network architectures which generally take a typical 10-year for a new generation of wireless networks to be standardized and deployed. Nowadays, the most emerging cellular network system is 5G [46]. But 5G has some challenges to think about [47]. In particular, the requirements for the 5G network system are high data rates (targeting 1 Gbps experienced users everywhere), ultra high capacity should be 1000-fold capacity/km^2, cost, a massive number of connections, and E2E latency should be less than 1 ms over the RAN. To facilitate these challenges of current network architectures, the most important need is to shift the design of current architectures for the next generation wireless networks. Moreover, the complementary concept of SDN, NFV has been presented to effectively separate the control functionalities from the hardware by simply decoupling the forwarding plane from the control plane. These functionalities will ensure the required flexibilities and adaptability of the ever-changing cellular network architectures with the introduction of the concept of SDN for 5G.

Though we presented the advantages of SDN for 5G, SDN was confronted with some challenges. First of all, security is a more challenging task that needs to be available everywhere within the SDN architecture because of (i) architecture and its controller, applications, devices, channels (TLS with plain text) and flow table, (ii) connected resources, (iii) services (to protect availability), and (iv) information. Furthermore, a reliable and balance controller is still out of scope because of lacking of robust and reliable framework policy. The framework policy should be very simple to maintain and implement, secure, and cost effective. An integration of SDN with NFV can be a new category for security deployment by decoupling control plane from forwarding plane or switching devices. In addition to security, link and controller availability, reliability, flexibility, controllers, and applications compatibility are considerable concerns. A centralized controller is not as fast as it is supposed to be though it can recover itself through a backup flows checking [48].

Operational, maintenance, and fixed costs are also another challenging topics for the deployment of 5G. The expenses can increase to reduce system blockage and maintain the availability though integration of SDN and NFV can reduce expenses. As a fully automated system with a centralized controller, SDN offers reduced human control and error free and fast configuration [48]. Despite these challenges, some remaining implementation affairs need to acknowledge such as flow tables and their large number of flow entries, flow level programming and controller programming, flow instructions, and actions.

NFV and SDN are independent and complementary to each other. But they can provide an open environment to fasten the innovation and can be easily integrated with new services and infrastructure like controlled and automated network resources. This combination can easily manage resources using its centralized controller. Packet-forwarding or processing is performed by NFV, while the controller can

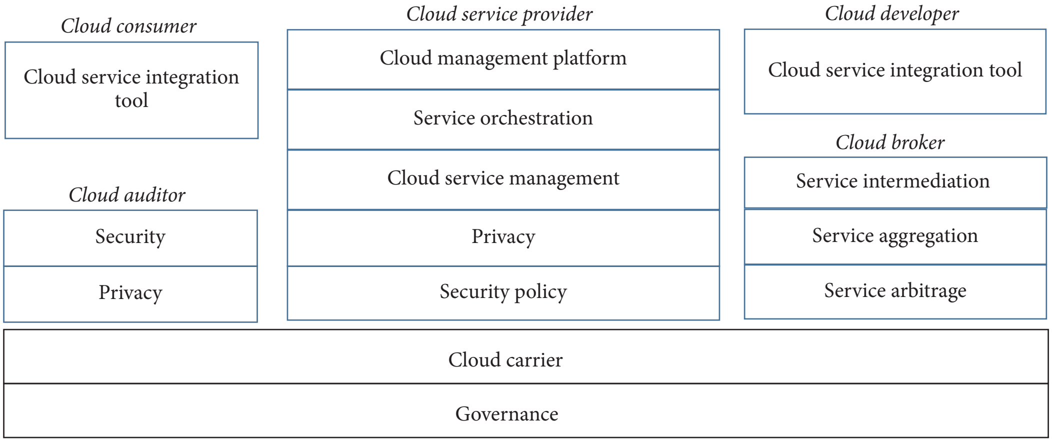


Figure 5: Cloud computing interface architecture.

control or update flow tables according to the needs of users or applications at any level. Here, NFV is responsible for creating or processing flow rules, and SDN is responsible for the management of the said rules. Integration of SDN and NFV will be a promising technology for 5G.

3. Cloud Computing

3.1. Cloud Computing for 5G Network. The main characteristics of 5G include high speed, low latency, and high capacity to support various real-time multimedia applications. 5G is being developed as a smart wireless network architecture using new models such as SDN or NFV for multidimensional massive data processing [49]. Network virtualization is a new concept that can create a big challenge for next generation networking based on IP networks, the Internet, and wireless technology. Virtualization creates connections between the communication and the computing domains. Service-Oriented Architecture (SOA) will be the main factor of network-as-a-service (NaaS), which is enabled by the convergence of networking and cloud computing. Network virtualization architecture with SOA has attracted wide interest from both the academia and the industry. However, some issues related to user requirements of QoS and QoE remain. There are more and more services considering cloud computing as the core backbone technology for service deployment and network implementation due to the scalability and flexibility. Cloud service is an important technology for the future, as it can reduce costs for service provider and customer by efficient resource allocation. Cloud computing becomes one of important reference architectures for 5G network due to the high data rate, high mobility, and centralization management services. It can be operated without direct installation consumers' systems. Cloud computing has been considered increasingly by both the academia and the industry. Due to development of technology and business trend of mobile service, the number of mobile devices is increasing more and more. The world is switching to compact devices with limited computing power, cloud computing will be the future of consumer technology. To provide a common protocol of management and operation for cloud service deployment, a complete and precise standard architecture is required. The researches on cloud computing issues are being carried out based on many cloud computing projects. They are attempting to standardize solutions related to architecture, operation, authentication, service, and cooperation integration of 5G network. The general interface architecture of cloud computing from proposals is selected from Figure 5. Based on the architecture and network function, cloud computing research can be classified by topology framework, architecture and service.

For topology frameworks, "CloudAudit: Automated Audit, Assertion, Assessment, and assurance" [50] was officially launched in 2010. It offers a cloud computing service architecture based on open, extensible, and secure interface and methodology. Cloud Standards Customer Council [51] deals with standards, security, and interoperability issues. Cloud Storage Initiative [52] discussed storage issues in cloud services by considering the adoption of cloud storage as a new delivery model. OASIS Identity in the cloud (IDCloud) [53] works on open standards for identity deployment, provisioning, and management in cloud computing. OpenStack [54] provided an open-source software API for private clouds. "Cloud Computing Interoperability Forum" [55], Open cloud Consortium [56] focused on cloud integration framework. MCC [57] proposed a new framework for 5G cloud computing by enhancing the traditional MCC architecture to satisfy the requirement of QoE in emotion-aware applications. It has three main components: mobile terminal, local cloudlet, and remote cloud. The proposed system can support the latest technological advances of 5G with computation-intensive affective computing, big data analysis, resource cognition-based emotion-aware feedback, and optimization of resource allocation under dynamic traffic load.

For service architecture, open cloud frameworks such as platform-as-a-service (PaaS) and infrastructure-as-a-service (IaaS) have been proposed by CloudFoundry [58] and DeltaCloud [52]. Open cloud computing Interface [59] provided interfaces for cloud resource management, including computing, storage, and bandwidth. Cloud Security Alliance [60] and Distributed Management Task Force (DMTF) [61] deal with cloud computing security. Open Data Center Alliance [62] develops usage models for cloud vendors with long-term data centers. The proposed cloud computing architecture in [63] uses an actor-based structure. It comprises six major actors: cloud consumer, cloud provider, cloud developer, cloud broker, cloud auditor, and cloud carrier. The actors have their own activities, requirements, and responsibilities. The associated cloud services are classified into four different groups: IaaS, PaaS, software as a service (SaaS), and anything as a service (XaaS).

For standard reference architecture, Standards Acceleration to Jumpstart Adoption of cloud computing [64] and The Open Group cloud Work Group [60] are examples of the use case creation of cloud computing standards. TM Forum cloud Services Initiative [65] suggested approaches to increase cloud computing adoption on different network service. CloudCommons [66] evaluated cloud service business performance based on the provided set of service measurement index (SMI). The proposed architecture focused mainly on commercial models, service functions, measurement of service-users preferences, and satisfaction indexes. Trusted Cloud Initiative (TCI) reference architecture was proposed by Cloud Security Alliance (CSA) in 2011 [67]. TCI uses four frameworks to define its security polity, namely, the Sherwood Business Security Architecture (SABSA), Information Technology Infrastructure (ITIL), the Open Group Architecture Framework (TOGAF), and Jericho. The proposed architecture includes methodology and supporting tools for security configuration, enterprise architecture, business plan, and risk management. The business requirements are based on different standards of control matrixes, payment, authentication, planning, design, and service development. The architecture is complex from the viewpoints of implementation and deployment because it combines several different frameworks, thus requiring developers to understand all frameworks. The standard of cloud computing is still an open issue because different industries have different precise definitions based on their own architectures [63]. However, the National Institute of Standard and Technology (NIST) [68] and IBM [69] are two typical cloud computing architectures that have been applied as references by both the industry and the academia. NIST introduced research on cloud computing architecture in September 2011 by suggesting a reference architecture including the main elements of cloud computing. The proposed architecture provides categories of functions, activities, and classification methods based on a tree structure. It describes the general concepts of technical functions and business models. The service management functions in the architecture require background knowledge. The architecture should have additional explanation and operational description for nonbackground users [63]. The detailed architecture proposed by IBM's research team is called Cloud Computing Reference Architecture (CCRA), and it is based on customer's demands of IBM's cloud products and services. Experience and research pertaining to cloud services were applied to devise a full cloud computing architecture. Compared with NIST, IBM CCRA has important advantages in terms of operation and management. System performance and scalability are considered based on the customer's cloud computing environment. Another strong point is the support system. IBM provides specific development and management tools to help customers deploy and manage their cloud services.

Cloud computing creates a new paradigm in networking technology with the concept of computing resource sharing. It can provide ubiquitous on-demand access with high flexibility, cost efficiency, and centralized management. Cloud computing has attracted considerable attention from and has had an impact on the ICT community. An increasing number of critical applications and services now support cloud computing architecture. Cloud computing is a promising architecture for future-generation networks. The distributed, dynamic, and heterogeneous characteristics of resource management are the main difference between cloud computing and the traditional service model, where the available architecture of the traditional network cannot adapt to new features. The resources used in cloud computing have different features. Hence, with the static QoS index strategy, system performance is not efficient.

3.2. Challenging Issues and Future Directions in 5G.

Cloud computing creates a new paradigm for networking technology with the concept of computing sharing and distributing resource. It can provide ubiquitous on-demand access with high flexibility, cost efficiency, and centralized management. Cloud computing has attracted considerable attention from many researchers and organization. With important contributions on architecture, cloud computing has had an impact on the ICT community. There are more and more critical applications and services which support cloud computing architecture. It will be promising architecture for future-generation of networking. Compare with traditional network and service architecture, cloud computing has advantages on distributed, dynamic, and heterogeneous characteristics of resource management. With the development of technology on semiconductor and human on demand, the traditional network architecture shows some limitations on mobility functions which cannot follow new features especially the static QoS index strategy.

With the advantage of higher capacity and powerful accessibility, 5G will be an enhanced technology with full on-demand mobile applications and services. In addition, it gains from the development of other services such as social networks, wearable devices, IoT, and cloud computing. Traditional network applications will be more human-centric on demand. A QoS model that can be configured dynamically based on the description of the required resource QoS is introduced in 5G by using three models: series, parallel, and hybrid [70]. The service architecture of cloud computing generally is categorized into three classes: SaaS, IaaS, and PaaS. The layered structure of cloud computing services is shown in Figure 6. SaaS includes applications such as Google Apps, Salesforce, and Microsoft Office 365. IaaS includes applications such as Amazon cloud Formation, Google Compute Engine, and Rackspace cloud. This service model defines the

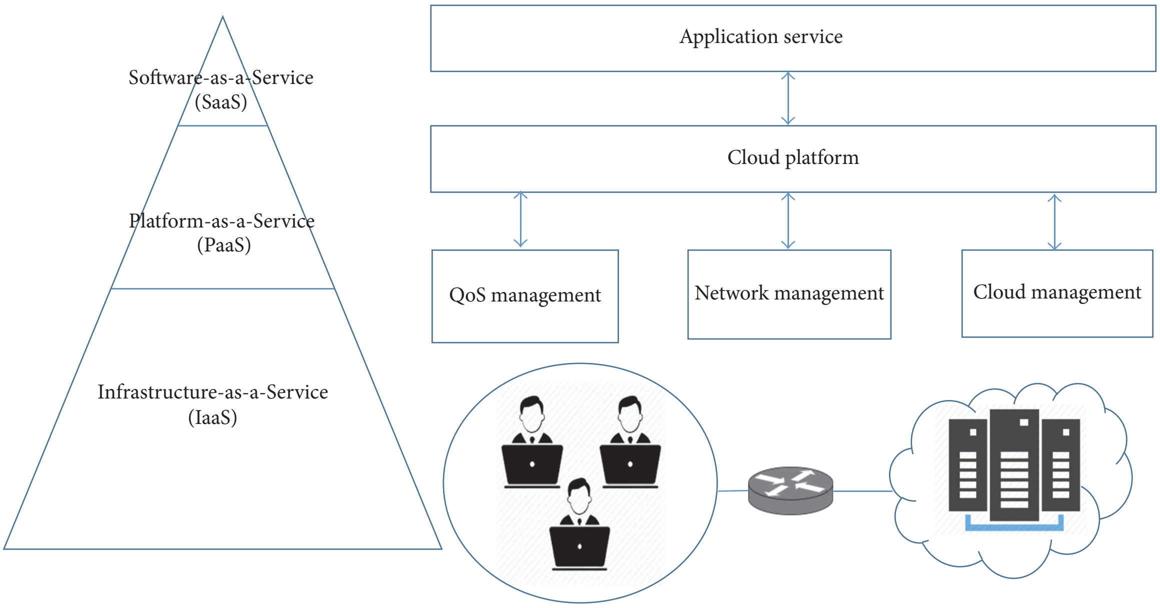


Figure 6: Cloud computing service layers.

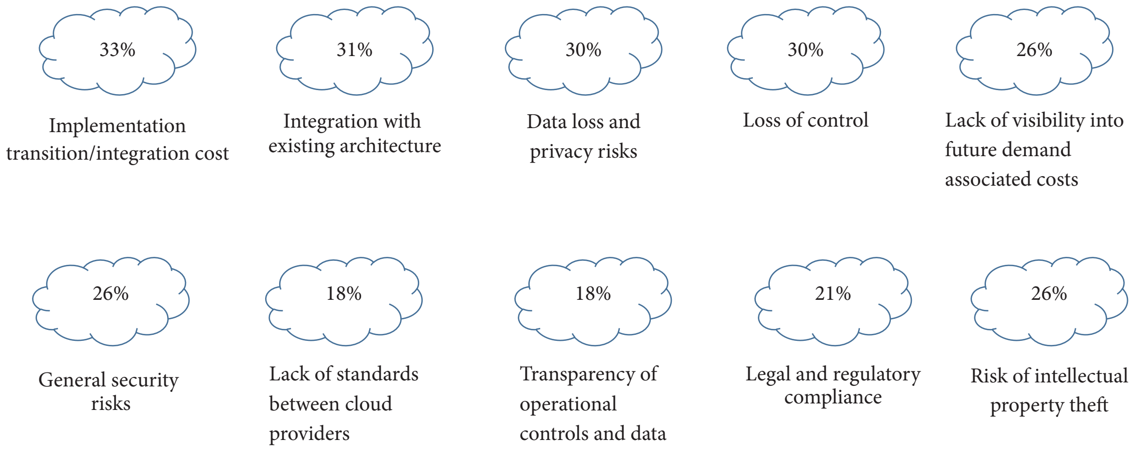


Figure 7: Challenges of loud computing service shifting.

service for users of servers and storage. IaaS supports users with an interface for virtual management and storage. PaaS includes applications such as Google App Engine, Microsoft Azure, and Amazon Elastic Beanstalk. Platform-as-a-Service provides access APIs, programming languages, middleware, and framework, which can be developed according to a user's applications without installing or configuring the operation environment. One of the most daunting challenges faced when working with PaaS is ensuring compatibility because there are no common features, API database type tools, and software architectures across various PaaS architectures.

Besides the advantages of cloud computing service from traditional network architecture, there also remain some certain challenges during the cloud shifting. Figure 7 shows the considerations of adopting cloud computing from implementation challenge survey by KPMG [71]. For 5G network, the challenge issues of cloud computing are considered as Security and Privacy, Quality of service, access time and Accessibility, Data access control, and transition to the cloud. About Security and Privacy issue, they are the most concerns for service providers when moving their data to the cloud. Although Security and Privacy issue in most cloud architectures is generally designed with high reliable and proficient model, it must show a fully secure scenario with different levels and strategies for customer's trust. Secondly, the Quality of service will be considered strongly before moving from traditional network architecture to cloud architecture. The cloud customer needs assurance that the business's data will be safe and available and reliable at all times. The performance of cloud infrastructure can be affected by the load, environment, number of users, and connection link technology. There should be some backup mechanisms to guarantee the data access. The research issues for access time and accessibility are related to infrastructure

Table 2: Leading organizations, institutions, and forums involved in IoT standardization.

International organizations	Regional and national organizations	Global standards collaboration	Forums	Clusters
(i) ITU-T [75–77] (ii) IEEE [74, 78–81] (iii) ISO [82] (iv) IEC (v) ISO/IEC JTC1 [83, 84]	(i) ETSI [85, 86] (ii) CEN (iii) CCSA (iv) ARIB (v) TIA (vi) TTA (vii) TTC (viii) GISFI	(i) MSTF	(i) oneM2M [87] (ii) W3C (iii) 3GPP (iv) NFC (v) ECMA (vi) IoT Forum (vii) Ipv6 Forum	(i) IERC

of cloud architecture. With data access control issue, there are some questions on the reliability of system such as backup strategies, storage structure, and security of data access. The transition from traditional network to cloud is important time of cloud customer. The first step in transitioning to the cloud is being able to identify the challenges and working conditions with cloud provider to navigate the barriers of cloud business model.

4. Internet of Things (IoT) for 5G

4.1. IoT Definition. IoT is a dynamic network of connected devices. The idea is to connect not only things but also people any time, any place, with anything and anyone, and so on. The definition of IoT has crossed the boundaries of traditional network. The International Telecommunications Union (ITU) has codified the concept of IoT [72] as the following definition:

> *"IoT is a global infrastructure for the information society, enabling advanced services by interconnecting (physical and virtual) things based on existing and evolving interoperable information and communication technologies".*

However, IoT has become hugely popular over the last decade. The dimensions and the scopes of IoT can be any Thing, any Place, any Time, any Body, and so on. Consequently, standardization is being demanded to establish interoperability among things with a view of transforming the world into a global village. The standardization efforts undertaken by different organizations and institutes are explained below.

4.2. Standardization Effort. Standards control any system to operate under fixed rules and regulation. Interoperability among the disciplines of any reference system depends on standards. Worldwide, numerous standardization authorities have initiated the creation of relevant standards during the last decade. Nevertheless, these efforts have had no impact in terms of unifying the standards into a single framework because IoT has become the storehouse of anything. The list of different organizations, institutions, and groups engaged in IoT standardization is given in Table 2 [73]. A few persuasive IoT-related standards by IEEE are listed in Table 3 [74].

4.3. IoT Architecture. The future is approaching a new paradigm of networks with huge numbers of devices. The idea of 5G (beyond 4G) refers to networks with improved QoS, huge capacity, enhanced data rate, and, overall, a feasible architecture to sustain the aforementioned features. The influential parts of 5G networks include D2D communication, which can be interpreted as the idea of IoT. IoT comprises the technologies of smart sensors, RFID, machine-to-machine (M2M), IP, communication systems, and so on. This part of the paper focuses on the different emerging IoT architectures suitable for future-generation 5G networks.

IoT architecture has evolved with the evolution of the Internet. The first phase of IoT evolution entailed communication among several computers through a computer network. However, the World Wide Web (WWW) was launched in 1991 to connect all computers worldwide [89, 90]. Further technological advances have connected the users of various types of electronic devices with computers under the same platform by connecting to the cloud network [91]. Finally, the idea of IoT was conceived to give shape to the world by connecting everything. IoT is the network that can adopt and connect anything that anyone can imagine [92].

IoT architectures can be classified into several types because it is absolutely difficult to merge the architectures proposed for various IoT applications into a single model [93]. A scheme for classifying IoT architectures is shown in Figure 8. Several authors have proposed three-layer-based simple IoT architectures that comprise an application layer, a network layer, and perception layer [94, 95]. Middleware-based IoT architectures consist of a greater number of layers, including the coordinate layer next to the middleware layer [96, 97]. In addition, the perception layer has a shared option for combining other edge technology and the access layer [98, 99]. Service-oriented architecture (SOA) has different layers, unlike middleware-based architecture. It has five layers, namely, the objects layer, object abstraction layer, service management layer, service composition layer, and application layer [100, 101]. However, common IoT networks have an architecture comprising five fundamental layers, namely, the objects layer, object abstraction layer, service management layer, application layer, and management layer.

The first and foremost layer is the *objects layer*, which is similar to the perception layer that embodies physical devices, and an IoT architecture might contain heterogeneous devices in the network. Next is the *object abstraction layer*,

Table 3: Standardization efforts for IoT by different groups of IEEE.

Group	Title of the Standardization Group
IEEE 802.11-2012	IEEE Standard for Information Technology–Telecommunications and information exchange between systems–Local and metropolitan area networks–Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications Amendment 10: Mesh Networking
IEEE 802.15.4-2011	IEEE Standard for Local and metropolitan area networks–Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs)
IEEE 802.15.4g-2012	IEEE Standard for Local and metropolitan area networks–Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs) Amendment 3: Physical Layer (PHY) Specifications for Low-Data-Rate, Wireless, Smart Metering Utility Networks
IEEE 802.15.7-2011	IEEE Standard for Local and Metropolitan Area Networks–Part 15.7: Short-Range Wireless Optical Communication Using Visible Light
IEEE 802.16-2012	IEEE Standard for Air Interface for Broadband Wireless Access Systems
IEEE 802.16p-2012	IEEE Standard for Wireless MAN-Advanced Air Interface for Broadband Wireless Access Systems – Amendment: Enhancements to Support Machine-to-Machine Applications
IEEE 802.16.1b-2012	IEEE Standard for Wireless MAN-Advanced Air Interface for Broadband Wireless Access Systems – Amendment: Enhancements to Support Machine-to-Machine Applications
IEEE 1609.11-2010	IEEE Standard for Wireless Access in Vehicular Environments (WAVE)–Over-the-Air Electronic Payment Data Exchange Protocol for Intelligent Transportation Systems (ITS)
IEEE 1888-2011	IEEE Standard for Ubiquitous Green Community Control Network Protocol
IEEE 1901-2010	IEEE Standard for Broadband over Power Line Networks: Medium Access Control and Physical Layer Specifications
IEEE 1905.1-2013	IEEE Draft Standard for a Convergent Digital Home Network for Heterogeneous Technologies
IEEE 11073-10103-2013	IEEE Standard for Health informatics – Point-of-care medical device communication – Nomenclature – Implantable device, cardiac

which is used for the conveying the data generated by the devices [102]. Various technologies are used for data transfer, for instance, the 5G network uses RFID, WiFi, Bluetooth, UWB, and ZigBee. Cloud computing technology, too, is deployed in this layer. Then, there is the *service management layer*, and it is responsible for application programmer management, which entails ensuring compatibility with any hardware platform by processing the generated data.

The *application layer* affords customers with services as requested. This layer includes different types of services such as smart city [103], smart wearable device [104], smart vehicle [105], smart home [106], smart healthcare [107], and industrial automation [108].

The self-characteristics of IoT should cope with the emerging future-generation 5G networks. The basic infrastructure of IoT has the characteristics of heterogeneous devices, resource-constrained, flexible infrastructure, dynamic network, distributed network, ultra-large-scale network, large number of events, spontaneous interaction, location awareness, and intelligence [109].

However, the idea of the future-generation 5G network possesses the characteristics of IoT networks. The simplified architecture of the 5G network is shown in Figure 9 to demonstrate that the emerging IoT architecture can deal with it. It is assumed that the 5G network might connect 50 billion devices to the cloud by 2020. The 5G network will cause a 10–100x increase in the number of devices, 10–100x increase in data rate, 1000x increase in data volume, 10x increase in device battery life, and 5x decrease in latency. The 5G network embraces some important technologies such as radio access, MIMO, mobility management, interference management, and massive spectrum to achieve compatibility with the IoT networks included in the METIS project. To handle different issues with these technologies, some mechanisms have been proposed in the METIS project [110]. D2D communications is one of the proposed methods that helps maintain an ultra-large-scale network using flexible infrastructure. However, massive machine communication (MMC), another solution, is the base of IoT in terms of interconnecting a huge number of devices across different smart technologies. Furthermore, moving network (MN), ultra-dense network (UDN), and ultra-reliable network (URN) are some other proposed solutions for mobility management, interference mitigation, capacity achievement, and so on.

Several studies on different types of services for IoT have been and are being carried out. A few vital architectures of the enabling technologies for the future-generation 5G network have been surveyed. M2M-based communication architecture for cognitive radio network has been demonstrated for IoT [111–113]. The architecture of the M2M network in IoT is shown in Figure 10. The relationship between the infrastructure layer and the application layer is maintained by the network layer, which is mainly a communication network. The infrastructure layer includes the M2M devices and gateway, while the application layer comprises the users, management interface, and the M2M server.

The M2M server is the core of the architecture, and it integrates the overall system for the required services such as traffic management, smart healthcare systems, and so forth. However, the object database (DB) also sends user information preserved in the DB by users in the form of SMS, email, video, and so on. The IMS server is connected to the M2M server, which is situated in the network layer. The GPRS server and gateway are the main components of the network layer that help the IMS server by collecting vehicle location in the system. The application layer and the network layer are connected through the Internet, while the infrastructure layer and the network layer are linked together by the gateway or some protocol. M2M devices update the M2M server through the network layer by using the information of M2M devices (i.e., body sensors, smart devices, other devices). The user interface provides access to the user and manages user

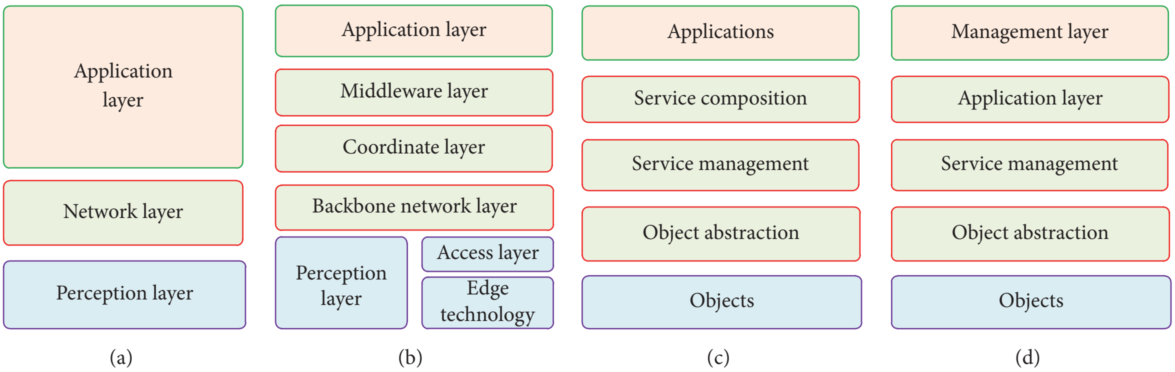


Figure 8: Categorized IoT architectures. (a) Three-layer-based simple architecture, (b) middleware-layer-based architecture, (c) service-oriented architecture (SOA) for IoT, and (d) five-layer architecture (adopted from [16]).

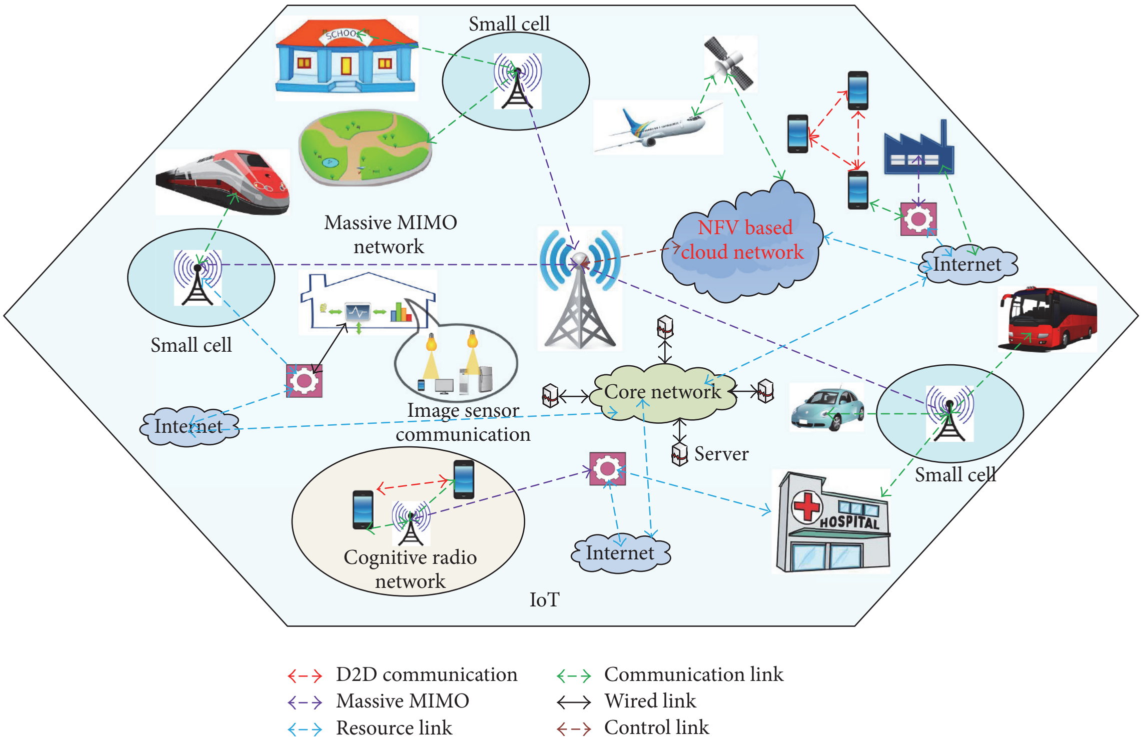


Figure 9: Simplified future-generation cellular 5G networks.

data. The network layer, which consists of a WPAN/WLAN network and an IoT gateway, connects the user interface with the management interface. The management interface monitors user data, takes initiatives as necessary, and informs the respective body about the situation [114–116].

4.4. Applications. IoT has become the source of many applications owing to its incredible potential and has given rise to numerous new application fields. It has brought revolutionary changes to our everyday life. It has connected everything, everyplace, and everybody to an inseparable framework, which has shrunken burdens many times over multiple associated systems. However, the applications of IoT are indescribable because IoT can contribute to almost every sector. Hence, the impact of IoT has surpassed our social and economic life and has entered our personal life. Figure 11 shows a summary of the numerous IoT applications. Note that it is quite impossible to outline all the applications in a frame. We describe a few the influential IoT applications below.

Smart Home. The idea of automation in home management and surveillance has brought personal life under the supervision of the IoT platform. Home appliances can be controlled from a remote place using IoT technology [117, 118]. Furthermore, human-machine interaction for the smart home environment is a new inclusion in IoT [92, 119, 120]. Wearable

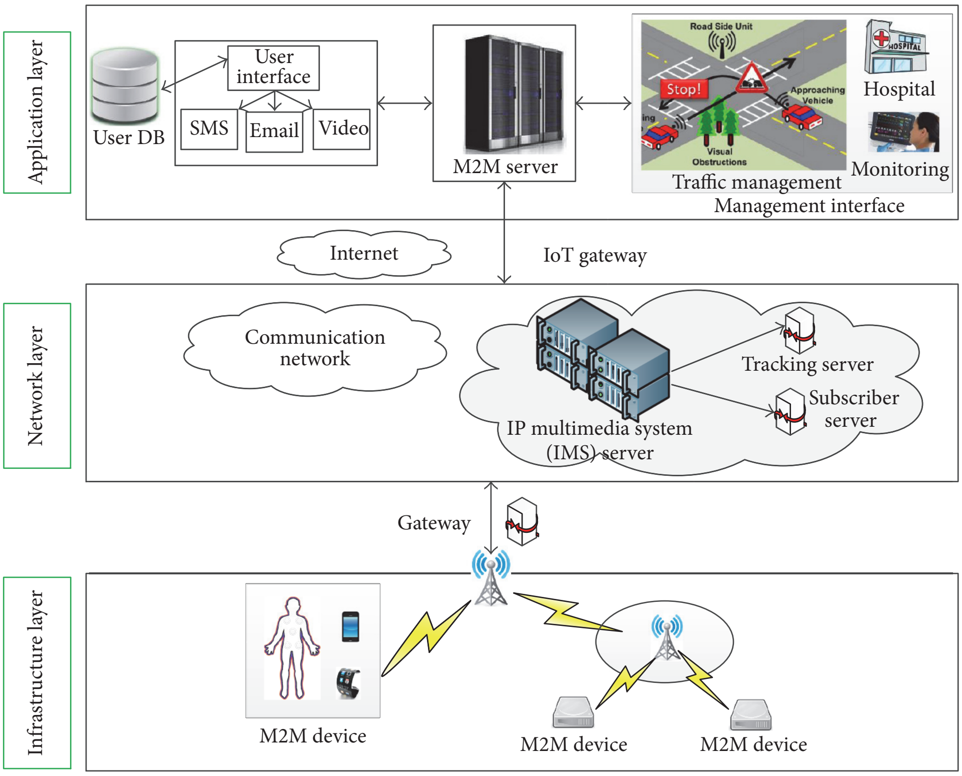


FIGURE 10: Generalized M2M network architecture in IoT.

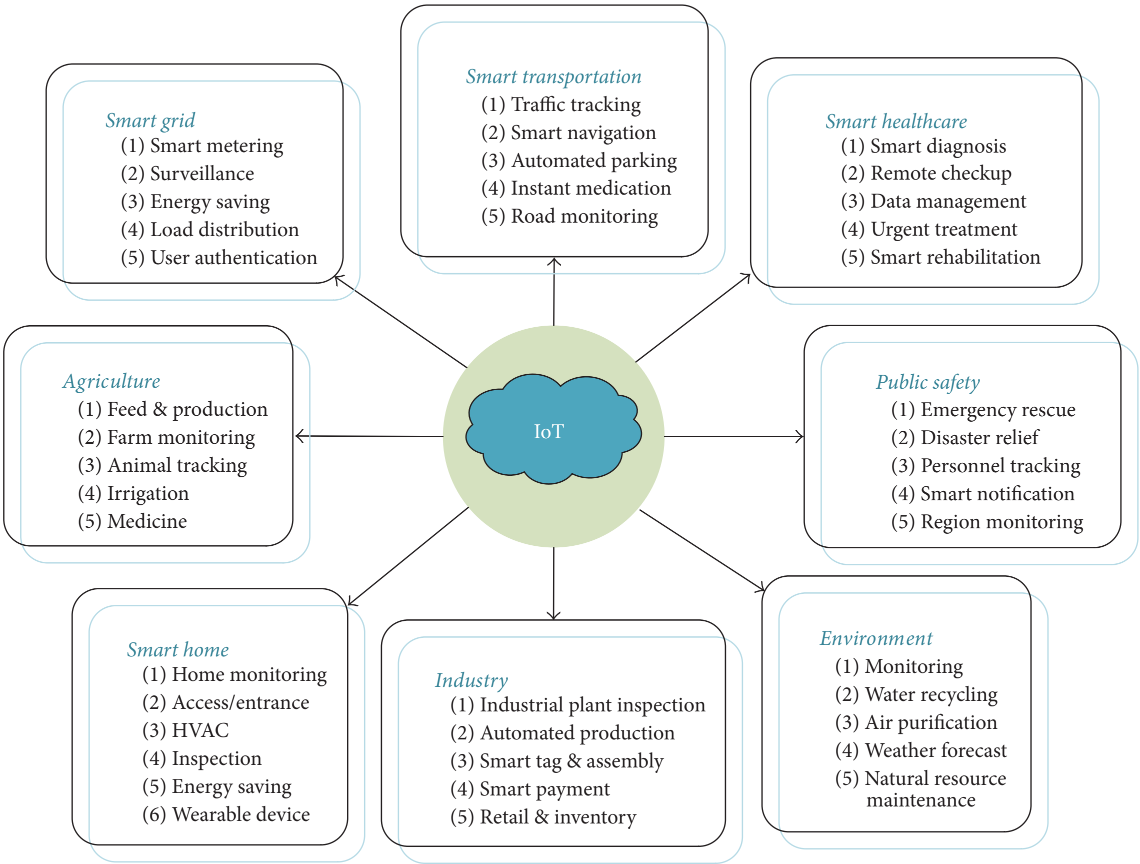


FIGURE 11: Prospective applications of IoT.

devices can be helpful in human-machine interaction and for home monitoring [121]. The environment of a smart home can be controlled effectively using the smart home technology in IoT [106].

Smart Grid. The energy consumption and distribution of a power plant and grid can be controlled by connecting them to IoT. The use of communication technology in the smart grid system establishes a link between user and system, helps broadcast urgent information to the customer, facilitates automatic device control, and so on [122]. However, the smart grid is closely related to the smart home system and can help reduce the energy consumed by such a home [123, 124].

Smart Transportation. IoT has brought of the idea of smart city to ameliorate the provision of basic services. Vehicle tracking is one of the promising applications of IoT that can lessen traffic congestion, enhance safety, schedule traffic time and smart vehicle parking, send information to travelers, and so on [125–127]. Cloud computing, the catalyst for IoT, has added a lot to the smart management and automation of smart parking systems [128].

Smart Healthcare. IoT has flourished smart healthcare systems by interconnecting various devices [129]. Smart healthcare systems cover healthcare records [130], remote prescription [131], patient observation [132], urgent treatment [133], smart diagnosis [16], and so on. Smart rehabilitation can be a part of a smart healthcare system to support the elderly and the disabled to get medication [134].

Public Safety. Safety and security are the most essential issues in a smart city. IoT contributes to emergency alarm systems, weather forecasts, disaster management, and emergency evacuation, and so on [135–137]. Moreover, intelligent algorithms and camera based system have progressed the safety system in a distinguished way [138].

Agriculture. Agriculture includes planting, farming, breeding, and animal rearing, and it has come under the purview of IoT lately [139]. The monitoring of farms, tracking of animals, and irrigation are the main aspects of IoT for agriculture [140, 141]. Moreover, feeding, vaccination, medication, rearing, and so on are vital applications of IoT in the agriculture sector [142].

Industry. The integration of different enterprises in the industry has brought about a radical change in this sector [143]. Automated monitoring systems for CO2, poisonous gases, and other gases can be among the great applications of IoT [144]. In addition, RFID and wireless sensors can be potential fields for automation in industries [145].

Environment. A physical environment may be arranged with smart devices, and the automation of homes, industries, traffic, and healthcare can be connected to physical entities to build a better world [146]. Disaster management, weather forecasting, and emergency alarm service are among some of the essential applications of IoT [147]. However, the concept of green IoT technology has been established to create a smarter environment [148, 149].

4.5. Challenges and Future Direction

4.5.1. Challenges. The definition of IoT has the significance of the challenges for 5G. The pivotal features of IoT such as heterogeneity, secure communications, system protocols, and so forth have commenced different challenges for IoT. Some of the key challenges are described below briefly.

Large Scale Storage. The property of heterogeneity creates a huge demand for storage of data. Besides, various types of data need to be categorized for the simplification of computation, data generation, and processing, which enhance the necessity of storage size of data.

Computation. One of the critical challenges of IoT, which emerged as one pivotal issue, is computation. The integration of heterogeneous device and functional variance of the devices has aggrandized the computation problem. The architecture of IoT demands a reliable and scalable computational methodology.

Ubiquitous Protocol Design. The architecture of IoT initializes a common platform for the devices with the different working mechanism. The D2D communication, an essential feature for IoT, produces a challenge for the IoT of 5G to build a pervasive protocol for connecting the heterogeneous devices. Every device connecting to each other should maintain a common protocol to make computation process simpler and to bring the characteristics of scalability.

Security and Privacy. Security, as well as privacy, has transformed into a major challenge for IoT. As cloud computing is used in IoT for storage of data, security has become one of the primary concern for the virtual storage. Moreover, lacking personal privacy between numerous devices has made the problem more critical because the establishment of personal privacy at each layer of the IoT architecture requires computing power constraints [128].

Reliability. Reliability arises as one of the major concerns recently. Due to the connectivity of everything, reliability in certain sectors can be defined as the most serious challenge. Public health such as emergency operation, critical treatment for diseases, smart transportation, and smart home are some of those types areas of IoT in which reliability plays a vital role.

Performance. Performance of IoT varies according to the several activities of the different layer of IoT. Particular functions in IoT need highly assured performance and QoS. Traffic mobility, real-time connections, and emergency services are some of them that have inaugurated the challenge of performance in IoT.

4.5.2. Future Direction. The introduction of cloud computing, big data, SDN, and so forth in the IoT area has initialized

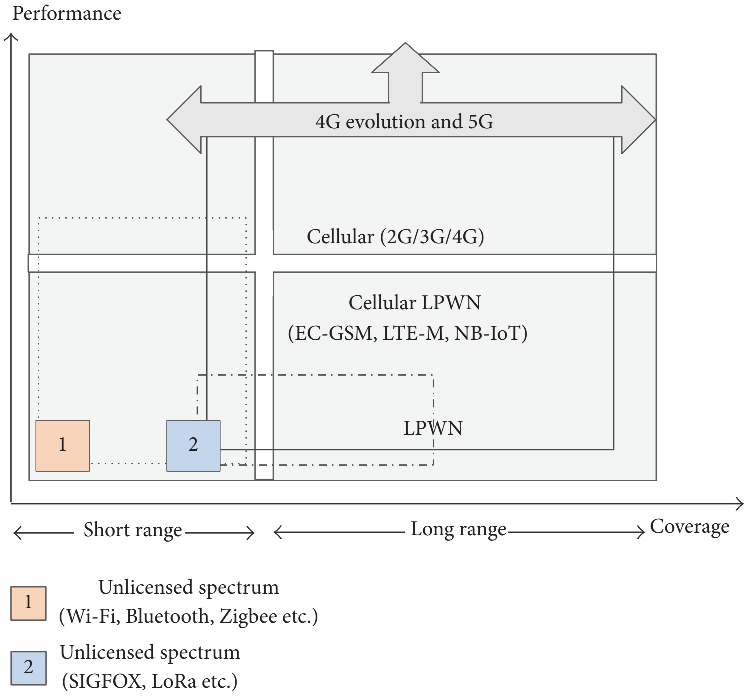


Figure 12: Technologies addressing different M2M segments [17].

a new era of research. The standardization effort of IoT has also imposed an influence on the future research of IoT. The methodology of integration of several devices of the different mechanism is one of the significant issues for future studies. Several system architectures have been proposed to make IoT architecture viable, but all of the systems have been proposed on the different platform. It is a matter of concern how all of the systems could work on the same platform with massive computational performance and energy efficiency. Security, privacy, and reliability are the next candidates for the common platform. Consequently, the integration architecture of big data, cloud computing, and SDN for IoT would be the most important issue for the future direction of research.

However, several other aspects of IoT can also be important parts of future research direction. Intelligent system for real-time data collection and processing in the IoT architecture could be one of the important research directions. Establishment of individual small social networks to work for different devices and combination of these networks to create a single platform for IoT could be another future direction for the research area. Besides, scalability, reliability, and flexibility are some other issues for future direction.

5. Mobile Access Networks for 5G

5.1. M2M (Machine-to-Machine) Communication. Legacy cellular networks have been developed to support high data rate and reliable communication. M2M environments are, however, very different from cellular networks, because low data rates and long latencies are desirable. The basic purpose of M2M communication is to transmit sensed data of small size with loose time constraints. To meet the characteristics of M2M communication, there are two categories of Radio Access Technologies (RATs) according to spectrum resources: cellular IoT and lower-powered wide-area network (LPWN). A classification of cellular IoT technologies is shown in Figure 12.

Cellular IoT involves modifying the legacy cellular network to accommodate IoT communication using licensed bands. The third-generation project partnership (3GPP) standardized long-term evolution machine-to-machine (LTE-M), which optimized the IoT protocol over the LTE system since Release 12 [150]. LTE-M reuses LTE PHY channels. LTE-M includes coverage enhancement, cost reduction, and improved battery life. Furthermore, it is able to cooperate within the legacy LTE network. However, it has a limitation in fulfilling all requirements of IoT communication because the nature of the LTE system is not suited for low data rates and long-range communication.

Therefore, 3GPP is currently studying and standardizing a narrow band radio interface called narrow band (NB) IoT. This technology started as a clean state standard to fulfill the requirements of IoT environments. NB-IoT reuses LTE core networks; thus, rapid deployment is possible in the market with only software modifications. In addition, NB-IoT supports various operation modes including in-band, guard-band, and standalone. NB-IoT requires only a narrowband carrier of 200 kHz with frequency division multiple access (FDMA) in uplink and orthogonal frequency division multiple access (OFDMA) in the downlink for 200,000 connections [88].

One of the benefits of using unlicensed spectrum is the ability to deploy a new service regardless of whether the

Table 4: Comparison of various RATs for IoT connectivity [88].

RAT	SIGFOX	LoRA	C-IoT	NB LTE-M R13	LTE-M R12/13	5G
Range	<13 km, 160 dB	<11 km, 157 dB	<15 km, 164 dB	<15 km, 164 dB	<11 km, 156 dB	<15 km, 164 dB
Spectrum Bandwidth	Unlicensed 900 MHz 100 Hz	Unlicensed 900 MHz, <500 kHz	Licensed 7–900 MHz <200 kHz or dedicated	Licensed 7–900 MHz <200 kHz or shared	Licensed 7–900 MHz <1.4 MHz or shared	Licensed 7–900 MHz shared
Data rate	<100 bps	<10 kbps	<50 kbps	<150 kbps	<1 Mbps	<1 Mbps
Battery life	>10 years	>10 years	>10 years	>10 years	>10 years	>10 years
Availability	Today	Today	2016	2016	2016	Beyond 2020

service provider is an Internet service provider (ISP). Such solutions include SIGFOX [151] and Long Range (LoRa) [152]. Table 4 shows a comparison of RATs for IoT in terms of transmission range, bandwidth, data rate, battery life, and availability. SIGFOX is growing rapidly in Europe. The main target of SIGFOX is ultra-low-end sensor systems with limited throughput demands (12 bytes per 1.6 sec frame, 140 transmissions per day). SIGFOX uses the 100 Hz ultra-narrow band and basic modulation of binary phase shift keying (BPSK). It has a unique feature that all data from devices are transmitted to the SIGFOX cloud through an SIGFOX gateway and any service provider can access the data using the SIGFOX open application programming interface (API). Thus, all communication services depend highly on the SIGFOX itself because of this architecture.

LoRa is being standardized by the LoRa alliance since 2015. It has a communication range of 10 miles and low power consumption, resulting in a maximum battery life of 10 years. The LoRa network architecture uses general frequency shift keying (GFSK) or LoRa modulation with the star-of-stars topology and a LoRa gateway, which relays data between an end device and the core network. Each device can establish multiple connections with two or more gateways. The main difference from SIGFOX is that LoRa follows the open ecosystem policy of the LoRa alliance. LoRa uses a narrow band of 125 kHz and transmits a payload of 50 bytes with chirp spread spectrum, which is similar to CDMA. LoRa also provides adaptive data rate (ADR) to improve power management and data rate simultaneously by dynamically adjusting the data rate and transmission power based on the analytic result of packet error rate, signal-to-noise ratio (SNR), and received signal strength indicator (RSSI).

5.2. Device-to-Device (D2D) Communication. Device-to-device (D2D) communication in cellular networks is an emerging technology that enables direct communication between user equipment (UE) with little or no help from the infrastructure such as eNodeB or core networks. D2D communication provides several advantages in terms of spectrum efficiency, power management, coverage expansion, and capacity improvement by reusing radio resources and allowing network functionalities to devices. Furthermore, D2D communication enables new services such as public safety services, location-based commercial proximity services, and traffic offloading [153]. Owing to these benefits, D2D communication is considered one of the key techniques. D2D communication can be classified into three types based on intervention from infrastructure with network control: autonomous D2D, network-assisted D2D, and network-controlled D2D.

In autonomous D2D, devices in the network work in a fully distributed manner to communicate and establish links with each other. It is similar to ad hoc or peer-to-peer (P2P) networking. Each device or cluster head handles all network functionalities, similar to self-organizing networks. Thus, this mode is suitable for disaster networks or public safety services because devices can communicate without any infrastructure. In the case of network-assisted D2D, the infrastructure supports some network functions including link management, synchronization, and security. The devices in the network basically construct a self-organizing network and retain control over D2D communication. The infrastructure mediates network nodes to improve network efficiency by reducing the control signaling overhead. In the case of network-controlled D2D, the infrastructure strongly controls the network from radio resource management to data communication. When the network is fully centralized, all D2D devices are allowed only for data communication. It is close to the legacy cellular mode.

We can categorize D2D communication types into in-band D2D and out-band D2D as well, regarding spectrum resources [154]. In in-band communication, cellular and D2D devices share the same spectrum band by reusing radio resources (underlay) or using dedicated resources (overlay). The advantage of this type of communication is that the infrastructure can have a high-level of control over the cellular spectrum, but there is additional interference from D2D communication to cellular communication, which requires an additional computation procedure for resource allocation, resulting in some overhead. In out-band D2D communication, different spectra from a cellular network (i.e., industrial, scientific, and medical (ISM) bands) are used; therefore, there is no interference between D2D and cellular communications. Because of this characteristic, D2D and cellular networks communicate simultaneously without any interruption. However, D2D devices may suffer from other network entities that access the ISM band, and QoS is lower compared to in-band D2D owing to the nature of unlicensed bands, limited transmission range, and low data rate.

3GPP started standard activities in Release 12 to enable D2D communication in LTE networks [155]. The D2D standard consists of two parts: device discovery and data communication. The purpose of device discovery is to find other neighboring devices within the transmission range for communication. There are two types of device discovery: type 1 and type 2 (2A and 2B). Type 1 is a collision-based procedure with non-UE-specific allocation. Thus, devices randomly select their radio resources for device discovery in every discovery period. In type 2, the network schedules discovery signal transmission for all UE. In particular, type 2B uses semipersistent allocation only for radio resource control-(RRC-) connected UE with predefined frequency hopping. Similarly, there are two data communication modes: mode 1 and mode 2. In the case of mode 1, an eNodeB allocates data resources to all UE. In particular, in the in-coverage scenario, eNodeB follows the same resource allocation procedure as the cellular mode. In the case of mode 2, a UE assigns its resources from the preconfigured resource pool autonomously. Thus, UE can select and communicate even in out-coverage or partial-coverage scenarios.

5.3. V2X Communication. The advent of autonomous cars, high-traffic information systems, and highly reliable safety services has led to the need for a new communication technology for vehicles with high reliability, high data rate, and low latency. This technology is called vehicle-to-everything (V2X) communication, and it includes vehicle-to-vehicle (V2V), vehicle-to-pedestrian (V2P), and vehicle-to-infrastructure (V2I) communication [156]. D2D communication for cellular networks is currently the most suitable option for enabling V2X communication because D2D provides short end-to-end latency and a long transmission range. When V2X communication is deployed in cellular networks, network deployment cost can be reduced and deployment time can be shortened by reusing the infrastructure of a legacy cellular network [18].

To enable road safety services and autonomous driving, vehicles need to exchange several pieces of information. ETSI defines message types for these use cases: cooperative awareness message (CAM) [157] and decentralized environmental notification message (DENM) [158]. CAM is defined for periodical broadcast of short messages to nearby vehicles. This message type delivers presence, position (i.e., GPS information), identifier, and basic status. DENM is transmitted when a specific event such as an accident or abnormal situation occurs to warn neighbor nodes of the event.

V2X communication is currently under standardization by 3GPP since 2015. SA1 defines use cases and service requirements, and SA2 studies network architecture to support use cases in-vehicle networks. RAN is working on radio resource management to satisfy the requirements of vehicular networks. In the US, system requirements for an on-board V2V safety service have been standardized in SAE J2945/1 [159]. ETSI documented Release 1 for a cooperative intelligent safety service message set and Release 2 for urban ITS applications.

V2X communication must support road safety services with high mobility. Therefore, it has very strict requirements compared to other communication technologies to enable highly reliable services. In vehicular networks, all vehicle nodes are moving at high speeds. Thus, local information becomes meaningless rather quickly. Furthermore, V2X communication guarantees the transmission of safety messages and maintains connectivity with neighbors to satisfy reliability requirements. Vehicles should react as fast as possible to accidents. To this end, ultra-low end-to-end delay is required, and a faster duty cycle is needed for communication and device discovery. Figure 13 shows an overview of the bandwidth and latency requirements for V2X communication.

The key performance indicators are as follows [18]:

(i) E2E delay: 10~100 ms

(ii) Reliability: 10^{-5}

(iii) Positioning accuracy: 30 cm

(iv) Data rate: 10~40 Mbit/s

5.4. Challenging Issues and Future Directions in 5G Mobile Access Networks

5.4.1. M2M Communication. The LTE standard was originally targeted to human-to-human (H2H) communication. The M2M communication in cellular IoT should support small data transmission with an irregular time interval. Current radio resource blocks are too large for M2M communication. Thus, new radio resource management schemes are needed to fulfill these requirements of M2M communication. If the M2M mechanism shares radio resource blocks with H2H communication, we will have to minimize impact and interference from M2M communication to H2H communication. The interference management among M2M devices is also a dormant concern due to the massive number of devices in M2M networks.

In M2M communication, cost efficiency is the most important factor owing to the large number of devices deployed in the network. Efficient power management to maximize battery life and network operation is also very crucial. The network capacity should be adequate to handle the massive number of simultaneous connection requests over a wide coverage area. The M2M communication should have capabilities to manage diverse use cases of IoT services. Also, data aggregation and data offloading concepts can be applied to M2M communication to enhance energy efficiency and communication.

5.4.2. D2D Communication. When applying D2D communication to legacy cellular networks, we need to manage the interference caused by D2D communication to minimize performance degradation of the cellular network. The network carefully handles the random mobility of UEs and random channel status for interference management. A power control mechanism is mainly used to coordinate the interference in

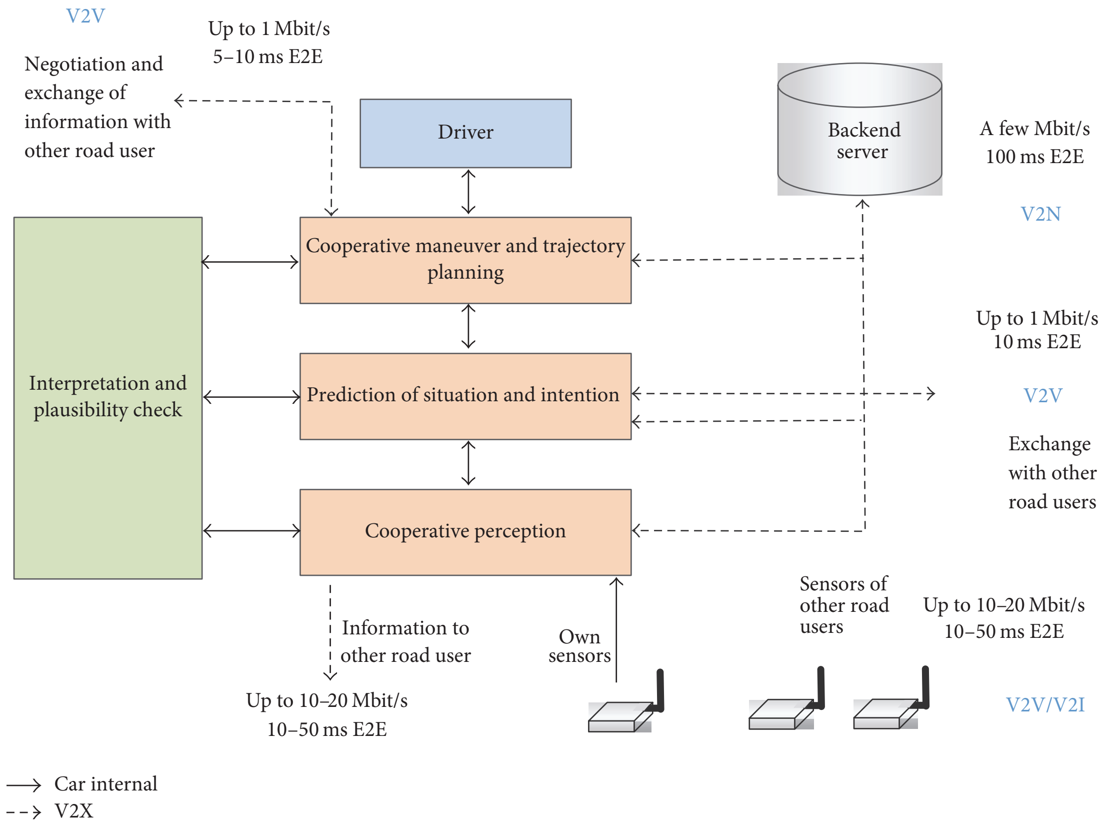

V2V
Up to 1 Mbit/s
5–10 ms E2E
Negotiation and exchange of information with other road user
Driver
Backend server
A few Mbit/s
100 ms E2E
V2N
Cooperative maneuver and trajectory planning
Interpretation and plausibility check
Prediction of situation and intention
Up to 1 Mbit/s
10 ms E2E
V2V
Exchange with other road users
Cooperative perception
Information to other road user
Up to 10–20 Mbit/s
10–50 ms E2E
Own sensors
Sensors of other road users
Up to 10–20 Mbit/s
10–50 ms E2E
V2V/V2I
Car internal
V2X

Figure 13: Connectivity demands of future connected vehicles [18].

D2D communication. Power control can be achieved either in a centralized way by eNodeB or in a distributed way by each UE. Fundamentally, a centralized approach is more efficient because all the network information including SNR and signaling power of UE is known, but information about additional control and computational overhead is needed to gather network information and adjust the signaling power of all UEs appropriately.

The most challenging issue in D2D communication is interference management. Without efficient interference management, D2D communication cannot coexist with a cellular network. The other issue is mode selection. For example, when the network allows D2D mode, the questions are which event triggers the mode selection procedure, what kind of parameter is used, and whether to use D2D. Data offloading needs to be studied further to increase network capacity and coverage expansion by a relay and power management.

5.4.3. V2X Communication. As mentioned earlier, cellular D2D communication can be an enabling technology for V2X communication. However, legacy D2D communication is unable to fulfill the requirements of V2X communication because of its limitations. D2D communication has a collision risk during side-link transmission. Moreover, there are many unsuitable mechanisms leading to latency that is too long for V2X communication, such as long duty cycles for device discovery, inefficient resource allocation and link adaptation, and slow connection setup procedure.

Thus, further studies are needed to apply D2D to V2X. The most important issue is reducing end-to-end latency. Network-assisted radio and link management can be a solution to improve the delay performance. Legacy device discovery mechanisms are too slow for vehicle networks. Thus, new faster device discovery mechanisms should be developed. In addition, a new protocol with a lower duty cycle is needed to minimize the latency and the network control mechanism used in D2D communication should be optimized to reduce control signaling overhead and interference. Additional study issues are supporting flexible retransmission and advanced collision resolution to guarantee reliable requirement and constructing a robust network by increasing link connectivity.

6. Conclusion

The expectation of future mobile system or next generation wireless networks comprises high-speed access providing without limitation of time and location. As a consequence, the NGN has to deal with the high data rate, real-time data handling, centralized views of the entire network with minimum delay, greater security, fewer data losses, and less error rate. The development of any technologies with high data traffic and high QoS of universal network infrastructures depends on the integration of new technologies or new services with the existing network infrastructure. In this survey, we have discussed the network architecture, service framework, and topologies that will play an important role to meet the requirements of future networking infrastructure that is 5G network. The requirement of 5G will be massive IoT connectivity, virtual experience and media, and real-time communication. So, the architecture of 5G will be such that the flexibility and scalability of the future network will be maximized. Therefore, the future network will depend on the combination of new technologies such as cloud computing, SDN, NFV, and E2E networking infrastructure. Besides, the integration of SDN with NFV will ensure dynamic data control, centralized network provisioning, and adaptation of new services and innovation. To the best of our knowledge, the survey of promising technologies for 5G networks has emphasized an absolute idea on the interesting attempts in network development trend based on standardization status. Along with this, the promising architectures and services of the SDN, cloud computing, and IoT have been provided. However, the contents presented in this survey are the first step toward the potential architectures and implementation works for 5G network and are not the end picture. After all, the realization of future network for 5G needs a lot of efforts in the research laboratories, industries, and companies.

Competing Interests

The authors declare that they have no competing interests.

Acknowledgments

This research was supported by the MSIP (Ministry of Science, ICT and Future Planning), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2016-R2718-16-0004) supervised by the IITP (National IT Industry Promotion Agency).

References

[1] P. Pirinen, "A brief overview of 5G research activities," in *Proceedings of the 1st International Conference on 5G for Ubiquitous Connectivity (5GU '14)*, pp. 17–22, November 2014.

[2] W. E. Dong, W. Nan, and L. Xu, "QoS-oriented monitoring model of cloud computing resources availability," in *Proceedings of the International Conference on Computational and Information Sciences (ICCIS '13)*, pp. 1537–1540, Hubai, China, June 2013.

[3] A. Aissioui, A. Ksentini, A. M. Gueroui, and T. Taleb, "Toward elastic distributed SDN/NFV controller for 5G mobile cloud management systems," *IEEE Access*, vol. 3, pp. 2055–2064, 2015.

[4] H. Wu, L. Hamdi, and N. Mahe, "TANGO: a flexible mobility-enabled architecture for online and offline mobile enterprise applications," in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC '14)*, pp. 2982–2987, Istanbul, Turkey, April 2014.

[5] W. Xia, Y. Wen, C. H. Foh, D. Niyato, and H. Xie, "A survey on software-defined networking," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 1, pp. 27–51, 2015.

[6] B. A. A. Nunes, M. Mendonca, X.-N. Nguyen, K. Obraczka, and T. Turletti, "A survey of software-defined networking: past, present, and future of programmable networks," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 3, pp. 1617–1634, 2014.

[7] R. Horvath, D. Nedbal, and M. Stieninger, "A literature review on challenges and effects of software defined networking," *Procedia Computer Science*, vol. 64, pp. 552–561, 2015.

[8] I. F. Akyildiz, P. Wang, and S.-C. Lin, "SoftAir: a software defined networking architecture for 5G wireless systems," *Computer Networks*, vol. 85, pp. 1–18, 2015.

[9] Open Networking Foundation (ONF), https://www.opennetworking.org.

[10] Open Networking Foundation (ONF), *OpenFlow-Enable SDN and Network Function Virtualization*, 2014.

[11] N. McKeown, T. Anderson, H. Balakrishnan et al., "OpenFlow: enabling innovation in campus networks," *ACM SIGCOMM Computer Communication Review*, vol. 38, no. 2, pp. 69–74, 2008.

[12] F. Hu, Q. Hao, and K. Bao, "A survey on software-defined network and OpenFlow: from concept to implementation," *IEEE Communications Surveys and Tutorials*, vol. 16, no. 4, pp. 2181–2206, 2014.

[13] L. I. B. López, A. L. V. Caraguay, L. J. G. Villalba, and D. López, "Trends on virtualisation with software defined networking and network function virtualisation," *IET Networks*, vol. 4, no. 5, pp. 255–263, 2015.

[14] T. Wood, K. K. Ramakrishnan, J. Hwang, G. Liu, and W. Zhang, "Toward a software-based network: integrating software defined networking and network function virtualization," *IEEE Network*, vol. 29, no. 3, pp. 36–41, 2015.

[15] "7 Advantages of Software Defined Networking," http://www.ingrammicroadvisor.com/data-center/7-advantages-of-software-defined-networking.

[16] N. L. Geller, D.-Y. Kim, and X. Tian, "Smart technology in lung disease clinical trials," *Chest*, vol. 149, no. 1, pp. 22–26, 2016.

[17] Ericsson, "Cellular Networks for Massive IoT," Ericsson White paper [Online], 2016, http://www.ericsson.com/news/160106-cellular-networks-massive-iot_244039856_c.

[18] 5G-PPP, "5G Automotive Vision—White Paper on Automotive Vertical sector," October 2015.

[19] OpenFlow, "OpenFlow switch specification version 1.5.0," 2014, https://www.opennetworking.org/images/stories/downloads/sdn-resources/onf-specifications/openflow/openflow-switch-v1.5.0.noipr.pdf.

[20] Infinera, "Transport SDN," 2013, [Online], http://www.infinera.com/go/sdn/index.php.

[21] A. Doria, J. H. Salim, R. Haas et al., "Forwarding and control element separation (ForCES) protocol specification," Internet Engineering Task Force, 2010, http://www.ietf.org/rfc/rfc5810.txt.

[22] B. Pfaff and B. Davie, "The Open vSwitch database management protocol," Internet Engineering Task Force, RFC 7047 (Informational), https://tools.ietf.org/pdf/draft-pfaff-ovsdb-proto-02.pdf.

[23] H. Song, "Protocol-oblivious forwarding: unleash the power of SDN through a future-proof forwarding plane," in *Proceedings of the 2nd ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking (HotSDN '13)*, pp. 127–132, August 2013.

[24] H. Song, J. Gong, J. Song, and J. Yu, "Protocol oblivious forwarding (POF)," 2013, http://www.poforwarding.org.

[25] M. Smith, M. Dvorkin, Y. Laribi, V. Pandey, P. Garg, and N. Weidenbacher, "OpFlex control protocol: Internet Engineering Task

Force," Internet Draft, April 2014, http://tools.ietf.org/html/draft-smith-opflex-00.

[26] G. Bianchi, M. Bonola, A. Capone, and C. Cascone, "OpenState: programming platform-independent stateful OpenFlow applications inside the switch," *SIGCOMM Computing Communication Review*, vol. 44, no. 2, pp. 44–51, 2014.

[27] M. Sune, V. Alvarez, T. Jungel, U. Toseef, and K. Pentikousis, "An OpenFlow implementation for network processors," in *Proceedings of the 3rd European Workshop on Software Defined Networks (EWSDN '14)*, pp. 123–124, Budapest, Hungary, September 2014.

[28] D. Parniewicz, R. D. Corin, L. Ogrodowczyk et al., "Design and implementation of an OpenFlow hardware abstraction layer," in *Proceedings of the ACM SIGCOMM Workshop on Distributed Cloud Computing*, pp. 71–76, Chicago, Ill, USA, August 2014.

[29] B. Belter, D. Parniewicz, L. Ogrodowczyk et al., "Hardware abstraction layer as an SDN-enabler for non-OpenFlow network equipment," in *Proceedings of the 3rd European Workshop on Software Defined Networks (EWSDN '14)*, pp. 117–118, IEEE, Budapest, Hungary, September 2014.

[30] B. Belter, A. Binczewski, K. Dombek et al., "Programmable abstraction of datapath," in *Proceedings of the 3rd European Workshop on Software-Defined Networks (EWSDN '14)*, pp. 7–12, September 2014.

[31] N. Feamster, J. Rexford, and E. Zegura, "The road to SDN: an intellectual history of programmable networks," *ACM SIGCOMM Computer Communication Review*, vol. 11, no. 12, 2013.

[32] J. Dix, "Clarifying the role of software-defined networking northbound APIs," May 2013, http://www.networkworld.com/article/2165901/lan-wan/clarifying-the-role-of-software-defined-networking-northbound-apis.html.

[33] NOX, http://archive.openflow.org/downloads/Workshop2009/OpenFlowWorkshop-MartinCasado.pdf.

[34] POX, http://searchsdn.techtarget.com/definition/POX.

[35] D. Erickson, https://openflow.stanford.edu/display/Beacon/Home.

[36] Floodlight is an Open SDN Controller, http://www.projectfloodlight.org/floodlight/.

[37] A. Tootoonchian, M. Ghobadi, and Y. Ganjali, "OpenTM: traffic matrix estimator for OpenFlow networks," in *Proceedings of the in 11th international conference on Passive and Active Measurement*, pp. 201–210, 2010.

[38] Y. S. P. Baskett, W. Zeng, and B. Guttersohn, "SDNAN: software defined networking in Ad hoc networks of smartphones," in *Proceedings of the IEEE 10th Consumer Communications and Networking Conference (CCNC '13)*, pp. 861–862, Las Vegas, Nev, USA, January 2013.

[39] M. Jarschel and R. Pries, "An OpenFlow-based energy efficient data center approach," *ACM SIGCOMM Computer Communication Review*, vol. 42, no. 4, pp. 87–88, 2012.

[40] Z. Kerravala, "As the value of enterprise networks escalates, so does the need for configuration management," Enterprise Computing & Networking, The Yankee Group Report, Boston, Mass, USA, 2004.

[41] H. Kim and N. Feamster, "Improving network management with software defined networking," *IEEE Communications Magazine*, vol. 51, no. 2, pp. 114–119, 2013.

[42] A. D. Ferguson, A. Guha, C. Liang, R. Fonseca, and S. Krishnamurthi, "Hierarchical policies for software defined networks," in *Proceedings of the 1st ACM International Workshop on Hot Topics in Software Defined Networks (HotSDN '12)*, pp. 37–42, Helsinki, Finland, August 2012.

[43] K. Giotis, C. Argyropoulos, G. Androulidakis, D. Kalogeras, and V. Maglaris, "Combining OpenFlow and sFlow for an effective and scalable anomaly detection and mitigation mechanism on SDN environments," *Computer Networks*, vol. 62, pp. 122–136, 2014.

[44] "The Zettabyte Era: Trends and Analysis, white paper," May 2015, http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/vni-hyperconnectivity-wp.html.

[45] H. E. Egilmez, S. T. Dane, K. T. Bagci, and A. M. Tekalp, "OpenQoS: an openflow controller design for multimedia delivery with end-to-end quality of service over software-defined networks," in *Proceedings of the Signal & Information Processing Association Annual Summit and Conference (APSIPA ASC '12)*, pp. 1–8, Hollywood, Calif, USA, December 2012.

[46] A. Gupta and R. K. Jha, "A survey of 5G network: architecture and emerging technologies," *IEEE Access*, vol. 3, pp. 1206–1232, 2015.

[47] P. K. Agyapong, M. Iwamura, D. Staehle, W. Kiess, and A. Benjebbour, "Design considerations for a 5G network architecture," *IEEE Communications Magazine*, vol. 52, no. 11, pp. 65–75, 2014.

[48] A. Lara, A. Kolasani, and B. Ramamurthy, "Network innovation using open flow: a survey," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 1, pp. 493–512, 2014.

[49] D. Wübben, P. Rost, J. S. Bartelt et al., "Benefits and impact of cloud computing on 5G signal processing: flexible centralization through cloud-RAN," *IEEE Signal Processing Magazine*, vol. 31, no. 6, pp. 35–44, 2014.

[50] CloudAudit: Automated Audit, Assertion, Assessment, and Assurance, http://www.cloudaudit.org.

[51] Cloud Standards Customer Council, [Online], http://www.cloud-council.org/.

[52] "DeltaCloud," http://incubator.apache.org/deltacloud.

[53] "OASIS Identity in the Cloud (IDCloud)," http://www.oasis-open.org.

[54] OpenStack, http://www.openstack.org.

[55] G. A. Lewis, "The role of standards in cloud-computing interoperability," Technical Note, Carnegie Mellon University, 2012.

[56] Open Cloud Consortium. [Online], http://opencloudconsortium.org.

[57] M. Chen, Y. Zhang, Y. Li, S. Mao, and V. C. M. Leung, "EMC: emotion-aware mobile cloud computing in 5G," *IEEE Network*, vol. 29, no. 2, pp. 32–38, 2015.

[58] CloudFoundry. [Online], https://www.cloudfoundry.org.

[59] Open Cloud Computing Interface. [Online], http://occi-wg.org.

[60] The Open Group Cloud Work Group, https://collaboration.opengroup.org/.

[61] Distributed Management Task Force (DMTF). [Online], http://www.dmtf.org.

[62] Open Data Center Alliance. [Online], http://www.opendatacenteralliance.org.

[63] Y. Amanatullah, C. Lim, H. P. Ipung, and A. Juliandri, "Toward cloud computing reference architecture: cloud service management perspective," in *Proceedings of the International Conference on ICT for Smart Society (ICISS '13)*, Jakarta, Indonesia, June 2013.

[64] Standards Acceleration to Jumpstart Adoption of Cloud Computing. [Online], http://www.nist.gov.

[65] TM Forum Cloud Services Initiative, http://www.tmforum.org.

[66] CloudCommons, *Introducing the Service Measurement Index*, Cloud Service Measurement Initiative Consortium, 2012.

[67] J. Archer, N. Puhlmann, A. Boehme, P. Kurtz, D. Cullinane, and J. Reavis, *Quick Guide to the Reference Architecture: Trusted Cloud Initiative*, Cloud Security Alliance, 2011.

[68] F. Liu, J. Tong, J. Mao et al., "NIST cloud computing reference architecture," Special Publication 500-292, National Institute of Standards and Technology, U.S. Department of Commerce, 2011.

[69] CCRA Team and M. Buzetti, *Cloud Computing Reference Architecture 2.0: Overview*, IBM Corporation, 2011.

[70] J. B. Abdo, J. Demerjian, H. Chaouchi, K. Barbar, and G. Pujolle, "Operator centric mobile cloud architecture," in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC '14)*, pp. 2982–2987, IEEE, Istanbul, Turkey, April 2014.

[71] The Cloud Takes Shape, *Global Cloud Survey: The Implementation Challenge*, KPMG International Cooperative, 2013.

[72] International Telecommunication Union, "Next Generation Networks—frameworks and functional architecture models—overview of the Internet of things".

[73] IERC (European Research Cluster on the Internet of Things) position paper, http://www.internet-of-things-research.eu/pdf/IERC_Position_Paper_IoT_Standardization_Final.pdf.

[74] IEEE Standards Association, "Internet of Things Related Standards," http://standards.ieee.org/innovate/iot/stds.html.

[75] Internet of Things Global Standards Initiative, http://www.itu.int/en/ITU-T/gsi/iot/Pages/default.aspx.

[76] Concluded Focus Groups, http://www.itu.int/en/ITU-T/focusgroups/Pages/concluded.aspx.

[77] ITU-T Focus Group on M2M Service Layer, "M2M service layer: Requirements and architectural framework," https://www.itu.int/dms_pub/itu-t/opb/fg/T-FG-M2M-2014-D2.1-PDF-E.pdf.

[78] IEEE Standards Association, "Internet of Things," http://standards.ieee.org/innovate/iot/index.html.

[79] IEEE Standards Association, "IEEE Standards Activities in the Intelligent Transportation Systems (ITS) Space (ICT Focus)," http://standards.ieee.org/develop/msp/its.pdf.

[80] IEEE Standards Association, "IEEE Standards Activities in the Network and Information Security (NIS) Space," http://standards.ieee.org/develop/msp/nis.pdf.

[81] IEEE Standards Association, "IEEE Standards Activities in the Smart Grid Space (ICT Focus)," May 2013, http://standards.ieee.org/develop/msp/smartgrid.pdf.

[82] Wikipedia, "ISO/IEC JTC 1," http://en.wikipedia.org/wiki/ISO/IEC_JTC_1.

[83] Wikipedia, "ISO/IEC JTC1/WG7," http://en.wikipedia.org/wiki/ISO/IEC_JTC_1/WG_7.

[84] Wikipedia, "ISO/IEC JTC 1/SC 31 Automatic identification and data capture techniques," https://en.wikipedia.org/wiki/ISO/IEC_JTC_1/SC_31_Automatic_identification_and_data_capture_techniques.

[85] ETSI Technology Clusters, http://www.etsi.org/technologies-clusters.

[86] ETSI, "Work programme 2013-2014," http://www.etsi.org/images/files/WorkProgramme/etsi-work-programme-2013-2014.pdf.

[87] Terms of Reference (ToR) for Technical Committee, "Smart M2M," https://portal.etsi.org/SmartM2M/SmartM2M_ToR.asp.

[88] Nokia, "LTE-M—optimizing LTE for the Internet of Things," White Paper, Nokia, 2015, http://networks.nokia.com/file/34496/lte-m-optimizing-lte-for-the-internet-of-things.

[89] N. Olifer and V. Olifer, *Computer Networks: Principles, Technologies and Protocols for Network Design*, John Wiley & Sons, Hoboken, NJ, USA, 2005, http://au.wiley.com/WileyCDA/WileyTitle/productCd-EHEP000983.html.

[90] C. Perera, A. Zaslavsky, P. Christen, and D. Georgakopoulos, "Context aware computing for the internet of things: a survey," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 1, pp. 414–454, 2014.

[91] S. Distefano, G. Merlino, and A. Puliafito, "A utility paradigm for IoT: the sensing cloud," *Pervasive and Mobile Computing*, vol. 20, pp. 127–144, 2015.

[92] B. Guo, D. Zhang, Z. Wang, Z. Yu, and X. Zhou, "Opportunistic IoT: exploring the harmonious interaction between human and the internet of things," *Journal of Network and Computer Applications*, vol. 36, no. 6, pp. 1531–1539, 2013.

[93] D. Singh, G. Tripathi, and A. J. Jara, "A survey of Internet-of-Things: future vision, architecture, challenges and services," in *Proceedings of the IEEE World Forum on Internet of Things (WF-IoT '14)*, pp. 287–292, IEEE, Seoul, South Korea, March 2014.

[94] R. Khan, S. U. Khan, R. Zaheer, and S. Khan, "Future Internet: the Internet of things architecture, possible applications and key challenges," in *Proceedings of the 10th International Conference on Frontiers of Information Technology (FIT '12)*, pp. 257–260, IEEE, Islamabad, Pakistan, December 2012.

[95] Z. Yang, Y. Yue, Y. Yang, Y. Peng, X. Wang, and W. Liu, "Study and application on the architecture and key technologies for IOT," in *Proceedings of the 2nd International Conference on Multimedia Technology (ICMT '11)*, pp. 747–751, Hangzhou, China, July 2011.

[96] G. Eleftherakis, D. Pappas, T. Lagkas, K. Rousis, and O. Paunovski, "Architecting the IoT paradigm: a middleware for autonomous distributed sensor networks," *International Journal of Distributed Sensor Networks*, vol. 11, no. 12, Article ID 139735, 17 pages, 2015.

[97] S. Martin, G. Diaz, I. Plaza, E. Ruiz, M. Castro, and J. Peire, "State of the art of frameworks and middleware for facilitating mobile and ubiquitous learning development," *The Journal of Systems and Software*, vol. 84, no. 11, pp. 1883–1891, 2011.

[98] P. Bellavista, R. Montanari, and S. K. Das, "Mobile social networking middleware: a survey," *Pervasive and Mobile Computing*, vol. 9, no. 4, pp. 437–453, 2013.

[99] V. Raychoudhury, J. Cao, M. Kumar, and D. Zhang, "Middleware for pervasive computing: a survey," *Pervasive and Mobile Computing*, vol. 9, no. 2, pp. 177–200, 2013.

[100] X. Qiu, H. Luo, G. Xu, R. Zhong, and G. Q. Huang, "Physical assets and service sharing for IoT-enabled Supply Hub in Industrial Park (SHIP)," *International Journal of Production Economics*, vol. 159, pp. 4–15, 2015.

[101] Z. Sheng, C. Mahapatra, C. Zhu, and V. C. Leung, "Recent advances in industrial wireless sensor networks toward efficient management in IoT," *IEEE Access*, vol. 3, pp. 622–637, 2015.

[102] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of things: a survey on enabling technologies, protocols, and applications," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 4, pp. 2347–2376, 2015.

[103] A. Gaur, B. Scotney, G. Parr, and S. McClean, "Smart city architecture and its applications based on IoT," *Procedia Computer Science*, vol. 52, pp. 1089–1094, 2015.

[104] P. Castillejo, J. F. Martinez, L. Lopez, and G. Rubio, "An internet of things approach for managing smart services provided by wearable devices," *International Journal of Distributed Sensor Networks*, vol. 9, no. 2, Article ID 190813, 2013.

[105] W. He, G. Yan, and L. D. Xu, “Developing vehicular data cloud services in the IoT environment,” *IEEE Transactions on Industrial Informatics*, vol. 10, no. 2, pp. 1587–1595, 2014.

[106] S. D. T. Kelly, N. K. Suryadevara, and S. C. Mukhopadhyay, “Towards the implementation of IoT for environmental condition monitoring in homes,” *IEEE Sensors Journal*, vol. 13, no. 10, pp. 3846–3853, 2013.

[107] S. M. R. Islam, D. Kwak, M. H. Kabir, M. Hossain, and K. S. Kwak, “The internet of things for health care: a comprehensive survey,” *IEEE Access*, vol. 3, pp. 678–708, 2015.

[108] Z. Bi, L. D. Xu, and C. Wang, “Internet of things for enterprise systems of modern manufacturing,” *IEEE Transactions on Industrial Informatics*, vol. 10, no. 2, pp. 1537–1546, 2014.

[109] M. A. Razzaque, M. Milojevic-Jevric, A. Palade, and S. Clarke, “Middleware for Internet of things: a survey,” *IEEE Internet of Things Journal*, vol. 3, no. 1, pp. 70–95, 2016.

[110] A. Osseiran, F. Boccardi, V. Braun et al., “Scenarios for 5G mobile and wireless communications: the vision of the METIS project,” *IEEE Communications Magazine*, vol. 52, no. 5, pp. 26–35, 2014.

[111] A. Aijaz and A.-H. Aghvami, “PRMA-based cognitive machine-to-machine communications in smart grid networks,” *IEEE Transactions on Vehicular Technology*, vol. 64, no. 8, pp. 3608–3623, 2015.

[112] Y. Zhang, R. Yu, M. Nekovee, Y. Liu, S. Xie, and S. Gjessing, “Cognitive machine-to-machine communications: visions and potentials for the smart grid,” *IEEE Network*, vol. 26, no. 3, pp. 6–13, 2012.

[113] A. Aijaz and A. H. Aghvami, “Cognitive machine-to-machine communications for Internet-of-things: a protocol stack perspective,” *IEEE Internet of Things Journal*, vol. 2, no. 2, pp. 103–112, 2015.

[114] S. M. R. Islam, D. Kwak, M. H. Kabir, M. Hossain, and K.-S. Kwak, “The Internet of things for health care: a comprehensive survey,” *IEEE Access*, vol. 3, pp. 678–708, 2015.

[115] Y. J. Fan, Y. H. Yin, L. D. Xu, Y. Zeng, and F. Wu, “IoT-based smart rehabilitation system,” *IEEE Transactions on Industrial Informatics*, vol. 10, no. 2, pp. 1568–1577, 2014.

[116] L. Catarinucci, D. de Donno, L. Mainetti et al., “An IoT-aware architecture for smart healthcare systems,” *IEEE Internet of Things Journal*, vol. 2, no. 6, pp. 515–526, 2015.

[117] L. C. De Silva, C. Morikawa, and I. M. Petra, “State of the art of smart homes,” *Engineering Applications of Artificial Intelligence*, vol. 25, no. 7, pp. 1313–1321, 2012.

[118] C. Chen, D. J. Cook, and A. S. Crandall, “The user side of sustainability: modeling behavior and energy usage in the home,” *Pervasive and Mobile Computing*, vol. 9, no. 1, pp. 161–175, 2013.

[119] K.-K. Du, Z.-L. Wang, and M. Hong, “Human machine interactive system on smart home of IoT,” *The Journal of China Universities of Posts and Telecommunications*, vol. 20, no. 1, pp. 96–99, 2013.

[120] C.-L. Wu and L.-C. Fu, “Design and realization of a framework for human-system interaction in smart homes,” *IEEE Transactions on Systems, Man, and Cybernetics Part A: Systems and Humans*, vol. 42, no. 1, pp. 15–31, 2012.

[121] Ó. D. Lara and M. A. Labrador, “A survey on human activity recognition using wearable sensors,” *IEEE Communications Surveys & Tutorials*, vol. 15, no. 3, pp. 1192–1209, 2013.

[122] E. Ancillotti, R. Bruno, and M. Conti, “The role of communication systems in smart grids: architectures, technical solutions and research challenges,” *Computer Communications*, vol. 36, no. 17-18, pp. 1665–1697, 2013.

[123] D.-M. Han and J.-H. Lim, “Smart home energy management system using IEEE 802.15.4 and zigbee,” *IEEE Transactions on Consumer Electronics*, vol. 56, no. 3, pp. 1403–1410, 2010.

[124] C. H. Liu, J. Fan, J. W. Branch, and K. K. Leung, “Toward QoI and energy-efficiency in internet-of-things sensory environments,” *IEEE Transactions on Emerging Topics in Computing*, vol. 2, no. 4, pp. 473–487, 2014.

[125] L. Calderoni, D. Maio, and S. Rovis, “Deploying a network of smart cameras for traffic monitoring on a ‘city kernel’,” *Expert Systems with Applications*, vol. 41, no. 2, pp. 502–507, 2014.

[126] S. P. Lau, G. V. Merrett, A. S. Weddell, and N. M. White, “A traffic-aware street lighting scheme for smart cities using autonomous networked sensors,” *Computers & Electrical Engineering*, vol. 45, pp. 192–207, 2015.

[127] N. Zheng and N. Geroliminis, “Modeling and optimization of multimodal urban networks with limited parking and dynamic pricing,” *Transportation Research Part B: Methodological*, vol. 83, pp. 36–58, 2016.

[128] A. Botta, W. D. Donato, V. Persico, and A. Pescapé, “Integration of cloud computing and internet of things: a survey,” *Future Generation Computer Systems*, vol. 56, pp. 684–700, 2016.

[129] M. R. Abdmeziem and D. Tandjaoui, “An end-to-end secure key management protocol for e-health applications,” *Computers & Electrical Engineering*, vol. 44, pp. 184–197, 2015.

[130] B. Xu, L. D. Xu, H. Cai, C. Xie, J. Hu, and F. Bu, “Ubiquitous data accessing method in iot-based information system for emergency medical services,” *IEEE Transactions on Industrial Informatics*, vol. 10, no. 2, pp. 1578–1586, 2014.

[131] A. Pantelopoulos and N. G. Bourbakis, “A survey on wearable sensor-based systems for health monitoring and prognosis,” *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews*, vol. 40, no. 1, pp. 1–12, 2010.

[132] S. R. Moosavi, T. N. Gia, A. M. Rahmani et al., “SEA: a secure and efficient authentication and authorization architecture for IoT-based healthcare using smart gateways,” *Procedia Computer Science*, vol. 52, pp. 452–459, 2015.

[133] R. Li, B. Lu, and K. D. McDonald-Maier, “Cognitive assisted living ambient system: a survey,” *Digital Communications and Networks*, vol. 1, no. 4, pp. 229–252, 2015.

[134] A. Hussain, R. Wenbi, A. L. Da Silva, M. Nadher, and M. Mudhish, “Health and emergency-care platform for the elderly and disabled people in the Smart City,” *Journal of Systems and Software*, vol. 110, pp. 253–263, 2015.

[135] S. J. Liu and G. Q. Zhu, “The application of GIS and IOT technology on building fire evacuation,” *Procedia Engineering*, vol. 71, pp. 577–582, 2014.

[136] Z. Yinghua, F. Guanghua, Z. Zhigang, H. Zhian, L. Hongchen, and Y. Jixing, “Discussion on application of IOT technology in coal mine safety supervision,” *Procedia Engineering*, vol. 43, pp. 233–237, 2012.

[137] E. Sun, X. Zhang, and Z. Li, “The internet of things (IOT) and cloud computing (CC) based tailings dam monitoring and pre-alarm system in mines,” *Safety Science*, vol. 50, no. 4, pp. 811–815, 2012.

[138] V. R. L. Shen, H.-Y. Lai, and A.-F. Lai, “The implementation of a smartphone-based fall detection system using a high-level fuzzy Petri net,” *Applied Soft Computing*, vol. 26, pp. 390–400, 2015.

[139] T. Ojha, S. Misra, and N. S. Raghuwanshi, “Wireless sensor networks for agriculture: the state-of-the-art in practice and

future challenges," *Computers and Electronics in Agriculture*, vol. 118, pp. 66–84, 2015.

[140] K. Han, D. Zhang, J. Bo, and Z. Zhang, "Hydrological monitoring system design and implementation based on IOT," *Physics Procedia*, vol. 33, pp. 449–454, 2012.

[141] A. S. Voulodimos, C. Z. Patrikakis, A. B. Sideridis, V. A. Ntafis, and E. M. Xylouri, "A complete farm management system based on animal identification using RFID technology," *Computers and Electronics in Agriculture*, vol. 70, no. 2, pp. 380–388, 2010.

[142] S. Sarangi, J. Umadikar, and S. Kar, "Automation of agriculture support systems using wisekar: case study of a crop-disease advisory service," *Computers and Electronics in Agriculture*, vol. 122, pp. 200–210, 2016.

[143] W. He and L. D. Xu, "Integration of distributed enterprise applications: a survey," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 1, pp. 35–42, 2014.

[144] M. Fagiani, S. Squartini, L. Gabrielli, S. Spinsante, and F. Piazza, "A review of datasets and load forecasting techniques for smart natural gas and water grids: analysis and experiments," *Neurocomputing*, vol. 170, pp. 448–465, 2015.

[145] L. D. Xu, W. He, and S. Li, "Internet of things in industries: a survey," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 4, pp. 2233–2243, 2014.

[146] M. S. Jamil, M. A. Jamil, A. Mazhar, A. Ikram, A. Ahmed, and U. Munawar, "Smart environment monitoring system by employing wireless sensor networks on vehicles for pollution free smart cities," *Procedia Engineering*, vol. 107, pp. 480–484, 2015.

[147] G. Xu, G. Q. Huang, and J. Fang, "Cloud asset for urban flood control," *Advanced Engineering Informatics*, vol. 29, no. 3, pp. 355–365, 2015.

[148] C. Zhu, V. C. M. Leung, L. Shu, and E. C. Ngai, "Green Internet of Things for smart world," *IEEE Access*, vol. 3, pp. 2151–2162, 2015.

[149] F. K. Shaikh, S. Zeadally, and E. Exposito, "Enabling technologies for green internet of things," *IEEE Systems Journal*, no. 99, pp. 1–12, 2015.

[150] 3GPP, "TR 36.888, Study on provision of low-cost Machine-Type Communications (MTC) User Equipments (UEs) based on LTE," [Online], https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=2578.

[151] SIGFOX, *SIGFOX White Paper*, SIGFOX, 2014.

[152] LoRa Alliance, "LoRaWAN- what is it," LoRa Alliance White paper, 2015. [Online], https://www.lora-alliance.org/portals/0/documents/whitepapers/LoRaWAN101.pdf.

[153] X. Lin, J. G. Andrews, A. Ghosh, and R. Ratasuk, "An overview of 3GPP device-to-device proximity services," *IEEE Communications Magazine*, vol. 52, no. 4, pp. 40–48, 2014.

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

[155] 3GPP TR 22.803, "Feasibility study for Proximity Services (ProSe) (Release 12)," v. 12.2.0, June 2012.

[156] C. Campolo, A. Molinaro, and R. Scopigno, *Vehicular Ad Hoc Networks: Standards, Solutions, and Research*, Springer, Berlin, Germany, 2015.

[157] ETSI, "Intelligent Transport Systems ITS; Vehicular communications; Basic set of applications; Part 2: specification of cooperative awareness basic service," EN 302 637-2, 2014.

[158] ETSI, "ITS; decentralized environmental notification messages (DENM)," EN 102 869-X, 2012.

[159] SAE International and DSRC Committee, "DSRC message communication minimum performance requirements: basic safety message for vehicle safety applications," SAE Draft Std. J2945.1 Revision 2.2, SAE, 2011.

Hindawi Publishing Corporation
Mobile Information Systems
Volume 2016, Article ID 6123234, 15 pages
http://dx.doi.org/10.1155/2016/6123234

Research Article

An Architecture of IoT Service Delegation and Resource Allocation Based on Collaboration between Fog and Cloud Computing

Aymen Abdullah Alsaffar,[1] Hung Phuoc Pham,[1] Choong-Seon Hong,[1] Eui-Nam Huh,[1] and Mohammad Aazam[2]

[1] Department of Computer Engineering, Kyung Hee University, Yongin-si, Seoul, Republic of Korea
[2] Department of Systems and Computer Engineering, Carleton University, Ottawa, ON, Canada

Correspondence should be addressed to Eui-Nam Huh; johnhuh@khu.ac.kr

Received 29 April 2016; Revised 25 July 2016; Accepted 25 August 2016

Academic Editor: Young-June Choi

Despite the wide utilization of cloud computing (e.g., services, applications, and resources), some of the services, applications, and smart devices are not able to fully benefit from this attractive cloud computing paradigm due to the following issues: (1) smart devices might be lacking in their capacity (e.g., processing, memory, storage, battery, and resource allocation), (2) they might be lacking in their network resources, and (3) the high network latency to centralized server in cloud might not be efficient for delay-sensitive application, services, and resource allocations requests. Fog computing is promising paradigm that can extend cloud resources to edge of network, solving the abovementioned issue. As a result, in this work, we propose an architecture of IoT service delegation and resource allocation based on collaboration between fog and cloud computing. We provide new algorithm that is decision rules of linearized decision tree based on three conditions (services size, completion time, and VMs capacity) for managing and delegating user request in order to balance workload. Moreover, we propose algorithm to allocate resources to meet service level agreement (SLA) and quality of services (QoS) as well as optimizing big data distribution in fog and cloud computing. Our simulation result shows that our proposed approach can efficiently balance workload, improve resource allocation efficiently, optimize big data distribution, and show better performance than other existing methods.

1. Introduction

Cloud computing is not only a technology that continuously advances for offering a variety of services and resources to many cloud consumers smart devices (e.g., IoT, smart wearable devices, smart phone, smart tablets, and smart home appliances) but also an enabling developer to develop more applications, tools, and services. Cloud computing architecture can empower ubiquitous, advantageous, and on-demand network access to a shared pool of configurable computing resources, providing many other benefits (e.g., storages, services, applications, networks, virtualized resources, large scale computation, schedulable virtual servers, high expansibility, computing power, low price services, virtual network, network bandwidth, and high reliability) [1–3]. One of the technologies that is gaining popularity is known as Internet of things (IoT). IoT is a technology that is still developing and enables many objects (e.g., thin-client, smart phone, smart tablets, smart home appliances, smart wearable devices, and sensor) to connect to Internet to perform variety of services (e.g., memory, storage space, processing, virtualization, resource allocation, services delegation, surfing, send/receive big data, and viewing social sites). Thus, smart devices services are present in every aspect of our daily life (e.g., health care, medicine treatment, education, and remotely controlled smart devices). Cloud computing technology is being widely used to support variety of cloud consumer devices, services, and applications.

Despite the wide utilization of cloud computing (e.g., services, applications, and resources), some of the services, applications, and smart devices are not able to fully benefit from this attractive cloud computing paradigm due to the following issues: (1) smart devices are lacking in their capacity (e.g., processing, memory, storage, battery, and resource allocation), (2) they are lacking in their network resources, and (3) the high network latency to centralized server in cloud is not efficient for delay-sensitive application, services, and resource allocations requests. According to [3], the number of devices that connected to Internet will exceed 24 billion by 2020. The rapid increase of number of Internet connected devices combined with the long distance between user smart devices and cloud computing, and the repeatedly requested services, will pose heavy burden to network performance and network bandwidth which in return will degrade cloud computing QoS as well. Moreover, the high network latency between user devices and cloud may not be ideal for delay-sensitive applications, services, and resources.

To resolve the abovementioned issues, we utilize fog computing which is a new paradigm that extends cloud computing resources and service to the edge of network. It is highly virtualized infrastructure that can provide networking services, computation, storage, memory between IoT devices, and traditional cloud computing environment [4]. Furthermore, fog computing is located in localized environment, making it closer to user location and giving it the advantage over cloud to provide variety of distributed applications [4]. The impressive advantages that fog computing offers over cloud computing will not only increase the number of requests services (i.e., delay-sensitive services, applications, and data) but also direct most of user requested services if not all to fog computing only. This will lead to unbalanced workload of services and degraded performance of fog performance, user requests, and the abandonment of IoT service from cloud computing.

Therefore, in this paper, we introduce new architecture of IoT service delegation and resource allocation based on collaboration between fog and cloud computing. We provide new algorithm that is decision rules of linearized decision tree based on three conditions (services size, completion time, and VMs capacity) for managing and delegating user request. Furthermore, we present our new strategy for data distribution optimization such as big data. Moreover, we present an algorithm to perform resource allocation in order to satisfy service level agreement (SLA) and quality of service (QoS). Our simulation shows that our approach can improve the efficiency of resource allocation and show better performance comparing with other approaches.

The rest of paper is organized as follows. In Section 2, we introduce related work. In Section 3, we introduce our system architecture and motivating scenario. In Section 4, we present our proposed mechanism for service delegation, resource allocation, and big data distribution processes. In Section 5, we present our implementation and analysis result. In Section 6, we present our conclusion and future work.

2. Related Work

There are many researches attempting to resolve the abovementioned issues. In [5], the author introduces efficient synchronization in cloud for number of hierarchy distributed file systems. The author deploys the conception of master-slave architecture to propagate data to reduce traffic. In [6], the author introduces method for resource scheduling which can be efficient in mitigating the impacts that influence application respond time and utilization of the system. In [7, 8], the authors introduce the impact of data transmission delay on the performance. In [9], the author introduces one method to make a parallel processing for big data which can increase the performance in federated cloud computing. However, these researches do not mention how much resources should be utilized.

Also, there are many completed researches that deal with resources allocation. In [10], the authors explain through their work that shared allocation is superior to dedicated allocation. Nevertheless, the authors do not perform experiment with arbitrary number of SLAs. Moreover, authors do not show how fast the server needs to be in order to guarantee quality of service (QoS). In [11, 12], the authors provide services to large number of SLAs as it is quite difficult to obtain performance between shared and reserved allocation. In [13], the author introduces a model which secure resource allocation in cloud computing, where the author designed fuzzy- logic based trust and reputation model.

Many researches have been done in order to provide efficient method to integrate mobile devices and cloud computing environment. In [14], the author presents concept where cloud computing is utilized in order to improve the capability of mobile devices. In [15], the author did some modification to Hyrax which enables mobile devices to utilize cloud computing platforms. The concept of deploying mobile devices as a provider of resources is presented even though the experiment was not integrated. In [16], the authors only concentrate on the use of partition policies to hold the effect of application on mobile devices. However, they did not resolve any other issues regarding mobile cloud computing or fog computing.

Fog computing technology is still in its early stage and needs more time to develop like cloud computing. To the best of our knowledge, there are not many researches considering collaboration of fog and cloud computing to provide efficient way of delegating IoT services between fog and cloud to better balance workload/requested services/resources. Furthermore, we introduce new methods to delegate services to multiple fog and cloud computing based on linearized decision tree which considers three conditions (service size, completion time, and VMs capacity). Moreover, we introduce new strategy for data distribution and introduce an algorithm to preform resource allocation to guarantee SLA and QoS.

3. Proposed Architecture and Scenario

In this section, we will introduce our new system architecture, explain its component and scenario, and explain the advantages and disadvantages of fog and cloud collaboration.

3.1. System Architecture. Figure 1 illustrates our proposed system architecture which consists of three layers; upper layer, middle layer, and lower layer. Table 1 illustrates our system components and explains their role.

Table 1: System component.

Component	Description
IoT devices	All smart devices that are capable of connecting to internet.
Cloud/fog broker	Responsible for receiving user request/services, providing services/search for VMs, and delegating service to other fog/cloud environments.
Cloud/fog computing server	Responsible for providing requested services/resource, processing them, and delivering them back to broker.
Services monitor server	Responsible for maintaining and storing record of current service and their progress and providing/checking available space for new services.
3rd-party cloud server	Responsible for providing services to fog broker and cloud broker.
VMs occupancy	Responsible for providing list of the current available VMs capacity and showing the used available VMs capacity.
Services map table	Responsible for presenting map of services and their divided chunks in the same and/in other fog computing environments as well as in cloud computing environment.

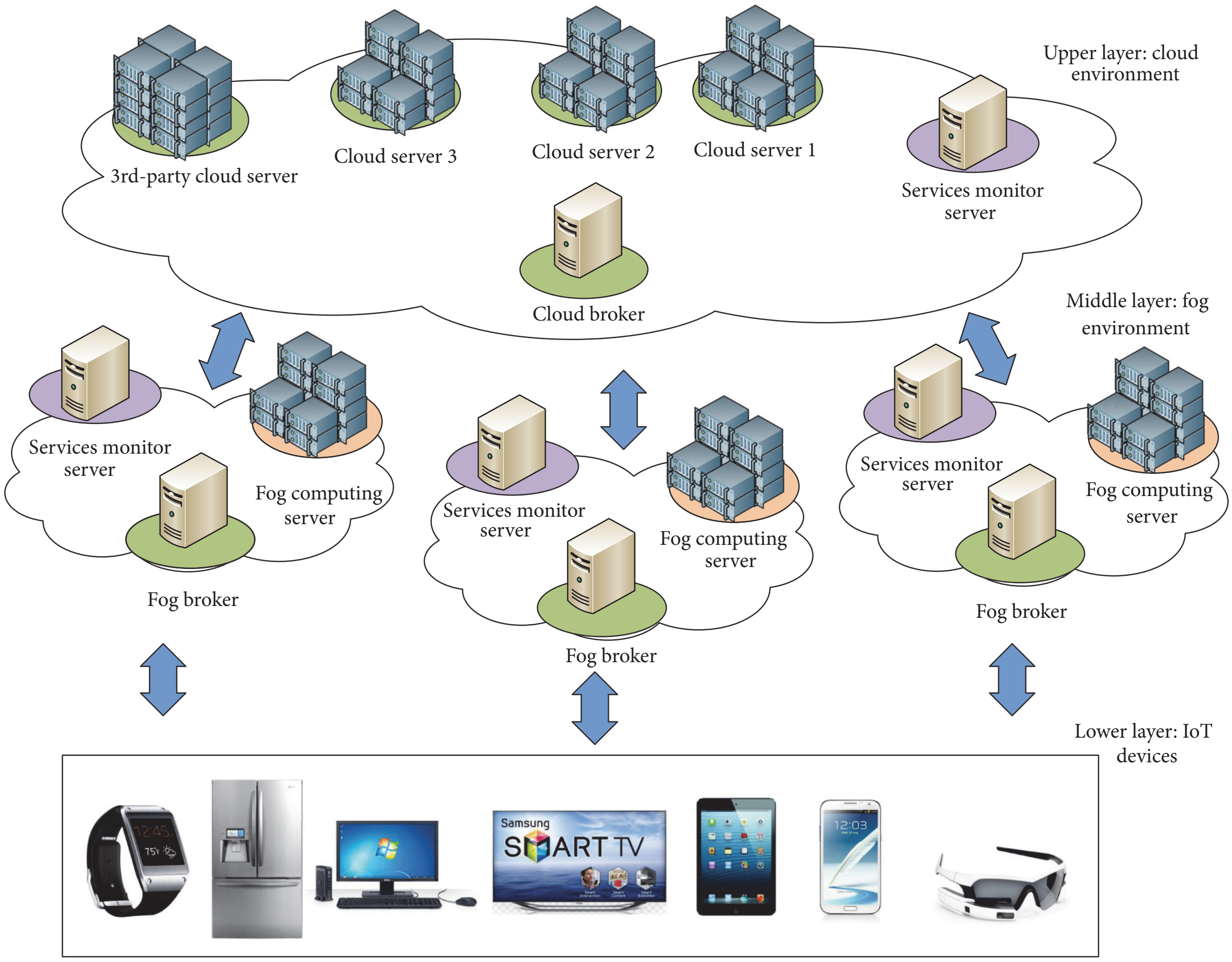


Figure 1: Proposed system architecture.

We provide detailed assumption for our architecture. Our assumption is as follows:

(1) We assume that there are 3 fog computing environments which have closer distance between each other and user smart devices as well as long distance between fog and cloud. Each fog computing environment will include fog broker to manage request services, obtain information of VMs capacity in other fog/cloud environments, and so forth. (Note: in case of larger network area, it is possible to have more than 3 fog computing environments.)

(2) We assume that all services will be requested from fog computing environment. Based on three conditions such as services size, completion time, and VMs capacity, fog broker will decide to process the current requested services in current fog or in other fog/cloud computing environments.

(3) We assume that there are delay-sensitive and nondelay-sensitive requested services, applications, and data.

(4) We assume that all services monitoring server in fog will sync their VM capacity status and current services processing with services monitoring server in cloud. When any fog needs more VM capacity, fog broker will obtain that information from services monitoring server in cloud which will reduce the search time of VMs capacity in other fog or cloud computing environments.

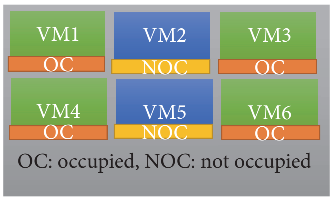
3.1.1. Upper Layer of System Architecture. The upper layer represents cloud computing environment. This layer consists of cloud broker, services monitor server, cloud servers, and 3rd-party cloud server. In case that there is no available capacity in fog to process service and the requested services that is needed later, then the fog can delegate these services to cloud computing.

3.1.2. Middle Layer of System Architecture. The middle layer represents fog computing environment and consists of three fog computing environments. Each fog computing environment consists of fog broker, fog computing server, and services monitor server. Each layer can be aware of each other through using unique communication address for each environment configured by policies or by the cloud itself. Note that it is possible to have more than 3 fog computing environments when we cover larger area/state. All ubiquitous and future services/resources can be requested from any fog environment as well as cloud based on service size (e.g., large or small), requested time (e.g., now or later), and VMs capacity (e.g., occupied or not occupied).

3.1.3. Lower Layer of System Architecture. The lower layer represents user smart devices. User smart devices can consist of smart phone, smart tablets, smart sensors, smart wearable devices, thin-client, smart home appliances, and so forth. Some of these smart devices have different specification and capabilities (e.g., computation process, storage space, screen resolution, and bandwidth). Fog computing can provide efficient way for them to perform these tasks over the Internet in less respond time and efficient performance.

3.1.4. The Role of Services Monitor Server. The services monitor server consists of two important components such as VMs occupancy table and services map table.

The VMs occupancy table is used to keep a list of VMs capacity and occupancy (e.g., occupied and not occupied) in each fog and cloud computing environment. The benefit of this table is to provide fast way for broker to decide on where to process the current service based on service size and the time needed to be completed. In case there is not enough capacity in current VMs at current fog, then the broker can send request to service monitor server in cloud which will store/keep record of all VMs capacity in other fog and cloud environments. Here, the broker can request to reserve these VMs for their current service from other fog/cloud environments. Figure 2 illustrates a sample of VMs occupancy table.



VMs number	Service ID	Fog/cloud	Parts	Progress	Exp_Finish_ Time
VM1	1035	Fog 1	2 out of 4	20%	00:00:00
VM6	1035	Fog 2	3 out of 4	40%	00:00:00
VM3	1040	Fog 1	5 out of 6	80%	00:00:00
VM4	1135	Cloud 1	1 out of 3	95%	00:00:00

Figure 2: Illustrating sample of VMs occupancy table.

Services map table is used to keep/store list of currently processed services and their location. For example, sometimes, service is requested from current fog/cloud environment and to complete this service we need the collaboration of 2 or 3 VMs; however, we only have 1 VM at current fog. As a result, the broker will search the cloud for not occupied VMs, reserve them, and request processing the rest of the service in cloud. Here, this table will list service ID, VMs number, location, progress time, expected finishing time, and the IP address which was used to send the services.

3.2. Scenario. As fog computing is gaining more popularity for being near the underlying network and extending cloud computing services/resources near to user location, envision some situation where IoT devices user can benefit from using fog computing environment to resolve any problem, especially when they are in public or at home. We can use the example of many IoT smart devices such as google glass, smart oven, and smart refrigerator. In the first example, the user wants to use their smart oven to cook some food. In order to do so, the smart oven wants to search the Internet to obtain the right temperature which is needed to cook the food. However, the smart oven has limited capability for searching the Internet. As a result, the smart oven can connect to local fog computing environment which in return searches for the right information, and finally, stores that information in the smart oven application. Furthermore, the user can input name of many foods and ask fog to search for the right recipe. Here, the smart oven receives the requested information in convenient and short time.

Another example could be google glass. Let us assume that google glass user is taking pictures which required obtaining information such as sightseeing and cloths. Google glass might have limited capabilities to do searching which might consume more power and need more resources, big storage, and higher bandwidth. Here, the user can take some pictures of video and send them to fog broker in fog computing for information searching. In case there are many services requested/assigned to current fog, fog broker will collaborate with other fog/cloud computing environments and delegate the services to them. In fog/cloud environment, the service will be divided into many chunks which will be assigned to VMs for processing. After processing is completed, these chunks will be sent to fog/cloud broker where they will be combined and sent back to user devices.

Our proposed scenario illustrates the advantages of utilizing collaboration of work between fog and cloud computing environment. This collaboration efficiently increases the chances of providing efficient method for services delegation, optimizing resources, and optimizing data distribution between fog and cloud computing environment.

3.3. Fog and Cloud Collaboration. In this section, we will introduce the advantages and disadvantages of fog and cloud computing collaboration.

Advantages of fog and cloud computing collaboration are as follows:

(1) Dividing the work load between fog and cloud leads to fast completion of requested tasks when there are many requests (e.g., video, movies, and clips) which are requested at the same time/peak time (e.g., World Cup show and Olympic games events).

(2) The collaboration between fog and cloud lead to better managing of network performance by dividing requested services to small parts and sending them through the network to different fog or cloud for processing. This will reduce the network overload which in return will reduce fog and cloud performance overload.

(3) Fast resources allocation for requested services leads to QoE and efficiently managing resources to a variety of fog and cloud consumers.

(4) Fog and cloud broker can communicate to manage and organize requested services and VMs capacity.

The disadvantages of fog and cloud computing collaboration are as follows:

(1) It might take more time to search for free VMs capacity from other fog or cloud computing environments. To solve this issue, we include in both fog and cloud environments services monitor server which keeps record of current free VMs and VMs status. When fog needs more VMs, fog broker will request VMs capacity of other fog environments from services monitoring server in cloud which will store VMs capacity and currently process services of all fog environments.

(2) Dividing many services to small parts and sending them to other fog or cloud environments might create larger table with larger size when it comes to request certain services that are larger in size.

Table 2: IoT services delegation constrain cases.

	Service size	Completion time	VMs capacity
Case 1	Small	Now	Occupied/not occupied
Case 2	Small	Later	Occupied/not occupied
Case 3	Large	Now	Occupied/not occupied
Case 4	Large	Later	Occupied/not occupied

4. Proposed Mechanism

In this section, we will introduce our methods which we used for IoT services delegation, optimizing resources allocation, and optimizing data distribution. Furthermore, we will provide algorithms and mathematical equations as well.

4.1. Services Delegation Process. In this section, we will explain our method which we used to delegate services to other fog environments and cloud computing environments. The delegation of any services requested from any fog/cloud environment is decision rules of linearized decision trees based on three conditions (service size, completion time, and VMs capacity). The requested services size can be small or large, the requested completion time can be now or later, and VMs capacity can be occupied or not occupied at current fog/cloud environment. We consider 4 cases in Table 2 and provide 2 algorithms that explain these cases process in detail. The cases are shown in Table 2.

Both of Algorithms 1 and 2 aim to find where to delegate the services for processing based on service size (e.g., small or large), services completion time (e.g., now or later), and VMs capacity (e.g., enough or not enough) and in some cases we include services that are in queue (e.g., services in queue, yes or no). As for service size, we can, for example, determine the size based on checking if the size is bigger than 500 mb or not. Moreover, we also aim to manage these services in fog and cloud environment. Figure 4 illustrates sequence flow diagram of any service that is requested from fog environment where there is enough VMs capacity. Figure 5 illustrates sequence flow diagram of services that is requested from fog environment where there is not enough VMs capacity.

4.2. Resource Allocation and Data Distribution Process. Many of formerly presented approaches utilize 1/m/1 model to provide solution to previously mentioned problem. Nevertheless, in our proposal, we utilize 1/m/m/1 for solving the same problem, where (1) refers to cloud broker, (m) refers to many paths, (m) refers to many fog brokers in fog environments, and (1) refers to IoT devices users. In detail, IoT devices will send service request to fog broker in fog environment. Fog broker will divide data into multiple blocks where they will be assigned to certain VMs. Each block will be divided into multiple chunks which will be sent to multiple processor for processing. After receiving the processed data, the processor combines them again into one big data and returns the result to user IoT devices.

By using this method, we reduce/eliminate the burden to the system when we process big data size. Therefore, we guarantee the availability of server in fog or cloud

```
Input: $S_s$ // service size (small or large), $S_{ct}$ // service completion time (now or later), VM$_c$ // VM capacity
        (occupied = not enough or not occupied = enough)
Output: service delegation/management location // fog or cloud
If (Service Size = small) && (Service completion time = now) && (VMs capacity = enough)
THEN
Divide requested services to small chunk
Calculate the required no. of VMs
Assign these chunks to the assigned VMs for processing
else if (Service Size = small) && (Service completion time = now) && (VMs capacity = not enough)
THEN
Divide requested services to small chunks
Calculate the required no. of VMs
Obtain list of available VMs capacity in other fog/cloud environment from Services Monitor Server in cloud.
Reserved the VMs and assign the chunks to them.
else if (Service Size = small) && (Service completion time = later) && (VMs capacity = enough) && (Services in Queue = no)
THEN
Process the requested service at current location (fog environment)
Divide requested services to small chunks
Calculate the required no. of VMs
Assign these chunks to the assigned VMs for processing
else if (Service Size = small) && (Service completion time = later) && (VMs capacity = enough) && (Services in Queue = yes)
delegate the requested services to be processed in cloud
Divide requested services to small chunks
Calculate the required no. of VMs
Assign these chunks to the assigned VMs for processing
end if
  end if
    end if
After completion the processing of all chunks,
return the chunks to broker for combining them and send the result to users IoT devices.
```

Algorithm 1: Finding IoT services delegation/management in fog/cloud based on three conditions (service size, completion time, and VMs capacity) for cases 1 and 2.

environment to process large number of requested services at peak and nonpeak time, guarantee fast respond time, and assure satisfying quality of services (QoS).

Next, we will explain the process of our work which consists of two stages. In stage 1, firstly, we determine the minimum number of VMs needed to do the job and their speed. Secondly, we divided and assigned data based on VMs capacity. In stage 2, firstly, we distribute data which has different capacity to processors. Secondly, after the completion of processing the divided chunks, they will return to cloud/fog broker to combine them and, finally, they will be sent to IoT devices user.

4.2.1. First Stage of Proposed Mechanism.

In the first stage, we determine the minimum number of VMs needed to do the job and their speed. Secondly, we divided and assigned data based on VMs capacity.

(A) Determine the Number of VMs and Speed. We use Algorithm 3 to determine the minimum number of VMs which is required to do the job depending on service level agreement (SLA). Furthermore, we use cumulative distribution function (CDF) $F(x)$ time respond which is available in [17]. The minimum number of VMs m keep increasing until $F(x)$ arrive at the desired targeted probability. As a result, we can receive the required m for SLA. Next, we present description of function $F(x)$ and it is as follows:

$$F(x) = \text{Probability}\ (\text{time of response} < x)$$
$$= \begin{cases} 1 - e^{-\mu x} - k\mu e^{-\mu x} x & \text{for } \sigma = m_i - 1 \\ 1 - e^{-\mu x} - k\mu e^{-\mu x (m_i - \sigma)} \left[\dfrac{1 - e^{-\mu x (1 - m_i + \sigma)}}{1 - m_i + \sigma} \right] & \text{for } \sigma \neq m_i - 1, \end{cases} \tag{1}$$

where $\sigma = \lambda/\mu$.

$$k = p(0) \frac{\sigma^{m_i} - \mu}{m_i!} * \frac{m_i}{(m_i - \sigma)},$$
$$P(0) = \left(\sum_{n=0}^{m-1} \frac{\sigma^n}{n!} + \frac{mp^m}{m!\,(m - \sigma)} \right)^{-1}. \tag{2}$$

λ is the arrival rate and μ is the service rate.

Fog computing infrastructure can provide diversity of services to satisfy a large number of SLAs through utilizing unique scheduling methods such as FCFS which is shown in Figure 6. Thus, we are recommending to allocate the VMs into two groups where the first group will be utilized for shared allocation (SA) $m_{\text{shared Allocation}}$ and the second group will be utilized for reserved allocation (RA) $m_{\text{reserved Allocation}}$.

```
Input: $S_s$ // service size (small or large), $S_{ct}$ // service completion time (now or later), $VM_c$ // VM capacity
        (occupied = not enough or not occupied = enough)
Output: service delegation/management location // fog or cloud
If (Service Size = Large) && (Service completion time = now) && (VMs capacity = enough)
THEN
  Process the requested service at current location (fog environment)
  Divide requested services to small chunks
  Calculate the required no. of VMs
  Assign these chunks to the assigned VMs for processing
else if (Service Size = Large) && (Service completion time = now) && (VMs capacity = not enough)
THEN
  Divide requested services to small chunks
  Calculate the required no. of VMs
  Obtain list of available VMs capacity in other fog/cloud environment from Services Monitor Server in cloud.
  Reserved the VMs and assign the chunks to VMs for processing
else if (Service size = Large) && (Service completion time = later) && (VMs capacity = not enough) &&
        (Services in Queue = yes)
THEN
  this services will be delegated to other fog/cloud environment
  Divide requested services to small chunks
  Calculate the required no. of VMs
  Assign these chunks to assigned VMs for processing.
end if
  end if
    end if
After completion the processing of all chunks,
return the chunks to broker for combining them and send the result to users IoT devices.
```

Algorithm 2: Finding IoT services delegation/management in fog/Cloud based on three conditions (service size, completion time, and VMs capacity) for cases 3 and 4.

```
Input:
  $\lambda$              // rate of arrival
  $\mu$              // rate of service
  SLA($x, z$)      // $x$: time of response
                 // $z$: probability target
Output: $m$          // required minimum no. of VMs
  Float $\sigma = \lambda/\mu$
  Function determineMinVM ($\sigma, \mu, x, z$) {
  If ($\sigma$ == (int) $\sigma$)    $m$ = (int) $\sigma$;
  Else $m$ = (int)Math.floor($\sigma$) + 1;
  While $F(x) \leq z$, $m$++;
  Return $m$; // required minimum no. of VMs }
```

Algorithm 3: Determining the number of VMs.

As for shared allocation, the arrival jobs of SLA are merged in a single steamed and served by m VMs.

And, as for reserved allocation, we provide one VM for each arriving job which is shown in Figure 7. Both fog and cloud computing will utilize the model for shared allocation and reserved allocation.

All of the SLAs in shared allocation will have the same CDF of response time and arrival rate $\lambda = \sum_{i=1}^{k} \lambda_i$. Thus, the minimum number of VMs $m_{\text{Shared Allocation}}$ to meet k SLAs is given by

$$m_{\text{Shared Allocation}} = \max\left(m_1, \ldots, m_i, \ldots, m_k\right). \quad (3)$$

Table 3: An example of proposed cases.

Cases	λ_1	x_1, y_1	λ_2	x_1, y_1	m_{Reserved}	m_{Shared}
Case 1	3.9	3, 0.7	3	10	10	11
Case 2	3.9	3, 0.85	3.9	12	12	10

The number of VMs which is required to satisfy SLA_i of user i is referred to as m_i. Let the smallest number of VMs which is required to meet k SLA in reserved allocation be $m_{\text{Shared Allocation}}$. As a result, $m_{\text{Reserved Allocation}}$ is given by

$$m_{\text{Reserved Allocation}} = \sum_{i=1}^{k} m_i. \quad (4)$$

In this case, when more than 1 user request services with the same SLAs, the shared allocation can provide the same or even enhanced performance than reserved allocation ($m_{\text{shared Allocation}} \leq m_{\text{Reserved Allocation}}$). However, in case that SLA_1 and SLA_2 are not the same for shared allocation, then it will be quite difficult to determine whether shared allocation is better than reserved allocation or the opposite. Table 3 will provide example of two cases for shared and reserved allocation.

Note that, in some cases, we have to consider the case where there are services in queue or not yet decided to where to process the requested services (e.g., in fog or in cloud).

```
Input:
$\lambda_1$, $\lambda_2$        // rate of arrival
$\mu$        // rate of service
SLA$_1$, SLA$_2$
$E$        // processing time expectation
Output:
SA, RA    //shared and reserved allocation strategy
Function determineAllocStrategy ($\lambda_1$, $\lambda_2$, SLA$_1$, SLA$_2$, $E$, $\mu$) {
Calculate SLA difference $D$
Get the corresponding angle $\alpha$ from the SLA difference table
If ($\mu \geq (1/E[T] + \lambda_1)$ && $\mu \geq (1/E[T] + \lambda_2)$)
If (Math.asin($\lambda_2$/sqrt ($\lambda_1 * \lambda_1 + \lambda_2 * \lambda_2$)) $\leq \alpha$)
Return RA // reserved allocation
Else
Return SA // shared allocation
Else
Return false {
```

Algorithm 4: Determining the allocation strategy.

Table 4: Service level agreement difference (SLA).

D	α
$(0, 20)$	0
$(20, 40)$	20
$(40, 66)$	50
$(66, 88)$	70

By examining both cases at Table 3, we notice that, in the first case, $m_{\text{Reserved Allocation}}$ has shown better performance than $m_{\text{shared Allocation}}$ and, in the second case, $m_{\text{shared Allocation}}$ has shown better performance than $m_{\text{Reserved Allocation}}$. We are trying to determine the best suitable strategy for shared and reserved allocation for the purpose of satisfying SLA$_1$ and SLA$_2$. Moreover, the VMs are able to guarantee the quality of services (QoS). Let the average number of VMs which is needed to meet a given SLA over considered arrival time be $E(\text{SLA})$:

$$E(\text{SLA}) = \frac{1}{k}\sum_{0}^{k}\int(k, x, y). \tag{5}$$

Let D refer to the difference between SLA$_1$ and SLA$_2$. As a result, D is given by

$$D = \left|E(\text{SLA}_1) - E(\text{SLA}_2)\right|. \tag{6}$$

Algorithm 4 illustrates our allocation strategy to satisfy service level agreements (SLA) and quality of services (QoS).

The relationship between D and angle α is explained in Table 4. As illustrated in Table 4, every D is fixed by the changes in arrival time of λ_1, λ_2 in (0, 30) and the average angle of SLA is different for every range.

We state angle α by the next formula:

$$\sin\alpha = \frac{\lambda_2}{\text{sqrt}(\lambda_1 * \lambda_1 + \lambda_2 * \lambda_2)}. \tag{7}$$

The next step is to discover the speed of VMs in order to guarantee the quality of services (QoS) for every requested service. We also deploy the little law which is explained in [18]:

$$E[N] = \frac{p}{(1-p)} \quad \text{where } p = \frac{\lambda}{\mu}. \tag{8}$$

In (8), we refer to the number of jobs in the system by $E[N]$. Equation (9) presents the expectation of processing time:

$$E[T] = \frac{E[N]}{\lambda} = \frac{p}{\lambda(1-p)} = \frac{1}{\mu(1-p)} = \frac{1}{\mu-\lambda}. \tag{9}$$

To satisfy the quality of services (QoS), we set the below formula:

$$\mu \geq \frac{1}{E[T]} + \lambda. \tag{10}$$

By using this formula, we are able to discover the VMs rate of service. Moreover, we introduce an example below to make it clear. For instance, let us assume that we want $E[T] \leq 10$ second, $\lambda = 1$ job/second, then the VMs rate which is needed is as follows:

$$\mu \geq \frac{1}{10} + 1\frac{11}{10}. \tag{11}$$

(B) Determine VMs Capacity. When the system receives any service, first, we have to find out if we need to process it at current location or delegate it to other fog/cloud computing environments based on the algorithm which we mentioned in Section 4.1. Then, we can determine VMs capacity. In order to determine the VMs capacity, we will sort, divide, and assign data to VMs current Capacity. We also set data priority by utilizing training data to sort out data. As a result, the data with higher priority will be transferred first and the one with lower priority will be transferred last.

We divide data to blocks of different sizes (e.g., bl_1, bl_2, and bl_n). In order to select the best VMs based on their capacities, we utilize greedy algorithm. Finally, the VMs with higher capacity will be assigned to the block with big size.

4.2.2. Second Stage of Proposed Mechanism

(A) Distribution of Data Block Process. We start distributing data block which has different capacities to processors. When we receive data, it will be divided into blocks of data. These blocks will be divided into small size which is known as chunks (e.g., $\mathrm{chk}_1, \mathrm{chk}_2, \dots, \mathrm{chk}_n$). Every chunk might have different size based on the strength of bandwidth.

Let us denote the chunk in each block by chk_i, the size of chunk by $w(\mathrm{ch}_i)$, and the bandwidth between VMs and processor by bw_i. Let $w(\mathrm{ch}_i)/b_i$ denote the time it takes to send data (chunk) from VMs to processor. Note that when we consider method of parallelization, the time it takes to send chunks of data to processors should be even:

$$\frac{w(\mathrm{chk}_1)}{\mathrm{bw}_1} = \frac{w(\mathrm{chk}_2)}{\mathrm{bw}_2} = \frac{w(\mathrm{chk}_3)}{\mathrm{bw}_3} = \cdots = \frac{w(\mathrm{chk}_i)}{\mathrm{bw}_i} = t, \tag{12}$$

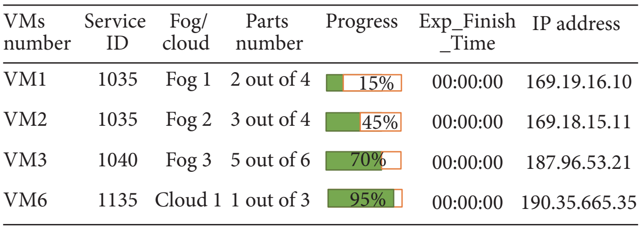
$$\text{Set } S = w(\text{block}) = \sum_{i=0}^{n} w(\mathrm{chk}) = t \sum_{i=0}^{n} b_i.$$

Thus,

$$w(\mathrm{chk}_i) = t * b_i = \frac{S}{\sum_{i=0}^{n} b_i} * b_i. \tag{13}$$

As it is shown above, we can determine the size of every chunk to adapt it with the bandwidth. The next process is to sort out the processor based on processor capacities. For example, if the chunk of data is bigger, then it will be sent to processor with higher capacity for processing it.

(B) The Merging of Data Block Process. In this section, we explain the process of merging data block after being processed. After the data block is being processed, it will be send back to fog broker in fog environment for merging process. In service monitor server, the services map table will keep a record of these blocks and the location where they were processed which is illustrated in Figure 3.



VMs number	Service ID	Fog/ cloud	Parts number	Progress	Exp_Finish _Time	IP address
VM1	1035	Fog 1	2 out of 4	15%	00:00:00	169.19.16.10
VM2	1035	Fog 2	3 out of 4	45%	00:00:00	169.18.15.11
VM3	1040	Fog 3	5 out of 6	70%	00:00:00	187.96.53.21
VM6	1135	Cloud 1	1 out of 3	95%	00:00:00	190.35.665.35

FIGURE 3: Illustrating sample of services map table.

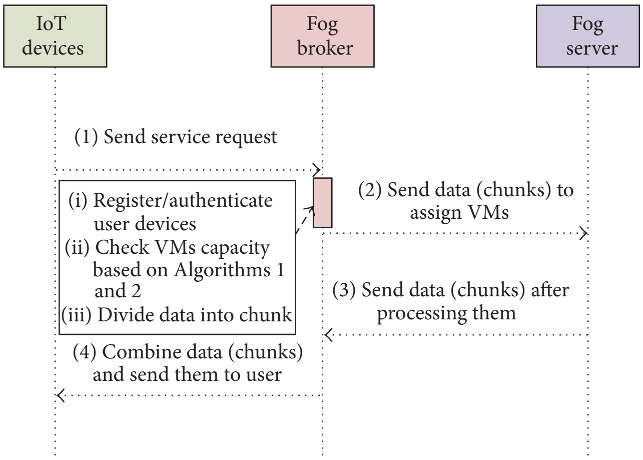

IoT devices
Fog broker
Fog server
(1) Send service request
(i) Register/authenticate user devices
(ii) Check VMs capacity based on Algorithms 1 and 2
(iii) Divide data into chunk
(2) Send data (chunks) to assign VMs
(3) Send data (chunks) after processing them
(4) Combine data (chunks) and send them to user

FIGURE 4: Sequence flow diagram of IoT service process in fog computing environment.

Figures 4 and 5 illustrate the concept where data is divided into chunk and then assigned to VMs in fog or cloud or both of them. Then, after processing these chunk, they are returned to fog broker to merge them and send them to user devices.

Fog computing will act as master which will receive all chunks of data to decrease the complexity that is due to the existing of firewall between processor in fog or cloud environment.

5. Implementation and Analysis

In this section, the numerical experiments results are presented to examine the efficiency of shared allocation (SA) and reserved allocation (RA) as well as comparing our approach's performance with other approaches in terms of processing time to transfer big multimedia data from fog/cloud broker to user smart devices. The comparison method uses one processor [19] to receive data from fog/cloud broker where in our case we use multiple processor.

5.1. Experiment Settings. The characteristics of our target system are illustrated in Table 5. In our PC, we used one Intel Core TM i7 965 and 8 GB RAM. The algorithm was simulated on CloudSim [20]. CloudSim is a framework for modelling and simulation of infrastructures and services in Java jdk-7u7-i586 and Netbeans-7.2.

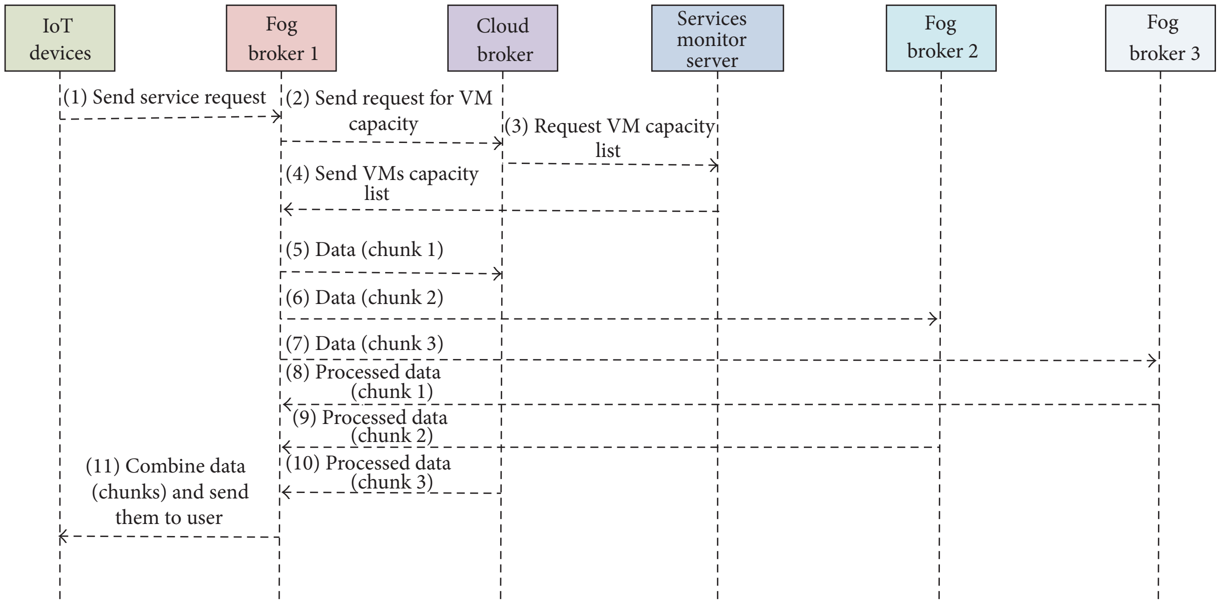

IoT devices
Fog broker 1
Cloud broker
Services monitor server
Fog broker 2
Fog broker 3
(1) Send service request
(2) Send request for VM capacity
(3) Request VM capacity list
(4) Send VMs capacity list
(5) Data (chunk 1)
(6) Data (chunk 2)
(7) Data (chunk 3)
(8) Processed data (chunk 1)
(9) Processed data (chunk 2)
(10) Processed data (chunk 3)
(11) Combine data (chunks) and send them to user

Figure 5: Sequence flow diagram of IoT service which is delegated to other fog/cloud computing environments.

Table 5: Characteristic of the target system.

Parameter	Value
Network	LAN
Topology	Connected
Operating system	Win7 Professional
Number of VMs	25
Number of fog	7
Number of smart devices	10
Bandwidth	[10~512] Mbps

Table 6: Setting for SLA.

Parameter	Value
Response time	[1~10]
Target time	[0.1~0.99]

Table 7: Speeds of requests and response services.

Parameter	Value
Arrival rate	[0.2~3.9]
Service rate	[1~4]

Every parameter in the simulation has different arrival rates λ, response times x, and target probabilities y. Some big files for the abovementioned algorithms are to estimate the required minimum number of VMs for two types of resource allocations and data distribution time. Table 6 illustrates setting for SLA and Table 7 illustrates the speeds of requests and response services.

The experiment result proves that shared allocation and reserved allocation almost have the same impact when SLA is the same for both of them with different arrival rate, response time, and target probability. We did our experiment in the same cases. However, different from other approaches, we used multiple SLA instead of one single SLA.

Figure 8 illustrates different response time of shared and reserved allocation. Our experiment result shows that when the smallest number of VMs decreases, the respond time for shared and reserved allocation increases. In addition, it shows that the probability is almost the same for shared and reserved allocation when we set different response time for shared and reserved allocation.

Figure 9 illustrates SLA different target probability for shared and reserved allocation. Our experiment result shows the minimum number of VMs which is required to meet the satisfaction of SLA. For instance, when the target probability to satisfy SLA is 0.4, we need minimum of 5 VMs for shared and reserved allocation. As a result, it can meet SLA different target probability for shared and reserved allocation.

Figure 10 illustrates different arrival rate of shared and reserved allocation. Our experiment result shows the minimum number of VMs that is required to satisfy SLA which is equivalent to different arrival rate. For instance, when the arrival rate of service is 2, we need minimum number of 3 VMs.

In the case where we consider using multiple SLAs, it is suggested that the strategy of shared and reserved allocation is more resource efficient compared to reserved allocation.

Figure 11 illustrates different SLAs of shared and reserved allocation. The result shows that reserved allocation uses more VMs than shared allocation when number of SLAs

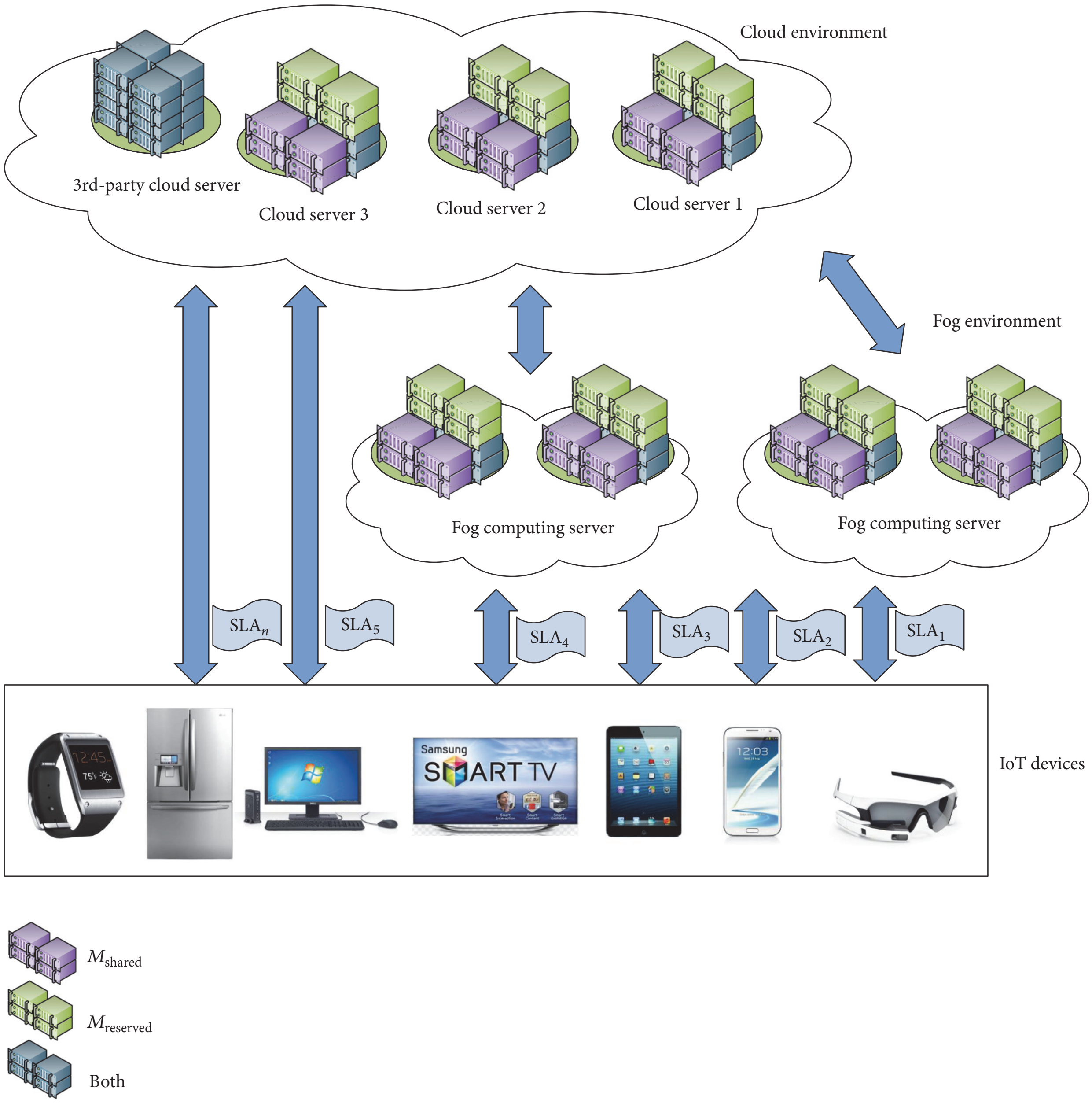


FIGURE 6: Illustrating our consideration of service level agreement (SLA).

decreases. As a result, reserved allocation can provide guarantee rate due to the offering of resources. For instance, when the number of SLAs is 1, then we need minimum number of less than 5 VMs to do the job. However, when the number of SLAs is 5, then the needed minimum number of VM for shared allocation is 10 VMs and more than 11 VMs for reserved allocation.

A comparison of processing time when sending big size of data to destination for our proposed system with other approaches [19] that utilize one single processor only is illustrated in Figure 12. For instance, by looking at Figure 12, we notice that our proposed approaches generate less processing time than other approaches when we try to send big size of data such as 400 mb. Moreover, our proposed approach shows better performance than other approaches which only use single processor [19]. Other approaches only use one processor where our approach uses multiple processor.

The result, concerning the number of fog/cloud computing environments with respect to IoT devices workload, is presented in Figure 13. We calculate the minimum number of fog/cloud computing environments which is able to satisfy IoT devices workload. The number of fog computing environments increases when the number of IoT devices workload increases and the same thing applies to cloud computing

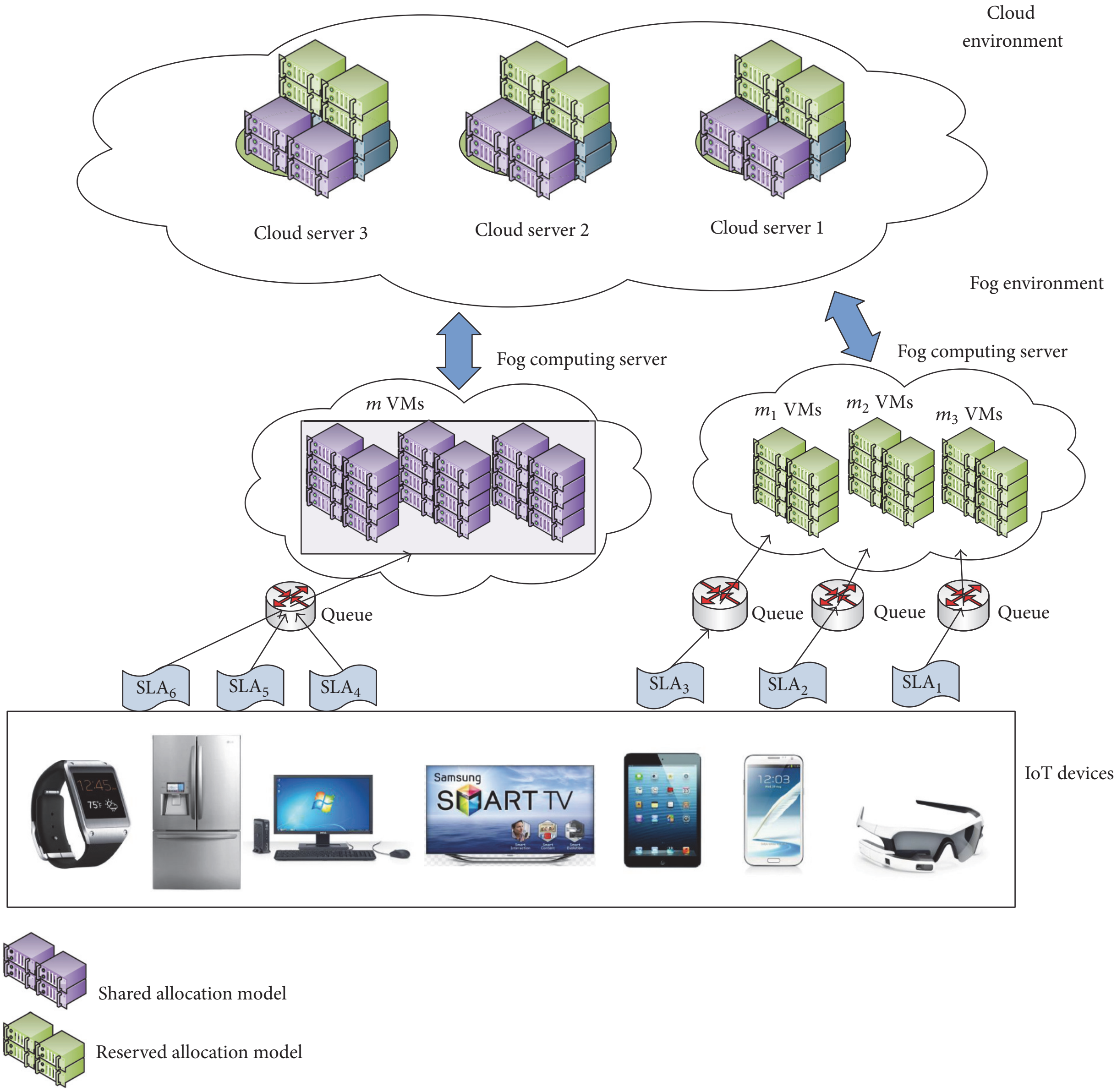


Figure 7: Illustrating our proposed strategy for resources allocation in shared allocation and reserved allocation.

when requested services are delegated to cloud computing. For instance, when the workload of IoT devices is 30 mb, then the minimum number of fog computing environments to satisfy IoT devices increase to 2 fog computing environments.

6. Conclusion

Smart IoT devices are growing rapidly and becoming smarter to access the Internet anytime, anywhere. Nevertheless, smart devices, services, and application are not able to fully benefit from this attractive cloud computing paradigm due to the following issues: (1) smart devices might be lacking in their capacity (e.g., processing, memory, storage, battery, and resource allocation), (2) they may be lacking in their network resources, and (3) the high network latency to centralized server in cloud might not be efficient for delay-sensitive application, services, and resource allocations requests. Moreover, sending or receiving big size of data from centralized server in cloud over the network degraded cloud performance and burden cloud network causing poor QoS, long response delay, and insufficient use of network resources. A localized environment such fog computing can be efficient in resolving the abovementioned issue. In spite of that, the rapid increasing number of services that will be requested from fog computing will generate overhead of services and less services requested from cloud which will result in poor management for both environment and poor QoS.

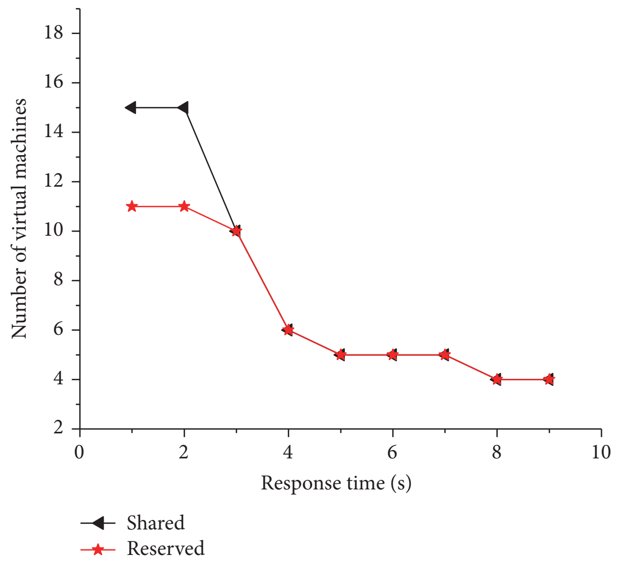


FIGURE 8: Showing different response time of shared and reserved allocation.

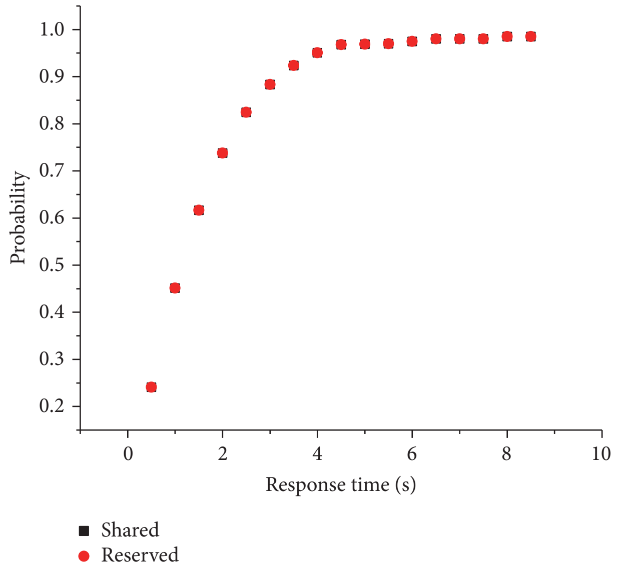


FIGURE 9: Showing SLA different target probability of shared and reserved allocation.

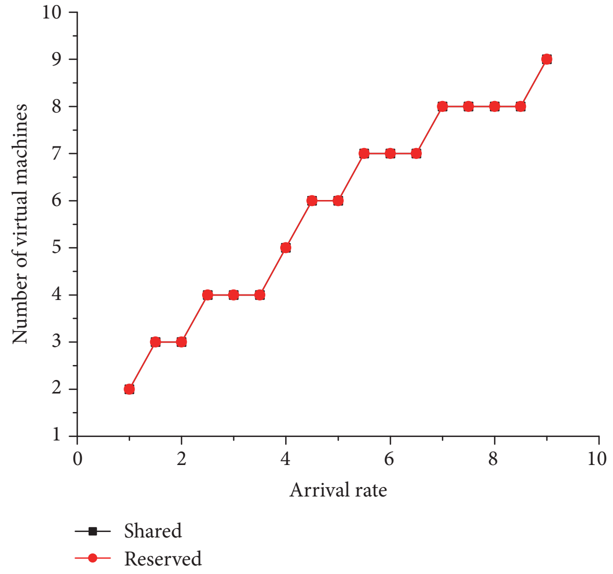


FIGURE 10: Showing the different arrival rate of shared and reserved allocation.

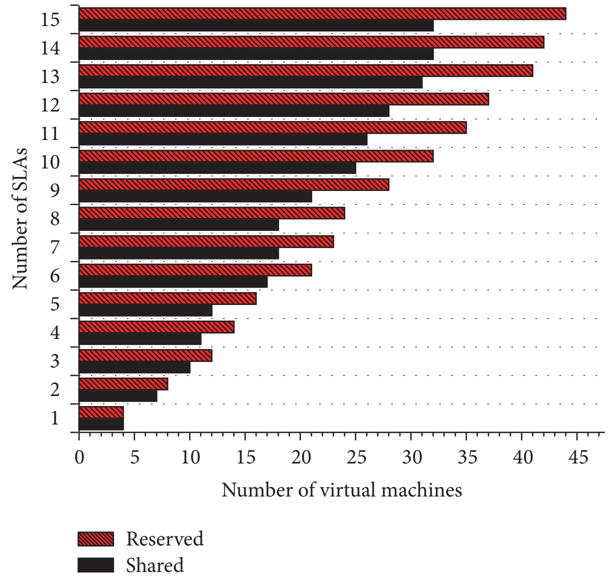


FIGURE 11: Showing different SLAs of shard and reserved allocation.

As a result, in this paper, we proposed an architecture of IoT service delegation and resource allocation based on collaboration between fog and cloud computing. We provide new algorithm that is decision rules of linearized decision tree based on three conditions (services size, completion time, and VMs capacity) for managing and delegating user request. Furthermore, we proposed new strategy for optimizing big data distribution in fog and cloud environment. Moreover, we propose algorithm to allocate resources to meet service level agreement (SLA) and QoS. Our simulation result shows that our proposed approach can improve services delegation, management, big data distribution, and resource allocation efficiently and show better performance than other existing methods.

Competing Interests

The authors declare that they have no competing interests.

Acknowledgments

This research is supported by the MSIP (Ministry of Science, ICT and Future Planning), Korea, under the ITRC (Information Technology Research Center) support Program (IITP-2016-(H8501-16-1015)) supervised by the IITP

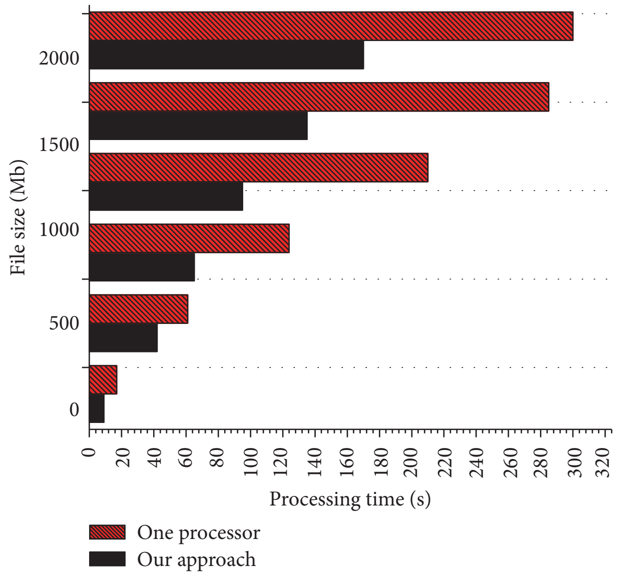


FIGURE 12: Showing comparison of other approaches (using one processor) with our approach (using multiprocessor).

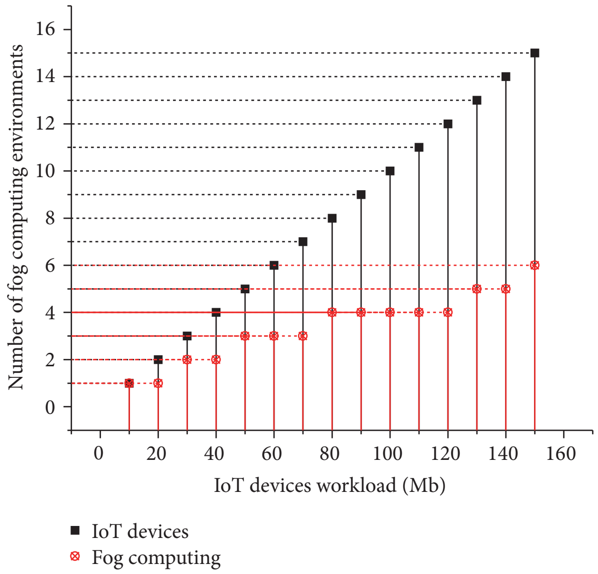


FIGURE 13: Showing the result of IoT devices workload comparing to the number of fog/cloud computing.

(Institute for Information and Communication Technology Promotion).

References

[1] Y. Pan and N. Hu, "Research on dependability of cloud computing systems," in *Proceedings of the 10th International Conference on Reliability, Maintainability and Safety (ICRMS '14)*, pp. 435–439, IEEE, Guangzhou, China, August 2014.

[2] W. Liu, "Research on cloud computing security problem and strategy," in *Proceedings of the 2nd International Conference on Consumer Electronics, Communications and Networks (CECNet '12)*, pp. 1216–1219, Yichang, China, April 2012.

[3] M. Aazam and E.-N. Huh, "Framework of resource management for intercloud computing," *Mathematical Problems in Engineering*, vol. 2014, Article ID 108286, 9 pages, 2014.

[4] M. Aazam and E. N. Huh, "Dynamic resource provisioning through fog micro datacenter," in *Proceedings of the 12th IEEE International Workshop on Managing Ubiquitous Communication and Services (MUCS '15)*, pp. 105–110, March 2015.

[5] S. Uppoor, M. D. Flouris, and A. Bilas, "Cloud-based synchronization of distributed file system hierarchies," in *Proceedings of the IEEE International Conference on Cluster Computing Workshops and Posters, Cluster*, pp. 1–4, September 2010.

[6] J. Delgado, S. M. Sadjadi, L. Fong, Y. Liu, N. Bobroff, and S. Seelam, "Efficiency assessment of parallel workloads on virtualized resources," in *Proceedings of the 4th IEEE International Conference on Utility and Cloud Computing (UCC '11)*, pp. 89–96, IEEE, Melbourne, Australia, December 2011.

[7] P. Fan, J. Wang, Z. Zheng, and M. R. Lyu, "Toward optimal deployment of communication-intensive cloud applications," in *Proceedings of the IEEE 4th International Conference on Cloud Computing (CLOUD '11)*, pp. 460–467, July 2011.

[8] M. Kwok, *Performance analysis of distributed virtual environments [Ph.D. thesis]*, University of Waterloo, Waterloo, Canada, 2006.

[9] G. Y. Jung, N. Gnanasambandam, and T. Mukherjee, "Synchronous parallel processing of big-data analytics services to optimize performance in federated clouds," in *Proceedings of the IEEE 5th International Conference on Cloud Computing (CLOUD '12)*, pp. 811–818, IEEE, Honolulu, Hawaii, USA, June 2012.

[10] Y. Hu, J. Wong, G. Iszlai, and M. Litoiu, "Resource provisioning for cloud computing," in *Proceedings of the Conference of the Center for Advanced Studies on Collaborative Research (CASCON '09)*, pp. 101–111, November 2009.

[11] J. Li, J. Chinneck, M. Woodside, and M. Litoiu, "Fast scalable optimization to configure service systems having cost and quality of service constraints," in *Proceedings of the 6th International Conference on Autonomic Computing (ICAC '09)*, pp. 159–168, June 2009.

[12] A. Lenk, M. Klems, J. Nimis, S. Tai, and T. Sandholm, "What's inside the cloud? An architectural map of the cloud landscape," in *Proceedings of the ICSE Workshop on Software Engineering Challenges of Cloud Computing (CLOUD '09)*, pp. 23–31, IEEE, Vancouver, Canada, May 2009.

[13] C. Kamalanathan, S. Valarmathy, and S. Kirubakaran, "Designing a fuzzy-logic based trust and reputation model for secure resource allocation in cloud computing," *The International Arab Journal of Information Technology*, vol. 13, no. 1, pp. 30–37, 2016.

[14] L. Xun, "From augmented reality to augmented computing: a look at cloud-mobile convergence," in *Proceedings of the International Symposium on Ubiquitous Virtual Reality (ISUVR '09)*, pp. 29–32, Gwangju, South Korea, July 2009.

[15] E. E. Marinelli, *Hyrax: cloud computing on mobile devices using MapReduce [M.S. thesis]*, Computer Science Department, CMU, Pittsburgh, Pa, USA, 2009.

[16] I. Giurgiu, O. Riva, D. Juric, I. Krivulev, and G. Alonso, "Calling the cloud: enabling mobile phones as interfaces to cloud applications," in *Middleware 2009*, J. M. Bacon and B. F. Cooper, Eds., vol. 5896 of *Lecture Notes in Computer Science*, pp. 83–102, Springer, New York, NY, USA, 2009.

[17] M. Andreolini, S. Casolari, and M. Colajanni, "Autonomic request management algorithms for geographically distributed internet-based systems," in *Proceedings of the 2nd IEEE International Conference on Self-Adaptive and Self-Organizing Systems (SASO '08)*, pp. 171–180, IEEE, Venice, Italy, October 2008.

[18] R. Sheldon, *Introduction to Probability Models*, Elsevier, 10th edition, 2010.

[19] H. C. Gonzalo and D. M. Lee, "A virtual cloud computing provider for mobile devices," in *Proceedings of the 1st ACM Workshop on Mobile Cloud Computing & Services: Social Networks and Beyond*, no. 6, pp. 1–5, San Francisco, Calif, USA, June 2010.

[20] Cloudsim, "A framework for modeling and simulation of cloud computing infrastructures and services," https://code.google.com/p/cloudsim/downloads/list.

Hindawi Publishing Corporation
Mobile Information Systems
Volume 2016, Article ID 3489193, 10 pages
http://dx.doi.org/10.1155/2016/3489193

Research Article

Multivariate Multiple Regression Models for a Big Data-Empowered SON Framework in Mobile Wireless Networks

Yoonsu Shin, Chan-Byoung Chae, and Songkuk Kim

School of Integrated Technology, Yonsei Institute of Convergence Technology, Yonsei University, Incheon, Republic of Korea

Correspondence should be addressed to Songkuk Kim; songkuk@yonsei.ac.kr

Received 22 April 2016; Accepted 26 July 2016

Academic Editor: Yeong M. Jang

In the 5G era, the operational cost of mobile wireless networks will significantly increase. Further, massive network capacity and zero latency will be needed because everything will be connected to mobile networks. Thus, self-organizing networks (SON) are needed, which expedite automatic operation of mobile wireless networks, but have challenges to satisfy the 5G requirements. Therefore, researchers have proposed a framework to empower SON using big data. The recent framework of a big data-empowered SON analyzes the relationship between key performance indicators (KPIs) and related network parameters (NPs) using machine-learning tools, and it develops regression models using a Gaussian process with those parameters. The problem, however, is that the methods of finding the NPs related to the KPIs differ individually. Moreover, the Gaussian process regression model cannot determine the relationship between a KPI and its various related NPs. In this paper, to solve these problems, we proposed multivariate multiple regression models to determine the relationship between various KPIs and NPs. If we assume one KPI and multiple NPs as one set, the proposed models help us process multiple sets at one time. Also, we can find out whether some KPIs are conflicting or not. We implement the proposed models using MapReduce.

1. Introduction

The technology of self-organizing networks (SON) has been developed to more economically manage wireless communication and mobile networks in increasingly complex environments [1, 2]. SON, however, do not fully handle data from all sources in mobile wireless networks such as mobile app-based data (mobile data) and channel baseband power (wireless communication information) [3, 4]. Thus, SON encounter challenges that hinder the current self-organizing networking paradigm from meeting the 5G requirements because 5G networks are more complex [4].

Engineers have thus come up with a big data-empowered SON (BSON), which develops a SON with big data in mobile wireless networks. BSON, currently a necessary technology for 5G [3–6], is still in its initial stage. Indeed, in its current iteration, it is insufficient for practical use. The BSON framework was proposed in [4] and includes the concrete concept of using big data in mobile wireless networks and applied them to SON. It ranks the key performance indicators (KPIs), selects network parameters (NPs) related to each KPI, and creates a Gaussian process regression model in which the KPI is the dependent variable and each NP related to this KPI is the independent variable. The Gaussian process regression models are then applied to the SON engine for management optimization. In this context, the KPIs include capacity, quality of service (QoS), capital expenditure (CAPEX), and operational expenditure (OPEX) from the perspective of a wireless communication operator. In addition, from a user perspective, the KPIs include seamless connectivity, spatiotemporal uniformity of service, demand for almost infinite capacity or zero latency, and cost of service. For instance, because 5G technology aims to connect everything such as automobile, wearable devices, and home network and to help human escape emergency situations, massive network capacity and zero latency are needed in a wireless ecosystem.

This BSON framework [4], however, has some aspects that need be improved. For example, the individual selection of NPs related to a KPI is considerably intricate because a typical 5G node is expected to have more than 2000 parameters. Moreover, in a single Gaussian process regression model, computing an exact KPI value according to the

NP values is difficult [7]. To address these problems, we proposed multiple regression models [8], which allow us easily distinguish the NPs related to each KPI from those unrelated ones. Simultaneously, we can generate models that can be immediately applied to the SON engine. Because of the many available NPs with massive values, we need to solve the issues concerning the multiplication of two large-sized matrices and the inverse of a large-sized matrix for multiple regressions using MapReduce. We describe and implement method that calculates matrices consisting of a KPI and NPs for multiple regression models using MapReduce [8].

These multiple regression models, however, suffer from weaknesses. We can calculate the relationship between only one KPI and the NPs at a single time. However, recognizing the relationship between various KPIs and all the NPs at a single time is important because some KPIs are conflicting, such as the relationship between QoS and CAPEX. If we want to know the relation between the KPIs and NPs, we need to individually calculate the multiple regressions of each KPI. Therefore, in this paper, we proposed improved models, namely, multivariate multiple regression models, that help us determine the relationship between the KPIs and NPs at a single time. We explain these models in the next section.

The remainder of this paper is organized as follows. Section 2 provides the background of big data in 5G and BSON framework, multiple regression models, MapReduce, and **LU** decomposition. Section 3 explores the proposed multivariate multiple regression models for the BSON framework and describes the implementation of these models using MapReduce. Section 4 presents the theoretical time complexity of the algorithms of these models. Section 5 presents the implementation of MapReduce and the execution time for executing these models in a cloud as a result. Finally, we conclude this paper in Section 6.

2. Background and Related Work

SON facilitates automatic operation of mobile wireless networks. It initially exploits big data in mobile wireless networks to improve the networks. This current research is devoted to BSON [3]. The researcher in [4] proposed the BSON framework.

2.1. SON. Operating mobile wireless networks is a challenging task, especially in cellular mobile communication systems due to their latent complexity. This complexity arises from the number of network elements and interconnections among their configurations. In heterogeneous networks, handling various technologies and their precise operational paradigms is difficult. Today, planning and optimization tools are typically semiautomated and the management tasks need to be closely supervised by human operators. This manual effort by a human operator is time-consuming, expensive, and error prone and requires a high degree of expertise. SON can be used to reduce the operating costs by reducing the tasks at hand and enhancing profit by minimizing human error. The next subsection details the SON taxonomies.

2.1.1. Self-Configuration. Configuration of base stations (eNBs), relay stations, and femtocells is required during deployment, extension, and upgrade of network terminals. Configurations may also be needed when a change in the system is required, such as failure of a node, drop in network performance, or change in service type. In future systems, the conventional process of manual configuration must be replaced with self-configuration. We can foresee that nodes in future cellular networks should be able to self-configure all their initial parameters including the IP addresses, neighbor lists, and radio-access parameters.

2.1.2. Self-Optimization. After the initial self-configuration phase, we need to continuously optimize the system parameters to ensure efficient performance of the system to maintain all optimization objectives. Optimization in the legacy systems can be done through periodic drive tests or analysis from log reports generated from network operating center. Self-optimization includes load balancing, interference control, coverage extension, and capacity optimization.

2.1.3. Self-Healing. Wireless cellular systems are prone to faults and failures due to component malfunctions or natural disasters. In traditional systems, failures are mainly detected by the centralized operation and maintenance (O&M) software. Events are recorded and necessary alarms are set off. When alarms cannot be remotely cleared, radio network engineers are usually mobilized and sent to cell sites. This process could take days or even weeks before the system returns to normal operation. In future self-organized cellular systems, this process needs to be improved by consolidating the self-healing functionality. Self-healing is a process that consolidates remote detection, diagnosis, and triggering of compensation or recovery actions to minimize the effect of faults in the mobile wireless network equipment.

2.2. Big Data in 5G and BSON. The massive amount of information comes from various elements in the mobile wireless networks, such as base stations, mobile terminals, gateways, and management entities, as shown in Figure 1 [3]. The authors in [4] classified the big data in cellular networks as follows.

2.2.1. Subscriber Level Data. This classification contains control data, contextual data, and voice data, which not only can be used to optimize, configure, and plan network-centric operations, but also are equally meaningful to support key business processes such as customer experience and retention enhancement.

2.2.2. Cell Level Data. This classification contains physical layer measurements that are reported by a base station and all user equipment within the coverage of this base station to the O&M center. The utilities of the cell level data can complement the subscriber level data. For example, minimization of drive test measurements, which contains the reference signal received power and reference signal received quality values of the serving and adjacent cells, are particularly useful for autonomous coverage estimation and optimization [9].

2.2.3. Core Network Level Data. This classification can be exploited to fully automate fault detection and troubleshoot

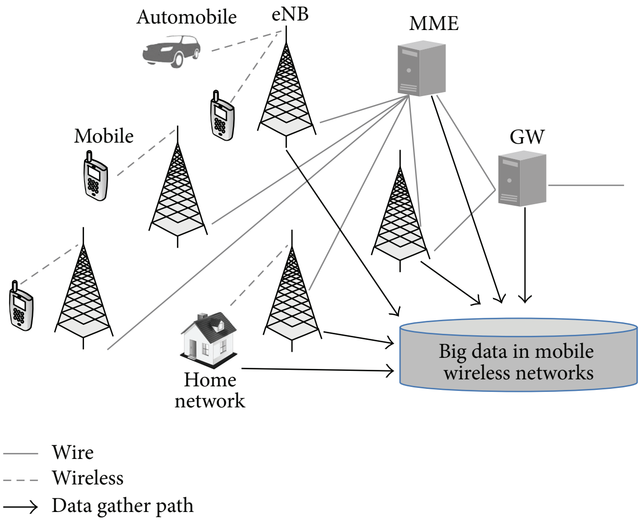


Figure 1: Big data gathering path in mobile wireless network architecture.

network level problems. The complexity of identifying problems in a core network is increased many times, particularly if the equipment used is supplied by different vendors that provide their own proprietary solutions for different network performance.

2.2.4. Additional Sources of Data. This classification contains the structured information already stored in the separate databases, including customer relationship management as well as billing data. This also includes unstructured information such as social media feeds, specific application usage patterns, and data from smart phone built-in sensors and applications.

As discussed in the Introduction, SON technology uses this aforementioned big data to improve itself. This process is facilitated using BSON. The three main features that make BSON distinct from the state-of-the-art SON are the following:

(i) full intelligence of the current network status,

(ii) capability in predicting user behavior,

(iii) capability in dynamically associating the network response to the NPs.

These three capabilities can go a long way in designing a SON that can meet the 5G requirements. The BSON framework shown in Figure 2 involves the following steps.

Step 1 (data gathering). This includes gathering of data from all sources of information into an aggregate data set.

Step 2 (transforming). This includes transforming the big data into right data.

The steps in this transformation are explained below. The underlying machine learning and data analytics are subsequently explained.

(1) *Classifying*. This means classifying the data with respect to key operational and business objectives (OBOs) in which accessibility, retainability, integrity, mobility, and business intelligence are present.

(2) *Unifying/Diffusing*. This means unifying multiple PIs into more significant KPIs.

(3) *Ranking*. This means ranking KPIs within each OBO with respect to their effect on that OBO.

(4) *Filtering*. This means filtering out KPIs that affect the OBO below a predefined threshold.

(5) *Relating*. This means, for each KPI, finding the NP that affects that KPI.

(6) *Ordering*. This means, for each KPI, ordering the associated NP with respect to the strength of their association.

(7) *Cross-Correlation*. This means, for each NP, determining a vector that quantifies its association with each KPI.

Step 3 (modeling). This includes developing a network behavior model by learning from the right data obtained in Step 2 using the Gaussian process regression and Kolmogorov-Wiener prediction.

Step 4 (running the SON engine). This includes using the SON engine on the model to determine a new NP and expected new KPIs.

Step 5 (validating). If the simulated behavior tallies with the expected behavior (KPIs), proceed with the new NPs.

Step 6 (relearning/improving). If the validation in Step 5 fails, make feedback to the concept drift block, which updates in turn the behavior model.

2.3. Multiple Regression Models [8, 10]. Step 2 (transforming) and Step 3 (modeling) presented in Section 2.2 (BSON framework) are replaced with the multiple regression models. The key factors in Step 2 (transforming) and Step 3 (modeling) are finding the associated NPs for each KPI and creating the model using a KPI and the associated NPs. They should, however, separately determine the associated NPs using machine-learning tools [11]. Moreover, calculating the accurate value of a KPI according to the change in the NP values is difficult. In other words, the model presented in Section 2.2 allows us to determine the value of a KPI according to only one NP because the model is merely a single regression model.

The single regression model shown in Figure 2 identifies the relationship between a KPI and only one NP. Of course, many single regression models exist according to the NPs, but calculating a KPI value when the NP values simultaneously change is difficult. In contrast, the multiple regression models shown in Figure 3 enable easy identification of the relationship between a KPI and the NPs.

We proposed the multiple regression models to enhance the previous BSON framework [8]. The multiple regression model is written in [10] as

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_m x_m + \varepsilon, \qquad (1)$$

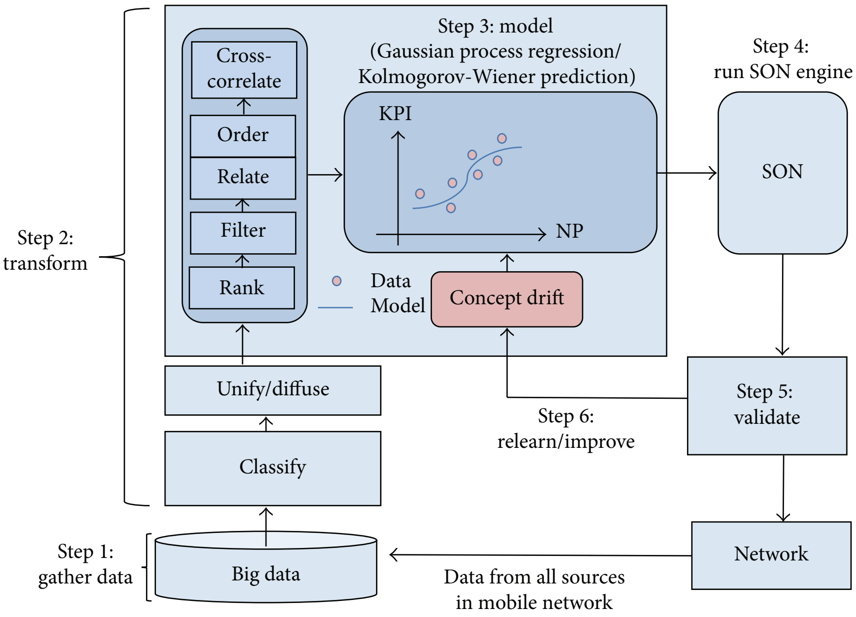


FIGURE 2: Big data-empowered SON framework [4].

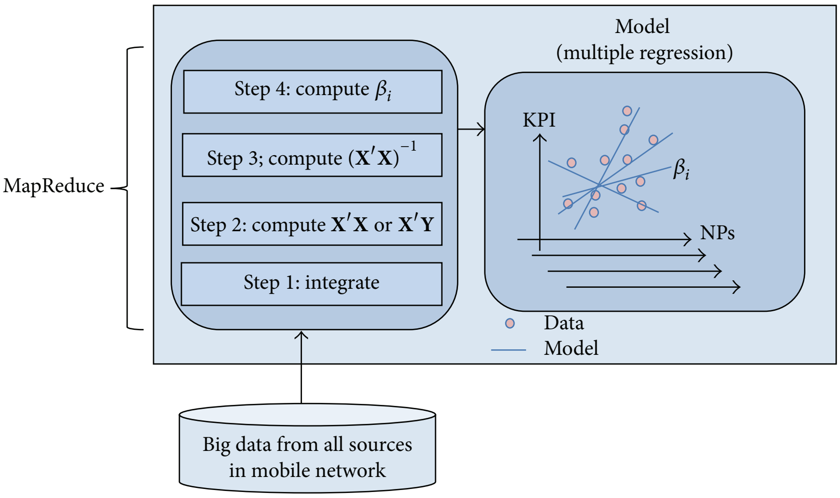

Model
(multiple regression)
Step 4: compute β_i
Step 3; compute $(\mathbf{X}'\mathbf{X})^{-1}$
Step 2: compute $\mathbf{X}'\mathbf{X}$ or $\mathbf{X}'\mathbf{Y}$
Step 1: integrate
MapReduce
KPI
β_i
NPs
Data
Model
Big data from all sources
in mobile network

FIGURE 3: Multiple regression models.

and it can be expressed as

$$\mathbf{Y} = \mathbf{X}\mathbf{B} + \mathbf{E}, \tag{2}$$

where

$$\mathbf{X} = \begin{bmatrix} 1 & \mathrm{NP}_1 & \mathrm{NP}_2 & \cdots \\ 1 & \cdot & & \\ 1 & \cdot & & \\ 1 & \cdot & & \end{bmatrix}, \qquad \mathbf{Y} = \begin{bmatrix} \mathrm{KPI}_k \\ \cdot \\ \cdot \\ \cdot \end{bmatrix}. \tag{3}$$

The elements in **X** and **Y** are the values of the NPs and KPI, and the parameter is estimated as

$$\widehat{\mathbf{B}} = \left(\mathbf{X}'\mathbf{X}\right)^{-1}\left(\mathbf{X}'\mathbf{Y}\right). \tag{4}$$

We can create multiple regression models ($\widehat{\mathbf{B}}$) by calculating the multiplication of $(\mathbf{X}'\mathbf{X})^{-1}$ and $(\mathbf{X}'\mathbf{Y})$. Figure 3 shows four steps to compute $\widehat{\mathbf{B}}$ using MapReduce, and we provided the detail of each step in [8].

2.4. Matrix Multiplication Using MapReduce [12, 13]. MapReduce is a computation method that has been implemented in several systems, including Google internal implementation and the popular open-source implementation Hadoop. (Hadoop can be obtained, along with Hadoop Distributed

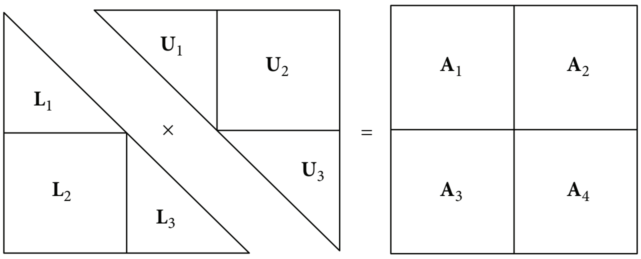


Figure 4: Block method for **LU** decomposition.

File System from the Apache Foundation.) We can use an implementation of MapReduce to manage many large-scale computations in a manner that is tolerant of hardware faults. Only two functions need to be written—Map and Reduce—while the system manages the parallel execution, coordinates tasks that execute Map or Reduce, and deals with the possibility that one of these tasks will fail to execute.

Matrix Multiplication with One MapReduce Step. If $\mathbf{M}$ is a matrix with element m_{ij} in row i and column j and $\mathbf{N}$ is a matrix with element n_{jk} in row j and column k, then the product, $\mathbf{P} = \mathbf{MN}$, is matrix $\mathbf{P}$ with element p_{ik} in row i and column k, where

$$p_{ik} = \sum_j m_{ij} n_{jk}. \tag{5}$$

We can possibly use only a single MapReduce pass to perform the matrix multiplication, $\mathbf{P} = \mathbf{MN}$. Here, we present an abstract of the Map and Reduce functions.

(1) *Map Function*. For each element m_{ij} of $\mathbf{M}$, we produce all key-value pairs $((i,k), (M, j, m_{ij}))$ for $k = 1, 2, \ldots$ up to the number of columns of $\mathbf{N}$. Similarly, for each element n_{ij} of $\mathbf{N}$, we produce all key-value pairs $((i,k), (N, j, n_{ij}))$ for $i = 1, 2, \ldots$ up to the number of columns of $\mathbf{M}$.

(2) *Reduce Function*. Each key (i,k) will have an associated list with all values (M, j, m_{ij}) and (N, j, n_{ij}), for all possible values of j. The jth values on each list must have their third components, namely, m_{ij} and n_{jk}, extracted and multiplied. Then, these products are added, and the result is paired with (i,k) in the output of the Reduce function.

2.5. Matrix Inversion Using MapReduce [14]. The **LU** algorithm splits the matrix into square submatrices and individually updates these submatrices. The block method splits the input matrix, as shown in Figure 4.

In this method, the lower triangular matrix **L** and the upper triangular matrix **U** are both split into three submatrices, whereas the original matrix **A** is split into four submatrices. These smaller matrices satisfy the following equations:

$$\begin{aligned}
\mathbf{L}_1\mathbf{U}_1 &= \mathbf{P}_1\mathbf{A}_1, \\
\mathbf{L}_1\mathbf{U}_2 &= \mathbf{P}_1\mathbf{A}_2, \\
\mathbf{L}_2'\mathbf{U}_1 &= \mathbf{A}_3, \\
\mathbf{L}_3\mathbf{U}_3 &= \mathbf{P}_2\left(\mathbf{A}_4 - \mathbf{L}_2'\mathbf{U}_2\right), \\
\mathbf{L}_2 &= \mathbf{P}_2\mathbf{L}_2',
\end{aligned} \tag{6}$$

where both $\mathbf{P}_1$ and $\mathbf{P}_2$ are permutations of the rows. The entire **LU** decomposition can be represented as

$$\mathbf{LU} = \begin{pmatrix} \mathbf{P}_1 & 0 \\ 0 & \mathbf{P}_2 \end{pmatrix} \mathbf{A} = \mathbf{PA}, \tag{7}$$

where $\mathbf{P}$ is also a permutation of the rows obtained by augmenting $\mathbf{P}_1$ and $\mathbf{P}_2$.

If submatrix $\mathbf{A}_1$ is sufficiently small (e.g., on the order of 10^3 or less), it can be very efficiently decomposed into $\mathbf{L}_1$ and $\mathbf{U}_1$ on a single node. If submatrix $\mathbf{A}_1$ is not small enough, we can recursively partition it into smaller submatrices, as shown in Figure 4. After obtaining $\mathbf{L}_1$ and $\mathbf{U}_1$, the elements of $\mathbf{L}_2'$ and $\mathbf{U}_2$ can be computed using the following two equations:

$$\begin{aligned}
\left[\mathbf{L}_2'\right]_{ij} &= \frac{1}{\left[\mathbf{U}_1\right]_{jj}}\left(\left[\mathbf{A}_3\right]_{ij} - \sum_{k=1}^{j-1}\left[\mathbf{L}_2'\right]_{ik}\left[\mathbf{U}_1\right]_{kj}\right), \\
\left[\mathbf{U}_2\right]_{ij} &= \frac{1}{\left[\mathbf{L}_1\right]_{ii}}\left(\left[\mathbf{A}_2\right]_{ij} - \sum_{k=1}^{j-1}\left[\mathbf{L}_1\right]_{ik}\left[\mathbf{U}_2\right]_{kj}\right).
\end{aligned} \tag{8}$$

We can compute $\mathbf{A}_4 - \mathbf{L}_2'\mathbf{U}_2$ using the $\mathbf{L}_2'$ and $\mathbf{U}_2$ matrices mentioned above. Subsequently, we can decompose it into $\mathbf{L}_3$ and $\mathbf{U}_3$.

3. Multivariate Multiple Regression Models for BSON Framework

The multiple regression models presented in Section 2.3 suffer from a shortcoming—they can calculate the relationship between only one KPI and NPs. Many KPIs exist, however, such as those from the operator perspective that include OPEX, CAPEX, QoS, and capacity and from the user perspective that include seamless connectivity, cost of service, capacity, and latency [4]. These are high-level KPIs; however, many precise technical KPIs also exist, such as the cell power and cell coverage. To reveal the relationship between the KPIs and NPs, we must calculate the multiple regression models several times for each KPI in the previous multiple regression models. This process is inconvenient and requires a long time.

Meanwhile, finding the conflicting or concordant relationship among KPIs is not easy when the NP values simultaneously change. As we mentioned earlier, we should perform multiple regressions several times for each KPI to finally learn the conflicting or concordant relationship among KPIs.

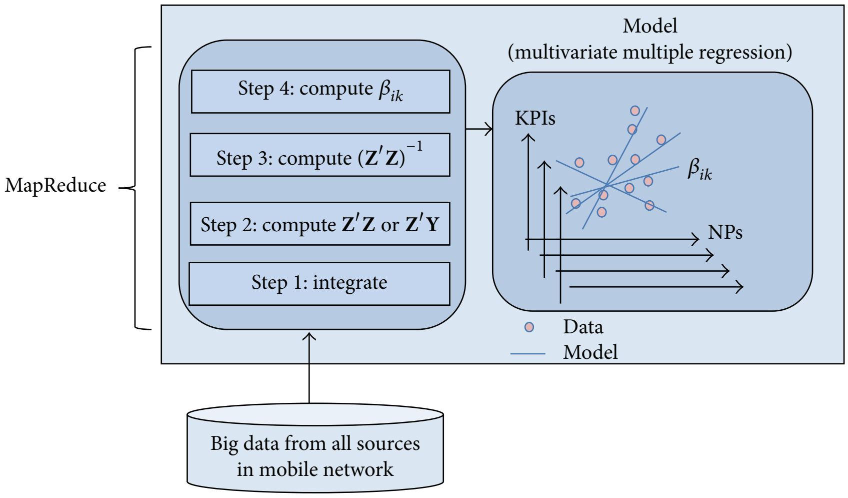

MapReduce
Step 4: compute β_{ik}
Step 3: compute $(\mathbf{Z}'\mathbf{Z})^{-1}$
Step 2: compute $\mathbf{Z}'\mathbf{Z}$ or $\mathbf{Z}'\mathbf{Y}$
Step 1: integrate
Model
(multivariate multiple regression)
KPIs
β_{ik}
NPs
Data
Model
Big data from all sources
in mobile network

Figure 5: Proposed multivariate multiple regression models.

In contrast, the proposed multivariate multiple regression models shown in Figure 5 allow simultaneous determination of the relationship between the KPIs and NPs.

To enhance the multiple regression models for BSON, we propose the multivariate multiple regression models. The multivariate multiple regression is expressed as follows [15, 16]:

$$Y_j = \beta_0 + \beta_1 z_{j1} + \beta_2 z_{j2} + \cdots + \beta_r z_{jr} + \varepsilon_j, \tag{9}$$

and it can also be expressed as

$$\mathbf{Y}_{n\times p} = \mathbf{Z}_{n\times(r+1)}\mathbf{B}_{(r+1)\times p} + \varepsilon_{n\times p}, \tag{10}$$

where

$$\mathbf{Z} = \begin{bmatrix} 1 & \mathrm{NP}_1 & \mathrm{NP}_2 & \cdots & \mathrm{NP}_r \\ 1 & \cdot & & & \\ 1 & \cdot & & & \\ 1 & \cdot & & & \end{bmatrix},$$
$$\mathbf{Y} = \begin{bmatrix} \mathrm{KPI}_1 & \mathrm{KPI}_2 & \cdots & \mathrm{KPI}_p \\ \cdot & & & \\ \cdot & & & \\ \cdot & & & \end{bmatrix}. \tag{11}$$

The elements in **Z** and **Y** are values of NPs and KPIs, and the parameter is estimated as

$$\widehat{\mathbf{B}}_{(r+1)\times p} = \left(\mathbf{Z}'\mathbf{Z}\right)^{-1}\left(\mathbf{Z}'\mathbf{Y}\right). \tag{12}$$

We can create multivariate multiple regression models ($\widehat{\mathbf{B}}$) by calculating the multiplication of $(\mathbf{Z}'\mathbf{Z})^{-1}$ and $(\mathbf{Z}'\mathbf{Y})$. Figure 5 shows four steps to compute $\widehat{\mathbf{B}}$ using MapReduce, and we specifically describe each step below.

Step 1 (integrating). Each message has limited information such as the location, time, reception sensitivity, cell power, mobile power, data traffic, and mobility status. Hence, we simultaneously integrate the whole messages to determine the values of the KPIs according to the NPs in the Map function. Then, we extract the values of the KPI and all the NPs in the Reduce function. The MapReduce key-value pair in Step 1 is presented in Algorithm 1.

In the Map function, the key is time, and the value is the name and value of each NP and KPI. When the Map tasks are all completed, the key-value pairs are grouped in terms of time. Thus, the input of the Reduce task contains the corresponding information and the key-value pairs are grouped according to each KPI (i.e., KPI_k) in the Reduce tasks. Therefore, we can simultaneously obtain the value of each KPI and NP as the output of the Reduce tasks.

For example, if we take one sample per minute for 1 hour, we can obtain 60 samples. Assuming that the numbers of NPs and KPIs are 30 and 10, respectively, then the orders of **Z** and **Y** are 60 × 30 and 60 × 10, respectively. Therefore, we can convert key (i.e., time_ℓ), NP_m elements and KPI_n elements in the Reduce function into the ℓth row of **Z** and **Y** and the mth column of **Z** and nth column of **Y**, respectively.

Algorithm 1 (the MapReduce key-value pair of Step 1).

The Map Function

{time, (NP_1, NP_1 value, NP_2, NP_2 value, …, KPI_1, KPI_1 value, KPI_2, KPI_2 value, …)}

The Reduce Function

{time, (NP_1, NP_1 value, NP_2, NP_2 value, …, KPI_1, KPI_1 value, KPI_2, KPI_2 value, …)}

Step 2 (computing $\mathbf{Z}'\mathbf{Z}$ and $\mathbf{Z}'\mathbf{Y}$). We compute $\mathbf{Z}'\mathbf{Z}$ and $\mathbf{Z}'\mathbf{Y}$ using the result in Step 1. Because the result in Step 1 includes the **Z** and **Y** matrices, we can easily compute $\mathbf{Z}'\mathbf{Z}$ and $\mathbf{Z}'\mathbf{Y}$ using MapReduce. As noted in Section 2.4, we can obtain matrix multiplication with one MapReduce step [12]. For instance, if we calculate matrix multiplication, $\mathbf{P} = \mathbf{MN}$, m_{ik} is used to obtain $p_{i1}, p_{i2}, \ldots, p_{ij}$ (j is the number of columns in **N**). Therefore, through m_{ik} forking off the jth elements in the Map function, we can calculate the element of $\mathbf{P}_{ij}$ in the Reduce function at the same time.

The MapReduce key-value pair in Step 2 is presented in Algorithm 2. Note that $Z'Z, Z'Y, Z', Z$, or Y are the names of these matrices and not of the entire matrix. Note also that k reaches up to the number of samples (i.e., time), i reaches up to the number of NPs plus one, and ℓ reaches up to the number of KPIs.

Algorithm 2 (the MapReduce key-value pair of Step 2).

The Map Function

$\{(Z'Z, i, j), (Z', k, z'_{ik})\}$ for $j = 1, 2, \ldots$ up to the number of columns of **Z**

$\{(Z'Z, i, j), (Z, k, z_{kj})\}$ for $i = 1, 2, \ldots$ up to the number of rows of $\mathbf{Z}'$

or

$\{(Z'Y, i, \ell), (Z', k, z'_{ik})\}$ for $\ell = 1, 2, \ldots$ up to the number of columns of **Y**

$\{(Z'Y, i, \ell), (Y, k, y_{i\ell})\}$ for $i = 1, 2, \ldots$ up to the number of rows of $\mathbf{Z}'$

The Reduce Function

$\{(Z'Z, i, j), (\mathbf{Z}'\mathbf{Z}_{ij}\ \text{value})\}$ or

$\{(Z'Y, i, \ell), (\mathbf{Z}'\mathbf{Y}_{i\ell}\ \text{value})\}$

Step 3 (computing $(\mathbf{Z}'\mathbf{Z})^{-1}$). To calculate the multivariate multiple regression, we compute $(\mathbf{Z}'\mathbf{Z})^{-1}$ using the result in Step 2. However, computing the inverse of a matrix using MapReduce is difficult when the order of the matrix is large. Fortunately, the authors in [14] proposed a method of matrix inversion using MapReduce. They proposed a block method for scalable matrix inversion using MapReduce. The block method enables parallel calculation of the **LU** decomposition. If the order of the matrices is not large ($\leq 10^3$), the matrix can be very efficiently decomposed into **L** and **U** on a single node. If the order of the matrices is not large, sequentially calculating the inverse of a matrix using **LU** decomposition in one node becomes easy. We can compute the **L** and **U** matrices using the following equations for the **LU** decomposition algorithm [14, 17]:

$$\begin{gathered} u_{ij} = a_{ij} - \sum_{k=1}^{j-1} \ell_{jk} u_{kj}, \\ \ell_{ij} = \frac{1}{u_{jj}} \left(a_{ij} - \sum_{k=1}^{j-1} \ell_{ik} u_{kj} \right). \end{gathered} \tag{13}$$

We can then easily compute $\mathbf{L}^{-1}$ using the following equations [14], and the inverse of the upper triangular matrix $(\mathbf{U}^{-1})$ can be equivalently computed. We invert upper triangular matrix, **U**, by calculating the inverse of $\mathbf{U}^{\mathbf{T}}$, which is a lower triangular matrix (**L**):

$$\left[\mathbf{L}^{-1}\right]_{ij} = \begin{cases} 0 & \text{for } i < j \\ \dfrac{1}{\left[\mathbf{L}_{ii}\right]} & \text{for } i = j \\ -\dfrac{1}{\left[\mathbf{L}_{ii}\right]} \displaystyle\sum_{k=j}^{i-1} \left[\mathbf{L}\right]_{ik} \left[\mathbf{L}^{-1}\right]_{kj} & \text{for } i > j. \end{cases} \tag{14}$$

The output key-value pair in Step 3 is presented in Algorithm 3. Note that $(Z'Z)^{-1}$ is the name of this matrix, and not of the entire matrix.

Algorithm 3 (the output key-value pair of Step 3).

$\{((Z'Z)^{-1}, i, j), ((\mathbf{Z}'\mathbf{Z})_{ij}^{-1}\ \text{value})\}$

Step 4 (computing $\widehat{\mathbf{B}}$). We compute $\widehat{\mathbf{B}} = (\mathbf{Z}'\mathbf{Z})^{-1}\mathbf{Z}'\mathbf{Y}$ using the results in Steps 2 and 3. We perform the multiplication of two matrices (i.e., $(\mathbf{Z}'\mathbf{Z})^{-1}$ and $\mathbf{Z}'\mathbf{Y}$) using MapReduce. We can also perform matrix multiplication using one MapReduce step such as in Step 2 [12].

The MapReduce key-value pair in Step 4 is presented in Algorithm 4. Note that $(Z'Z)^{-1}$ and $(Z'Y)$ are the names of these matrices and not of the entire matrix in the Map function. In the Reduce function, the jth element of $(\mathbf{Z}'\mathbf{Z})^{-1}$ multiplies the jth element of $\mathbf{Z}'\mathbf{Y}$ in same (i, k) key; then all the results are added. The result is the (i, k) element of $\widehat{\mathbf{B}}$. In the Reduce function, note that i reaches up to the number of NPs plus one, and k reaches up to the number of KPIs.

Algorithm 4 (the MapReduce key-value pair of Step 4).

The Map Function

$\{(i, k), ((Z'Z)^{-1}, j, (\mathbf{Z}'\mathbf{Z})_{ij}^{-1})\}$ for $k = 1, 2, \ldots$ up to number of rows of $(\mathbf{Z}'\mathbf{Y})$

or

$\{(i, k), ((Z'Y), j, (\mathbf{Z}'\mathbf{Y})_{jk})\}$ for $i = 1, 2, \ldots$ up to number of rows of $(\mathbf{Z}'\mathbf{Z})^{-1}$

The Reduce Function

$\{(i, k), (\beta_{(i-1)k})\}$

We can recognize that estimated parameters (i.e., $\beta_{(i-1)k}$) separate the NPs from the NPs unrelated to a KPI. If $\beta_{(i-1)k}$ is close to zero at KPI_k, then NP_{i-1} is unrelated to KPI_k. In addition, we can identify whether a conflicting or concordant relationship among KPIs exists. For example, if the sign of all row elements of β_{ip} and β_{iq} for KPI_p and KPI_q are totally different, these KPIs are conflicting. Otherwise, they are concordant.

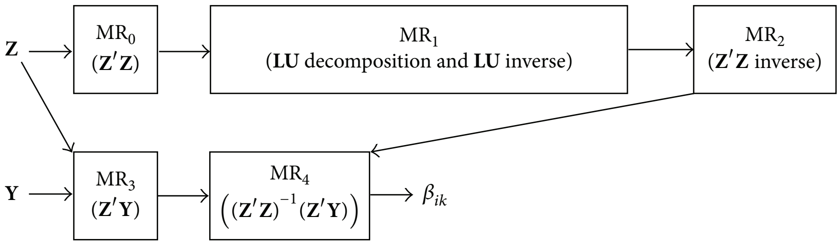

Z
Y
MR$_0$ (Z′Z)
MR$_1$ (LU decomposition and LU inverse)
MR$_2$ (Z′Z inverse)
MR$_3$ (Z′Y)
MR$_4$ $((\mathbf{Z}'\mathbf{Z})^{-1}(\mathbf{Z}'\mathbf{Y}))$
β_{ik}

Figure 6: MapReduce pipeline for estimate parameter (β_{ik}).

Table 1: Time complexity of matrix multiplication.

Matrix	Input order	Output order	Time complexity
$\mathbf{Z}' \times \mathbf{Z}$	$(r+1)\times n$ $n \times (r+1)$	$(r+1)\times(r+1)$	$O(r^2 n)$
$(\mathbf{Z}'\mathbf{Z})^{-1}$	$(r+1)\times(r+1)$	$(r+1)\times(r+1)$	$O(r^3)$
$\mathbf{Z}' \times \mathbf{Y}$	$(r+1)\times n$ $n \times p$	$(r+1)\times p$	$O(nrp)$
$(\mathbf{Z}'\mathbf{Z})^{-1} \times (\mathbf{Z}'\mathbf{Y})$	$(r+1)\times(r+1)$ $(r+1)\times p$	$(r+1)\times p$	$O(r^2 p)$

4. Time Complexity of Multiple Regression Models

We calculate the time complexity of the multivariate multiple regression models. The result of the multivariate multiple regression models can be obtained as the product of $(\mathbf{Z}'\mathbf{Z})^{-1}$ and $\mathbf{Z}'\mathbf{Y}$. The time complexity of $\mathbf{Z}'\mathbf{Z}$ is $O(r^2 n)$ because the order of $\mathbf{Z}$ is $n \times (r+1)$. The time complexities of $\mathbf{Z}'\mathbf{Y}$, $(\mathbf{Z}'\mathbf{Z})^{-1}$ and $(\mathbf{Z}'\mathbf{Z})^{-1}\mathbf{Z}'\mathbf{Y}$ are $O(n \times r \times p)$, $O(r^3)$, and $O(r^2 \times p)$, respectively, as listed in Table 1 [18, 19]. Thus, the entire time complexity of the multivariate multiple regression models is $O(r^3)$ when $n < r$.

We can reduce this time complexity using distributed programming such as MapReduce. Let $\mathbf{T}(\mathbf{L})$ be the time complexity of N tasks. $\mathbf{T}(\mathbf{L})$ can then be presented as follows, assuming an ideal case without consideration of a network bottleneck:

$$\mathbf{T}(\mathbf{L}) = \frac{\mathbf{T}(\mathbf{1})}{\mathbf{L}}. \tag{15}$$

Thus, the time complexity of the N tasks is $O(r^3/\mathbf{L})$, and if $\mathbf{L}$ is sufficiently large, we can obtain almost constant or linear time complexity, which shows that the time complexity of the proposed models is equal to that of the multiple regression models [8].

5. Implementation in MapReduce

We implemented our models using Hadoop 2.7.1 [20, 21]. All experiments were performed in our laboratory cluster, which has 32 machines. Each machine has four CPU cores and 24 GB of memory, where each CPU is an Intel® Xeon® CPU X5650 at 2.67 GHz.

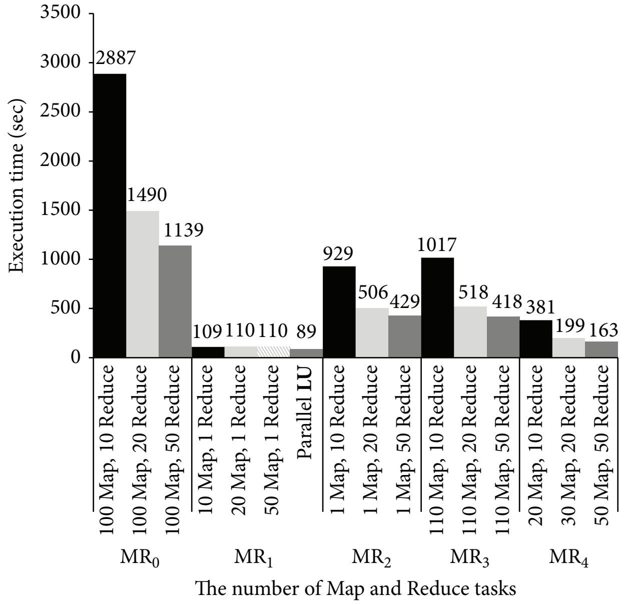


Figure 7: Execution time for calculating each phase (MR$_i$) according to the number of tasks.

For implementation in MapReduce, several phases were required. Thus, we had a pipeline of MapReduce jobs as shown in Figure 6. MR$_i$ is one MapReduce job. Three phases are required to calculate $(\mathbf{Z}'\mathbf{Z})^{-1}$.

In MR$_0$, we computed the product of $\mathbf{Z}'$ and $\mathbf{Z}$. In MR$_1$, we computed the $\mathbf{L}$ and $\mathbf{U}$ matrices using (13). In addition, in MR$_1$, we can easily compute $\mathbf{L}^{-1}$ using (14), and the inverse of the upper triangular matrix ($\mathbf{U}^{-1}$) can be equivalently computed. We inverted upper triangular matrix, $\mathbf{U}$, by calculating the inverse of $\mathbf{U}^{\mathbf{T}}$, which is a lower triangular matrix ($\mathbf{L}$). Finally, in MR$_2$, we computed $(\mathbf{Z}'\mathbf{Z})^{-1}$ as the product of $\mathbf{U}^{-1}$ and $\mathbf{L}^{-1}$.

Meanwhile, MR$_3$ is required to calculate $\mathbf{Z}'\mathbf{Y}$. From the output of MR$_2$ and MR$_3$, we can calculate estimated parameters (i.e., β_{ik}) as the product of $(\mathbf{Z}'\mathbf{Z})^{-1}$ and $\mathbf{Z}'\mathbf{Y}$. In reference to Section 3, Step 1 phase creates $\mathbf{Z}$ and $\mathbf{Y}$. Step 2 presents MR$_0$ and MR$_3$. Step 3 presents MR$_1$ and MR$_2$. Finally, Step 4 presents MR$_4$.

In this implementation, we compared the execution time according to the number of MapReduce jobs as shown in Figure 7. We used 600 × 400 matrix as input $\mathbf{Z}$ and 600 × 100

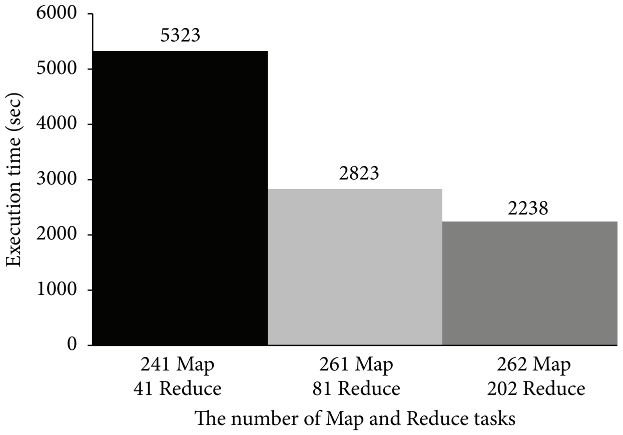


Figure 8: Execution time for calculating estimated parameter (β_{ik}).

matrix as input **Y**. Thus, the order of estimated parameter (i.e., β_{ik}) was 400 × 100. In a practical experiment, we need to calculate a large order of matrices. Much time, however, is needed to calculate matrix multiplication in our cloud when the matrices are in a large order. Hence, we reduced the order of matrices and simply compared the execution times according to the number of tasks.

Figure 7 shows the execution time for calculating each phase (i.e., MR_i). In Figure 7, the execution times of MR_0, MR_2, MR_3, and MR_4 are linearly reduced when the number of Reduce tasks increases from 10 to 20. They, however, later gradually decreased when the number of Reduce tasks increases from 20 to 50 because network bottleneck, communication cost, or additional management time exists [22, 23].

To the left of the three bars in MR_1 in Figure 7, we can see the execution time for calculating MR_1 on a single node. No reduction in the execution time can be observed by increasing the Map tasks. Thus, if we want to reduce the execution time in MR_1, we need to use parallel **LU** decomposition.

The last bar in MR_1 in Figure 7 shows the execution time for calculating MR_1 using parallel **LU** decomposition as presented in Section 2.5. On a single node (i.e., one Reduce), this process takes approximately 110 s to calculate the **LU** decomposition of a 400 × 400 matrix and to obtain the inverse of **L** and **U** matrices, whereas, on parallel **LU**, we split the 400 × 400 matrix into four submatrices, from $\mathbf{A}_1$ to $\mathbf{A}_4$ (the order of each matrix is 200 × 200), and then obtain $\mathbf{L}_1$, $\mathbf{L}_2$, $\mathbf{L}_3$, $\mathbf{U}_1$, $\mathbf{U}_2$, and $\mathbf{U}_3$ as presented in Section 2.5. We need two MapReduce phases and require 89 s to calculate the results to be the same as those in a single node.

Figure 8 shows the total execution time to obtain estimated parameter (i.e., β_{ik}). By increasing the number of tasks, the execution time is reduced. If we can increase the task capacity by building additional machines in a cluster, we may be able to calculate matrix operations faster than we can currently perform. In addition, we can easily perform numerous matrix operations using MapReduce.

Figure 9 shows the comparison of the execution times to calculate MR_3 and MR_4 when we use multiple regression and multivariate multiple regression models. The reason why we compare these two models using only MR_3 and MR_4 is that MR_0, MR_1, and MR_2 of the two models are the same. We consider only one KPI at a time in the multiple regression models. Thus, the order of the **Y** matrix is 600 × 1. In the multivariate multiple regression models, we consider 100 KPIs; thus, the order of the **Y** matrix is 600 × 100. Given the complexity of matrix multiplication, it is likely that the execution time for MR_3 and MR_4 in the multiple regression models is 100 times faster than that in the multivariate multiple regression models.

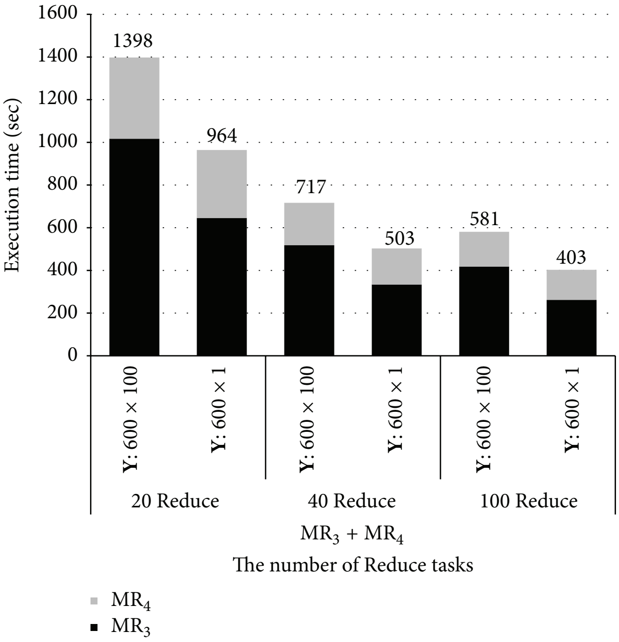


Figure 9: Execution time for calculating MR_3 and MR_4 according to the number of tasks when the order of **Y** is 600 × 100 or 600 × 1.

In Figure 9, however, the execution time for MR_3 and MR_4 when the order of **Y** matrix is 600 × 1 is about 1.4 times faster than that when the order of **Y** matrix is 600 × 100. This happens because a minimum amount of time is needed for MapReduce execution, which includes time for forking Map, sorting, and merging Reduce. Therefore, in this case, multivariate multiple regression models are more efficient than multiple regression models.

6. Conclusion

In BSON, recent research has indicated that a framework using machine-learning tools and the Gaussian process regression model facilitates a more automatic operation of SON. This approach suffers from some limitations. However, although it determines NPs individually related to a KPI, it cannot inform us of the exact value of the KPI according to the change in the NP values. Therefore, we have proposed the multiple regression models to easily determine the relationship between a KPI and the NPs [8]. These multiple regression models, however, were found to have their own shortcomings. If we want to identify the relationship between various KPIs and NPs, we must calculate the multiple regression models several times.

To eliminate these limitations, we have proposed in this paper multivariate multiple regression models. These models separate the NPs unrelated to a KPI from those that are related and allow us to determine at once the relationship between various KPIs and NPs. If $\beta_{(i-1)k}$ is close to zero at KPI_k, then NP_{i-1} is unrelated to KPI_k. Further, we can identify if two KPIs (e.g., KPI_p and KPI_q) are conflicting if the signs of all the row elements of β_{ip} and β_{iq} are entirely different.

We implemented these proposed models using MapReduce. By increasing the number of tasks, the execution time was reduced. We have also shown through experiments that the proposed multivariate multiple regression models are more efficient than the multiple regression models, as shown in Figure 9. Naturally, this approach suffers from limitations, such as communication cost. However, using distributed programming such as MapReduce, we can easily simultaneously calculate numerous matrix operations. We can also possibly achieve faster and more frequent calculations by introducing additional machines in a cluster. In our future work, we will analyze the proposed models using real big data in mobile wireless networks.

Competing Interests

The authors declare that they have no competing interests.

Acknowledgments

This research was supported by the MSIP (Ministry of Science, ICT and Future Planning), Republic of Korea, under the IT Consilience Creative Program (IITP-2015-R0346-15-1008) supervised by the IITP (Institute for Information & Communications Technology Promotion) and the ICT R&D Program of MSIP/IITP (B0126-15-1017).

References

[1] O. G. Aliu, A. Imran, M. A. Imran, and B. Evans, "A survey of self organisation in future cellular networks," *IEEE Communications Surveys and Tutorials*, vol. 15, no. 1, pp. 336–361, 2013.

[2] S. Hamalainen, H. Sanneck, and C. Sartori, *LTE Self-Organizing Networks (SON)*, John & Wiley Sons, Ltd, New York, NY, USA, 2012.

[3] N. Baldo, L. Giupponi, and J. Mangues-Bafalluy, "Big data empowered self organized networks," in *Proceedings of the 20th European Wireless Conference (EW '14)*, pp. 181–188, May 2014.

[4] A. Imran, A. Zoha, and A. Abu-Dayya, "Challenges in 5G: how to empower SON with big data for enabling 5G," *IEEE Network*, vol. 28, no. 6, pp. 27–33, 2014.

[5] E. J. Khatib, R. Barco, P. Munoz, I. D. La Bandera, and I. Serrano, "Self-healing in mobile networks with big data," *IEEE Communications Magazine*, vol. 54, no. 1, pp. 114–120, 2016.

[6] E. J. Khatib, R. Barco, A. Gómez-Andrades, P. Muñoz, and I. Serrano, "Data mining for fuzzy diagnosis systems in LTE networks," *Expert Systems with Applications*, vol. 42, no. 21, pp. 7549–7559, 2015.

[7] C. K. Williams and C. E. Rasmussen, *Gaussian Processes for Regression*, MIT Press, Cambridge, Mass, USA, 1996.

[8] Y. Shin, C.-B. Chae, and S. Kim, "Multiple regression models for a big data empowered SON framework," in *Proceedings of the 7th International Conference on Ubiquitous and Future Networks (ICUFN '15)*, pp. 982–984, IEEE, Sapporo, Japan, July 2015.

[9] Ö. F. Çelebi, E. Zeydan, Ö. F. Kurt et al., "On use of big data for enhancing network coverage analysis," in *Proceedings of the 20th International Conference on Telecommunications (ICT '13)*, pp. 1–5, Casablanca, Morocco, May 2013.

[10] R. J. Freund, D. Mohr, and W. J. Wilson, *Statistical Methods*, Elsevier/Academic Press, Amsterdam, Netherlands, 3rd edition, 2010.

[11] I. Witten and F. Eibe, *Data Mining: Practical Machine Learning Tools and Techniques*, Morgan Kaufmann, 2005.

[12] A. Rajaraman and J. D. Ullman, *Mining of Massive Datasets*, Cambridge University Press, Cambridge, UK, 2011.

[13] J. Dean and S. Ghemawat, "MapReduce: simplified data processing on large clusters," *Communications of the ACM*, vol. 51, no. 1, pp. 107–113, 2008.

[14] J. Xiang, H. Meng, and A. Aboulnaga, "Scalable matrix inversion using mapreduce," in *Proceedings of the 23rd ACM Symposium on High-Performance Parallel and Distributed Computing (HPDC '14)*, pp. 177–190, ACM, Vancouver, Canada, June 2014.

[15] J. P. Stevens, *Applied Multivariate Statistics for the Social Sciences*, Routledge, 2012.

[16] M. Bilodeau and D. Brenner, *Theory of Multivariate Statistics*, Springer Science & Business Media, 2008.

[17] J. Kiusalaas, *Numerical Methods in Engineering with MATLABR*, Cambridge University Press, 2010.

[18] D. Serre, *Matrices*, vol. 216 of *Graduate Texts in Mathematics*, Springer, New York, NY, USA, 2nd edition, 2010.

[19] S. Skiena, *The Algorithm Design Manual*, Springer Science+Business Media, Berlin, Germany, 1998.

[20] Hadoop: Open source implementation of MapReduce, http://hadoop.apache.org.

[21] V. K. Vavilapalli, A. C. Murthy, C. Douglas et al., "Apache hadoop YARN: yet another resource negotiator," in *Proceedings of the 4th Annual Symposium on Cloud Computing (SoCC '13)*, ACM, October 2013.

[22] A. D. Sarma, F. Afrati, S. Salihoglu, and J. Ullman, "Upper and lower bounds on the cost of a map-reduce computation," *Proceedings of the VLDB Endowment*, vol. 6, no. 4, pp. 277–288, 2013.

[23] Q. He, T. Shang, F. Zhuang, and Z. Shi, "Parallel extreme learning machine for regression based on MapReduce," *Neurocomputing*, vol. 102, pp. 52–58, 2013.

Hindawi Publishing Corporation
Mobile Information Systems
Volume 2016, Article ID 2517052, 9 pages
http://dx.doi.org/10.1155/2016/2517052

Research Article

mCSQAM: Service Quality Assessment Model in Mobile Cloud Services Environment

Young-Rok Shin and Eui-Nam Huh

Department of Computer Science and Engineering, Kyung Hee University, Yongin, Republic of Korea

Correspondence should be addressed to Eui-Nam Huh; johnhuh@khu.ac.kr

Received 26 April 2016; Accepted 2 August 2016

Academic Editor: Yeong M. Jang

Cloud computing is high technology that extends existing IT capabilities and requirements. Recently, the cloud computing paradigm is towards mobile with advances of mobile network and personal devices. As concept of mobile cloud, the number of providers rapidly increases for various mobile cloud services. Despite development of cloud computing, most service providers used their own policies to deliver their services to user. In other words, quality criteria for mobile cloud service assessment are not clearly established yet. To solve the problem, there were some researches that proposed models for service quality assessment. However, they did not consider various metrics to assess service quality. Although existing research considers various metrics, they did not consider newly generated Service Level Agreement. In this paper, to solve the problem, we proposed a mobile cloud service assessment model called mCSQAM and verify our model through few case researches. To apply the mobile cloud, proposed assessment model is transformed from ISO/IEC 9126 which is an international standard for software quality assessment. mCSQAM can provide service quality assessment and determine raking of the service. Furthermore, if Cloud Service Broker includes mCSQAM, appropriate services can be recommended for service users using user and service conditions.

1. Introduction

Cloud Service Providers (CSPs) should provide reliable and consistent quality of services to Cloud Service Customers (CSCs). For that reason, CSPs also need to suggest a quality assessment model based on credible quality indicators that can be measureable quantitatively. Mobile cloud service is a service that can provide various activities through mobile devices such as smartphones and tablet PCs. using cloud storage or computing resources [1]. Most interests and developments of cloud computing were focused on the computing resources between enterprises and research institutes. Due to advances of mobile network and personal mobile devices, many customer's requirements are increased rapidly to use content sharing service such as social network service (SNS). In other words, the number of cloud computing based services increases rapidly and it becomes possible to use internet easily in mobile service environments using smartphone [2].

Despite the development of mobile cloud services, there are still problems. There are no defined metrics as standardization and quality assessment model for the mobile cloud service. So, CSPs perform quality assessment according to their own policy. In this case, only providers can get the benefit because it is very difficult to assess the service quality systematically. Therefore, it is necessary to quantitatively measure service quality using quality indicators and assessment model for mobile cloud services.

In general, mobile cloud service provides web-based applications to user. For that reason, mobile cloud services can be applied assessment indicators of ISO/IEC 9126 that is an international standard for software quality assessment. Thus, we determine service quality metrics depending on the mobile cloud service features. And we also propose service quality assessment model named mobile cloud service quality assessment model (mCSQAM) that allows (1) providing the function for service quality assessment as quality metric priority and determining raking of the services. And (2) mCSQAM can recommend appropriate services using user and service conditions with Cloud Service Broker (CSB) in collaborative cloud computing environment. In this paper, we also perform the initial validation of the proposed model by evaluating a case study based on mobile cloud service quality value (mCSQV).

The rest of this paper is organized as follows. In Section 2, we present related works and existing researches for mobile cloud quality assessment. Section 3 describes terms that related mobile cloud computing such as Mobile device, mobile cloud computing, and mobile cloud service. Section 4 presents international standard ISO/IEC 9126 and its quality model and metrics in order to select and match features of mobile cloud services. After reviewing and selecting the metrics from ISO/IEC 9126, our proposed service assessment model named mCSQAM presents with case study examples in Section 5. Finally, we conclude our work by presenting a summary and describing future works in Section 6.

2. Related Work

Although the user centric infrastructures are founded for cloud, cloud service is not exactly defined and standardized about its concept and domain. For that reason, Service Level Agreement (SLA) is generally used to guarantee the quality of cloud services. SLA is a part of a standardized service contract where a service is formally defined. Particular aspects of the service agreed between CSP and CSC. A common feature of SLA is a contracted delivery time. In South Korea, one of the standard bodies named Telecommunications Technology Association (TTA) defined an association standard related Cloud Computing SLA. In the standard document that is established from TTA, availability, performance, security, serviceability, and so forth are categorized as cloud service quality characteristics that were suggested as quality metrics [3].

According to [3], a lot of cloud computing features were defined and applied to cloud services. Even with these efforts from TTA, any factors still did not exactly apply for cloud computing and imbalance contract can be caused. Therefore, we propose mobile cloud service quality assessment model from also CSC's perspective in this paper. To solve the aforementioned problem, we refer to ISO/IEC 9126 as international standard. ISO/IEC 9126 defined a model to perform quality assessment of general software by ISO. Functionality, reliability, efficiency, usability, maintainability, and portability are the main characteristics for measuring and assessment of software quality in that international standard. Furthermore, 6 main characteristics include various subcharacteristics [4]. Although it has been developed systematically for a long time, the main purpose of ISO/IEC 9126 is software quality assessment. And ISO/IEC 9126 also has too many quality characteristic categories and metrics are defined; it is difficult to apply it to mobile cloud service quality assessment directly. For that reason, we propose mCSQAM that considered mobile cloud service features which are referred to as ISO/IEC 9126.

There are various researches on going for the cloud service quality assessment. ISO/IEC 25010 is used for establishing quality model in [5]. And service quality model was also proposed in [6] which described how cloud services are well responded. Furthermore, several frameworks were also proposed for cloud service quality evaluation. References [7, 8] were proposed a framework of cloud service quality evaluation system for activating cloud service ecosystem and service delivery. And another framework was named QoE4CLOUD [9] that divided 4 service layers to consider and assess the quality. And QoS and QoE metrics were defined in [10–13] using a quality model for SaaS cloud computing.

However, the existing researches and researches have some problems as follows. For quality assessment, the quality model must consider various metrics and scenarios, However, [6] considered only reliability for service quality assessment. Similarly, [7, 8] focused on security for quality assessment. Furthermore, when performing quality assessment, the quality model has to reflect dramatically changed service conditions and user requirements. However, [11, 12] are not considered to newly generate SLA that will have different quality metrics and weight than the previous one. Also, [9, 10, 13] just suggested quality metrics for SaaS cloud computing without including the method for quality assessment. Thus, we also propose a mobile cloud service quality assessment model named mCSQAM with suggesting quality metrics based on international standard ISO/IEC 9126.

3. Define Related Terms of Mobile Cloud Computing

In Section 3, we introduce brief and clear definitions about mobile device, mobile cloud computing, and mobile cloud service as follows.

3.1. Mobile Device. Mobile device is defined as devices that have mobility and portability and can use internet generally. The mobile devices have limited hardware conditions.

3.2. Mobile Cloud Computing. Mobile cloud computing means the overall technology to provide services from cloud to mobile device. The mobile cloud is generally composed of Data Storage Server and Data Processing Server. This configuration is responsible for infrastructure. Although the mobile device has fewer resources itself, service customer can use additional functions in cloud server. Thus, mobile device must work given operations simply with its own resources. The following theorems are definitions of mobile cloud computing.

(i) Mobile cloud computing is the technical and functional supporting about processing of mobile cloud services.

(ii) The supporting components for platform service are server, storage, network, controlling device, and so forth.

(iii) There are several types like IaaS, PaaS, and SaaS based on a range of support in platform.

3.3. Mobile Cloud Service. CSCs can use a lot of contents and operation software on their mobile devices using their cloud service via internet. Like this, the mobile cloud service means supporting manner and mode of service through cloud infrastructure. Generally, the mobile cloud service requires

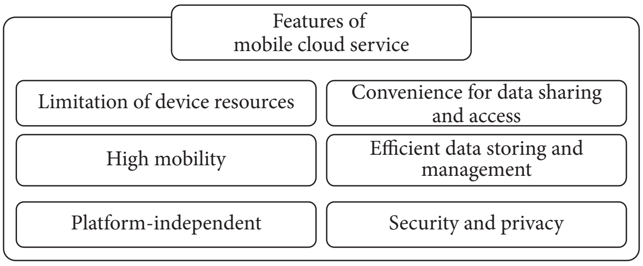

Features of mobile cloud service
Limitation of device resources
Convenience for data sharing and access
High mobility
Efficient data storing and management
Platform-independent
Security and privacy

Figure 1: Features of mobile cloud service.

assurance (1) in *functionality* for suitable, interoperable services and devices, accurate service delivery, and secure information system and communication, (2) in *efficiency* for in-time response and resource provisioning on mobile nodes, (3) in *usability* for operable environment of mobile services, and (4) in *reliability* for fault tolerant services and resources. For that reason, we derive the features of mobile cloud services as in Figure 1 and determine the metrics using the derived features for quality assessment in Section 4.

4. Define Quality Metrics for Service Assessment Model

In this section, we review main characteristics of ISO/IEC 9126 for establishing quality assessment model. And we determine quality metrics for our mobile cloud service quality assessment model that includes 4 main metrics and 8 submetrics from quality model of ISO/IEC 9126. Because quality model in ISO/IEC 9126 is for software, we need to transform the quality model that considered features of mobile cloud service. As a result, we determined finally the metrics for our quality model after matching with features of mobile cloud service as in Figure 2 and the following are the description of the finally determined metrics.

4.1. Functionality. Functionality is a set of attributes that bear on the existence of a set of functions and their specified properties. In this paper, functionality denoted by FU that is the metric for the providing degree of functionality meets expressed or suggested needs in a certain condition when the services are provided. In other words, it is a metric about accuracy and suitability to measure whether the mobile cloud services are correctly provided. As the user needs, mobile cloud service has the responsibility of serving accurate outputs and making it easy to complete the function. We choose 4 submetrics such as suitability (SU), accuracy (AC), interoperability (IO), and security (SEC).

Suitability is an attribute that bears on the presence and appropriateness of a set of functions for specified tasks. For calculating the value of this metric, we define a term SU denoting suitability as in the following equation:

$$\mathrm{SU} = 1 - \frac{\text{Number of missing functions}}{\text{Number of required functions}}. \qquad (1)$$

Accuracy (accurateness) is an attribute that bears on the provision of right or agreed results or effects. For calculating value of this metric, we define a term AC denoting accuracy as in the following equation:

$$\mathrm{AC} = 1 - \frac{\text{Number of exceed expectations}}{\text{Number of attempts for data processing}}. \qquad (2)$$

Interoperability is an attribute that bears on its ability to interact with specified systems. For calculating the value of this metric, we define a term IO denoting interoperability as in the following equation:

$$\mathrm{IO} = 1 - \frac{\text{No. of failures when data exchanges}}{\text{No. of total data exchanges}}. \qquad (3)$$

Security is an attribute that bears on its ability to prevent unauthorized access or alteration, whether accidental or deliberate, to programs or data. For calculating the value of this metric, we define a term SEC denoting security as in the following equation when the problem happens:

$$\mathrm{SEC} = \frac{\text{Number of provided functions}}{\text{Number of required functions}}. \qquad (4)$$

4.2. Reliability. Reliability is a set of attributes that bear on the capability of software to maintain its level of performance under stated conditions for a stated period of time. Reliability denoted by RE is the metric that most mobile cloud services served on the mobile devices and all the user data will be stored in cloud storage through network. Reliability is an important metric for the service quality evaluation as mobile cloud services are depending on network conditions. Reliability has several submetrics such as maturity, fault tolerance, and recoverability. Fault tolerance is the property that enables a system to continue operating property in the event of the failure of (or one or more faults within) some of its components.

In the reliability, we define a term of FT denoting fault tolerance that can be calculated as in the following equation:

$$\mathrm{FT} = 1 - \frac{\text{Number of system errors of network error}}{\text{Number of network error happen}}. \qquad (5)$$

4.3. Usability. Usability is a set of attributes that bear on the effort needed for use and on the individual assessment of such use by a stated or implied set of users. Usability denoted by US is the degree to which a software or service can be used by specified users to achieve quantified objectives with effectiveness, efficiency, and satisfaction in a quantified context of use. For the mobile cloud service, usability is the metric to evaluation of learnability, operability, understandability, and so forth. When consumer uses mobile cloud services, it must be easy to control and access the service and give user satisfaction. We just choose operability (OP) for mCSQAM. Operability is an attribute that bears on the users' effort for operation and operation control. As in following equation, how operability can measure many proper functions are provided to user through their mobile cloud service operation and control:

$$\mathrm{OP} = \frac{\text{Number of functions below expectation}}{\text{Number of total service functions}}. \qquad (6)$$

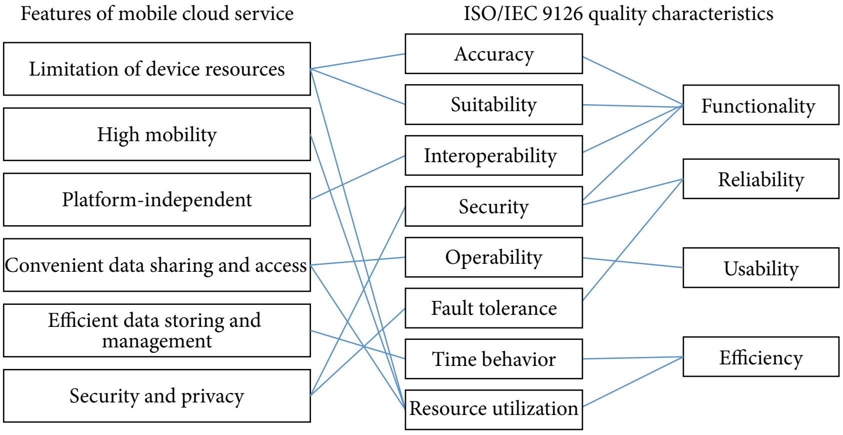


Figure 2: Mapping of mobile cloud service features and ISO/IEC 9126 quality characteristics.

4.4. Efficiency. Efficiency is a set of attributes that bear on the relationship between the level of performance of the software and the amount of resources used, under stated conditions. Efficiency is a metric to measure the relative performance for used service amount in regulated condition. And this is the metric to assess time behavior (TB) and resource utilization (RU) for mobile cloud services. Time behavior is an attribute that bears on response and processing times and on throughput rates in performance of its function. So, this metric measures a ratio of an execution time for a total invocation time. We define a term TB that denotes time behavior to calculate value of this metric as in the following equation:

$$\text{TB} = \frac{\text{Number of responses in average expection time}}{\text{Number of responses in measuring time}}. \quad (7)$$

Resource utilization is an attribute that bears on the amount of resource used and the duration of such use in performing its function. And this metric measures a ratio of an amount of allocated resources for the predefined resources. So, for calculating the value of this metric, we defined a term RU that denotes resource utilization as in the following equation:

$$\text{RU} = \frac{\text{Number of alarms or errors in service running}}{\text{Number of service requests}}. \quad (8)$$

When the mobile cloud service is provided, clients must satisfy the requirements such as response time and throughput by well utilizing the resources. Due to poor resource provisioning, the service quality does not need to degrade SLA. For that reason, time behavior and resource utilization are chosen for calculating value of efficiency.

4.5. Portability and Maintainability. Portability is a set of attributes that bear on the ability of software to be transferred from one environment to another. Portability is the usability of the same software in different environment. The prerequirement for portability is the generalized abstraction between the application logic and system interfaces. When software or service with the same functionality is produced for several platforms, portability is the key issue for development cost reduction. And its metrics are for evaluation of adaptability and installability in the mobile cloud service. It must be effectively adapted on various service environments and devices and it also should be easy to install and remove. Maintainability is a set of attributes that bear on the effort needed to make specified modifications. Maintainability is also important metric for the mobile cloud services. In this metric, Analyzability, Changeability, Stability, Testability and Maintainability compliance are included for its submetrics. However, the submetrics are difficult to map to mobile cloud service requirements. Thus, for that reason, we exclude portability and maintainability in quality evaluation model, mCSQAM.

5. The Proposed Method: mCSQAM

In this paper, we propose a quality assessment model to validate service quality and recommend the best service by cloud broker environments which is easier way to customer to deliver proper services among many cloud providers.

5.1. Analytic Hierarchy Process (AHP). Analytic Hierarchy Process (AHP) [14] is known as one of the effective Multicriteria Decision Making (MCDM) methods that were formerly developed by Thomas L. Saaty. AHP provides alternatives through reasonable evaluation that provides systematic analysis and stepwise derivation with pairwise comparison about various measures. Utilizing mathematical methodology, quantitative evaluation measures cannot be only considering but also qualitative assessment measures by AHP. Furthermore, it has been widely used for decision-making with various and complex measures due to simple calculation and easy understanding.

To make a decision in an organized way to generate priorities, we need to decompose the decision into the following 4 steps:

(i) Define the problem and determine the kind of knowledge sought.

(ii) Structure the decision hierarchy from the top with the goal of the decision, then the objectives from a broad perspective, through the intermediate levels (criteria on which subsequent elements depend) to the lowest level which usually is a set of the alternatives.

(iii) Construct a set of pairwise comparison matrixes. Each element in an upper level is used to compare the elements in the level immediately below with respect to it.

(iv) Use the priorities obtained from the comparisons to weigh the priorities in the level immediately below. Do this for every element. Then, for each element in the level below add its weight values and obtain its overall or global priority. Continue this process of weighting and adding until final priorities of the alternatives in the bottom mist level are obtained.

5.2. System Model for mCSQAM. Figure 3 shows components of our system model that includes Quality Monitor (QM), Quality Assessment Performer (QAP), Quality Balancer (QB), and SLA Generator. The role of QM is to measure given quality metrics with storing to DB and to propagate the result of monitoring to QAP. After receiving the result of quality monitoring, QAP calculates service quality using AHP method. By using the assessment result, QB controls quality metrics weight for balanced service quality. After determining the quality metric weight, SLA is generated newly to be utilized between CSP and CSC. Furthermore, the newly generated SLA is also used for the next time quality assessment.

5.3. Scenarios for Evaluation of mCSQAM. We evaluate our model and show the result of quality assessment using generating service scenarios, whose services have different quality related components as shown in Figure 4. For the case study, we assume that the service quality value (Q_i) of each mobile cloud service as schematically given as shown Figure 4. And we also assume that each service has different weight (W_i) to assess Q_i in detail for the case study. Thus, FU, RE, US, and EF have different weight values as user requirements. Through the above assumptions, we can find which metric is more effective to 4 different mobile cloud services quality and compare them relatively.

The following steps are showing how we conduct quality assessment procedure.

Step 1 (applying weight to each quality metric). We assigned a weight value to each submetric in Functionality (FU) as shown in Table 1.

Submetrics of efficiency (EF) also are assigned as shown in Table 2.

Table 1: Weight value for submetrics of FU.

Quality metric	Weight
SU	0.4
AC	0.3
IO	0.2
SEC	0.1

Table 2: Weight value for submetrics of EF.

Quality metric	Weight
TB	0.3
RU	0.7

Reliability (RE) and Usability (US) have just 1 submetric, so the weight value of each submetric is 1. Although weight value can vary depending on user requirements, it is difficult to use the weight for general cases. Thus, we assume that the metrics have fixed weight at the design time in this paper. However, in order to compare the result of quality assessment in different cases and apply properly to the real case, we evaluate our model by assigning different weight values with 4 different scenarios. To generate different scenario cases, we assign weight values of 4 main metrics, FU, RE, US, and EF, such that the important one was set to 0.4 and others were set to 0.2 in each case. So now we can have 4 different weighted scenario services as shown in Figure 4. After applying weights, mCSQV (mobile cloud service quality value) is finally calculated using the product of service quality measure value (Q_i) and weight value (W_i) as in the following equation:

$$\text{mCSQV} = \sum Q_i \times W_i. \tag{9}$$

Step 2 (calculating the result after applying weight for sub–metrics). Submetrics of FU have weight values as in Table 1. In evaluation, weight values of submetrics are randomly determined as shown (10). Our model can support adaptation of dynamic changes by user or service requirements. After applying the above settings, we have the quality value as in following equation for 4 different scenario cases. Only for consideration to functionality (FU), Service 2 provides the best quality, and the results tell that the mobile cloud services are ranked by Service 2 > Service 4 > Service 1 > Service 3:

$$\begin{aligned} \text{mCSQV}_{\text{FU}} &= \begin{bmatrix} 0.222 & 0.296 & 0.148 & 0.333 \\ 0.235 & 0.314 & 0.216 & 0.235 \\ 0.268 & 0.250 & 0.268 & 0.214 \\ 0.172 & 0.276 & 0.241 & 0.310 \end{bmatrix} \\ &\quad \times \begin{bmatrix} 0.4 \\ 0.3 \\ 0.2 \\ 0.1 \end{bmatrix} \\ &= \begin{bmatrix} 0.230 & 0.290 & 0.202 & 0.278 \end{bmatrix}. \end{aligned} \tag{10}$$

The weight of FT, submetric of RE, is 1 as having only one submetric. After the calculation of this case, each service has a value of 0.212, 0.364, 0.152, and 0.273. If a user or a service considered only reliability, the best service is Service 2. And

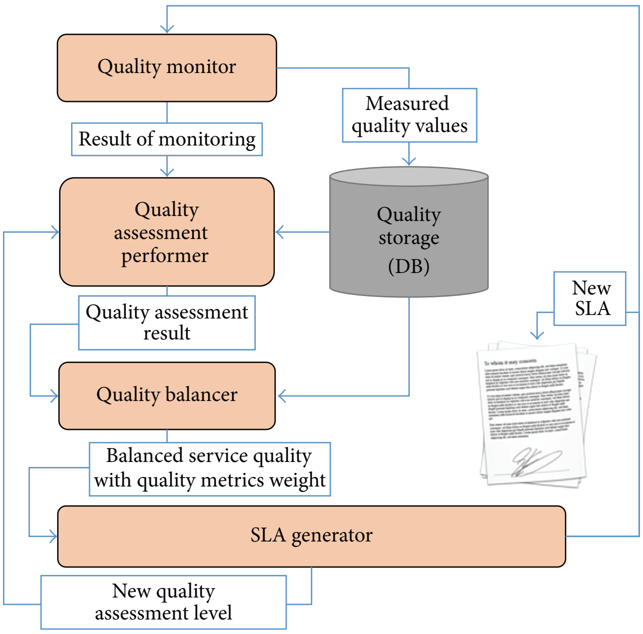

Quality monitor
Measured quality values
Result of monitoring
Quality assessment performer
Quality storage (DB)
New SLA
Quality assessment result
Quality balancer
Balanced service quality with quality metrics weight
SLA generator
New quality assessment level

Figure 3: Components of mCSQAM.

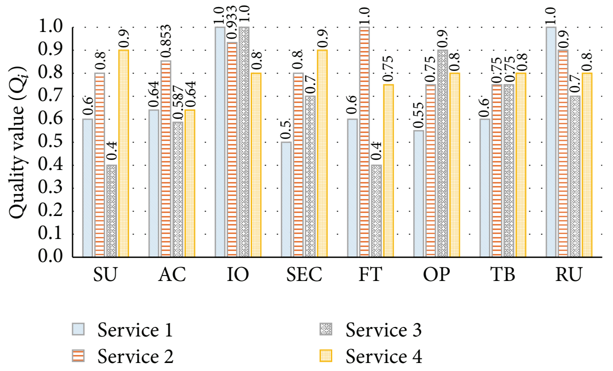


Figure 4: Quality measure value for each mobile cloud service.

the mobile cloud services are ranked by Service 2 > Service 4 > Service 1 > Service 3 in this case:

$$\text{mCSQV}_{\text{RE}} = \begin{bmatrix} 0.230 & 0.290 & 0.202 & 0.278 \end{bmatrix}. \qquad (11)$$

The weight of OP, submetric of US, is 1 as a single submetric. After the calculation of this case, each service has the value as in the following equation for 4 different scenario cases. If the user or service considers only usability, the best service is 3 and the mobile cloud services are ranked by Service 3 > Service 4 > Service 2 > Service 1 in this case:

$$\text{mCSQV}_{\text{US}} = \begin{bmatrix} 0.183 & 0.250 & 0.300 & 0.267 \end{bmatrix}. \qquad (12)$$

The value 0.3 is assigned to TB for its weight. And the value 0.4 is also assigned to RU for its weight. After the calculation of this case, each service has the value as in the following equation for 4 different scenario cases. Considering only efficiency, Service 1 has the best quality and the mobile cloud services are ranked by Service 1 > Service 2 > Service 4 > Service 3:

$$\text{mCSQV}_{\text{EF}} = \begin{bmatrix} 0.207 & 0.294 \\ 0.259 & 0.265 \\ 0.259 & 0.206 \\ 0.276 & 0.235 \end{bmatrix} \times \begin{bmatrix} 0.3 \\ 0.7 \end{bmatrix} \qquad (13)$$

$$= \begin{bmatrix} 0.268 & 0.263 & 0.222 & 0.247 \end{bmatrix}.$$

Figure 5 shows the quality value after applying submetric weight for 4 scenarios. We can see in Figure 5 that Service 2 is the best for FU and RE, Service 3 for US, and Service 4 for EF. However, Figure 5 shows that just one main quality metric is considered and determined the ranking of services. To get the comprehensive result of quality assessment to select the best service, we also need to consider different weight for main quality metrics, FU, RE, US, and EF.

Step 3 (calculating the result after applying weight for main metrics). For calculating final service assessment value, make a matrix resulted in previous steps as in the following matrix. After multiplying the weight of main metrics to the matrix, we can get the final service assessment value:

$$\text{mCSQV} = \begin{bmatrix} 0.230 & 0.212 & 0.183 & 0.268 \\ 0.290 & 0.364 & 0.250 & 0.263 \\ 0.202 & 0.152 & 0.300 & 0.222 \\ 0.278 & 0.273 & 0.267 & 0.247 \end{bmatrix}. \qquad (14)$$

For calculating the final service quality assessment, we assigned different weight for each case. According to the user

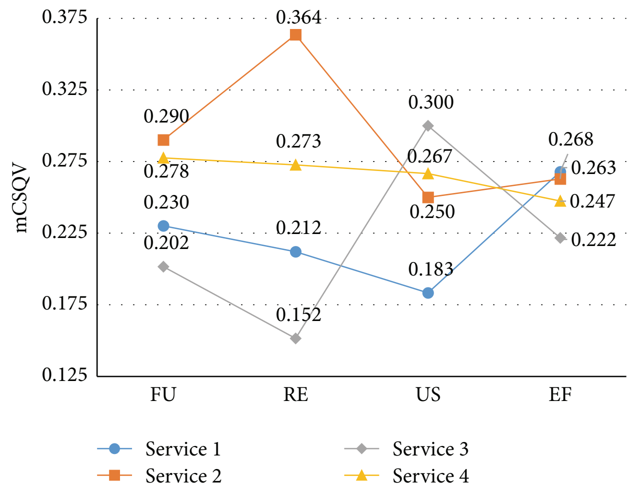


Figure 5: mCSQV comparison after applying submetrics weight.

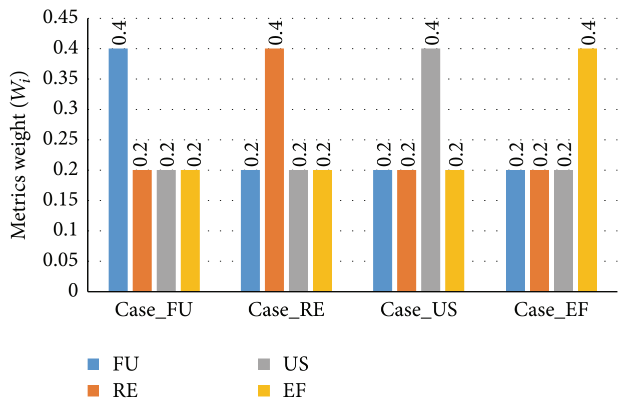


Figure 6: Main metrics weight for each case.

requirement, if there is a user who considered functionality (FU) has the highest weight for final service quality assessment. Likewise, if a user wants reliable service, set the RE's weight as the highest. And usability and efficiency are same as the above cases.

Figure 6 shows the assigned weights for each case that the weight considered the most important is 0.4 which is twice bigger than others. And the weight of others is set equally as 0.2.

When the functionality is considered as the most important metric at Case_FU in Figure 6, only FU's weight is 0.4 and others are 0.2. To get the final service quality assessment value, multiply weight vector to matrix as follows:

$$\mathrm{mCSQV}_{\mathrm{Case\_FU}} = \begin{bmatrix} 0.230 & 0.212 & 0.183 & 0.268 \\ 0.290 & 0.364 & 0.250 & 0.263 \\ 0.202 & 0.152 & 0.300 & 0.222 \\ 0.278 & 0.273 & 0.267 & 0.247 \end{bmatrix} \times \begin{bmatrix} 0.4 \\ 0.2 \\ 0.2 \\ 0.2 \end{bmatrix} = \begin{bmatrix} 0.225 & 0.291 & 0.215 & 0.268 \end{bmatrix}. \quad (15)$$

The result of quality assessment: the best quality Service is 2 and mobile cloud services are ranked as Service 2, Service 4, Service 1, and Service 3. In other words, if functionality is considered at the service selection process, users need to choose the service 2.

If a user considers that reliability is the most important, the weight is set at Case_RE in Figure 6. As the result, Service 2 is also the best quality assessment value as 0.306. The result of assessment: mobile cloud services are ranked as Service 2 (0.306), Service 4 (0.267), Service 1 (0.221), and Service 2 (0.205)

$$\mathrm{mCSQV}_{\mathrm{Case\_RE}} = \begin{bmatrix} 0.230 & 0.212 & 0.183 & 0.268 \\ 0.290 & 0.364 & 0.250 & 0.263 \\ 0.202 & 0.152 & 0.300 & 0.222 \\ 0.278 & 0.273 & 0.267 & 0.247 \end{bmatrix} \times \begin{bmatrix} 0.2 \\ 0.4 \\ 0.2 \\ 0.2 \end{bmatrix} = \begin{bmatrix} 0.221 & 0.306 & 0.205 & 0.267 \end{bmatrix}. \quad (16)$$

3rd case is for usability and the weight vector is set to Case_US in Figure 6. As the result of quality assessment, Service 2 is ranked first and received 0.283. Service 4 received 0.266, ranked second. Services 3 and 1 are ranked as third and fourth services and each service receives 0.235 and 0.215:

$$\mathrm{mCSQV}_{\mathrm{Case\_US}} = \begin{bmatrix} 0.230 & 0.212 & 0.183 & 0.268 \\ 0.290 & 0.364 & 0.250 & 0.263 \\ 0.202 & 0.152 & 0.300 & 0.222 \\ 0.278 & 0.273 & 0.267 & 0.247 \end{bmatrix} \times \begin{bmatrix} 0.2 \\ 0.2 \\ 0.4 \\ 0.2 \end{bmatrix} = \begin{bmatrix} 0.215 & 0.283 & 0.235 & 0.266 \end{bmatrix}. \quad (17)$$

The last considered that efficiency is the most important case. For this metric, the weight vector is consisting of Case_EF in Figure 6. In this case, Service 2 shows the best

Table 3: Comparison of quality measuring methods.

Methods	Total number of metrics	Quality model hierarchy architecture	Mobile environment support
[6]	1	X	X
[7]	5 (focused on security)	X	X
[8]	10 (focused on performance)	X	X
QoE4CLOUD [9]	0 (not defined)	○ 4-level (hierarchical framework)	△ (SaaS based cloud service only)
[10]	0 (not defined)	X	△ (SaaS based cloud service only)
ADVISE [11, 12]	2 (focused on elasticity)	X	△ (IaaS performance only)
[13]	0 (only suggested using SMI, not defined)	X	△ (SaaS based cloud service only)
[15]	10 or more (security, QoS, and software)	X	△ (SaaS based cloud service only)
Proposed mCSQAM	8 submetrics in 4 main metrics	○ 2-level (hierarchical quality model)	○

"○" means that the research is well supported and "△" means that the research is partly supported hierarchy architecture or mobile environment. By contrast, "X" means that the research is not supported hierarchy architecture or mobile environment.

quality in the assessment result. As a result, mobile cloud services are ranked as Service 2, Service 4, and Service 3:

$$\mathrm{mCSQV}_{\mathrm{Case\_EF}} = \begin{bmatrix} 0.230 & 0.212 & 0.183 & 0.268 \\ 0.290 & 0.364 & 0.250 & 0.263 \\ 0.202 & 0.152 & 0.300 & 0.222 \\ 0.278 & 0.273 & 0.267 & 0.247 \end{bmatrix} \times \begin{bmatrix} 0.2 \\ 0.2 \\ 0.2 \\ 0.4 \end{bmatrix} = \begin{bmatrix} 0.232 & 0.286 & 0.219 & 0.262 \end{bmatrix}. \tag{18}$$

From previous results including case of metric weight set to 1, we can observe that it is difficult to determine synthetically which service is the best. So, we applied different weights for each case and it derives the result of mobile cloud service quality assessment like Figure 7. As the comprehensive assessment result, Service 2 shows the best quality in the all cases.

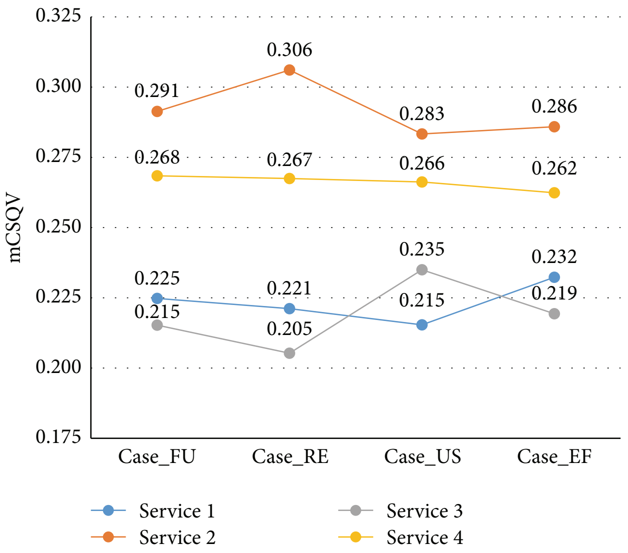


Figure 7: Service ranking for each case.

Many quality models are proposed for measuring cloud services. So, we compared our proposed mCSQAM with other existing quality evaluation methods or models. Table 3 shows comparison results among quality models.

Quality model was mentioned in [9, 10]. However, they did not include metrics definition for quality model. So, they cannot measure service quality accurately and details. Other researches [7, 8, 11, 12] defined 2~10 quality measures for their model but focus on only one characteristic. In contrast, our proposed model, mCSQAM, includes 8 submetrics within 4 main metrics.

And our proposed model, mCSQAM, considers categorizing quality measuring level to 4 main metrics and 8 submetrics. When applying the hierarchy architecture, we expect that quality assessment will be able to assess more accurate result. In contrast, there are no models that considered hierarchy architecture for quality assessment model except QoE4CLOUD [9]. In [9], QoE4CLOUD was proposed as framework that includes 4 layers which are System/Hardware QoS, Network QoS, Application QoS (QoE), and Business QoS (QoBiz). Even though the layers were separated for various QoSs, the framework is not enough to assess service quality that the metrics were not defined clearly.

There are some researches [6–8] that were not considered mobile cloud service environment. Even though other researches [9–13, 15] considered, on general, cloud services, it was only focused on SaaS or IaaS except mobile environment. So, our model is appropriate for mobile cloud services in order to assess quality evaluation model.

6. Conclusion and Future Work

Cloud computing has become an important and its paradigm is towards to mobile cloud with mobile network. Currently, there are many Cloud Service Providers who offer different services with different quality attributes with their own policy. With the growing number of cloud offerings, there are some researches for quality assessment of cloud services. However, most researches about quality assessment are not considered characteristics of mobile environment.

To solve the aforementioned problem, we determined quality metrics with properties of mobile cloud service from ISO/IEC 9126. ISO/IEC 9126 was established as international standard for software quality assessment. However, it is difficult to apply directly to use on mobile cloud services; we also propose mobile cloud service quality assessment model named mCSQAM that was transformed to ISO/IEC 9126 quality model.

In this paper, this work presents the first architecture, mCSQAM, to systematically measure quality metrics selected in Section 4 and rank the mobile cloud services based on these metrics. For verification of our quality assessment model, we proposed an Analytic Hierarchy Process (AHP) based method which can assess the mobile cloud services based on different services depending on quality requirements.

We believe the mCSQAM represents a significant next step towards enabling accurate quality measurement. And we also expect that the mCSQAM with Cloud Service Broker can provide recommendations service through appropriate mobile cloud service selection for Cloud Service Customers. However, the quality metrics in our proposed model are measured quantitatively on system side. For that reason, our model needs extension and supplement qualitative assessment in near future. So, we will consider Service Measurement Index (SMI) from Cloud Service Measurement Initiative Consortium.

Competing Interests

The authors declare that they have no competing interests.

Acknowledgments

This research is supported the MSIP (Ministry of Science, ICT & Future Planning), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2016-(H8501-16-1015)) supervised by the IITP (Institute for Information & communication Technology Promotion).

References

[1] X. Li, H. Zhang, and Y. Zhang, “Deploying mobile computation in cloud service,” in *Proceedings of the 1st International Conference on Cloud Computing (CloudCom ’09)*, pp. 301–311, Beijing, China, December 2009.

[2] N. Fernando, S. W. Loke, and W. Rahayu, “Mobile cloud computing: a survey,” *Future Generation Computer Systems*, vol. 29, no. 1, pp. 84–106, 2013.

[3] Telecommunications Technology Association (TTA), “Quality Factor for Establishing Cloud Computing Service Level Agreement,” 2010.

[4] International Organization for Standardization (ISO), “ISO/IEC 9126: Information Technology-Software Quality Characteristics and Metrics,” 1997.

[5] A. Ravanello, J.-M. Desharnais, L. E. B. Villalpando, A. April, and A. Gherbi, “Performance measurement for cloud computing applications using ISO 25010 standard characteristics,” in *Proceedings of the Joint Conference of the 24th International Workshop on Software Measurement (IWSM ’14) and the 9th International Conference on Software Process and Product Measurement (Mensura ’14)*, pp. 41–49, Rotterdam, The Netherlands, October 2014.

[6] Z. Raghebi and M. R. Hashemi, “A new trust evaluation method based on reliability of customer feedback for cloud computing,” in *Proceedings of the 10th International ISC Conference on Information Security and Cryptology (ISCISC ’13)*, pp. 1–6, IEEE, Yazd, Iran, August 2013.

[7] H. Jeon and K.-K. Seo, “A framework of cloud service quality evaluation system for activating cloud service ecosystem,” *Advanced Science and Technology Letters*, vol. 35, pp. 97–100, 2013.

[8] H. Jeon, Y.-G. Min, and K.-K. Seo, “A framework of performance measurement of cloud service infrastructure system for service delivery,” in *Proceedings of the Advacnced Science and Technology Letters (Cloud and Super Computing 2014 Conference)*, vol. 46, pp. 142–145, December 2014.

[9] E. Kafetzakis, H. Koumaras, M. A. Kourtis, and V. Koumaras, “QoE4CLOUD: a QoE-driven multidimensional framework for cloud environments,” in *Proceedings of the International Conference on Telecommunications and Multimedia (TEMU ’12)*, pp. 77–82, Chania, Greece, August 2012.

[10] S. Shah and S. Buch, “Identification of cloud computing service quality indicators with its expected involvement in cloud computing services and its performance issues,” *International Journal on Recent and Innovation Trends in Computing and Communication*, vol. 3, no. 7, pp. 4569–4572, 2015.

[11] G. Copil, D. Trihinas, H.-L. Truong et al., “ADVISE-a framework for evaluating cloud service elasticity behavior,” in *Proceedings of the 12th International Conference on Service-Oriented Computing (ICSOC ’14)*, pp. 275–290, Paris, France, November 2014.

[12] G. Copil, H.-L. Truong, D. Moldovan et al., “Evaluating cloud service elasticity behavior,” in *Proceedings of 12th International Conference on Service-Oriented Computing (ICSOC ’14)*, pp. 275–290, November 2014.

[13] S. Al-Shammari and A. Al-Yasiri, “Defining a metric for measuring QoE of SaaS cloud computing,” in *Proceedings of the 15th Annual Post Graduate Symposium on the Convergence of Telecommunications, Networking Broadcasting (PGNET ’14)*, Liverpool, UK, June 2014.

[14] T. L. Saaty, “Decision making with the analytic hierarchy process,” *International Journal of Services Sciences*, vol. 1, no. 1, pp. 83–98, 2008.

[15] S. Banerjee and S. Jain, “A survey on Software as a Service (SaaS) using quality model in cloud computing,” *International Journal of Engineering and Computer Science*, vol. 3, no. 1, pp. 3598–3602, 2014.

Hindawi Publishing Corporation
Mobile Information Systems
Volume 2016, Article ID 1708970, 12 pages
http://dx.doi.org/10.1155/2016/1708970

Research Article

Securing SDN Southbound and Data Plane Communication with IBC

JunHuy Lam, Sang-Gon Lee, Hoon-Jae Lee, and Yustus Eko Oktian

Department of Ubiquitous IT, Division of Computer & Information Engineering, Dongseo University, Busan 617-716, Republic of Korea

Correspondence should be addressed to Sang-Gon Lee; nok60@dongseo.ac.kr

Received 21 March 2016; Revised 22 June 2016; Accepted 4 July 2016

Academic Editor: Juan C. Cano

In software-defined network (SDN), the southbound protocol defines the communication between the control plane and the data plane. The agreed protocol, OpenFlow, suggests securing the southbound communication with Transport Layer Security (TLS). However, most current SDN projects do not implement the security segment, with only a few exceptions such as OpenDayLight, HP VAN SDN, and ONOS implementing TLS in the southbound communication. From the telecommunication providers' perspective, one of the major SDN consumers besides data centers, the data plane becomes much more complicated with the addition of wireless data plane as it involves numerous wireless technologies. Therefore, the complicated resource management along with the security of such a data plane can hinder the migration to SDN. In this paper, we propose securing the distributed SDN communication with a multidomain capable Identity-Based Cryptography (IBC) protocol, particularly for the southbound and wireless data plane communication. We also analyze the TLS-secured Message Queuing Telemetry Transport (MQTT) message exchanges to find out the possible bandwidth saved with IBC.

1. Introduction

Software-defined network (SDN) is a new network technology that separates the intelligence of the network by decoupling the control and data planes. In order to achieve that, a new network element was introduced into the network, the SDN controller. It centralizes the network control plane, manages the network data plane, and provides the platform that eases the development of the management plane or, in other words, the network applications. The network switches that take on the role of the network's data plane become forwarding devices in SDN; they forward the packets in accordance with the flow tables received from the SDN controller unquestioningly.

SDN introduces three new protocols into the network, namely, the northbound protocol, east/west-bound protocol, and southbound protocol. The conceptual view of SDN with both the wired and wireless data planes is as shown in Figure 1.

The northbound protocol is used by the management plane or network applications to communicate with the control plane (the SDN controllers) to perform tasks such as load balancing via load adaption [1] and Quality of Service (QoS) [2]. The security risks and requirements of the northbound communication are dependent on the network application.

In the SDN environment, security applications or tools can also be used to provide network security from the management plane. SDN allows applications to monitor the network traffic and have a network-wide view. Hence, identity revocation can be carried out easily with applications or tools that detect malicious nodes such as the intrusion detection system (IDS) [3], Distributed Denial-of-Service (DDoS) detection [4], and network monitoring [5, 6].

The east/west-bound protocol is used for the communication within the control plane or, specifically, the communication between the SDN controllers and the data stores. Unfortunately, the SDN controllers currently available are vendor specific as those of the time of writing. They have neither the agreed east/west-bound protocols nor the security for them, with Open Network Operating System (ONOS) being the exception [7]. However, the security of this communication is especially important for the distributed SDN. It ensures that no malicious controllers are snooping for network information or even driving the network.

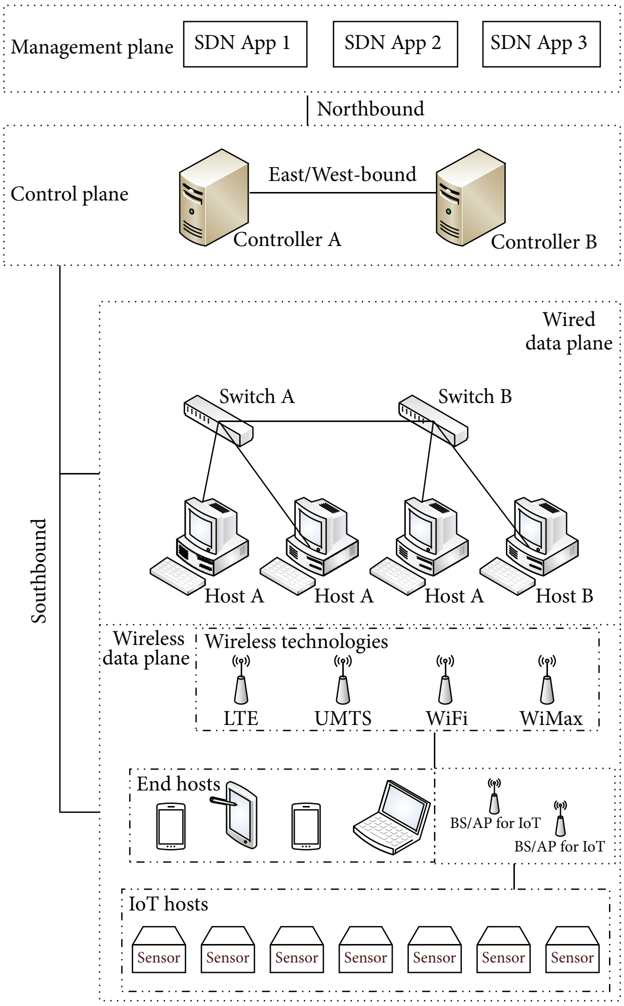


FIGURE 1: The conceptual view of SDN with both wired and wireless data planes.

In the single controller SDN implementation, a compromise of the controller will enable the attacker to control the entire network. However, in the distributed SDN, the network is spread across all the available controllers within the network. In order to minimize the effect following the compromise of a single controller in the distributed SDN, the east/west-bound communication has to be protected. If it is left unprotected, the malicious controller will have the ability to manipulate all other controllers in the entire network.

The attacker can also insert a malicious data store into the network to obtain a duplicate or backup copy of the network information from other data stores within the network. By utilizing this information, the attacker will then be able to learn the network topology and carry out relevant attacks accordingly. Worse still, the attacker can inject the desired flow through a malicious data store.

The southbound protocol defines the communication between the control plane and the data plane. There is a commonly agreed protocol for the southbound communication which is the OpenFlow protocol [8], standardized by the Open Networking Foundation (ONF). However, other southbound protocols are also available if OpenFlow does not suit a particular purpose. For instance, Cisco's Application Centric Infrastructure (ACI) [9] is another alternative for OpenFlow. However, using such proprietary protocols will hence limit the device's vendor choices.

In order to protect the network from being driven by a malicious controller or to prevent a malicious switch or network device from obtaining any network information, the southbound communication must be operated in a secure channel. OpenFlow suggested securing the southbound communication with Transport Layer Security (TLS).

Projects that do not implement TLS are prone to man-in-the-middle attacks [10]. Attackers will be able to penetrate the OpenFlow networks while remaining undetected. Despite the security risks involved, it is still not implemented in many SDN projects with exceptions such as OpenDayLight [11], HP Virtual Application Network (VAN) SDN [12], and ONOS [13].

One of the main reasons for why TLS is not used by SDN network administrators is that the steps required to configure it correctly can be quite tedious [10]. The TLS implementation requires a Certificate Authority (CA) to generate the CA's key, certificates for the controllers, switches, and then the signing of these certificates with the CA's key. The certificates and devices' keys will then be deployed to the respective devices prior to the actual network deployment. This tedious process is a hindrance for them when adopting the TLS to secure the communication channel.

As illustrated in Figure 1, there are two general types of data planes, the wired and wireless variants. SDN was initially built for the wired data plane and the data center is the main consumer of it. As the SDN technology grows, telecommunication providers became interested in it and started experimenting with it in their network with AT&T supporting both OpenDayLight and ONOS while Verizon, China Unicom, NTT Communication, and SK Telecom are backing ONOS.

However, the original design neglected the wireless data plane that was powered by numerous wireless technologies such as the Fourth-Generation Long Term Evolution (4G-LTE) [14], Universal Mobile Telecommunication System (UMTS) [15], Wireless Fidelity (WiFi) [16], and Worldwide Interoperability for Microwave Access (WiMax) [17]. Hence, it is difficult to manage the heterogeneous wireless data plane [18] even with the current advancement in SDN technology. It is even more complicated when it comes to the security management between these wireless technologies.

Unlike its wired counterpart, the security for the wireless data plane cannot be neglected because anyone within the coverage area of the wireless technology can tap into the network and perform any malicious activities to disrupt the network. Therefore, security becomes a crucial criterion for the wireless data plane before they can deploy it for real usage.

By replacing TLS with Identity-Based Cryptography (IBC) [19], the steps required for system setup will be greatly simplified, improving both the performance availability and bandwidth availability as well as minimizing storage and management of the public keys and therefore saving on costs.

Our Contributions. In this paper, we proposed securing the SDN communication with IBC protocol. To the best of our knowledge, this is the first work that allows multidomain secure communication with IBC in SDN along with its data plane. Our contributions are listed in detail as follows:

(1) We described the security risks involved in SDN and its data plane as well as the reasons it needs to be protected.

(2) We described the reasons the other proposal is insufficient to protect the SDN and why IBC is a preferable method.

(3) We presented the proposal to secure the SDN and its data plane, specifically the wireless data plane, with a multidomain capable IBC protocol.

(4) We provided some application scenarios in which the multidomain IBC protocol can be utilized. We also described how it allows multidomain communication and switch migration in the distributed SDN which previously could not be performed.

(5) We described the application of IBC in helping the communication within the data plane and an analysis to show the possible bandwidth saved with IBC.

2. Background

2.1. Southbound Security. The de facto standard of the SDN southbound protocol, OpenFlow [20, 21], suggested that the southbound communication should be secured with TLS. Projects that do not implement TLS are prone to man-in-the-middle attacks [10]. Adversaries will be able to penetrate the OpenFlow networks while remaining undetected.

In order to prevent the compromise of a controller from the southbound communication channel and a malicious switch or network device from obtaining or modifying any network information, the southbound communication must be operated in a secure channel. Despite this requirement, it is not implemented in many SDN projects because TLS is only an optional feature in OpenFlow specifications and the complicated certificate management. Besides that, the security also comes at a cost to the bandwidth due to the exchange of the certificates for authentication purposes.

2.2. Data Plane Security. The data plane can be further divided into two categories, the wired and wireless data plane. Both data planes may share some security concerns but there are also security concerns that are specific to either one of the data planes. The detailed security concerns will be discussed as follows.

2.2.1. Wired Data Plane. As illustrated in Figure 1, the wired data plane is much simpler compared to the wireless data plane. It involves only the switches, hosts, or any other devices that are connected through the switch. Even if the east/west-bound and southbound communication channels are secure, it does not guarantee that the communication between the devices within the data plane is secure.

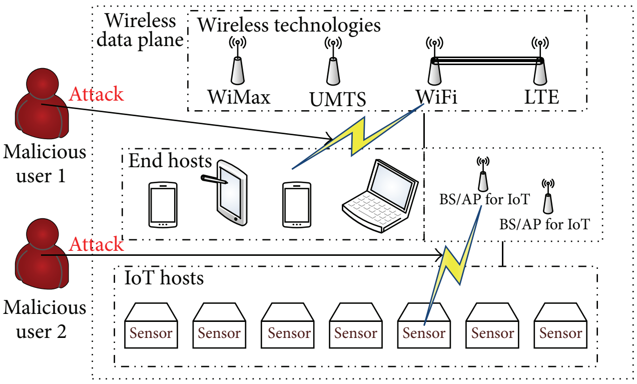


FIGURE 2: Possible attacks within the wireless data plane.

Possible attacks that can be launched within the data plane are Denial-of-Service (DoS) attack and man-in-the-middle attacks that intercept messages from the insecure communication channel and so forth.

2.2.2. Wireless Data Plane. With the growing use of mobile and Internet of Things (IoT) devices, the wireless data plane becomes enormous in size and involves complicated topologies. The capability crisis arises when mobile devices grow at a pace that exceeds the wireless spectrum capability. Therefore, the wireless resource management becomes crucial in order to sustain the network performance for the vast wireless data plane.

This also drives telecommunication providers to look for alternatives to fully utilize their network resources especially for the wireless portion as the wireless bandwidth is limited and expensive. One such solution for them will be to manage their wireless data plane with SDN.

Besides the wireless resource management, certificate management will also be involved if TLS were to be used to secure the data plane. The certificate management can be very complicated due to the enormous amount of devices involved. Besids that, it is also bandwidth consuming to perform the TLS handshake that involves certificate exchanges for authentication.

Figure 2 shows two sample attacks that can happen within the wireless data plane. Unlike the wired data plane, the malicious user does not need to have physical access to the switch to perform any malicious activity. As long as the malicious user is within the wireless coverage range, he/she can simply intercept the wireless communication or even modify the information if the wireless data plane is not protected. This compromises both the data integrity and confidentiality, and hence security is not a luxury feature but a necessity in the wireless communication especially for IoT devices that are deeply involved in personal privacy or even life threatening in the case of IoT devices used in healthcare.

2.3. IoT Application Protocols. In order for IoT devices to be compliant with the one Machine-to-Machine (oneM2M) [22] IoT standard, two widely used application protocols can be used to facilitate the communication within the wireless data plane: Message Queuing Telemetry Transport (MQTT) [23] and Constrained Application Protocol (CoAP) [24].

2.3.1. MQTT. MQTT is a lightweight asynchronous publish-subscribe messaging protocol that relies on the MQTT broker to facilitate the messages between the publisher and subscriber. MQTT employs Transmission Control Protocol (TCP) to provide a reliable communication channel between the IoT devices. The small header of MQTT protocol (2 bytes only) allows the message delivery with minimal bandwidth and yet reliable connection.

A simple architecture that relies on the MQTT protocol consists of only three main components, broker, publisher, and subscriber. Broker, as the name implies, is a messaging agency or server that distributes the messages between the publisher and subscriber. On the client side, the publisher will send the message to the broker on a particular topic while the subscriber that subscribed to the particular topic will receive the message from the broker. The MQTT protocol also allows multiple subscribers on a topic but multiple publishers on a single topic are not recommended even though it is possible to do so because the subscriber cannot differentiate the source of the message.

2.3.2. CoAP. Similar to MQTT, CoAP [25] is also a widely used lightweight protocol for IoT devices but the similarity ends here. Unlike MQTT, CoAP is based on Representational State Transfer (REST) architecture [26]. Therefore, CoAP relies on the four REST verbs to perform the Create, Read, Update and Delete (CRUD) operations as follows:

(i) POST: create a new resource identifier (ID).

(ii) GET: read/retrieve the information of the resource ID.

(iii) PUT: update the state of the resource ID.

(iv) DELETE: delete a resource ID.

CoAP employs User Datagram Protocol (UDP) to deliver the messages between the IoT devices and uses Uniform Resource Identifier (URI) to address the REST verbs to a particular resource ID. These REST verbs allow it to be integrated with the web, mobile, or even desktop applications easily.

2.4. Revocation. In the Public Key Cryptography (PKC), there are cases that the CA has to revoke the certificates even before they expire. This revocation can happen due to certificate loss by the user, a compromised certificate, an employee that has left the company and hence no longer have the right to use the certificate to access company information, and so forth. Certificate revocation can be done in one of the two common methods [27]:

(1) Certificate Revocation Lists (CRL).

(2) Online Certificate Status Protocol (OCSP).

CRL contains a list of revoked certificates that have yet to expire along with the reasons for revocation. In the case of the long certificate validity, for instance 1 year, this list will be likely very long assuming that the CA has issued a lot of certificates. Users will be required to obtain the complete list from the CA in order to verify the status of the certificate that they are going to use. Hence, this is an inefficient revocation method that involves a lot of network bandwidth to transmit the long list of revoked certificate from the CA.

In OCSP, the user will send a certificate status request to the CA and the CA will check it against its revocation list before informing the user on the certificate's validity. This method reduces the network bandwidth consumption but increases the CA computation cost and the CA will be required to be online to perform verification for the users.

Cooper [27] also worked on the attacks of the revocation list by manipulating the reason codes. If the categorization of the reason codes were not carefully analyzed, the user might be able to abuse it, for example, categorization of the seriousness of the revocation as the key is no longer needed or the key was compromised. The key that is no longer needed might not be handled as quickly as the compromised key and hence this gives the attacker time to use the key as long as it is not updated to the user.

Similar to the PKC, in order to prevent the key misuse of IBC, key revocation has to be done on a compromised node, failed node, and so forth. In the IBC of SDN, key revocation can be performed using one of the two methods:

(i) A trusted third party, mediator.

(ii) A network application on the management plane.

A trusted third party, mediator, was used in PKC [28] and IBC [29] to revoke any malicious user in the system. In this mechanism, the PKG generates a private key of the user and then splits it into two portions, for instance, portion A and portion B. PKG will then send portion A to the user and portion B to the mediator. Similar to the PKG, the mediator is a second trusted party that keeps portion B of the private keys for the users. Besides that, the mediator will keep track of malicious users in the system.

When a user requires his private key, he will send an encrypted message to the mediator for partial decryption. The mediator will then check whether the user is malicious or not. If the user has been compromised, the mediator will deny the operation and the user will be revoked from the system and unable to decrypt any message. If the user is genuine, the mediator will then send the partially decrypted message to the user and the user will then be able to decrypt the partially decrypted message and obtain the plaintext message.

However, the mediator method can be bandwidth consuming since the user and mediator are required to transmit the encrypted message and partially decrypted message through the network. Besides that, the mediator requires extra computing resources in order to perform the partial decryption and has to be online at all times. Figure 3 illustrates the key revocation with the help of the mediator.

The network application on the management plane [30] can be in the form of a firewall, intrusion detection system (IDS) or intrusion prevention system (IPS) application, identity management application, or solely revocation control application. It oversees the network-wide view as per Figure 1

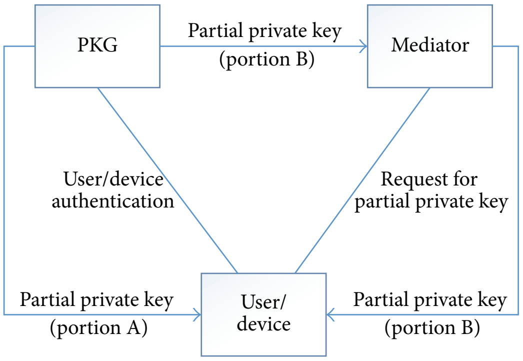

PKG
Partial private key (portion B)
Mediator
User/device authentication
Request for partial private key
Partial private key (portion A)
User/ device
Partial private key (portion B)

Figure 3: Key revocation with mediator.

(SDN app) and hence will be able to monitor the activities of each node easily.

When the network application detects any malicious behavior in a particular node, it will notify the controller to place the malicious node in a sandbox by blocking all network flows towards the switch. The controller can do so by performing flow removals towards the malicious node. The controller (PKG for the node within its domain) can then analyze the misbehavior. If it is proven safe to continue the communication, the controller can then generate a new private key for the affected switch and distribute this new key to the node.

This network application method might increase the controller's load with the addition of network application and flows removal but a distributed SDN is able to distribute the load with the load balancing mechanism. Hence, this method is preferable in the SDN environment especially where distributed SDN is concerned.

2.5. Identity-Based Cryptography (IBC). IBC was first proposed by Shamir in the form of an identity-based signature scheme [31] in 1985. His idea of IBC was then implemented by Sakai et al. [32] and Boneh and Franklin [33] for the encryption scheme with pairing in the years 2000 and 2001, respectively. Both their implementations went on to form the basis of many IBC researches thereafter, with more being based on the latter.

Similar to the PKC of TLS, IBC requires a Trusted Authority (TA) to act as a Private Key Generator (PKG) that generates keys for the users. In the SDN environment, controllers can also act as PKGs for the switches that are located within its domain.

In PKC, CA is used to generate the public and private key pairs whereas the PKG of IBC generates only the private keys. In IBC, public keys will be derived from the identity of the user; in this case, the user's identity can be in the form of the Media Access Control (MAC) address or any other network identities of the controllers and switches.

With IBC, the users or, in this case, the controllers, switches, or data stores, do not need to store every single public key of every user in the domain or obtain a particular public key from the TA on demand. This in turn saves storage space or network bandwidth that otherwise can decrease the network performance or translate into high system setup costs.

Smart [34], whose research was based on the implementation of Boneh and Franklin, initiated the usage of IBC in key agreement protocol. Chen and Kudla [35] improved Smart's protocol by solving the key escrow problem, allowing for communication between users of multiple TAs and providing forward secrecy.

2.6. Related Research. Santos et al. [36] proposed applying IBC to secure the communications between Master Controller-Secondary Controller (MC-SC), SC-SC, and the client- and server-side or their framework. In their proposal, the IBC protocol was based on the Sakai et al. protocol [32]. However, two issues become apparent if this were to be implemented for practical uses. In their proposal, the Type 1 pairing was used to establish the key. According to the research by Chen et al. [37] and Chatterjee et al. [38], Type 1 pairing is suitable for security levels of up to 80 bits; for security levels higher than 80 bits, the performance will degrade significantly. For details on the 4 pairing types, please refer to [37, 38].

Depending on the usage of the SDN, a security level of 80 bits might be sufficient for a network that manages time-sensitive data or data that might be useless after a short period of time [39]. For a SDN that manages time-insensitive data, a security level of higher than 80 bits should be used. Therefore, the key agreement protocol should be able to support other pairing types.

Another significant disadvantage of their proposal also lies in the key agreement protocol. It does not allow communication between devices that have obtained their private keys from different PKGs. Hence, it becomes impractical especially when it is to be used in the distributed SDN environment as a single PKG might not be sufficient in providing private keys to the entire network. This disadvantage also limits the scalability of the network.

3. SDN Security with IBC

Identity-based key agreement protocol is used to establish the symmetric session key that will secure the SDN communication. Due to the high amount of traffic that will be encrypted, the symmetric key is more preferable to the asymmetric one. The asymmetric keys will be used to derive the symmetric key for session communication.

Our proposal [40] employs the pairing-based key agreement protocol with separate TAs introduced by Chen and Kudla [35] which was originally not meant for SDN. In this implementation, it assumes that the different PKGs share the same domain parameters which is plausible in the case of the SDN setup.

Although it is worth noting that the controller can act as a PKG for all the devices located within the network at the same time, due to the importance of PKG in IBC, it is advisable to have different PKGs generating the private keys for the controllers and switches. By doing so, it can provide better protection to the control plane when the controllers'

A ($S_{\text{A}} = s_1 Q_{\text{A}}$, $Q_{\text{A}} = H_1$(A's ID))		**B** ($S_{\text{B}} = s_2 Q_{\text{B}}$, $Q_{\text{B}} = H_1$(B's ID))
Choose $a \in_R \mathbb{Z}_q^*$		Choose $b \in_R \mathbb{Z}_q^*$
Compute $T_{\text{A}} = aP$, $W_{\text{A}} = aP_{\text{pub,ctlr2}}$		Compute $T_{\text{B}} = bP$, $W_{\text{B}} = bP_{\text{pub,ctlr1}}$
	$\xrightarrow{T_{\text{A}}}$	
	$\xleftarrow{T_{\text{B}}}$	
$K_{\text{AB}} = \widehat{e}(S_{\text{A}}, T_{\text{B}})\widehat{e}(Q_{\text{B}}, W_{\text{A}})$		$K_{\text{BA}} = \widehat{e}(S_{\text{B}}, T_{\text{A}})\widehat{e}(Q_{\text{A}}, W_{\text{B}})$
$K_{\text{AB}} = \widehat{e}(s_1 Q_{\text{A}}, bP)\widehat{e}(Q_{\text{B}}, as_2 P)$		$K_{\text{BA}} = \widehat{e}(s_2 Q_{\text{B}}, aP)\widehat{e}(Q_{\text{A}}, bs_1 P)$
$K_{\text{AB}} = \widehat{e}(Q_{\text{A}}, P)^{s_1 b}\widehat{e}(Q_{\text{B}}, P)^{s_2 a}$	$=$	$K_{\text{BA}} = \widehat{e}(Q_{\text{B}}, P)^{s_2 a}\widehat{e}(Q_{\text{A}}, P)^{s_1 b}$

Box 1: Illustration of the establishment of the IBC key agreement protocol.

PKGs are disconnected from the network, thus lowering the risk of the PKGs being exposed or attacked.

However, the controllers' PKGs will be required to reconnect to the network when the private keys expire (can be set at a fixed interval) or when a malicious controller is detected (can be triggered by a network application) whereby a new set of private keys will be needed to secure the control plane.

(1) System Setup. Suppose there are two PKGs, PKG_1 and PKG_2, that generate the private keys for the controllers of the SDN. Each has a public/private key pair, $(P, s_1 P \in G_1, s_1 \in_R \mathbb{Z}_q^*)$ and $(P, s_2 P \in G_1, s_2 \in_R \mathbb{Z}_q^*)$, respectively, where P and G_1 have been globally agreed on.

Controller A, $\text{controller}_{\text{A}}$, is registered under PKG_1 with its private key, $S_{\text{A}} = s_1 Q_{\text{A}}$, where $Q_{\text{A}} = H_1$ ($\text{controller}_{\text{A}}$'s ID).

Controller B, $\text{controller}_{\text{B}}$, is registered under PKG_2 with $S_{\text{B}} = s_2 Q_{\text{B}}$, where $Q_{\text{B}} = H_1$ ($\text{controller}_{\text{B}}$'s ID). (Note that controllers A and B can also act as PKGs for the switches that are located within their respective domains.) H_1 is a cryptographic hash function; $H_1 : \{0,1\}^* \rightarrow G_1$.

(2) Key Establishment. If controller A wants to communicate with controller B, the IBC key agreement protocol will be initiated to establish the shared session keys. Box 1 illustrates the key establishment of the protocol. Each of controllers A and B picks nonce at random, a and $b \in_R \mathbb{Z}_q^*$, and computes $T_{\text{A}} = aP$, $W_{\text{A}} = aP_{\text{pub,ctlr2}}$ and $T_{\text{B}} = bP$, $W_{\text{B}} = bP_{\text{pub,ctlr1}}$, respectively, where $P_{\text{pub,ctlr1}} = s_1 P$ and $P_{\text{pub,ctlr2}} = s_2 P$. These computed values will then be exchanged between the two controllers.

At the end of the protocol, controller A computes the shared key; $K_{\text{AB}} = \widehat{e}(S_{\text{A}}, T_{\text{B}})\widehat{e}(Q_{\text{B}}, W_{\text{A}})$, and controller B computes the shared key; $K_{\text{BA}} = \widehat{e}(S_{\text{B}}, T_{\text{A}})\widehat{e}(Q_{\text{A}}, W_{\text{B}})$:

$$\therefore K_{\text{BA}} = K_{\text{AB}}. \quad (1)$$

Then, the shared session key can be generated by hashing the key; $SK = h_2(K_{\text{AB}})$, where h_2 is a secure hash function for the purpose of key derivation. However, this session key does not offer TA forward secrecy and the key escrow issue still exists.

If TA forward secrecy is required and key escrow is not allowed, the previously generated ephemeral keys can also be used to generate a variant of the shared session key; $SK = h_2(K_{\text{AB}}, abP)$ as suggested by Chen and Kudla [35]. These shared session keys that have been established by the IBC key agreement protocol will then be used to provide message confidentiality.

3.1. Southbound Security. Controllers that were registered under their respective PKGs can also act as PKGs for the switches located within their domain for southbound communication. Southbound security is more straightforward as it does not usually involve multiple domains. However, during the switch migration from one controller to another, interdomain communication is still required so as to hand over the switch swiftly. Therefore, the same key agreement protocol can also be used for southbound communications.

To apply the IBC key agreement protocol to southbound security, the role of the PKGs will be transferred to the controllers themselves, that is, to generate the private keys for the switches. This reduces the load of the PKGs that manages the control plane and also isolates the two communication channels.

3.2. Data Plane Security

3.2.1. Wired Data Plane. Multidomain key agreement is also helpful in the wired data plane to allow the communication between hosts that obtained their private keys from different controllers through the southbound communication. The key agreement protocol enables the multidomain communication without keeping multiple public keys for each domain. This allows the data plane devices to establish the session by using the identity information and the exchanged parameters.

3.2.2. Wireless Data Plane. The wireless data plane involves multiple wireless technologies and it gets complicated when the end hosts try to establish the communication through heterogeneous backend infrastructure. In order for the heterogeneous communication to take place, a multidomain domain key agreement protocol is used for such a communication.

Despite the multiple wireless technologies used, the underlying protocol to exchange messages may be the same. For example, the two popular communication protocols used by IoT devices, MQTT and CoAP, are able to work with the TCP and UDP, respectively, regardless of the underlying wireless technologies used. Therefore, application of the IBC protocol to either MQTT or CoAP will be able to provide the

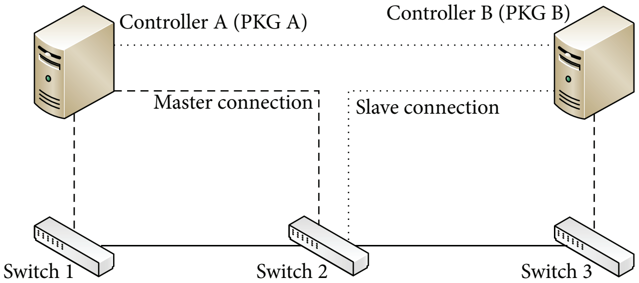


FIGURE 4: Switch migration for switch 2.

necessary security for the communication between the IoT devices.

4. Application Scenarios

4.1. Southbound Communication. Figure 4 shows the switch migration for switch 2. Switch 2 is allowed to migrate between controllers A and B that also act as the PKGs for switch 2. The load balancing application will notify the respective controller that will be taking over the switch for migration and key establishment will be performed between the switch and controller B.

During this transition period, switch 2 is still able to communicate with switch 1 or switch 3 by using the old key (the key obtained from controller A) and switch over to the new key (the key obtained from the controller that will be taken over) once the key establishment and handover process are completed. When switch migration is completed, the switch can then dispose of the old key and proceed with communication using the new key.

This eases the switch migration process with a single identity information and saves the computing resources at the controller for certificate issuance and management. Besides, it also reduces the bandwidth required for the new certificate distribution in TLS.

4.2. Data Plane Communication

4.2.1. Wired Data Plane. Figure 5 illustrates the interdomain communication within the wired data plane that was made possible with an interdomain key agreement protocol. With the help of the protocol, it allows the communication between switch 1-switch 2 and host A-host B to be established without having a second public key or identity in this case.

Unlike TLS, switch 1 and host A do not need to have controller B issue a new public key to them in order to derive the session key. The same goes to switch 2 and host B without needing a second public key from controller A. Therefore, the key agreement protocol saves computing resources at the controller that generates and manages the certificate. Besides, it also saves the bandwidth that will be used to distribute the certificate.

4.2.2. Wireless Data Plane. The wireless data plane involves heterogeneous wireless technologies and hence the advantage of this multidomain capable IBC key agreement protocol actually benefits this data plane communication the most. The certificate management was simplified with the controller managing only the private keys for each of these devices instead of managing multiple certificates for each wireless technology.

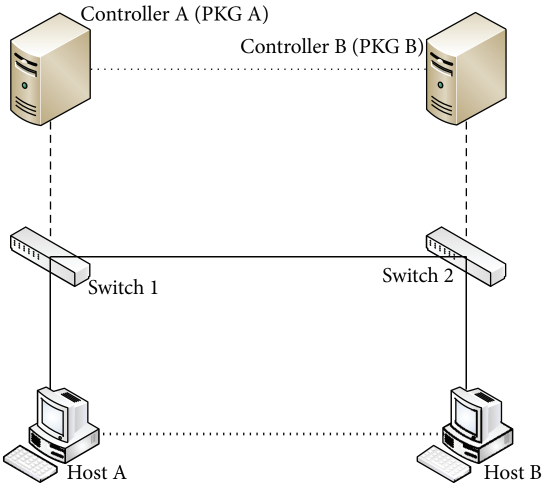


FIGURE 5: Interdomain data plane communication.

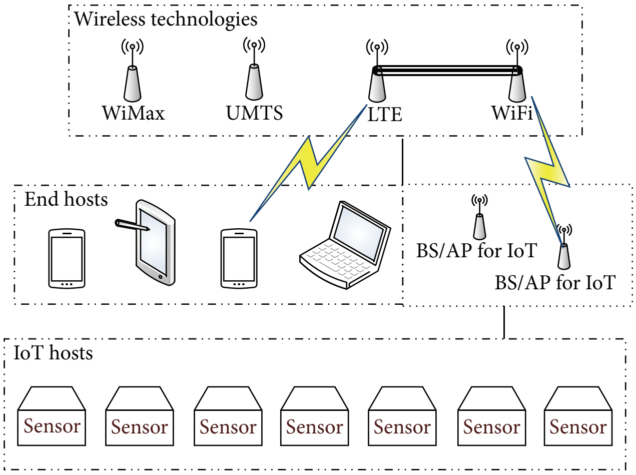


FIGURE 6: Heterogeneous wireless communication.

Figure 6 illustrates the heterogeneous wireless communication where the mobile device is able to communicate with the base station for IoT devices through the two different wireless technologies. The mobile device connects to the network via Long Term Evolution (LTE) while the base station connects to the network via Wireless Fidelity (WiFi). The communication between the LTE base station and WiFi access point is possible with only the identity information of the mobile devices. It saves the hassle of having multiple certificates for a single device.

Besides that, the IBC key agreement protocol reduces the bandwidth consumption as compared to the TLS for no certificate exchanges will be required. The bandwidth saved

Table 1: Security comparisons between SDN projects.

SDN projects	Security		
	East/west-bound	Southbound	Data plane
OpenDayLight	—	TLS	TLS
HP VAN SDN	—	TLS	TLS
ONOS	TLS	TLS	TLS
D-SDN (Santos et al. [36])	IBC (single domain)	IBC (single domain)	IBC (single domain)
Our proposal	IBC (multidomain)	IBC (multidomain)	IBC (multidomain)

can then accommodate more wireless hosts with the same infrastructure which in turn saves cost.

5. Analysis and Discussions

OpenDayLight and HP VAN SDN implemented TLS to secure the southbound security but neglected the east/west-bound security which is also crucial for a distributed SDN. ONOS secured both the southbound and east/west-bound communications but the system setup is still rather inconvenient since the user is still required to deploy the keys manually.

Santos et al. [36] proposed securing the communications between MC-SC, SC-SC, and the client- and server-side or their framework with IBC. However, there are several flaws in their proposal and their scheme is not suitable for a distributed SDN when interdomain communication is required.

Table 1 is a comparison between the securities of different SDN projects. To simplify the comparison, SDN projects that do not implement a secure channel were excluded from this table. Do note that the protocol can also be added to any open source SDN project as a security module.

Since TCP is the preferred transport protocol to provide reliable communication channel, we chose the MQTT protocol which is one of the most commonly used communication protocols in the world of IoT (relies on TCP) besides CoAP (relies on UDP) and secured it with TLS. Then, we analyzed the communication between the MQTT broker (server) and MQTT client (publisher) to find out the possible bandwidth saved by using IBC instead of TLS.

In the communication analysis, we used a network sniffing tool, Wireshark [41], to monitor the network traffic and the exchanged information between the MQTT client (publisher) and MQTT broker. Figure 7 illustrates a complete TLS handshake for the MQTT protocol and details of handshake messages will be shown in Figure 7.

Figure 8 shows the first message exchange initiated by the MQTT client (in this case, the publisher) to the MQTT broker. The communication started with the TLS handshake mechanism, client hello. In this message, it sends the client's nonce, cipher suites, compression methods, extensions, and any other special features that the client supports to the server or in this case the MQTT broker.

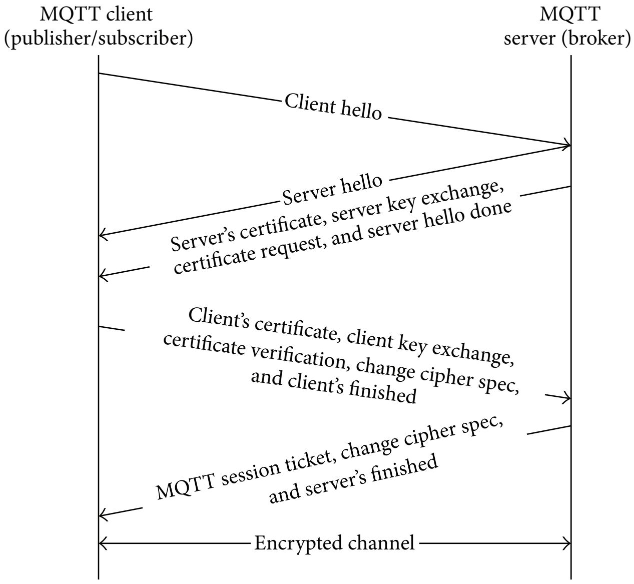
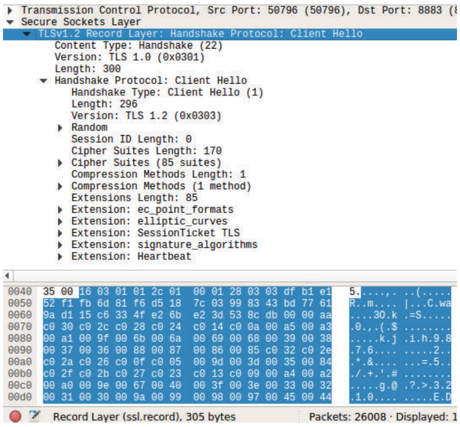

MQTT client
(publisher/subscriber)
MQTT
server (broker)
Client hello
Server hello
Server's certificate, server key exchange,
certificate request, and server hello done
Client's certificate, client key exchange,
certificate verification, change cipher spec,
and client's finished
MQTT session ticket, change cipher spec,
and server's finished
Encrypted channel

Figure 7: MQTT-TLS handshake mechanism.

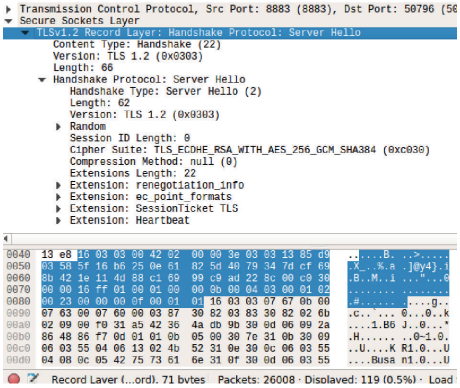


```
▸ Transmission Control Protocol, Src Port: 50796 (50796), Dst Port: 8883 (8
▾ Secure Sockets Layer
  ▾ TLSv1.2 Record Layer: Handshake Protocol: Client Hello
      Content Type: Handshake (22)
      Version: TLS 1.0 (0x0301)
      Length: 300
    ▾ Handshake Protocol: Client Hello
        Handshake Type: Client Hello (1)
        Length: 296
        Version: TLS 1.2 (0x0303)
      ▸ Random
        Session ID Length: 0
        Cipher Suites Length: 170
      ▸ Cipher Suites (85 suites)
        Compression Methods Length: 1
      ▸ Compression Methods (1 method)
        Extensions Length: 85
      ▸ Extension: ec_point_formats
      ▸ Extension: elliptic_curves
      ▸ Extension: SessionTicket TLS
      ▸ Extension: signature_algorithms
      ▸ Extension: Heartbeat

0040  35 00 16 03 01 01 2c 01  00 01 28 03 03 df b1 e1   5.....,. ..(.....
0050  52 f1 fb 6d 81 f6 d5 18  7c 03 99 83 43 bd 77 61   R..m.... |...C.wa
0060  9a d1 15 c6 33 4f e2 6b  e2 3d 53 8c db 00 00 aa   ....3O.k .=S.....
0070  c0 30 c0 2c c0 28 c0 24  c0 14 c0 0a 00 a5 00 a3   .0.,.(.$ ........
0080  00 a1 00 9f 00 6b 00 6a  00 69 00 68 00 39 00 38   .....k.j .i.h.9.8
0090  00 37 00 36 00 88 00 87  00 86 00 85 c0 32 c0 2e   .7.6.... .....2..
00a0  c0 2a c0 26 c0 0f c0 05  00 9d 00 3d 00 35 00 84   .*.&.... ...=.5..
00b0  c0 2f c0 2b c0 27 c0 23  c0 13 c0 09 00 a4 00 a2   ./.+.'.# ........
00c0  00 a0 00 9e 00 67 00 40  00 3f 00 3e 00 33 00 32   .....g.@ .?.>.3.2
00d0  00 31 00 30 00 9a 00 99  00 98 00 97 00 45 00 44   .1.0.... .....E.D

Record Layer (ssl.record), 305 bytes        Packets: 26008 · Displayed: 1
```

Figure 8: MQTT: client hello message from the MQTT publisher.

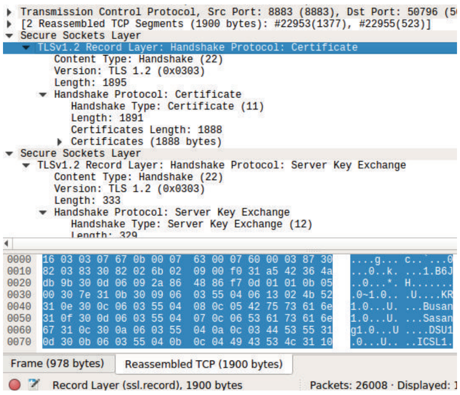
Similar to the client hello message, Figure 9 shows the server hello message which is the server's response upon receiving the client hello message. In this message, the server (MQTT broker) will choose the best or most suitable cryptography setting such as the most secure cipher suite supported by the client, compression method used, or any other extensions that were included in the client hello message.

If the IBC protocols were to be applied here, the size of the cipher suites, compression methods, or other features supported might be much lesser initially and hence a smaller client hello message can be used. However, if the IBC protocol is adopted by more organizations, it might grow to a size similar to the TLS protocol. Hence, the bandwidth saved in client hello message can be negligible then. On the other hand, the message length of the server hello message should be similar whether it is in TLS or IBC mode since it only chooses one of the cryptography settings from each type.



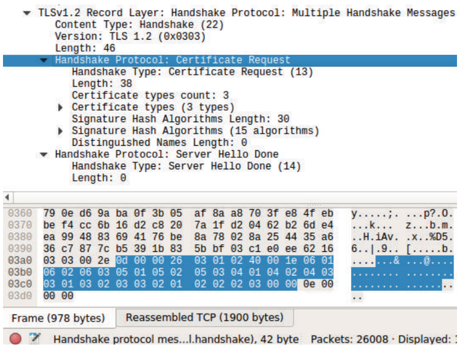
```
▸ Transmission Control Protocol, Src Port: 8883 (8883), Dst Port: 50796 (50
▾ Secure Sockets Layer
  ▾ TLSv1.2 Record Layer: Handshake Protocol: Server Hello
      Content Type: Handshake (22)
      Version: TLS 1.2 (0x0303)
      Length: 66
    ▾ Handshake Protocol: Server Hello
        Handshake Type: Server Hello (2)
        Length: 62
        Version: TLS 1.2 (0x0303)
      ▸ Random
        Session ID Length: 0
        Cipher Suite: TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 (0xc030)
        Compression Method: null (0)
        Extensions Length: 22
      ▸ Extension: renegotiation_info
      ▸ Extension: ec_point_formats
      ▸ Extension: SessionTicket TLS
      ▸ Extension: Heartbeat

0040  13 e8 16 03 03 00 42 02  00 00 3e 03 03 13 85 d9   .......B. ..>.....
0050  03 58 5f 16 b6 25 0e 61  82 5d 40 79 34 7d cf 69   .X_..%.a .]@y4}.i
0060  8b 42 1e 11 4d 88 c1 69  99 c9 ad 22 8c 00 c0 30   .B..M..i ..."...0
0070  00 00 16 ff 01 00 01 00  00 0b 00 04 03 00 01 02   ........ ........
0080  00 23 00 00 00 0f 00 01  01 16 03 03 07 67 0b 00   .#...... .....g..
0090  07 63 00 07 60 00 03 87  30 82 03 83 30 82 02 6b   .c..`... 0...0..k
00a0  02 09 00 f0 31 a5 42 36  4a db 9b 30 0d 06 09 2a   ....1.B6 J..0...*
00b0  86 48 86 f7 0d 01 01 0b  05 00 30 7e 31 0b 30 09   .H...... ..0~1.0.
00c0  06 03 55 04 06 13 02 4b  52 31 0e 30 0c 06 03 55   ..U....K R1.0...U
00d0  04 08 0c 05 42 75 73 61  6e 31 0f 30 0d 06 03 55   ....Busa n1.0...U

Record Layer (...ord), 71 bytes    Packets: 26008 · Displayed: 119 (0.5%) · Load
```

Figure 9: Server hello message from MQTT broker.



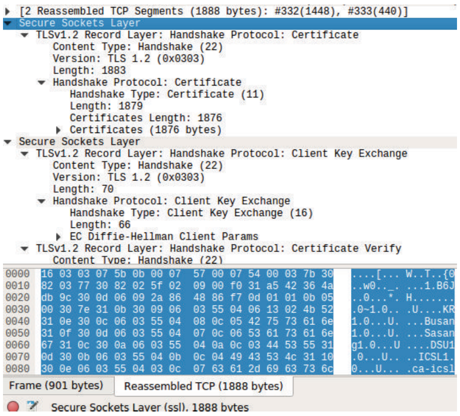
```
▸ Transmission Control Protocol, Src Port: 8883 (8883), Dst Port: 50796 (5
▸ [2 Reassembled TCP Segments (1900 bytes): #22953(1377), #22955(523)]
▾ Secure Sockets Layer
  ▾ TLSv1.2 Record Layer: Handshake Protocol: Certificate
      Content Type: Handshake (22)
      Version: TLS 1.2 (0x0303)
      Length: 1895
    ▾ Handshake Protocol: Certificate
        Handshake Type: Certificate (11)
        Length: 1891
        Certificates Length: 1888
      ▸ Certificates (1888 bytes)
▾ Secure Sockets Layer
  ▾ TLSv1.2 Record Layer: Handshake Protocol: Server Key Exchange
      Content Type: Handshake (22)
      Version: TLS 1.2 (0x0303)
      Length: 333
    ▾ Handshake Protocol: Server Key Exchange
        Handshake Type: Server Key Exchange (12)
        Length: 329

0000  16 03 03 07 67 0b 00 07  63 00 07 60 00 03 87 30   ....g... c..`...0
0010  82 03 83 30 82 02 6b 02  09 00 f0 31 a5 42 36 4a   ...0..k. ...1.B6J
0020  db 9b 30 0d 06 09 2a 86  48 86 f7 0d 01 01 0b 05   ..0...*. H.......
0030  00 30 7e 31 0b 30 09 06  03 55 04 06 13 02 4b 52   .0~1.0.. .U....KR
0040  31 0e 30 0c 06 03 55 04  08 0c 05 42 75 73 61 6e   1.0...U. ...Busan
0050  31 0f 30 0d 06 03 55 04  07 0c 06 53 61 73 61 6e   1.0...U. ...Sasan
0060  67 31 0c 30 0a 06 03 55  04 0a 0c 03 44 53 55 31   g1.0...U ....DSU1
0070  0d 30 0b 06 03 55 04 0b  0c 04 49 43 53 4c 31 10   .0...U.. ..ICSL1.

Frame (978 bytes)   Reassembled TCP (1900 bytes)

Record Layer (ssl.record), 1900 bytes    Packets: 26008 · Displayed: 1
```

Figure 10: MQTT broker sends its certificate to the client.



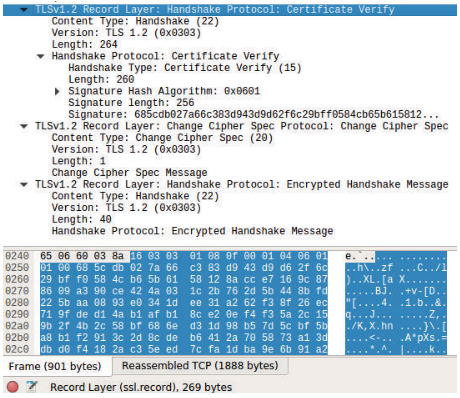
```
  ▾ TLSv1.2 Record Layer: Handshake Protocol: Multiple Handshake Messages
      Content Type: Handshake (22)
      Version: TLS 1.2 (0x0303)
      Length: 46
    ▾ Handshake Protocol: Certificate Request
        Handshake Type: Certificate Request (13)
        Length: 38
        Certificate types count: 3
      ▸ Certificate types (3 types)
        Signature Hash Algorithms Length: 30
      ▸ Signature Hash Algorithms (15 algorithms)
        Distinguished Names Length: 0
    ▾ Handshake Protocol: Server Hello Done
        Handshake Type: Server Hello Done (14)
        Length: 0

0360  79 0e d6 9a ba 0f 3b 05  af 8a a8 70 3f e8 4f eb   y.....;. ...p?.O.
0370  be f4 cc 6b 16 d2 c8 20  7a 1f d2 04 62 b2 6d e4   ...k...  z...b.m.
0380  ea 99 48 83 69 41 76 be  8a 78 02 8a 25 44 35 a6   ..H.iAv. .x..%D5.
0390  36 c7 87 7c b5 39 1b 83  5b bf 03 c1 e0 ee 62 16   6..|.9.. [.....b.
03a0  03 03 00 2e 0d 00 00 26  03 01 02 40 00 1e 06 01   .......& ...@....
03b0  06 02 06 03 05 01 05 02  05 03 04 01 04 02 04 03   ........ ........
03c0  03 01 03 02 03 03 02 01  02 02 02 03 00 00 0e 00   ........ ........
03d0  00 00                                              ..

Frame (978 bytes)   Reassembled TCP (1900 bytes)

Handshake protocol mes...l.handshake), 42 byte    Packets: 26008 · Displayed: 1
```

Figure 11: MQTT broker request for client's certificate.



```
▸ [2 Reassembled TCP Segments (1888 bytes): #332(1448), #333(440)]
▾ Secure Sockets Layer
  ▾ TLSv1.2 Record Layer: Handshake Protocol: Certificate
      Content Type: Handshake (22)
      Version: TLS 1.2 (0x0303)
      Length: 1883
    ▾ Handshake Protocol: Certificate
        Handshake Type: Certificate (11)
        Length: 1879
        Certificates Length: 1876
      ▸ Certificates (1876 bytes)
▾ Secure Sockets Layer
  ▾ TLSv1.2 Record Layer: Handshake Protocol: Client Key Exchange
      Content Type: Handshake (22)
      Version: TLS 1.2 (0x0303)
      Length: 70
    ▾ Handshake Protocol: Client Key Exchange
        Handshake Type: Client Key Exchange (16)
        Length: 66
      ▸ EC Diffie-Hellman Client Params
  ▾ TLSv1.2 Record Layer: Handshake Protocol: Certificate Verify
      Content Type: Handshake (22)

0000  16 03 03 07 5b 0b 00 07  57 00 07 54 00 03 7b 30   ....[... W..T..{0
0010  82 03 77 30 82 02 5f 02  09 00 f0 31 a5 42 36 4a   ..w0.._. ...1.B6J
0020  db 9c 30 0d 06 09 2a 86  48 86 f7 0d 01 01 0b 05   ..0...*. H.......
0030  00 30 7e 31 0b 30 09 06  03 55 04 06 13 02 4b 52   .0~1.0.. .U....KR
0040  31 0e 30 0c 06 03 55 04  08 0c 05 42 75 73 61 6e   1.0...U. ...Busan
0050  31 0f 30 0d 06 03 55 04  07 0c 06 53 61 73 61 6e   1.0...U. ...Sasan
0060  67 31 0c 30 0a 06 03 55  04 0a 0c 03 44 53 55 31   g1.0...U ....DSU1
0070  0d 30 0b 06 03 55 04 0b  0c 04 49 43 53 4c 31 10   .0...U.. ..ICSL1.
0080  30 0e 06 03 55 04 03 0c  07 63 61 2d 69 63 73 6c   0...U... .ca-icsl

Frame (901 bytes)   Reassembled TCP (1888 bytes)

Secure Sockets Layer (ssl), 1888 bytes
```

Figure 12: MQTT client sends its certificate to the MQTT broker.



```
  ▾ TLSv1.2 Record Layer: Handshake Protocol: Certificate Verify
      Content Type: Handshake (22)
      Version: TLS 1.2 (0x0303)
      Length: 264
    ▾ Handshake Protocol: Certificate Verify
        Handshake Type: Certificate Verify (15)
        Length: 260
      ▸ Signature Hash Algorithm: 0x0601
        Signature length: 256
        Signature: 685cdb027a66c383d943d9d62f6c29bff0584cb65b615812...
  ▾ TLSv1.2 Record Layer: Change Cipher Spec Protocol: Change Cipher Spec
      Content Type: Change Cipher Spec (20)
      Version: TLS 1.2 (0x0303)
      Length: 1
      Change Cipher Spec Message
  ▾ TLSv1.2 Record Layer: Handshake Protocol: Encrypted Handshake Message
      Content Type: Handshake (22)
      Version: TLS 1.2 (0x0303)
      Length: 40
      Handshake Protocol: Encrypted Handshake Message

0240  65 06 60 03 8a 16 03 03  01 08 0f 00 01 04 06 01   e.`..... ........
0250  01 00 68 5c db 02 7a 66  c3 83 d9 43 d9 d6 2f 6c   ..h\..zf ...C../l
0260  29 bf f0 58 4c b6 5b 61  58 12 8a cc e7 16 9c 87   )..XL.[a X.......
0270  86 09 a3 90 ce 42 4a 03  1c 2b 76 2d 5b 44 8b fd   .....BJ. .+v-[D..
0280  22 5b aa 08 93 e0 34 1d  ee 31 a2 62 f3 8f 26 ec   "[....4. .1.b..&.
0290  71 9f de d1 4a b1 af b1  8c e2 0e f4 f3 5a 2c 15   q...J... .....Z,.
02a0  9b 2f 4b 2c 58 bf 68 6e  d3 1d 98 b5 7d 5c bf 5b   ./K,X.hn ....}\.[
02b0  a8 b1 f2 91 3c 2d 8c de  b6 41 2a 70 58 73 a1 3d   ....<-.. .A*pXs.=
02c0  db d0 f4 18 2a c3 5e ed  7c fa 1d ba 9e 6b 91 a2   ....*.^. |....k..

Frame (901 bytes)   Reassembled TCP (1888 bytes)

Record Layer (ssl.record), 269 bytes
```

Figure 13: MQTT client sends the certificate verification message to the broker.

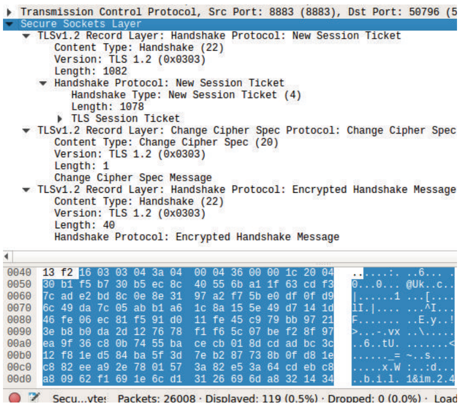
Therefore, the bandwidth saved in the client hello and server hello messages are negligible.

After the MQTT broker (server) sends the server hello message to the client, it will proceed to send its own certificate to the client as shown in Figure 10 and send a certificate request message to the client as shown in Figure 11 so that the server and client can authenticate each other prior to sending any actual data. If IBC protocol is used here, these two messages will no longer be needed because the client will be able to derive the "certificate" from the server's identity and the same goes to the server where it will be able to derive the "client's certificate" with the client's identity.

Figure 10 shows that the certificate handshake message used 1900 bytes from the bandwidth while the certificate request message in Figure 11 used 42 bytes of data, resulting in a total of 1942 bytes saved. The bandwidth saved here is significant as the size of the entire client hello and server hello messages is only 376 bytes. By switching it to the IBC protocol, the bandwidth saved here can actually accommodate for another 5 pairs of server-client hello message exchanges.

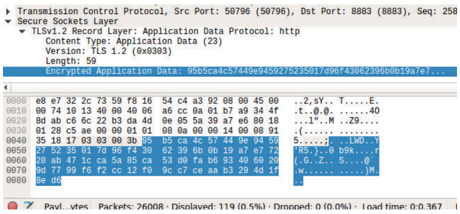
Figure 12 shows the MQTT client (in this case, publisher) sending its own certificate to the MQTT broker upon receiving the certificate request message from the broker and the information required to verify the client was sent in the certificate verification message as shown in Figure 13 to the broker. Again, if the IBC protocol was to be applied here, these handshake messages will be redundant as the broker is able to derive the "client's certificate" from the client's identity and, with the exchanged nonce and derived "certificates," the broker will be able to verify the client without needing the certificate verification message as well.

Figure 12 shows that the client's certificate used 1888 bytes while the certificate verification message in Figure 13 used 269 bytes of data. Again, a total of 2157 bytes can be saved by switching to the IBC protocol.



```
▸ Transmission Control Protocol, Src Port: 8883 (8883), Dst Port: 50796 (5
▾ Secure Sockets Layer
  ▾ TLSv1.2 Record Layer: Handshake Protocol: New Session Ticket
      Content Type: Handshake (22)
      Version: TLS 1.2 (0x0303)
      Length: 1082
    ▾ Handshake Protocol: New Session Ticket
        Handshake Type: New Session Ticket (4)
        Length: 1078
      ▸ TLS Session Ticket
  ▾ TLSv1.2 Record Layer: Change Cipher Spec Protocol: Change Cipher Spec
      Content Type: Change Cipher Spec (20)
      Version: TLS 1.2 (0x0303)
      Length: 1
      Change Cipher Spec Message
  ▾ TLSv1.2 Record Layer: Handshake Protocol: Encrypted Handshake Message
      Content Type: Handshake (22)
      Version: TLS 1.2 (0x0303)
      Length: 40
      Handshake Protocol: Encrypted Handshake Message

0040  13 f2 16 03 03 04 3a 04  00 04 36 00 00 1c 20 04   ......:. ..6... .
0050  30 b1 f5 b7 30 b5 ec 8c  40 55 6b a1 1f 63 cd f3   0...0... @Uk..c..
0060  7c ad e2 bd 8c 0e 8e 31  97 a2 f7 5b e0 df 0f d9   |......1 ...[....
0070  6c 49 da 7c 05 ab b1 a6  1c 8a 15 5e 49 d7 14 1d   lI.|.... ...^I...
0080  46 fe 06 ec 81 f5 91 d0  11 fe 45 c9 79 bb 97 21   F....... ..E.y..!
0090  3e b8 b0 da 2d 12 76 78  f1 f6 5c 07 be f2 8f 97   >...-.vx ..\.....
00a0  ea 9f 36 c8 0b 74 55 ba  ce cb 01 8d cd ad bc 3c   ..6..tU. .......<
00b0  12 f8 1e d5 84 ba 5f 3d  7e b2 87 73 8b 0f d8 1e   ......_= ~..s....
00c0  c8 82 ee a9 2e 78 01 57  3a 82 e5 3a 64 cd eb c8   .....x.W :..:d...
00d0  a8 09 62 f1 69 1e 6c d1  31 26 69 6d a8 32 14 34   ..b.i.l. 1&im.2.4

Secu...vte:   Packets: 26008 · Displayed: 119 (0.5%) · Dropped: 0 (0.0%) · Load
```

Figure 14: Session ticket from MQTT broker.



```
▸ Transmission Control Protocol, Src Port: 50796 (50796), Dst Port: 8883 (8883), Seq: 258
▾ Secure Sockets Layer
  ▾ TLSv1.2 Record Layer: Application Data Protocol: http
      Content Type: Application Data (23)
      Version: TLS 1.2 (0x0303)
      Length: 59
      Encrypted Application Data: 95b5ca4c57449e9459275235017d96f43062396b0b19a7e7...

0000  e8 e7 32 2c 73 59 f8 16  54 c4 a3 92 08 00 45 00   ..2,sY.. T.....E.
0010  00 74 10 13 40 00 40 06  a6 cc 0a 01 b7 a9 34 4f   .t..@.@. ......4O
0020  8d ab c6 6c 22 b3 da 4d  0e 05 5a 39 a7 e6 80 18   ...l"..M ..Z9....
0030  01 28 c5 ae 00 00 01 01  08 0a 00 00 14 00 08 91   .(...... ........
0040  35 18 17 03 03 00 3b 95  b5 ca 4c 57 44 9e 94 59   5.....;. ..LWD..Y
0050  27 52 35 01 7d 96 f4 30  62 39 6b 0b 19 a7 e7 72   'R5.}..0 b9k....r
0060  28 ab 47 1c ca 5a 85 ca  53 d0 fa b6 93 40 60 20   (.G..Z.. S....@`
0070  9d 77 99 f6 f2 cc 12 f0  9c c7 ce aa b3 29 4d 1f   .w...... .....)M.
0080  8e d6                                              ..

Payl...ytes   Packets: 26008 · Displayed: 119 (0.5%) · Dropped: 0 (0.0%) · Load time: 0:0.367
```

Figure 15: Encrypted channel.

Finally, Figure 14 shows the session establishment of the server-client pair while Figure 15 shows the first application data exchange with the established session's cryptographic parameters. The bandwidth required for these message exchanges should be similar whether it is in TLS or IBC protocol if the same handshake mechanism and symmetric cryptography are used because the size of a session ticket and change cipher spec message is not affected by the cryptography protocol.

The server's finished or the encrypted handshake message shown in Figure 14 and the first encrypted data in Figure 15 will have a similar length regardless of whether TLS or IBC was used because TLS and IBC are used to derive the symmetric key with the provided asymmetric keys via the handshake mechanism and the symmetric key cryptography used to encrypt these messages will be the same for both TLS and IBC. Hence, no bandwidth can be saved here.

By referring to Figure 7, the size of each handshake message is listed as follows:

Client hello message: 305 bytes (refer to Figure 8).

Server hello message: 71 bytes (refer to Figure 9).

Server's certificate: 1900 bytes (refer to Figure 10).

Server key exchange: 338 bytes (5 bytes of header + length, refer to Figure 10).

Certificate request and server hello message done: 51 bytes (refer to Figure 11).

Client's certificate: 1888 bytes (refer to Figure 12).

Client key exchange: 75 bytes (refer to Figure 12).

Certificate verification message: 269 bytes (refer to Figure 13).

Change cipher spec (client): 6 bytes (refer to Figure 13).

Client's finished: 45 bytes (refer to Figure 13).

New session ticket: 1087 bytes (refer to Figure 14).

Change cipher spec (server): 6 bytes (refer to Figure 14).

Server's finished: 45 bytes (refer to Figure 14).

Complete handshake: 6086 bytes.

Based on the analysis of the possible bandwidth that can be saved with IBC, it shows that removing the certificate related messages (server's certificate, certificate request, client's certificate, and certificate verification) from the TLS handshake alone can save up to 4099 bytes per communicating pair. A complete handshake requires 6086 bytes and 4099 bytes are more than half of the bandwidth saved.

Hence, this can lead to lower power consumption as less data are required to be exchanged between the server and client. The bandwidth saved here will also allow the same network infrastructure to accommodate for more communicating pairs and hence improve performance.

Besides that, the multidomain IBC protocol also allows the MQTT client that has the key generated by a controller of a different domain to communicate through the MQTT broker of another domain.

6. Conclusion

There are several notable benefits when securing the SDN communication with IBC; steps required for system setup will be significantly simplified while performance and network bandwidth vastly improved. Furthermore, as a smaller storage space is needed to store the keys, all these will then translate to a decrease in cost.

Besides that, IBC also speeds up the key exchange process since the two communicating parties do not need to obtain each other's public key from the CA to derive the session key. This reduces the time for the key setup and hence less time is spent on the key exchange process.

With this IBC scheme, even the nodes of different subdomains will be able to derive the shared session key. This not only improves the network scalability, but also eases switch migration in the southbound communication and enables interdomain data plane communication.

Lastly, the analysis shows the possible bandwidth that can be saved by switching to IBC; certificate exchanges are the most bandwidth consuming part during a handshake process. By removing the use of certificates in TLS, bandwidth consumption is reduced by as much as 4099 bytes per communicating pair during the handshake process. In other words, more than half of the bandwidth is saved as a complete handshake requires 6086 bytes.

This will lead to lower power consumption of the IoT devices and higher network performance as it can now accommodate more IoT devices without upgrading the network infrastructure.

Disclosure

Part of this paper was presented in the International Conference on Ubiquitous and Future Networks (ICUFN) 2015.

Competing Interests

The authors declare that they have no competing interests.

Acknowledgments

This research was supported by Basic Science Research Program through National Research Foundation of Korea funded by the Ministry of Education, Science and Technology (Grant no. NRF-2014R1A1A2060021). The third author (Hoon-Jae Lee) was supported by the National Research Foundation of Korea, NRF2011 Project (Grant no. NRF-2011-0023076) and NRF2016 Project (Grant no. NRF-2016R1D1A1B01011908).

References

[1] A. Dixit, F. Hao, S. Mukherjee, T. V. Lakshman, and R. Kompella, "Towards an elastic distributed SDN controller," in *Proceedings of the 2nd ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking (HotSDN '13)*, pp. 7–12, Hong Kong, August 2013.

[2] M. F. Bari, S. R. Chowdhury, R. Ahmed, and R. Boutaba, "PolicyCop: an autonomic QoS policy enforcement framework for software defined networks," in *Proceedings of the Workshop on Software Defined Networks for Future Networks and Services (SDN4FNS '13)*, pp. 1–7, Trento, Italy, November 2013.

[3] R. Skowyra, S. Bahargam, and A. Bestavros, "Software-Defined IDS for securing embedded mobile devices," in *Proceedings of the 2013 IEEE High Performance Extreme Computing Conference (HPEC '13)*, pp. 1–7, Waltham, Mass, USA, September 2013.

[4] R. Braga, E. Mota, and A. Passito, "Lightweight DDoS flooding attack detection using NOX/OpenFlow," in *Proceedings of the 35th Annual IEEE Conference on Local Computer Networks (LCN '10)*, pp. 408–415, Denver, Colo, USA, October 2010.

[5] J. R. Ballad, I. Rae, and A. Akella, "Extensible and scalable network monitoring using openSAFE," in *Proceedings of the Internet Network Management Conference on Research on Enterprise Networking (INM/WREN '10)*, p. 8, 2010.

[6] C. Yu, C. Lumezanu, Y. Zhang, V. Singh, G. Jiang, and H. V. Madhyastha, "FlowSense: monitoring network utilization with zero measurement cost," in *Proceedings of the 14th International Conference on Passive and Active Measurement (PAM '13)*, vol. 7799, pp. 31–41, Springer, Berlin, Germany, 2013.

[7] J. H. Lam, S.-G. Lee, H.-J. Lee, and Y. E. Oktian, "TLS channel implementation for ONOS's east/west-bound communication," in *Electronics, Communications and Networks V*, vol. 382 of *Lecture Notes in Electrical Engineering*, pp. 397–403, Springer, Singapore, 2016.

[8] OpenFlow, https://www.opennetworking.org/sdn-resources/technical-library.

[9] "Cisco Application Centric Infrastructure: Use ACI as a Technology-Based Catalyst for IT Transformation White Paper," http://www.cisco.com/c/en/us/solutions/collateral/data-center-virtualization/application-centric-infrastructure/white-paper-c11-734501.html.

[10] K. Benton, L. J. Camp, and C. Small, "OpenFlow vulnerability assessment," in *Proceedings of the 2nd ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking (HotSDN '13)*, pp. 151–152, Hong Kong, August 2013.

[11] Open Daylight, "Cross Project: Open Daylight Security Analysis," https://wiki.opendaylight.org/view/CrossProject:OpenDaylight_Security_Analysis.

[12] HP, *HP VAN SDN Controller Administrator Guide*, revision 2, 1st edition, 2014, http://h20564.www2.hp.com/hpsc/doc/public/display?docId=c04003114.

[13] ONOS, "Secure OpenFlow connection using TLS/SSL," https://jira.onosproject.org/browse/ONOS-1319.

[14] Y. Zaki, "Long Term Evolution (LTE)," in *Future Mobile Communications: LTE Optimization and Mobile Network Virtualization*, vol. 1 of *Advanced Studies Mobile Research Center Bremen*, pp. 13–33, Springer, Wiesbaden, Germany, 2013.

[15] M. Stasiak, M. Głąbowski, A. Wiśniewski, and P. Zwierzykowski, "Universal mobile telecommunication system," in *Modeling and Dimensioning of Mobile Networks: From GSM to LTE*, John Wiley & Sons, Chichester, UK, 2010.

[16] M. D. Aime, G. Calandriello, and A. Lioy, "Dependability in wireless networks: can we rely on WiFi?" *IEEE Security & Privacy*, vol. 5, no. 1, pp. 23–29, 2007.

[17] S. W. Peters and R. W. Heath Jr., "The future of WiMAX: multihop relaying with IEEE 802.16j," *IEEE Communications Magazine*, vol. 47, no. 1, pp. 104–111, 2009.

[18] F. Bari and V. C. M. Leung, "Automated network selection in a heterogeneous wireless network environment," *IEEE Network*, vol. 21, no. 1, pp. 34–40, 2007.

[19] C. Peng, Q. Zhang, and C. Tang, "Improved TLS handshake protocols using identitybased cryptography," in *Proceedings of the 2009 International Symposium on Information Engineering and Electronic Commerce (IEEC '09)*, pp. 135–139, Ternopil, Ukraine, May 2009.

[20] N. McKeown, T. Anderson, H. Balakrishnan et al., "OpenFlow: enabling innovation in campus networks," *ACM SIGCOMM Computer Communication Review*, vol. 38, no. 2, pp. 69–74, 2008.

[21] S. J. Vaughan-Nichols, "OpenFlow: the next generation of the network?" *IEEE Computer*, vol. 44, no. 8, pp. 13–15, 2011.

[22] Standards for M2M and the Internet of Things: oneM2M Release 1 Specifications, http://www.onem2m.org/technical/published-documents.

[23] D. Locke, "MQ Telemetry Transport (MQTT) V3.1 Protocol Specification," August 2010, http://www.ibm.com/developerworks/webservices/library/ws-mqtt/index.html.

[24] Z. Shelby, K. Hartke, and C. Bormann, "The constrained application protocol (CoAP)," RFC 7252, 2014, https://datatracker.ietf.org/doc/draft-ietf-core-coap/.

[25] C. Bormann, A. P. Castellani, and Z. Shelby, "CoAP: an application protocol for billions of tiny internet nodes," *IEEE Internet Computing*, vol. 16, no. 2, pp. 62–67, 2012.

[26] R. T. Fielding and R. N. Taylor, "Principled design of the modern Web architecture," in *Proceedings of the ACM 22nd International Conference on Software Engineering (ICSE '00)*, pp. 407–416, Limerick, Ireland, June 2000.

[27] D. A. Cooper, "A closer look at revocation and key compromise in public key infrastructures," in *Proceedings of the 21st National Information Systems Security Conference*, pp. 555–565, October 1998.

[28] D. Boneh, X. Ding, G. Tsudik, and C. M. Wong, "A method for fast revocation of public key certificates and security capabilities," in *Proceedings of the 10th Conference on USENIX Security Symposium*, vol. 10, pp. 297–308, Berkeley, Calif, USA, 2001.

[29] X. Ding and G. Tsudik, "Simple identity-based cryptography with mediated RSA," in *Topics in Cryptology—CT-RSA 2003: The Cryptographers' Track at the RSA Conference 2003 San Francisco, CA, USA, April 13–17, 2003 Proceedings*, vol. 2612 of *Lecture Notes in Computer Science*, pp. 193–210, Springer, Berlin, Germany, 2003.

[30] C. Yoon, T. Park, S. Lee, H. Kang, S. Shin, and Z. Zhang, "Enabling security functions with SDN: a feasibility study," *Computer Networks*, vol. 85, pp. 19–35, 2015.

[31] A. Shamir, "Identity-based cryptosystems and signature schemes," in *Advances in Cryptology: Proceedings of the CRYPTO '84*, Section I, pp. 47–53, Springer, 1985.

[32] R. Sakai, K. Ohgishi, and M. Kasahara, "Cryptosystems based on pairing," in *Proceedings of the Symposium on Cryptography and Information Security (SCIS '00)*, Okinawa, Japan, January 2000.

[33] D. Boneh and M. Franklin, "Identity-based encryption from the weil pairing," in *Proceedings of the 21st Annual International Cryptology Conference*, Santa Barbara, Calif, USA, August 2001.

[34] N. P. Smart, "Identity-based authenticated key agreement protocol based on Weil pairing," *Electronics Letters*, vol. 38, no. 13, pp. 630–632, 2002.

[35] L. Chen and C. Kudla, "Identity based authenticated key agreemet protocols from pairings," in *Proceedings of the 16th IEEE Computer Security Foundations Workshop*, vol. 2, pp. 219–233, Pacific Grove, Calif, USA, July 2003.

[36] M. A. S. Santos, B. A. A. Nunes, K. Obraczka, T. Turletti, B. T. De Oliveira, and C. B. Margi, "Decentralizing SDN's control plane," in *Proceedings of the 39th Annual IEEE Conference on Local Computer Networks (LCN '14)*, pp. 402–405, Edmonton, Canada, September 2014.

[37] L. Chen, Z. Cheng, and N. P. Smart, "Identity-based key agreement protocols from pairings," *International Journal of Information Security*, vol. 6, no. 4, pp. 213–241, 2007.

[38] S. Chatterjee, D. Hankerson, and A. Menezes, "On the efficiency and security of pairing-based protocols in the type 1 and type 4 settings," in *Arithmetic of Finite Fields: Third International Workshop, WAIFI 2010, Istanbul, Turkey, June 27–30, 2010. Proceedings*, vol. 6087 of *Lecture Notes in Computer Science*, pp. 114–134, Springer, Berlin, Germany, 2010.

[39] N. P. Smart, V. Rijmen, B. Warinschi et al., "Algorithms, Key Sizes and Parameter Report—2013 recommendations," European Union Agency for Network and Information Security (ENISA), version 1.0, October 2013.

[40] J.-H. Lam, S.-G. Lee, H.-J. Lee, and Y. E. Oktian, "Securing distributed SDN with IBC," in *Proceedings of the 7th International Conference on Ubiquitous and Future Networks (ICUFN '15)*, pp. 921–925, Sapporo, Japan, July 2015.

[41] Wireshark, https://www.wireshark.org/.

Hindawi Publishing Corporation
Mobile Information Systems
Volume 2016, Article ID 2368427, 11 pages
http://dx.doi.org/10.1155/2016/2368427

Research Article

Data-Driven Handover Optimization in Next Generation Mobile Communication Networks

Po-Chiang Lin, Lionel F. Gonzalez Casanova, and Bakary K. S. Fatty

Department of Communications Engineering, Yuan Ze University, Taoyuan 320, Taiwan

Correspondence should be addressed to Po-Chiang Lin; pclin@saturn.yzu.edu.tw

Received 1 April 2016; Revised 3 July 2016; Accepted 5 July 2016

Academic Editor: Young-June Choi

Network densification is regarded as one of the important ingredients to increase capacity for next generation mobile communication networks. However, it also leads to mobility problems since users are more likely to hand over to another cell in dense or even ultradense mobile communication networks. Therefore, supporting seamless and robust connectivity through such networks becomes a very important issue. In this paper, we investigate handover (HO) optimization in next generation mobile communication networks. We propose a data-driven handover optimization (DHO) approach, which aims to mitigate mobility problems including too-late HO, too-early HO, HO to wrong cell, ping-pong HO, and unnecessary HO. The key performance indicator (KPI) is defined as the weighted average of the ratios of these mobility problems. The DHO approach collects data from the mobile communication measurement results and provides a model to estimate the relationship between the KPI and features from the collected dataset. Based on the model, the handover parameters, including the handover margin and time-to-trigger, are optimized to minimize the KPI. Simulation results show that the proposed DHO approach could effectively mitigate mobility problems.

1. Introduction

The first generation (1G) mobile communication systems enabled the release from traditional wireline place-to-place communications to wireless person-to-person communications. As technologies and demands evolve, now the fifth generation (5G) mobile communication systems aim to connect anything to anything all over the world.

One of the engineering requirements for 5G is to achieve 1000x data rate [1]. There are three categories of technologies that would be combined together to achieve the 1000x capacity gain:

(1) Network densification and traffic offloading to improve the area spectral efficiency.

(2) Millimeter wave (mmWave) technologies and access to unlicensed spectrum to increase bandwidth.

(3) Massive Multiple-Input and Multiple-Output (massive MIMO), Multiuser MIMO (MU-MIMO), and Coordinated Multipoint (CoMP) to increase spectral efficiency.

Among these categories of technologies, network densification and traffic offloading are expected to provide the majority (40x to 50x) of the required capacity gain. Network densification is a straightforward but effective method to increase the network capacity by making cell size smaller. The advantages of cell shrinking include frequency reuse and reduction of resource competition among users within each cell.

However, network densification leads to mobility problems since users are more likely to hand over to another cell in dense or even ultradense mobile communication networks. Therefore, supporting seamless and robust connectivity through such networks becomes a very important issue. Moreover, as the number of base stations increases, the installation, configuration, and maintenance efforts also increase. Mitigating these efforts to decrease the capital expenditure (CAPEX) and operational expenditure (OPEX) is also a critical issue [2].

In next generation mobile communication networks, optimizing the handover (HO) parameters to improve the system performance is critical. The objective of handover optimization, an important part of the Self-Organizing Network (SON), is to provide fast and seamless handover from one cell to another while simultaneously keeping network management simple [3, 4]. The main goals of handover optimization include minimizing call drops, minimizing radio link failures (RLF), minimizing unnecessary handovers, and minimizing idle mode problems.

In this paper, we investigate the handover optimization problem in next generation mobile communication networks. We propose a data-driven handover optimization (DHO) approach, which aims to mitigate mobility problems. The DHO approach collects data from the mobile communication measurement results and provides a model to estimate the relationship between the key performance indicator (KPI) and features from the collected dataset. Based on the model, the handover parameters, which include the handover margin (HOM) and time-to-trigger (TTT), are optimized to minimize the KPI, which is the weighted average of various mobility problem ratios. Simulation results show that the proposed DHO approach could effectively mitigate mobility problems.

The major contributions of this paper are threefold:

(1) We consider several features, including the location, moving speed, and moving direction of mobile stations, HOM, and TTT. The proposed approach is thus context-aware and adaptive to mobile communication environments.

(2) We classify the mobility problems into five different types and design the mobility problem identification mechanism based on the combination of three conditional tests.

(3) We adopt the neural network model to estimate the KPI function and to optimize the handover parameters HOM and TTT.

The rest of this paper is organized as follows. In Section 2 we present a review of related work. The HO problem formulation is described in Section 3. In Section 4, we provide the detailed description of the proposed DHO approach. Performance evaluation and discussions are given in Section 5. Finally, Section 6 concludes this paper.

2. Related Work

There exist several works for the handover optimization in the long-term evolution (LTE) mobile communication networks. Jansen et al. proposed a self-optimizing algorithm that tunes the handover parameters of an LTE base station in order to diminish negative effects, such as call dropping and handover failures [5]. Their algorithm picks the optimal hysteresis and time-to-trigger combination for the current network status. The authors also proposed a weighted performance based handover optimization algorithm (WPHPO) that tunes the hysteresis and time-to-trigger in iterative steps [6]. Li et al. proposed a dynamic hysteresis-adjusting (DHA) method, which uses the network-allowed maximum RLF ratio as a key indicator [7]. Lobinger et al. investigated the coordination of handover parameter optimization and load balance in LTE SON [8]. Capdevielle et al. investigated the joint interference management and handover optimization in LTE small cells networks [9]. They focus on two challenges for LTE small cell deployment, intercell interference management, and mobility management.

Several works focus on the heterogeneous/inter-RAT handover. Ali and Saquib investigated cellular/wireless LAN handover analysis for different mobility models in order to exploit the heterogeneity of the wireless environment [10]. They provided a medium selection method for vertical handover. Awada et al. investigated the cell-pair specific optimization of the inter-RAT handover parameters in SON [11]. They provided the configuration paradigms for the inter-RAT handover thresholds. Giacomini and Agarwal use QoS reputation and GM$(1,1)$ prediction to make decision on vertical handover [12]. They built on a novel reputation based vertical handover decision rating system in the handover decision making progress. López-Pérez et al. characterized the relation between handover failure and ping-pong rates in a heterogeneous network scenario [13]. Rath and Panwar proposed a new prefetch-based fast handover procedure that is designed to overcome the higher latency caused by the use of the public internet to connect the femtocell base station with the mobile core network [14]. Xenakis et al. proposed a novel handover decision policy for the two-tier LTE network with the goal to reduce the power consumption at mobile stations [15]. The proposed method can adapt to the handover hysteresis margin with respect to the prescribed SINR target and measurement results.

Some works introduced new concepts and architectures to enhance user mobility in mobile communication networks. Taleb et al. investigated the support of highly mobile users [16]. They introduced a data anchor gateway relocation method based on the information, including user mobility, history information, and user activity patterns, and proposed a handover management policy to select a target base station. Imran et al. investigated how to empower SON with big data for enabling 5G in [17]. They first characterized big data in next generation mobile communication networks and then provided challenges in SON for enabling 5G, including underutilized intelligence, need for self-coordination, need for more transparent SON, scalability, energy efficiency, and need for a paradigm shift from reactive to proactive SON. The authors also proposed a framework for big data empowered SON (BSON) which takes on the challenges mentioned above. In this paper we leverage from their basic idea and investigate the HO problem which is one of the most important issues in SON.

3. Problem Formulation

We formulate the HO problem as follows. Suppose that $\mathbf{X} = [\mathbf{X}_D, \mathbf{X}_M]$ is a feature vector which includes two types of variables, $\mathbf{X}_D$ and $\mathbf{X}_M$. To be specific, $\mathbf{X}_M$ denotes the measurement result, which includes the location of MS, moving speed of MS, moving direction of MS, and other useful counters and flags described in Section 4.1. $\mathbf{X}_D$ denotes the decision variables of the HO problem, which is the handover parameter vector with two elements, HOM and TTT.

The details of these two handover parameters are described as follows.

(i) Handover Margin (HOM) (Also Known as Hysteresis). HOM is defined as the threshold of the difference in Signal to Interference plus Noise Ratio (SINR) between the serving and the target cells. When the SINR from the target cell is HOM better than that from the serving cell, the HO condition is satisfied.

(ii) Time-to-Trigger (TTT). An HO request would not be sent immediately when the corresponding HO condition mentioned above is satisfied. Instead, the HO condition has to be fulfilled for a certain period denoted as TTT. When the HO condition keeps satisfied for TTT, an HO request is triggered.

The proper values of HOM and TTT are critical to seamless connectivity. Large HOM and TTT values lead to more stable behavior, but they may delay the HO decisions unnecessarily which may cause problems. On the other hand, small HOM and TTT values avoid long delay to trigger HO request, but they may cause ping-pong HO and unnecessary HO.

The HO problem is formulated as

$$\mathbf{X}_{D,\mathrm{opt}} = \arg\min_{\mathbf{X}_D} Y, \tag{1}$$

where Y is the corresponding key performance indicator (KPI), which depends on the feature vector $\mathbf{X} = [\mathbf{X}_D, \mathbf{X}_M]$. The KPI should be designed to measure the mitigation of mobility problems.

There are four main challenges in the HO problem:

(1) There are various kinds of mobility problems in next generation mobile communication networks [18]. Identifying the mobility problems would be the first priority in the HO problem.

(2) The design of the KPI needs comprehensive and careful considerations for these mobility problems.

(3) The KPI Y can be formulated as a function of $\mathbf{X}$; that is, $Y = f(\mathbf{X})$. However, the exact form of $f(\mathbf{X})$ is unknown. Therefore, one of the challenges we take on in this work is to provide a good estimation $\widehat{f}(\mathbf{X})$ to the function $f(\mathbf{X})$. With a good estimate $\widehat{f}(\mathbf{X})$ we can make predictions of the KPI Y at some points $\mathbf{X} = \mathbf{x}$, where $\mathbf{x}$ is a vector of some specific values of $\mathbf{X}$.

(4) The decision variables $\mathbf{X}_D$ consist of two elements, HOM and TTT. Mitigating the mobility problems by joint-optimizing HOM and TTT would be another challenge to the HO problem.

In Section 4 we describe the proposed DHO approach to deal with these main challenges.

4. Proposed DHO Approach

In this section, we provide the detailed description of the proposed data-driven handover optimization (DHO) approach. Figure 1 shows the system architecture of the proposed DHO approach. First, measurement results are collected from the mobile communication network. The dataset collected in the database includes the feature vector described in Section 3 and various counters and flags that are used to identify the mobility problems. The details of these counters and flags are provided in Section 4.1. The DHO approach then analyzes and processes the data to calculate the mobility problem ratio (MPR) of each mobility problem and obtain the KPI, which is saved in the database denoted as “right data” in Figure 1. The KPI function estimation engine uses the processed “right data” to provide the KPI function estimate. After that, the HO parameter optimization engine uses the KPI function estimate to optimize the HO parameters including HOM and TTT. Finally, the optimal values of HOM and TTT are applied to corresponding base stations.

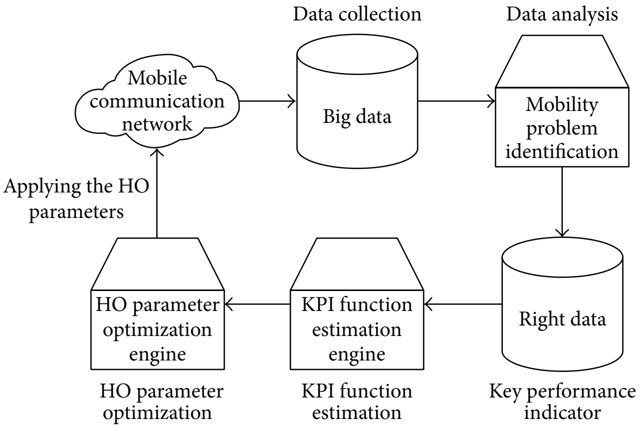

Data collection
Data analysis
Mobile communication network
Big data
Mobility problem identification
Applying the HO parameters
HO parameter optimization engine
KPI function estimation engine
Right data
HO parameter optimization
KPI function estimation
Key performance indicator

FIGURE 1: System architecture of the proposed DHO approach.

In the following paragraphs, we provide the detailed description of each building block in Figure 1.

4.1. Mobility Problem Identification. There would be five different mobility problems (MPs) in next generation mobile communication networks.

(1) Too-Late HO. Too-late HO trigger timing leads to low SINR level of the serving cell. A radio link failure (RLF) between the mobile station (MS) and the serving bast station (BS) occurs, and then the MS try to reestablish a connection to a nearby BS.

(2) Too-Early HO. Too-early HO trigger timing leads to a low SINR level of the target cell. An RLF between the MS and the target BS occurs, and then the MS try to reestablish a connection to the original serving BS.

(3) HO to Wrong Cell. HO from a source cell A to a target cell B which provides unstable SINR leads to an RLF. At the same time, if there is another suitable cell C for the MS to reconnect to after the RLF, the MS reestablishes connection with cell C that is neither the serving cell A nor the target cell B. In this situation, instead of the failed HO from cell A to cell B, HO from cell A directly to cell C would be a better decision.

Table 1: Mobility problem identification.

Mobility problem	Close to last HO?	RLF status?	New target = previous serving?
Too-late HO	N	Y	Don't care
Too-early HO	Y	Y	Y
HO to wrong cell	Y	Y	N
Ping-pong HO	Y	N	Y
Unnecessary HO	Y	N	N

(4) Ping-Pong HO. Cell A hands over a MS to cell B, and cell B hands over the same MS back to cell A shortly after. Since data transmission is temporarily blocked during the connection transferring to the target cell, these two HOs, despite both being successful, should be avoided.

(5) Unnecessary HO. Cell A hands over a MS to cell B and cell B hands over the same MS to another cell C shortly after. Those two HOs, despite both being successful, can be combined into one HO from cell A directly to cell C in order to avoid unnecessary blocking of data transmission during the connection transferring time.

In order to identify the mobility problems mentioned above, we design the mobility problem identification criteria, as Table 1 shows. The mobility problem identification criteria depend on three conditional tests:

(1) Close to last HO?

Whether the occurrence time of the current HO event is close to that of the previous HO event?

(2) RLF status?

Is there an RLF for the current connection?

(3) New target = previous serving?

Whether the new target cell is the same as the previous serving one?

Table 1 shows that each of the five mobility problems would be identified based on the test results of the three conditional tests mentioned above.

In the proposed DHO approach, we design the following counters and flags to identify the mobility problems:

(i) lastHoCounter: to indicate the time interval since last HO.

(ii) lastHoThreshold: to define a threshold value, combined with lastHoCounter to identify whether the occurrence time of the current HO event is close to that of the previous HO event.

(iii) rlfStatus: to indicate whether an RLF occurs for the current connection.

(iv) newTagertCellId: to indicate the new target cell ID.

(v) previousServingCellId: to indicate the previous serving cell ID, combined with the newTagertCellId to indicate whether the new target cell is the same as the previous serving one.

4.2. Design of KPI. In the HO problem, the KPI Y is defined as the weighted average of the various mobility problem ratios (MPRs). There are five different kinds of mobility problems. The KPI Y is calculated as

$$Y = \frac{\sum_{k=1}^{5} w_k R_k(\mathbf{X})}{\sum_{k=1}^{5} w_k}, \qquad (2)$$

where w_k is the weight of the mobility problem k, which is determined by mobile network operators, and $R_k(\mathbf{X})$ is the MPR of the mobility problem k, which is a function of $\mathbf{X}$. MPR R_k is defined as the probability of the event where the mobility problem (MP) k happens. Consider

$$R_k(\mathbf{X}) = \Pr(\mathrm{MP}_k) = \frac{N_k}{N_{\text{total}}}, \qquad (3)$$

where N_k denotes the number of MP_k, and N_{total} denotes the total number of HO. Each BS uses a sliding time window to collect the statistics of N_k and N_{total} in order to obtain each MPR and the KPI Y.

4.3. KPI Function Estimation. In order to solve the HO problem, as (1) shows, some previous solutions focus on analyzing and obtaining the KPI function $Y = f(\mathbf{X})$. When the analytical formula $f(\mathbf{X})$ is known, solving the equation $\partial f/\partial \mathbf{X}_D = 0$ would get the optimal handover parameters HOM and TTT. However, note that the function f consists of complex joint effects of the feature vector $\mathbf{X} = [\mathbf{X}_D, \mathbf{X}_M]$. Under these complex effects, it is difficult to analyze and obtain an accurate KPI function.

In this paper we use machine learning techniques to estimate the KPI function. To be specific, we use the multilayer perception (MLP), which is one of the most popular neural networks, to perform the KPI function estimation [19].

The MLP consists of an input layer, one or more hidden layers, and an output layer. In this paper we adopt the MLP with one hidden layer consisting of four hidden neurons. The feature vector $\mathbf{X}$ is applied to the input layer and propagates through the network layer by layer from left to right. Each arrow in this figure represents an adjustable synaptic weight. From our comprehensive experiments, this architecture is empirically sufficient to model the relationship between KPI and the feature vector, whereas the other experiments with more hidden neurons provide similar results. A simulation result about the mean-square-error (MSE) of the KPI function estimation with various numbers of hidden neurons is shown in Figure 2. The epoch limitation is set as 1000. This figure shows that four hidden neurons are sufficient to improve the performance of modeling to the degree of $1E{-}10$, whereas the performance of two neurons only reaches the degree of $1E - 6$. Applying more hidden neurons does not significantly improve the MSE.

In MLP, the backpropagation (BP) algorithm is used to adjust the neural network parameters in order to minimize the error between the target (measured) KPI and the response from the function estimate $\widehat{f}$. The detailed operation of the BP algorithm includes the forward pass and the backward pass. We provide the detailed descriptions of the forward pass and the backward pass in the following paragraphs.

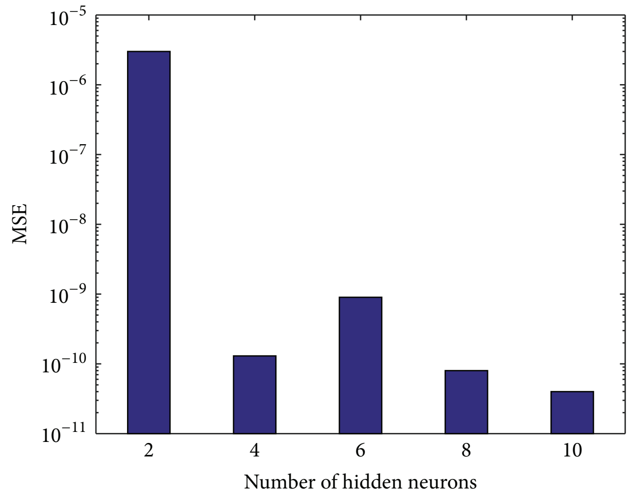


Figure 2: MSE of the KPI function estimation with various number of hidden neurons.

4.3.1. Forward Pass. The purpose of the forward pass is to get the actual response of the MLP in accordance with the specified input signal. In the forward pass the feature vector $\mathbf{X}(n)$ at iteration n is applied to the input layer and propagates through the network layer by layer from left to right. The top part of Figure 3 shows the signal flow of the forward pass. $X_1(n)$ denotes the HOM, and $X_2(n)$ denotes the TTT. $X_3(n)$ to $X_m(n)$ denote the other features. When the signal propagates through a synapse, its value is multiplied by the synaptic weight. The input of a neuron is the synaptic weighted sum of the output of its previous layer. For example, the input of the jth neuron in the lth layer is

$$v_j^{(l)}(n) = \sum_{i=0}^{m_{l-1}} w_{ji}^{(l)}(n)\, y_i^{(l-1)}(n), \qquad (4)$$

where $i = 0$ represents the bias term. The input of this neuron is applied to the activation function $\varphi(\cdot)$. The activation function should be differentiable everywhere. We adopt the commonly used sigmoid function as

$$y_j^{(l)}(n) = \varphi\left(v_j^{(l)}(n)\right) = \frac{1}{1 + \exp\left(-v_j^{(l)}(n)\right)}. \qquad (5)$$

The output of this neuron propagates through the network to the next layer, and the same operation is performed again. Finally the signal aggregates in the output layer, and the estimated KPI function $\widetilde{Y}(n)$ is obtained. During the forward pass the synaptic weights are all fixed.

4.3.2. Backward Pass. The purpose of the backward pass is to adjust the synaptic weights in order to minimize the error between the actual response $\widetilde{Y}(n)$ and the desired response $Y(n)$. The error signal $e(n)$ is defined by

$$e(n) = Y(n) - \widetilde{Y}(n). \qquad (6)$$

The instantaneous square error $E(n)$ is defined as

$$E(n) = \frac{1}{2} e^2(n) = \frac{1}{2}\left(Y(n) - \widetilde{Y}(n)\right)^2. \qquad (7)$$

The goal of the back propagation algorithm is to minimize $E(n)$. To achieve this goal, the back propagation algorithm applies a respective correction $\Delta\omega_{ji}^{(l)}(n)$ to every synaptic weight $\omega_{ji}^{(l)}(n)$. The correction $\Delta\omega_{ji}^{(l)}(n)$ is designed to be proportional to the gradient $\partial E(n)/\partial\omega_{ji}^{(l)}(n)$; that is,

$$\Delta\omega_{ji}^{(l)}(n) = -\eta_a \frac{\partial E(n)}{\partial\omega_{ji}^{(l)}(n)}. \qquad (8)$$

The minus sign comes from the fact that lower $E(n)$ is preferred. η_a is the adaptive learning rate proposed by Behera et al. in [20], which is expressed as

$$\eta_a = \mu \frac{\|\tilde{y}\|^2}{\|\mathbf{J}^T \tilde{y}\|^2}, \qquad (9)$$

where $\tilde{y} = Y(n) - \widetilde{Y}(n)$ and $\mathbf{J} = \partial\widetilde{Y}/\partial\mathbf{W}_l$. While considering the weight correction for $l = 2$, that is, the output layer, according to the chain rule of calculus, the gradient $\partial E(n)/\partial\omega_{1i}^{(2)}(n)$ can be written as

$$\frac{\partial E(n)}{\partial\omega_{1i}^{(2)}(n)} = \frac{\partial E(n)}{\partial e(n)} \frac{\partial e(n)}{\partial\widetilde{Y}(n)} \frac{\partial\widetilde{Y}(n)}{\partial v_1^{(2)}(n)} \frac{\partial v_1^{(2)}(n)}{\partial\omega_{1i}^{(2)}(n)}. \qquad (10)$$

Applying (4)–(7) to the four partial derivatives at the right of the equal sign in (10), it can be rewritten as

$$\begin{aligned} \frac{\partial E(n)}{\partial\omega_{1i}^{(2)}(n)} &= -e(n)\,\varphi'\left(v_1^{(2)}(n)\right) y_i^{(1)}(n) \\ &= -\delta_1^{(2)}(n)\, y_i^{(1)}(n), \end{aligned} \qquad (11)$$

where $\delta_j^{(l)}(n)$ is the local gradient which is defined as

$$\delta_j^{(l)}(n) = -\frac{\partial E(n)}{\partial v_j^{(l)}(n)}. \qquad (12)$$

While considering the weight correction for $l = 1$, that is, the hidden layer, by iterative backpropagation and chain rule manners, the local gradient $\delta_j^{(1)}(n)$ is

$$\delta_j^{(1)}(n) = \varphi'\left(v_j^{(1)}(n)\right) \delta_1^{(2)}(n)\, \omega_{1j}^{(2)}(n). \qquad (13)$$

The derivation of the local gradients at each layer is shown in the bottom part of Figure 3. Finally, every synaptic weight is corrected following this formula:

$$\omega_{ji}^{(l)}(n+1) = \omega_{ji}^{(l)}(n) + \Delta\omega_{ji}^{(l)}(n). \qquad (14)$$

The forward pass and the backward pass are executed iteratively until the square error $E(n)$ reaches some satisfactory level or when the number of epoch exceeds some threshold. After that, the handover parameter optimization is executed.

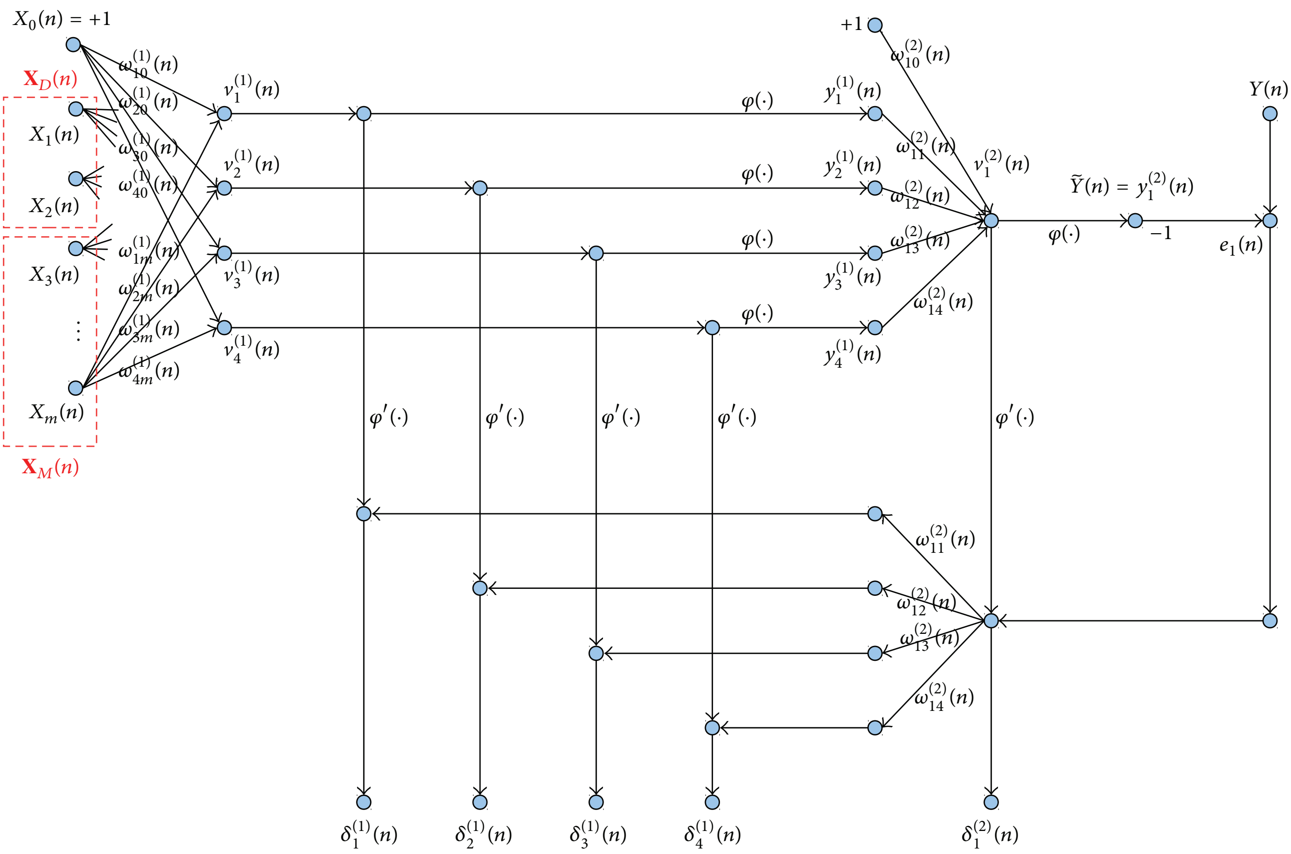


FIGURE 3: The forward pass and the backward pass of the BP algorithm.

4.4. Handover Parameter Optimization. In this paper, the handover optimization problem is solved by the gradient descent algorithm [21]. Here the handover parameters $\mathbf{X}_D$, including HOM and TTT, are adjusted adaptively. To find the local minimum of the KPI function $Y \approx \widehat{f}(\mathbf{X})$ using gradient descent, one takes steps proportional to the gradient. Suppose that at the nth time of adjustment, the handover parameters are $\mathbf{X}_D(n)$ and the KPI is $Y(n)$. At the next time of adjustment, the handover parameters $\mathbf{X}_D$ are set as

$$\mathbf{X}_D(n+1) = \mathbf{X}_D(n) + \Delta\mathbf{X}_D(n). \tag{15}$$

$\Delta\mathbf{X}_D(n)$ depends on the gradient of estimated KPI $Y(n)$ with respect to $\mathbf{X}_D(n)$; that is,

$$\Delta\mathbf{X}_D(n) = \mu\frac{\partial Y(n)}{\partial \mathbf{X}_D(n)}, \tag{16}$$

where μ is the constant adjustment rate.

How to obtain the gradient $\partial Y/\partial\mathbf{X}_D$ is a problem left. The original BP algorithm described in Section 4.3 only consists of two passes: the forward pass and the backward pass. In order to obtain the gradient $\partial Y/\partial\mathbf{X}_D$, the proposed DHO approach modifies the original BP algorithm to add the third pass: the adjustment pass. We provide the detailed description of the adjustment pass in the following paragraphs.

The purpose of the adjustment pass is to obtain the estimated gradient $\partial\widetilde{Y}(n)/\partial\mathbf{X}_D(n)$ which is the key to adjust the handover parameter in order to minimize the KPI. The synaptic weights that have been adjusted well in the backward pass in accordance with (14) are set as fixed in the adjustment pass. In the following we show how to obtain the estimated gradient $\partial\widetilde{Y}(n)/\partial\mathbf{X}_D(n)$. The signal flow of the adjustment pass is depicted in Figure 4.

First of all, we define the local gradient $\lambda_j^{(l)}(n)$ of the adjustment pass as

$$\lambda_j^{(l)}(n) = \frac{\partial\widetilde{Y}(n)}{\partial v_j^{(l)}(n)}. \tag{17}$$

This local gradient $\lambda_j^{(l)}(n)$ is similar to the local gradient $\delta_j^{(l)}(n)$ of the backward pass which is defined in (12), except that the target of the partial derivative is changed to $\widetilde{Y}(n)$. While considering the output layer, the local gradient $\lambda_1^{(2)}(n)$ is

$$\lambda_1^{(2)}(n) = \frac{\partial\widetilde{Y}(n)}{\partial v_1^{(2)}(n)} = \varphi'\left(v_1^{(2)}(n)\right). \tag{18}$$

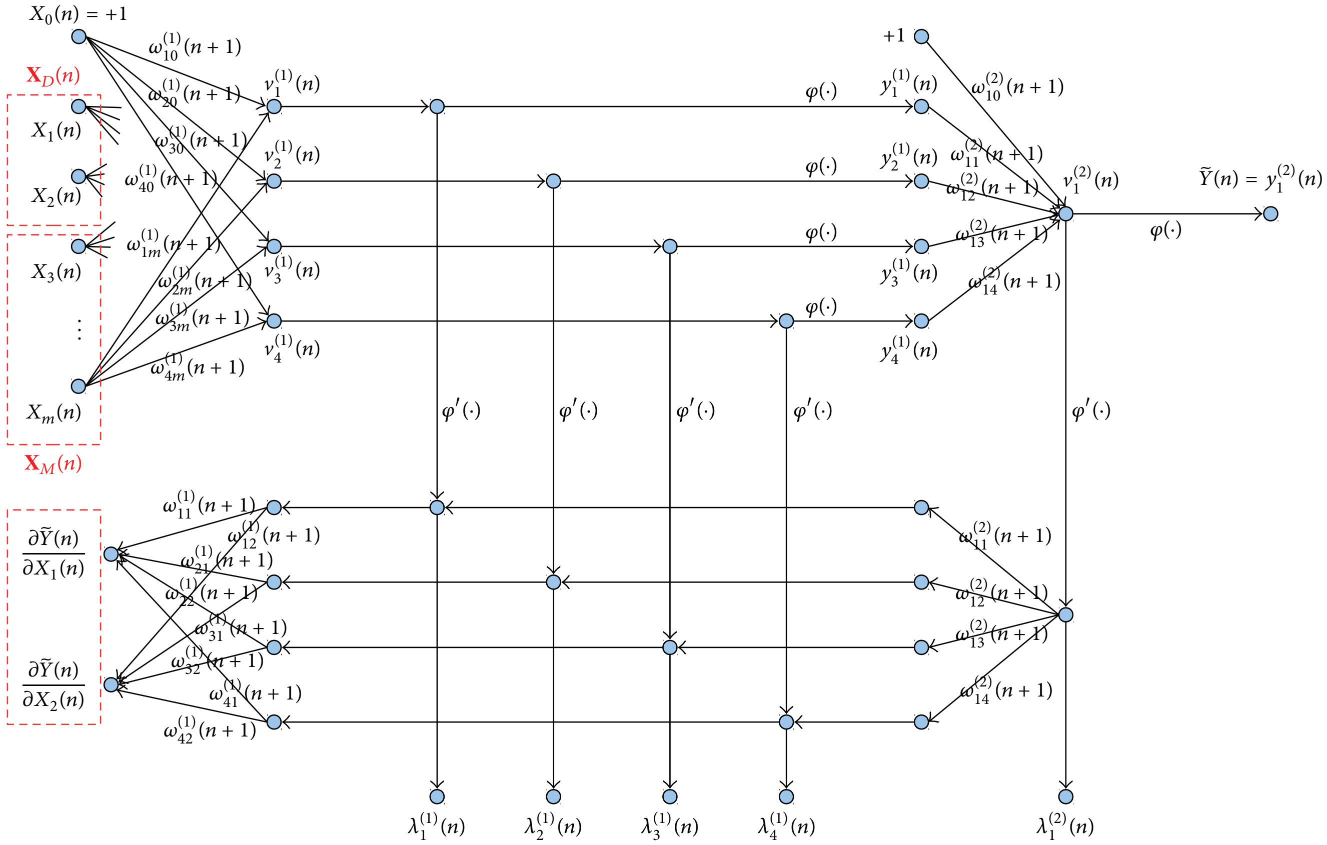


FIGURE 4: Adjustment pass of the proposed DHO approach.

While considering the hidden layer, the local gradient $\lambda_j^{(1)}(n)$ is

$$\lambda_j^{(1)}(n) = \frac{\partial \tilde{Y}(n)}{\partial v_j^{(1)}(n)} = \frac{\partial \tilde{Y}(n)}{\partial v_1^{(2)}(n)} \frac{\partial v_1^{(2)}(n)}{\partial y_j^{(1)}(n)} \frac{\partial y_j^{(1)}(n)}{\partial v_j^{(1)}(n)}$$
$$= \lambda_1^{(2)}(n)\,\omega_{1j}^{(2)}(n+1)\,\varphi'\left(v_j^{(1)}(n)\right). \tag{19}$$

Using the results from (18) and (19), the gradient can be written as follows:

$$\frac{\partial \tilde{Y}(n)}{\partial X_i(n)} = \frac{\partial \tilde{Y}(n)}{\partial v_1^{(2)}(n)} \frac{\partial v_1^{(2)}(n)}{\partial X_i(n)} = \lambda_1^{(2)}(n) \frac{\partial v_1^{(2)}(n)}{\partial X_i(n)}, \tag{20}$$

where the last equality comes from (18). Since

$$v_1^{(2)}(n) = \sum_{j=0}^{4} \omega_{1j}^{(2)}(n+1)\, y_j^{(1)}(n), \tag{21}$$

the second term at the most right of (20) can be written as

$$\begin{aligned}\frac{\partial v_1^{(2)}(n)}{\partial X_i(n)} &= \sum_{j=0}^{4} \omega_{1j}^{(2)}(n+1) \frac{\partial y_j^{(1)}(n)}{\partial X_i(n)} \\ &= \sum_{j=0}^{4} \omega_{1j}^{(2)}(n+1) \frac{\partial y_j^{(1)}(n)}{\partial v_j^{(1)}(n)} \frac{\partial v_j^{(1)}(n)}{\partial X_i(n)} \\ &= \sum_{j=1}^{4} \omega_{1j}^{(2)}(n+1) \frac{\partial y_j^{(1)}(n)}{\partial v_j^{(1)}(\mathrm{n})} \frac{\partial v_j^{(1)}(n)}{\partial X_i(n)} \\ &= \sum_{j=1}^{4} \omega_{1j}^{(2)}(n+1)\, \varphi'\left(v_j^{(1)}(n)\right) \omega_{j1}^{(1)}(n+1),\end{aligned} \tag{22}$$

where the third equality comes from the fact that $y_0^{(1)}(n)$ is always equal to 1 and its partial derivative can be ignored from the summation. Combining (19) and (22) into (20), the gradient $\partial \tilde{Y}(n)/\partial X_i(n)$ is obtained as

$$\frac{\partial \tilde{Y}(n)}{\partial X_i(n)} = \sum_{j=1}^{4} \lambda_j^{(1)}(n)\, \omega_{j1}^{(1)}(n+1). \tag{23}$$

The derivation of the local gradients at each layer and the gradient $\partial \tilde{Y}(n)/\partial X_i(n)$ is depicted in Figure 4. Based on the

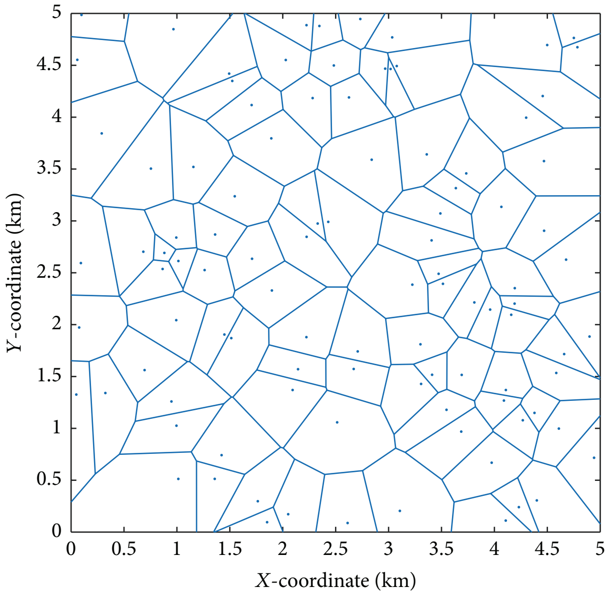


Figure 5: Topology of the simulated mobile communication network.

result from (23), the handover parameter X_i is adjusted as (15) and (16) show.

In order to cope with the time-varying channel conditions, we use online training and operate the three passes iteratively to model the instantaneous relationship $Y = f(\mathbf{X})$ and optimize the handover parameters, HOM and TTT.

5. Performance Evaluation and Discussion

In this section we describe the simulation environment, simulation results, and discussions.

In our simulation, a mobile communication network which consists of 100 small cells is built in a 5 km × 5 km area. Figure 5 shows the coordinates of these small cell base stations. Each cell has 5 MHz bandwidth with 25 resource blocks and 2 GHz carrier frequency. Each resource block consists of 12 subcarriers of size 15 kHz each. A time slot is 0.5 ms in duration and the transmission time interval (TTI) is 1 ms.

In the simulation, 100 MSs are uniformly distributed over the area with random initialized positions. The mobility trace of these 100 MSs is generated by using the Manhattan mobility model, which is widely used for urban areas and suitable to implement realistic simulations [22–24]. In the Manhattan mobility model, MSs move in either horizontal or vertical directions on an urban map. At each intersection, an MS employs a probabilistic approach to choose the direction of its movements. The probabilities of going straight, turning left, and turning right are 0.5, 0.25, and 0.25, respectively. The blue line in Figure 5 shows an illustrative moving path of an MS in the mobile communication network. While considering the KPI, the weights of mobility problems are set to be equal.

An RLF will occur if the SINR level of the serving cell drops below −6.5 dB before handover execution is completed [18]. We set this as our SINR threshold value such that if the SINR from the source BS is lower than the threshold, an RLF will occur. The simulation time is set as 5000 seconds. The operating point with 0 dB HOM and 0 s TTT is the starting operating point for all cells in the network.

In this paper we use LTE-Sim, an open source framework to simulate LTE networks, as our basic simulation tool [25]. The design and development of LTE-Sim support the following aspects:

(i) The Evolved Universal Terrestrial Radio Access (E-UTRAN) and the Evolved Packet System (EPS).

(ii) Single and heterogeneous multicell environments.

(iii) Multiusers environment.

(iv) User mobility.

(v) Handover procedures.

(vi) Frequency reuse techniques.

(vii) Quality-of-service (QoS) management.

(viii) Scheduling strategies.

However, LTE-Sim still lacks some important functionalities and features that are required in this work. Therefore, we modified LTE-Sim from several aspects. Our modification to LTE-Sim includes the following:

(1) Add:

(i) radio link failure,

(ii) handover failure,

(iii) HOM and TTT,

(iv) proposed DHO approach.

(2) Modify:

(i) mobility pattern: using the seed to generate the same sequence of random numbers each time we run the simulation.

The simulation parameter values are shown in Table 2 unless otherwise specified.

Figure 6 shows the simulation result when the TX power is set as 20 dBm. The horizontal axis denotes the mobility problem index. Indexes 1 to 5 mean the mobility problem of too-late HO, too-early HO, HO to wrong cell, ping-pong HO, and unnecessary HO, respectively. Index 6 means the KPI, which is the weighted average of the five MPRs. The vertical axis means the corresponding values of MPR or KPI. The blue bars show the performance for a static HO parameter setting with 0 dB HOM and 0 s TTT and the green ones for another static setting with 10 dB HOM and 5.12 s TTT. The yellow bars show the performance of the proposed DHO approach. The results shown in Figure 6 validate that the proposed DHO approach can effectively mitigate the mobility problems and achieve better MPRs and KPI.

Figure 7 shows the simulation result when the TX power is set as 43 dBm. Since the TX power is higher than the one in the previous scenario in Figure 6, it is less likely that an

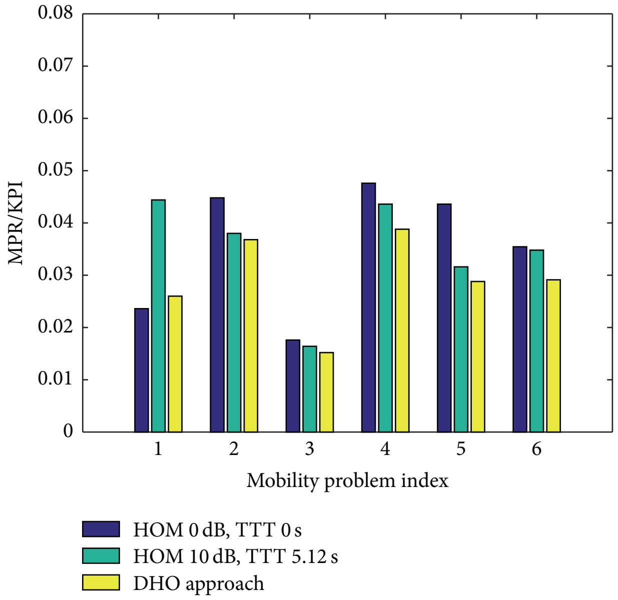


Figure 6: Simulation result when the TX power is set as 20 dBm.

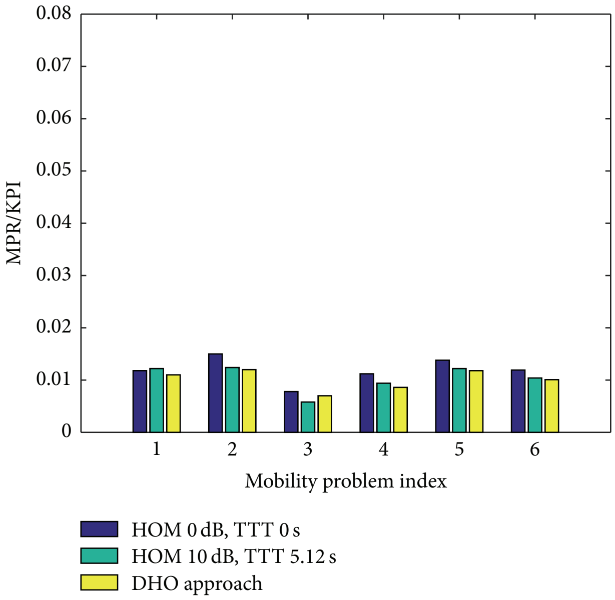


Figure 7: Simulation result when the TX power is set as 43 dBm.

Table 2: Simulation parameters.

Parameter	Values
Number of small cell base stations	100
Number of mobile stations	100
eNB Tx power	15 dBm/20 dBm/43 dBm
Carrier frequency	2 GHz
MS speed	30 km/h
Mobility model	Manhattan
TTI	1 ms
Subcarrier spacing	15 KHz
Resource block	180 KHz
Super frame time	10 ms
Noise figure	2.5
Noise spectral density	−174 dBm
SINR threshold	−6.5 dB
Simulation time	5000 s
Handover delay	30 ms
System bandwidth	5 MHz 25 RBs/TTI

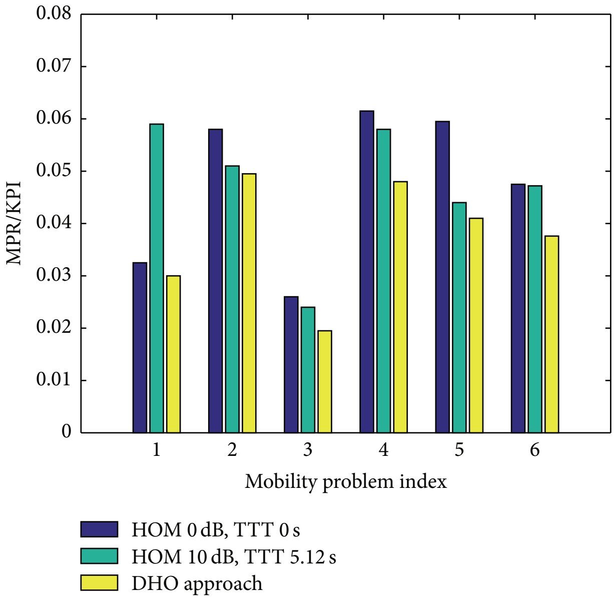


Figure 8: Simulation result when the TX power is set as 15 dBm.

RLF occurs. Therefore, the MPRs and KPI in Figure 7 are lower than those in Figure 6. Moreover, the proposed DHO approach can still effectively mitigate the mobility problems and provide better connectivity to MSs.

Figure 8 shows the simulation result when the TX power is set as 15 dBm. Since the TX power is lower than the one in the previous scenario in Figure 6, it is more likely that an RLF occurs. Therefore, the MPRs and KPI in Figure 8 are higher than those in Figure 6. Moreover, the proposed DHO approach can still effectively mitigate the mobility problems and provide better connectivity to MSs.

We compare the KPI improvement under the various values of TX power. The KPI improvement is calculated as the difference between the initial-case KPI (HOM 0 dB, TTT 0 s) and the KPI of the proposed DHO method, divided by the initial-case KPI. Figure 9 shows that the KPI improvement ranges from 15% to 20%. It also shows that the KPI improvement is more significant when the TX power is lower; that is, the transmission range of the base station is smaller.

6. Conclusion

In this work, we investigate the mobility problems in next generation mobile communication networks. There are totally five different mobility problems, each with different

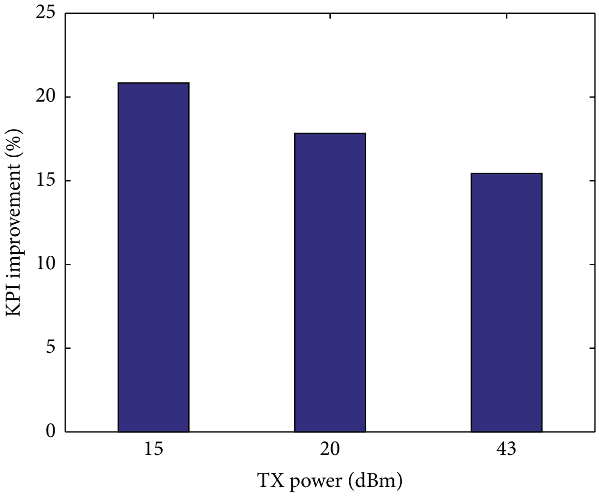


Figure 9: KPI improvement with various TX power.

characteristics. We propose a data-driven handover optimization (DHO) approach, which aims to mitigate mobility problems. The DHO approach consists of four parts:

(1) Mobility problem identification.

(2) KPI design.

(3) KPI function estimation.

(4) Handover parameter optimization.

The DHO approach collects data from the mobile communication measurement results and provides a model to estimate the relationship between the KPI and features from the collected dataset. Based on the model, the handover parameters, including the HOM and TTT, are optimized to minimize the KPI, which is a weighted average of different mobility problem ratios. Simulation results show that the proposed DHO approach could effectively mitigate mobility problems and provide better connectivity.

Competing Interests

The authors declare that they have no competing interests.

Acknowledgments

This work was supported in part by Ministry of Science and Technology (MOST), Taipei, Taiwan, under Grants nos. MOST 103-2221-E-155-005, MOST 104-2221-E-155-017, and MOST 105-2218-E-155-001.

References

[1] J. G. Andrews, S. Buzzi, W. Choi et al., “What will 5G be?” *IEEE Journal on Selected Areas in Communications*, vol. 32, no. 6, pp. 1065–1082, 2014.

[2] K. Kitagawa, T. Komine, T. Yamamoto, and S. Konishi, “A handover optimization algorithm with mobility robustness for LTE systems,” in *Proceedings of the IEEE 22nd International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC ’11)*, pp. 1647–1651, Toronto, Canada, September 2011.

[3] 3GPP, “Self-configuring and self-optimizing network (SON) use cases and solutions,” TR 36.902, V9.3.1, 2011.

[4] A. Awada, B. Wegmann, I. Viering, and A. Klein, “A SON-based algorithm for the optimization of inter-RAT handover parameters,” *IEEE Transactions on Vehicular Technology*, vol. 62, no. 5, pp. 1906–1923, 2013.

[5] T. Jansen, I. Balan, J. Turk, I. Moerman, and T. Kürner, “Handover parameter optimization in LTE self-organizing networks,” in *Proceedings of the IEEE 72nd Vehicular Technology Conference Fall (VTC 2010-Fall ’10)*, pp. 1–5, IEEE, Ottawa, Canada, September 2010.

[6] T. Jansen, I. Balan, S. Stefanski, I. Moerman, and T. Kürner, “Weighted performance based handover parameter optimization in LTE,” in *Proceedings of the IEEE 73rd Vehicular Technology Conference (VTC ’11)*, pp. 1–5, May 2011.

[7] W. Li, X. Duan, S. Jia, L. Zhang, Y. Liu, and J. Lin, “A dynamic hysteresis-adjusting algorithm in LTE self-organization networks,” in *Proceedings of the IEEE 75th Vehicular Technology Conference (VTC Spring ’12)*, pp. 1–5, Yokohama, Japan, May 2012.

[8] A. Lobinger, S. Stefanski, T. Jansen, and I. Balan, “Coordinating handover parameter optimization and load balancing in LTE self-optimizing networks,” in *Proceedings of the IEEE 73rd Vehicular Technology Conference (VTC ’11)*, pp. 1–5, May 2011.

[9] V. Capdevielle, A. Feki, and E. Sorsy, “Joint interference management and handover optimization in LTE small cells network,” in *Proceedings of the IEEE International Conference on Communications (ICC ’12)*, pp. 6769–6773, IEEE, Ottawa, Canada, June 2012.

[10] T. Ali and M. Saquib, “WLAN/cellular handover analysis for different mobility models,” in *Proceedings of the IEEE International Conference on Communications (ICC ’12)*, pp. 2758–2762, Ottawa, Canada, June 2012.

[11] A. Awada, B. Wegmann, I. Viering, and A. Klein, “Cell-pair specific optimization of the inter-RAT handover parameters in SON,” in *Proceedings of the IEEE 23rd International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC ’12)*, pp. 1168–1173, Sydney, Australia, September 2012.

[12] D. Giacomini and A. Agarwal, “Vertical handover decision making using QoS reputation and GM(1,1) prediction,” in *Proceedings of the IEEE International Conference on Communications (ICC ’12)*, pp. 5655–5659, IEEE, Ottawa, Canada, June 2012.

[13] D. López-Pérez, I. Guvenc, and X. Chu, “Theoretical analysis of handover failure and ping-pong rates for heterogeneous networks,” in *Proceedings of the IEEE International Conference on Communications (ICC ’12)*, pp. 6774–6779, IEEE, Ottawa, Canada, June 2012.

[14] A. Rath and S. Panwar, “Fast handover in cellular networks with femtocells,” in *Proceedings of the International Conference on Communications (ICC ’12)*, pp. 2752–2757, IEEE, Ottawa, Canada, June 2012.

[15] D. Xenakis, N. Passas, and C. Verikoukis, “A novel handover decision policy for reducing power transmissions in the two-tier LTE network,” in *Proceedings of the IEEE International Conference on Communications (ICC ’12)*, pp. 1352–1356, Ottawa, Canada, June 2012.

[16] T. Taleb, K. Samdanis, and A. Ksentini, “Supporting highly mobile users in cost-effective decentralized mobile operator

networks," *IEEE Transactions on Vehicular Technology*, vol. 63, no. 7, pp. 3381–3396, 2014.

[17] A. Imran, A. Zoha, and A. Abu-Dayya, "Challenges in 5G: how to empower SON with big data for enabling 5G," *IEEE Network*, vol. 28, no. 6, pp. 27–33, 2014.

[18] H.-D. Bae, B. Ryu, and N.-H. Park, "Analysis of handover failures in LTE femtocell systems," in *Proceedings of the Australasian Telecommunication Networks and Applications Conference (ATNAC '11)*, pp. 1–5, IEEE, Melbourne, Australia, November 2011.

[19] S. Haykin, *Neural Networks. A Comprehensive Foundation*, Prentice-Hall, 1999.

[20] L. Behera, S. Kumar, and A. Patnaik, "On adaptive learning rate that guarantees convergence in feedforward networks," *IEEE Transactions on Neural Networks*, vol. 17, no. 5, pp. 1116–1125, 2006.

[21] J. A. Snyman, *Practical Mathematical Optimization: An Introduction to Basic Optimization Theory and Classical and New Gradient-Based Algorithms*, vol. 97 of *Applied Optimization*, Springer, New York, NY, USA, 2005.

[22] M. Gudmundson, "Cell planning in Manhattan environments," in *Proceedings of the 42nd Vehicular Technology Conference*, pp. 435–438, IEEE, Denver, Colo, USA, May 1992.

[23] P.-E. Ostling, "Implications of cell planning on handoff performance in manhattan environments," in *Proceedings of the 5th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications: Wireless Networks—Catching the Mobile Future*, vol. 2, pp. 625–629, The Hague, The Netherlands, September 1994.

[24] M. Karabacak, D. Wang, H. Ishii, and H. Arslan, "Mobility performance of macrocell-assisted small cells in Manhattan model," in *Proceedings of the 79th IEEE Vehicular Technology Conference (VTC-Spring '14)*, pp. 1–5, Seoul, South Korea, May 2014.

[25] G. Piro, L. A. Grieco, G. Boggia, F. Capozzi, and P. Camarda, "Simulating LTE cellular systems: an open-source framework," *IEEE Transactions on Vehicular Technology*, vol. 60, no. 2, pp. 498–513, 2011.

Hindawi Publishing Corporation
Mobile Information Systems
Volume 2016, Article ID 3057397, 11 pages
http://dx.doi.org/10.1155/2016/3057397

Research Article

Hierarchical Brokering with Feedback Control Framework in Mobile Device-Centric Clouds

Chao-Lieh Chen,[1] Chao-Chun Chen,[2] and Chun-Ting Chen[1]

[1] *Department of Electronic Engineering, National Kaohsiung First University of Science and Technology (First Tech), No. 1 University Road, Yanchao District, Kaohsiung 824, Taiwan*
[2] *Department of Computer Science and Information Engineering and Institute of Manufacturing Information and Systems, National Cheng Kung University, Tainan, Taiwan*

Correspondence should be addressed to Chao-Lieh Chen; frederic@ieee.org

Received 11 April 2016; Accepted 13 June 2016

Academic Editor: Young-June Choi

We propose a hierarchical brokering architecture (HiBA) and Mobile Multicloud Networking (MMCN) feedback control framework for mobile device-centric cloud (MDC2) computing. Exploiting the MMCN framework and RESTful web-based interconnection, each tier broker probes resource state of its federation for control and management. Real-time and seamless services were developed. Case studies including intrafederation energy-aware balancing based on fuzzy feedback control and higher tier load balancing are further demonstrated to show how HiBA with MMCN relieves the embedding of algorithms when developing services. Theoretical performance model and real-world experiments both show that an MDC2 based on HiBA features better quality in terms of resource availability and network latency if it federates devices with enough resources distributed in lower tier hierarchy. The proposed HiBA realizes a development platform for MDC2 computing which is a feasible solution to User-Centric Networks (UCNs).

1. Introduction

A user possesses many mobile devices nearby to enjoy proactively provided services. Wearable and portable devices together with proximity and remote servers process large amount of requests to complete a service. Therefore, an instance of cloud computing methodology to federate these devices and servers is desired especially to the 5th-generation (5G) era that cooperation among massive devices is one of technology focuses. Moreover, the latest surveys [1–5] on User-Centric Network (UCN) advocate self-organizing autonomic networks in which users cooperate through sharing of network services and resources. The self-organizing autonomic architecture, no matter whether ad hoc or infrastructure-based, federates nominal low cost devices and makes users a new type of resource provider and stakeholder in addition to content consumer or producer. Based on sociological relations, mobile devices proactively join the UCN and become network elements (such as router or gateway) to share bandwidth among themselves and to obtain more reliable network connection allowing free roaming. The user-provided networks (UPNs) proposed in [2] are one of the examples. The contribution [3] based on software-defined networking (SDN) explores cooperation among wireless mobile devices to share network transmission bandwidth. Routing, opportunistic relaying, and sensing of hybrid types of information [5] through spontaneous networks [4] are projection of user behaviors in their social networks.

The UCNs emphasize three main key techniques [1]: (1) understanding user context, (2) profiling and predicting user interests, and (3) personalizing content delivery. Therefore, the state observation of UCN underlying a specific service is essential for constructing control and management planes. However, depending on resource distribution among attending mobile devices and data centers, the state space can be too large to be observed. Then, design of control and management algorithms becomes complex and challenging.

In this paper, we propose HiBA architecture with hierarchical brokering and feedback control framework MMCN for both control and management planes in UCNs. The idea comes from the hierarchical fuzzy control [6] where a complex control problem with huge state space is conquered. In a hierarchical control system, the cross product of state spaces of all control hierarchies remains huge while each hierarchy adopts only a few fuzzy rules to perform simple control tasks.

A series of contributions adopting MDC2 for improving load balance [7], user-centric security [8], access control [9] in the cloud as well as MDC2 applications of real-time seamless video sharing [8], and sociological ontology-based hybrid recommendation [10] were proposed. In these contributions, we see that MDC2 is a feasible solution to provide services in UCN. In this paper, HiBA with feedback control framework generalizes the brokering and state observation. Availability and network latency are analyzed and experimental HiBA comprising real-world mobile devices and servers are realized. We prove that a UCN with huge state space is observable and hence controllable through HiBA cloud networking. In this paper, we further extend our previous presentation [11] to demonstrate algorithm embedding using the MMCN development platform in both lower and higher tiers of HiBA. We study energy-aware balancing in the cloudlet tier where dense mobile devices federate. In addition, we study the load balancing in bigger data centers where large amount of requests usually arrives in a short time. Through both theoretical analysis and real-world experiments, we conclude that a HiBA federation with MMCN is a platform for developing distributed algorithms in future 5G's massive machine-to-machine (M2M) communications and UCNs.

This paper is organized as follows. Section 2 depicts the HiBA architecture and the feedback control framework. Mathematical analysis on availability and network latency is provided in Section 3. Section 4 demonstrates real-world experiments in which the HiBA federates heterogeneous mobile and fixed devices in three tiers using different network interfaces. Energy-aware balancing for a cloudlet and load balancing for a bigger data center are both conducted. We also present conclusions at the end of Section 5.

2. System Architecture

Unlike traditional data centers that span trees of VMs in a top-down manner from a primary computing domain, usually called domain zero, in a physical machine (PM), the proposed HiBA architecture is initiated from physical mobile devices at the bottom tier, called the cloudlet tier. Each broker dynamically connects and adaptively controls lower tier PMs or VMs and thus can avoid drawbacks, such as the degradation of the aggregate available bandwidth and circuitous communication path of VM spanning trees [12]. The system includes resources in the cloudlets then associates cloudlets with private and public clouds to further scale up the cloud federation. On the management aspect, we exploit adaptive feedback control framework to tackle the uncertain dynamics of mobile ad hoc networks in order to adapt to the dynamics such as dynamic joining and leaving a broker's domain caused by user mobility, device failure, or physical network handoff [8].

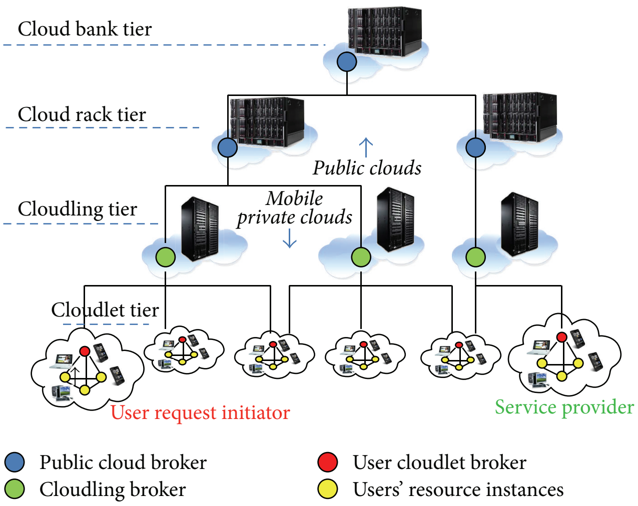


Figure 1: Hierarchical brokering architecture (HiBA) for mobile device-centric cloud computing.

2.1. Hierarchical Brokering. Figure 1 shows the HiBA architecture, which is comprised of four tiers. Tier 0, called the cloudlet tier, contains groups of mobile user devices. Each cloudlet is managed by a cloudlet broker maintaining network connectivity and member association and also performing QoS control. A cloudlet is dynamically organized by the broker according to resource instant status including CPU, memory, network capability, and energy consumption. The resources underlying a tier is abstracted as a JSON or XML list in its broker's database accessible to federation members through RESTful API. Thus, each entry of the list includes a URL containing available RESTful command or a URL of filename and file attributes. Fields of security configurations, NAT port, and GPS coordinates are optionally included in each entry [8]. A tier-0 device does not have a VM member except itself and it aggregates its resources to be shared with other members in the resource list. Each broker hierarchically aggregates the resource lists from members and uploads updates to its upper tier broker.

We continue using the cloudlet federation procedure in our previous research [8]. A cloudlet broker is either dynamically elected among user devices according to the resources status or selected by its upper tier (tier 1) broker, called cloudling broker. A cloudling broker is a small data center or private cloud proximate to a cluster of cloudlets that associate with the cloudling. It is similar to the small data center proposed in [13, 14], owned by a smaller business unit such as a coffee shop or a clinic. We differ from [13, 14] in that our cloudlets are autonomously grouped by user devices which also share resources with others. A tier-n broker associates with a tier-$(n + 1)$ broker to scale the federation in depth or alternatively to include more tier-$(n - 1)$ federations to scale in width. In the proposed architecture, each broker only recognizes next lower tier brokers while further lower tiers are transparent to it. Furthermore, each broker has the same feedback control framework for managing lower tier devices' network attachments and resource associations as well as for QoS control on requesting and executing services.

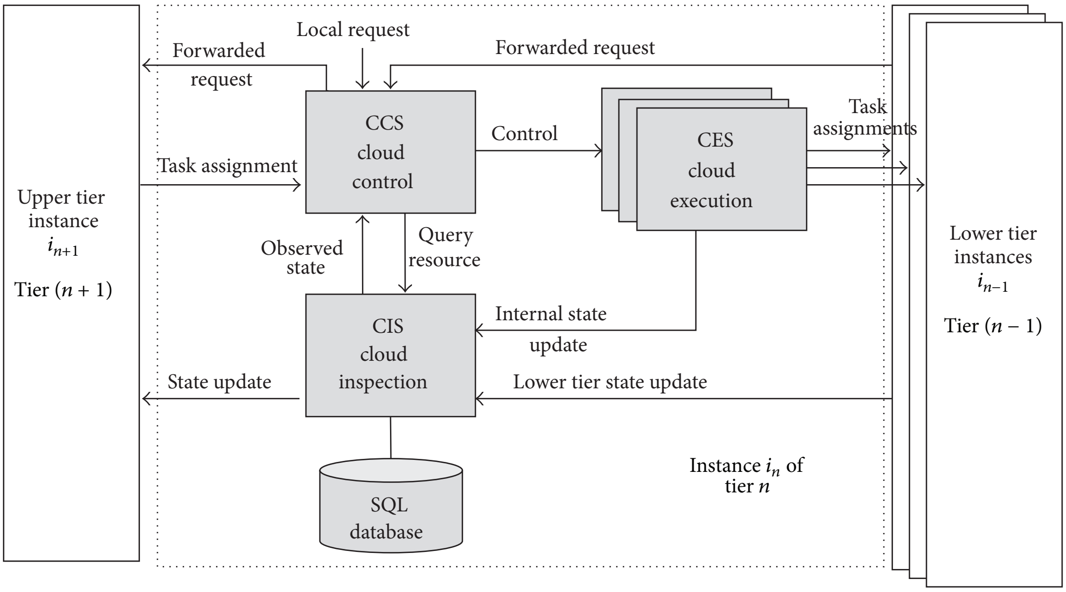


Figure 2: The feedback control framework MMCN in a HiBA broker.

2.2. Feedback Control Framework. Figure 2 shows the feedback control framework MMCN of each broker in the HiBA virtualized network. The feedback loop is used to adaptively manage and control the cloud federation governed by the broker. This management includes network attachments and resource associations caused by user mobility and VM migration [8]. In future application, the framework is ready for developing hierarchical control algorithms such as request differentiation and task scheduling as well as real-time QoS control and job tracking once a service is started. A feedback control loop comprises three subsystems including Cloud Inspection Subsystem (CIS), Cloud Control Subsystem (CCS), and Cloud Execution Subsystem (CES) which are analogous to observer, controller, and plant, respectively, in a classic feedback control system. When performing real-time control for QoS, multiple lower tier instances of the feedback loop are allocated. A tier-n CCS determines the number of tier-$(n-1)$ instances to be allocated; then the tier-n CES invokes the required instances each of which is constituted by the other set of CIS, CCS, and CES at the tier-$(n-1)$.

A CCS processes tasks assigned by its upper tier broker and processes requests from lower tiers. Then according to the database and states tracked by the CIS subsystem, it determines the controls including admission of member joining, allocating member federations for task execution, schedules of tasks, and service level agreements (SLAs) of tasks to be assigned to member federations. A number of CES instances physically accomplish the network attachment for the admission control as well as the recruiting of members associated with corresponding SLAs to complete current schedule. A CCS produces these determinations as controls to CES subsystems while the executions are left to CESs and associated member federations. The CIS observes the state updates from lower tiers for CCS's reference when determining these controls.

2.3. Algorithm Embedding. It is obvious that this framework is feasible for future development of hierarchical control and management algorithms including admission control, request differentiation, task partition, adaptive resource allocation, and dynamic task scheduling regarding SLAs assigned by upper tier broker. For example, on receiving a new request, a priority queue of requests is adapted with new set of priorities. The CCS checks with the CIS to determine whether to forward the request to upper tier broker or to reduce the request into smaller tasks such that the CESs can further assign the task partitions to lower tier members just as the processing of task assignments from upper tier broker's CES. The CES's makes lower tiers transparent to the CCS and upper tiers as if it is the single plant of the CCS controller. Since it is the CESs and associated member federations executing tasks rather than the CCS, the CCS is able to process the management and control algorithms without waiting for the end of the current job execution. This is easy to be realized by programming multiple threads each of which realizes CCS, CESs, and CIS subsystems.

In summary, the CCS of a tier-n broker forwards requests to tier-$(n+1)$ if it is not able to process them according to the CIS database. The tier-$(n+1)$ broker assigns jobs to other tier-n brokers through CES if the requested resources are sharable in these tier-n members and below. Therefore, the sharing becomes intercloud service of tier-n as well as intracloud service of tier-$(n+1)$.

3. Performance Analysis

The baseline performance for benchmarking is to evaluate the HiBA architecture and feedback control framework without optimization on specific management and control. That is, the queues are simple FIFOs without priority adaptation and all devices in all tiers randomly generate requests. The

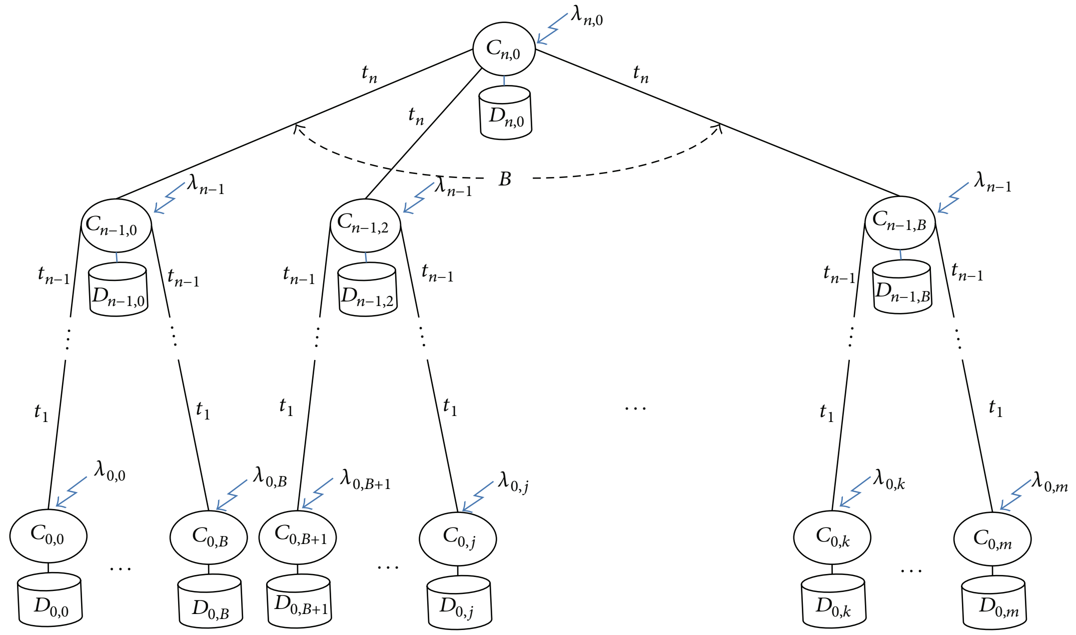


FIGURE 3: Performance model of the HiBA architecture.

performance model of the virtualized network is shown in Figure 3. Suppose that the maximum number of members of a HiBA broker is B; a broker j in tier i has computing capability C_{ij}, which is also the minimum length of the *Task Queue*. The capability C_{ij} also represents the maximum number of tasks that can be processed by a broker in tier i under the condition that no task is dropped. The resources are assumed to be uniformly distributed in each tier. In tier i, the average amount of contents and resources is D_{ij}. A tier-i broker has its amount of local requests in a Poisson distribution with mean λ_{ij}. This simulates the reinitiated requests caused by request partitions and task handoffs. t_n denotes the mean initial latency caused by network delay to forward requests and the waiting time that requests stay in the *Request Queue* until being partitioned into smaller tasks. In this paper, we derive baseline performances on availability and latency.

3.1. Resource Availability. We define availability, a_{ij}, as the ratio of computing capability C_{ij} over the number of accepted requests R_{ij} to a broker j in tier i. That is,

$$a_{ij} = \min\left(\frac{C_{ij}}{R_{ij}}, 1\right). \tag{1}$$

In a HiBA architecture of n tiers, the expected total resource amount summing from the records in the CIS databases is approximated as

$$D_{\text{total}} = D_n + \sum_{i=1}^{n}\left(\prod_{k=0}^{i-1} B_{n-k}\right) D_{n-i} = \sum_{i=0}^{n} B^i D_{n-i} \tag{2}$$

supposing that $D_i = E[D_{ij}]$. Thus, the expected total resource amount of the subtree rooted at a tier-i node is

$$D_{i,\text{total}} = \sum_{k=0}^{i} B^k D_{n-k}. \tag{3}$$

For any request, the probability that the requested resource r_{ij} is available in local node ij is approximated by

$$P\left(r_{ij}\right) = P\left\{r_{ij} \in \mathbf{D}_{ij}\right\} = \frac{D_{ij}}{D_{\text{total}}}, \tag{4}$$

where $\mathbf{D}_{ij}$ is the set of resources in node ij. Suppose that the sample space is the union of $\mathbf{D}_{ij}$ in the whole HiBA architecture. Then, the worst case of availability occurs when network latency is much smaller than the mean time interval $1/\lambda_{ij}$ between two consequent requests. That is, requests from other cloud federations including those in lower and higher tiers arrive in current tier instantly with negligible network delay. Thus, the maximal number of requests arriving node ij is

$$R_{ij} = P\left(r_{ij}\right) \sum_{k=0}^{n} B^k \lambda_{n-k,j}. \tag{5}$$

Then we obtain the worst case availability as

$$\begin{aligned} a_{ij} &= \min\left(\frac{C_{ij}}{P\left(r_{ij}\right)\sum_{k=0}^{n} B^k \lambda_{n-k,j}}, 1\right) \\ &= \min\left(\frac{C_{ij}}{D_{ij}} \cdot \frac{D_{\text{total}}}{\sum_{k=0}^{n} B^k \lambda_{n-k,j}}, 1\right). \end{aligned} \tag{6}$$

From (6), we see that when n is smaller, resources (including contents) are more centralized with larger D_{ij} and if D_{total} remains large, this causes lower availability. Thus there are various means to increase availability. One is to increase capability C_{ij}, though this will increase costs. Alternatively, branch amount B of each tier can be increased, though this also will increase computing capability and costs. Third, by adopting HiBA, we put user devices in tier 0 or tier 1 (once the user device is elected cloudlet broker). However, the capabilities C_{0j} and C_{1j} of thin devices may be small, and so D_{0j} and D_{1j} are also expected to be small. This indicates that availability remains close to 1 if resource distribution D_{ij} is proportional to the capability C_{ij} with the ratio of total number of requests over total resource amount. Therefore, the optimal resource distribution to low tier brokers and mobile devices both offloads data centers' computing overhead and increases user satisfaction.

3.2. Latency. Social networks can cause locality, such that many requests do not travel to their destinations over long distances in terms of either network relay counts or the terrestrial radio barriers. Thus initial latency of a service is reduced. Lower tier brokers have smaller resource granules, while locality based on social network provides access proximity and thus results in shorter latency. We estimate the latency of a request traversing the HiBA architecture as follows. Suppose that a request initiated from a tier-i node arrives at its destination possessing the required resources in tier $l, i < l$. The expected latency of the request is

$$T_{i,l} = P\left(\overline{r}_{i<l}\right) \sum_{k=i}^{l} \left(t_{N,k} + t_{D,k} + t_{C,k}\right), \tag{7}$$

where $P(\overline{r}_{i<l})$ is the probability that the requested resource is *not* found in tiers lower than l and $t_{N,k}, t_{D,k}, t_{C,k}$ are latencies caused, respectively, by network communication, request queuing plus database querying, and computing for the brokering at each tier-k broker on the path to tier l. Supposing that request arrival is independent of resource distribution, we have

$$P\left(\overline{r}_{i<l}\right) = \frac{\lambda_i}{\lambda_{\text{total}}} \left(1 - \frac{D_{l-1,\text{total}}}{D_{\text{total}}}\right). \tag{8}$$

Thus, the estimated latency is

$$T_{i,l} = \frac{\lambda_i}{\lambda_{\text{total}}} \left(1 - \frac{D_{l-1,\text{total}}}{D_{\text{total}}}\right) \left(\sum_{k=i}^{l} t_{N,k} + t_{D,k} + t_{C,k}\right). \tag{9}$$

If a request is originated from a lower tier user device, that is, $i = 0$ or 1, the effective way to reduce expected latency $T_{0,l}$ or $T_{1,l}$ is to reduce $P(\overline{r}_{i<l})$. This means increasing $D_{l-1,\text{total}}$ by distributing resources to lower tiers, that is, increasing D_k for $k < l$. However, this also increases both database searching time $t_{D,k}$ and computation time $t_{C,k}$. Considering network delay $t_{N,k}$, we can expect that, in the virtualized network, the physical distance of a cloud server is increasing as k is getting larger. A high tier link in the virtualized HiBA network actually contains many more physical relaying hops than a lower tier link. If request is sent using TCP protocol, the network latency is increasing much more than that caused by database searching and brokerage computing, as k increases. From (9), we still effectively decrease the latency by increasing $D_{l-1,\text{total}}$. This can be expected especially for multimedia content sharing since, according to social network relations, contents are stored in proximate cloudlets or cloudlings which are lower tiers (small l).

3.3. Development Platform for MDC2 Control and Management. From the availability and latency analysis, we see that a HiBA with feedback framework is a development platform. It is easy to observe the performance when developing algorithms for admission control, request differentiation, task partition, adaptive resource allocation, and dynamic task scheduling regarding SLAs assigned by upper tier broker. For example, the admission control in federating a tier affects resource distribution D_{ij} and according to the number of network hops between a broker and a member, the network latency $t_{N,k}$ in (9) also differs. The request differentiation and queuing mechanism directly affect $T_{i,l}$ since $t_{D,k}$ includes the queue delay that a request stays in the *Request Queue*. Task partitioning, resource allocation, and scheduling further affect the efficiency processing requests and consequently they further affect $t_{D,k}$ and resultant $T_{i,l}$. Globally, they also affect arrival rate λ_i at tier i. Exploiting hierarchical feedback control, it is easy to ensure availability by performing proper management while tracking the latency by tuning control regarding the deadline specified in the SLA of each task.

4. Case Studies

To demonstrate algorithm embedding using the MMCN development platform in both lower and higher tiers, we study energy-aware balancing among mobile devices in the cloudlet tier as well as the load balancing in bigger data centers where large amount of requests would arrive in a short time, that is, large λ_{ij}. The energy-aware balancing is critical in crowd-sensing applications especially when the sensing data amount is large such as using cameras for image data collecting. The load balancing is essential for bigger data centers especially when database access frequency is high. Both of the balancing algorithms are effective in the Industry 4.0 era when crowd-sensing and big data analytics are deployed.

4.1. Energy-Aware Balancing. The energy-aware balancing is based on unsupervised fuzzy feedback control where the reference command is also adapted according to the feedback state of the federation self-organized by mobile devices. The principal idea comes from the energy proportional routing [15] that the lifetime of a clustering-based sensor network is prolonged if member nodes' proportions of consumed energy in the remaining are close to the cluster's. We exploit the energy proportion of the cloudlet federation as the adaptive reference to be tracked by the fuzzy feedback control system. Therefore, the energy sharing is unsupervised because of no given objective of the control a priori.

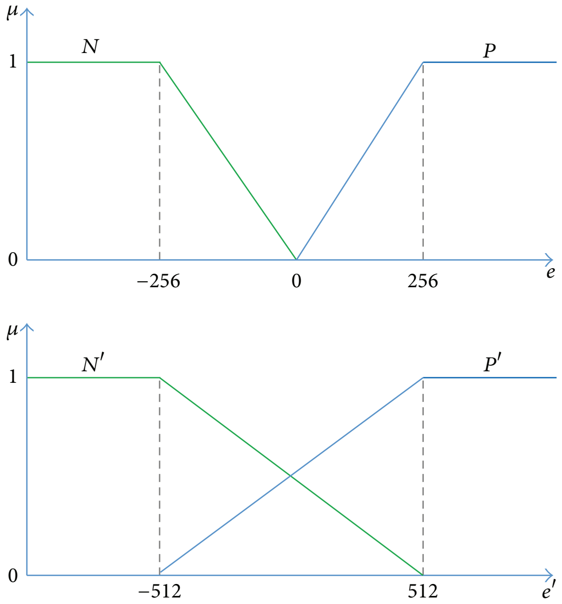


Figure 4: Membership functions for fuzzy sets N, P, N', and P'.

Suppose that, in a cloudlet tier of N member nodes, data transmission is observed by the cloudlet broker in each round t (discrete time). We first define symbols as follows:

(i) k: member node identification, $1 \leq k \leq N$.

(ii) $D_k(t)$: data transmission amount being assigned to node k at round t.

(iii) $E_k(t)$: remaining energy of node k at t.

(iv) $X_k(t) = D_k(t)/E_k(t)$, representing the ratio of overhead to the remaining energy (current capability). This is the indication of how node k is loaded with respect to the remaining energy.

(v) $Th(t) = \sum_{k=1}^{N} D_k(t) / \sum_{k=1}^{N} E_k(t)$, representing how the whole cloudlet federation is loaded.

(vi) $e_k(t) = X_k(t) - Th(t)$, representing the difference of loading between node k and the whole federation.

(vii) $e'_k(t) = e_k(t) - e_k(t-1)$ representing how the difference changes.

(viii) $r_k(t)$: the ratio of data amount assigned to node k at t.

We define fuzzy rules as follows:

(R1) If $e_k(t)$ is P and $e'_k(t)$ is P', then $r_k(t+1)$ is $S_k(t)$.

(R2) If $e_k(t)$ is P and $e'_k(t)$ is N', then $r_k(t+1)$ is $M_k(t)$.

(R3) If $e_k(t)$ is N and $e'_k(t)$ is P', then $r_k(t+1)$ is $M_k(t)$.

(R4) If $e_k(t)$ is N and $e'_k(t)$ is N', then $r_k(t+1)$ is $L_k(t)$.

The membership functions for fuzzy sets N, P, N', and P' are configured in Figure 4.

We realize the "and" operator with t-norm "minimum." That is, the matching degrees μ_1, μ_2, μ_3, and μ_4 of premise part of rules (R1) to (R4), respectively, are

$$\begin{aligned} \mu_1(t) &= \min\left(\mu_P(e(t)), \mu_{P'}\left(e'(t)\right)\right), \\ \mu_2(t) &= \min\left(\mu_P(e(t)), \mu_{N'}\left(e'(t)\right)\right), \\ \mu_3(t) &= \min\left(\mu_N(e(t)), \mu_{P'}\left(e'(t)\right)\right), \\ \mu_4(t) &= \min\left(\mu_N(e(t)), \mu_{N'}\left(e'(t)\right)\right). \end{aligned} \tag{10}$$

Obtaining the matching degrees μ_1, μ_2, μ_3, and μ_4 of the four fuzzy rules, respectively, we perform the defuzzification equivalently applying the Takagi-Sugeno inference method. The inference result, adequate ratio of data amount to transmit, is

$$\begin{aligned} & r_k(t+1) \\ & \quad = \frac{\mu_1(t) S(t) + \mu_2(t) M(t) + \mu_3(t) M(t) + \mu_4(t) L(t)}{\mu_1(t) + \mu_2(t) + \mu_3(t) + \mu_4(t)} \end{aligned} \tag{11}$$

by configuring the conclusion part membership functions as dynamic singletons as follows:

$$\begin{aligned} L(t) &= r_k(t-1) + \Delta \\ M(t) &= r_k(t-1) \\ S(t) &= r_k(t-1) - \Delta, \end{aligned} \tag{12}$$

where Δ is the ratio tuning amount for each round. Finally, the data amount assigned to node k is determined as

$$D_k(t+1) = \frac{r_k(t)}{\sum_{i=1}^{N} r_i(t)} \mathbf{D}(t), \tag{13}$$

where $\mathbf{D}(t)$ is the total data amount to be transmitted from this cloudlet at time t. In the above fuzzy inference algorithm, each member node tracks the dynamic control goal $Th(t)$ and the proportion of energy to be consumed in the remaining energy approaches the proportion regarding the whole federation. Therefore, the intrafederation energy sharing is achieved by offloading transmission jobs using the proposed algorithm. When each node k is a lower tier federation, the offloading is hierarchically extended through the MMCN networking where the data amount $D_k(t)$ is determined and assigned by CCS and the energy status updates $E_k(t)$'s are received and aggregated by CIS. This shows the scalability of the HiBA architecture.

4.2. Load Balancing. The load balancing is required in a bigger federation organized in a higher tier of HiBA because a higher federation always has larger amount of request arrivals. As shown in Figure 5, the framework of the proposed load-balanced cloud service interface consists of three components which realizes CES, CCS, and CIS, respectively. The details of components are presented as follows:

(1) Cloud Service Interface Node (CSIN): it is a virtual machine that provides cloud service interface and is

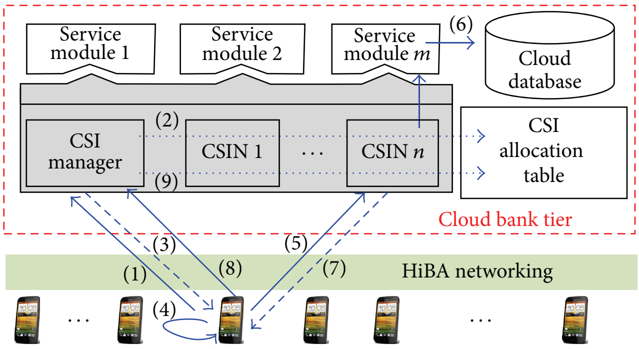


Figure 5: Framework of load-balanced cloud service interface.

denoted as CSIN_i in Figure 5. A CSIN can receive members' requests and obtain results by interacting with requested cloud service module. In our design, each cloud service interface in CSIN is implemented as a RESTful API for universally communicating with different types of mobile devices. A CSIN is then the realization of CES in the MMCN framework.

(2) CSI manager: it is designed to manage the usage status of CSINs and matches mobile users to CSINs. The efficiency of the load balance depends on the matching method of the CSI Manager. A CSI manager is then the realization of CCS in the MMCN framework.

(3) CSI allocation table: it maintains the serving status between mobile users and CSINs and is used by the CSI manager for assisting the matching decision for the new-arrived mobile users. This table is stored in the cloud database. A CSI allocation table is then the realization of CIS in the MMCN framework.

The processing flow of using a cloud service for a mobile user is designed for achieving the load balance. The idea of committing requests requires three phases: (1) service registration (Steps (1)-(2)), (2) service execution (Steps (3)–(7)), and (3) service deregistration (Steps (8)-(9)). The detailed steps, also illustrated in Figure 5, are presented as follows:

(1) Mobile user MU_i sends a request to CSI manager for allocating a CSI address.

(2) CSI manager searches for a CSI machine with least number of serving users, say, CSIN_n, in the CSI allocation table, and then registers (MU_i, CSIN_n) into the CSI allocation table.

(3) CSI manager informs MU_i that CSIN_n can serve his/her cloud service requests in the following period.

(4) MU_i configures the cloud service interface by replacing the IP attribute in the service template with the IP address of CSIN_n.

(5) MU_i connects to CSIN_n through the configured URL of the cloud service interface and sends a request.

(6) CSIN_n delivers the request to the associated cloud service which may access the cloud databases if required. After processing, the results will be sent back to CSIN_n.

(7) CSIN_n sends the results to MU_i. (If more requests are required to processed, Steps (3)–(7) are repeated.)

(8) MU_i deregisters CSIN_n to CSI manager.

(9) CSI manager removes the registration record of MU_i from the CSI allocation table.

Notice that the matching rule for a new mobile user and CSINs in the current design uses least-user-first basis, as shown in Step (2). The matching principle can be modified according to different demands. In the experiments, we will show that the current design already obtains splendid performance. More study on customizing matching methods is one of our future research directions. That is, although five machines are used in our CSI (more than the single-machine CSI), the improvement is by a factor of one hundred, indicating the superior performance of our proposed load-balanced CSI mechanism under the HiBA architecture.

5. Real-World Experiments

5.1. HiBA Baseline Performance. We conduct an experiment demonstrating performance difference in terms of availability and latency when distributions of virtualized resource instances (VRIs) differ. We leave development of enhancing algorithms for specific purposes in the CCS as future work which will seek performance in terms of various metrics. A total of 50,000 VRIs are distributed in a 3-tier HiBA federation comprising Android smart phones, tablets, and VMs in desktop PCs. Figure 6 is the HiBA topology and related specifications of the machines are shown in Table 1. Each device is labeled (i, j) if it is the jth node at tier i. Device (1, 3) connects to (2, 0) through a 3G access point with VPN tunneling. Device (0, 11) connects to (1, 0) with USB and shares Internet connection from (1, 0). These machines have different computing ability, though request and task queues being of the same size of 20 to preserve the differences of computing ability. Requests are randomly generated in each device. Mean request arrivals (λ) are 3, 1.5, and 1 which equivalently mean that request intervals ($1/\lambda$) are 333 ms, 666 ms, and 1 s, respectively. The resource distributions have four cases. The first case is that the tier-2 device possess all the 50,000 VRIs while 16,000 VRIs are duplicated to each tier-1 federations. In the remaining three cases, we, respectively, duplicate 28000, 32000, and 36000 VRIs to each tier-1 federation. The experiment is conducted in a heterogeneous networking environment connecting machines by different communication technologies shown in Table 1.

The results of the experiment are shown in Figures 7 and 8. Both Figures 7 and 8 reveal that distributing resources in lower tiers provides better performance. Figures 8(a) and 8(b) show the latency differences caused by the resource distributions that 16000, 32000, and 40000 VRIs are duplicated in each cloudling tier. The results before all brokers fully loaded are also given with respective zoom-in charts. When devices all continuously issue requests at a high frequency (3 requests per second), the queue delays are obviously high and we see a few requests that are dropped. Requests from tier-1 devices have lower loss rate since the resources requested are

Table 1: Machines in the offloading experiments using heterogeneous communication technologies.

(Tier, device)	Specifications	Physical network link
Tier 2/(2, 0)	Desktop PC: Intel Core i5-3470 CPU 3.2 GHz, 4 G RAM	Ethernet
Tier 1/(1, 0), (1, 1)	Desktop PC: Intel Core i7-4770 CPU 3.4 GHz, 8 G RAM	
Tier 1/(1, 2)		
Tier 1/(1, 3)	Laptop: Intel Core i7-2820QM CPU 2.3 GHz, 8 G RAM	Down: WiFi; UP: 3G VPN
Tier 0/(0, 0~3)	Desktop PC: Intel Pentium D CPU 3.4 GHz, 1 G RAM	Ethernet
Tier 0/(0, 4~6)		
Tier 0/(0, 7)		
Tier 0/(0, 8)	Tablet: ASUS Fonepad Intel Atom Z2420, 1.2 GHz	WiFi
Tier 0/(0, 9)	Cell Phone S3: SAMUNG Exynos 4412, 1.4 GHz	
Tier 0/(0, 10)	Tablet: ASUS Nexus7 NVIDIA Tegra 3 4-core	
Tier 0/(0, 11)	Tablet: WS-170 Qualcomm QCT MSM8x60 surf/1.5 GHz duo-core	USB ←→ PC ←→ Ethernet
Tier 0/(0, 12~13)	Virtual Machines (VirtualBox)	Ethernet
Tier 0/(0, 14)	Tablet: Samsung Galaxy CPU GT-I8260	WiFi

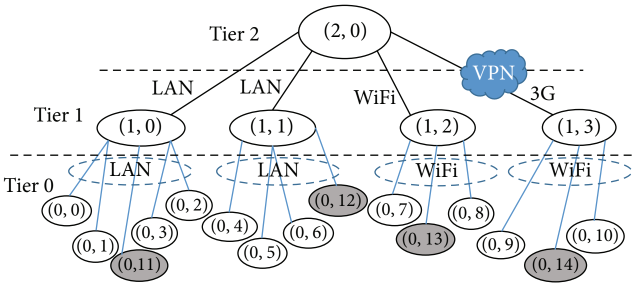


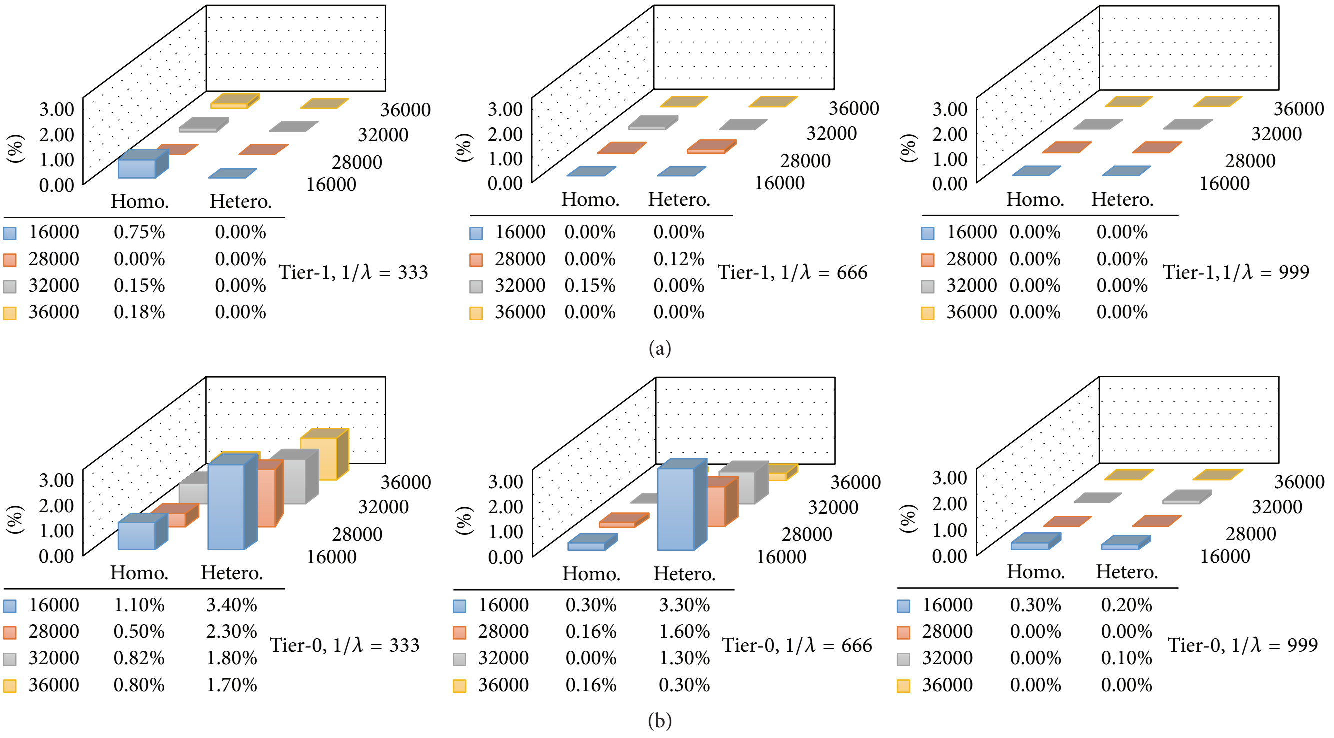
Figure 6: Real-world HiBA topology for offloading performance measurement. Gray nodes are local VMs of respective tier-1 brokers. Diverse physical network links are exploited in tiers of federations.

obtained in a shorter time. For each physical machine, the CPU utilization of the MMCN brokering itself is smaller than 2% depending on the number of cores. The mean brokering computation and database querying time is estimated to be about 500 ms while the WLAN delay time is smaller than 5 ms. In the heterogeneous network case, the latency difference will be larger if upper tiers are physically at a long distance from the cloudlings and cloudlets. If the request interval is higher than the processing time, the latency tends to be smaller. When more VRIs are duplicated in lower tier devices, we also see that the latencies are smaller.

In this paper, we leave algorithms, such as request partition, network embedding, and federation for specific performance metrics, as future work. Instead, we implement real-world HiBA to prove that, with feedback framework, it is a new design paradigm and development platform for these algorithms.

5.2. Energy-Aware Balancing. In the energy-aware balancing experiment we exploit four mobile devices of different types from different manufacturers. The cloudlet is self-organized according to the broker election protocol in [8]. The elected broker is ASUS-Fonepad. The other hierarchical federations are in Figure 6. Each device, including the broker itself, in each round updates its energy status to the CIS of the broker (ASUS-Fonepad). Each mobile device in the cloudlet is recharged to 100% of respective battery energy capacity. The data and total data amount to be transmitted in each round are randomly generated with mean of 2.4 GB prior to the experiment. For each round t, total data amount is the same for both balancing and nonbalancing cases. In nonbalancing case, the data amounts are evenly assigned to members. In balancing case, the data amounts are determined by the fuzzy controller in Section 4.1. The data amount assignment is of unit 100 MB. When any device has remaining energy lower or equal to 40%, the experiment is terminated. The initial singletons are configured as $S(0) = 3$, $M(0) = 5$, and $L(0) = 7$ ($\Delta = 2$). The result of energy-aware balancing is shown in Figure 9. The respective control systems' behaviors are in Figure 10. The fuzzy controllers of mobile devices all track the federation's data amount to energy proportion (Th). We see that no matter how the performance of respective device varies, a device with better transmission capability will share its energy with others. Tablet PCs are usually with higher capacity batteries. Without energy sharing, the remaining energy of a tablet drops more slowly than a mobile phone. With energy sharing, tablets undertake more transmission jobs and become with higher energy drop rate, and the mobile phones become with lower energy drop rate. However, we see that the whole cloudlet federation has longer lifetime. The experiment can also be applied to energy consuming tasks in addition to data transmission.

5.3. Load Balancing. The load-balanced cloud service interface (CSI) in the form of RESTful APIs is implemented by using Jersey and is deployed on Microsoft Windows Azure public cloud platform. We conduct an experiment for comparing the proposed CSI to that of single machine. In our experiment, five CSI machines are used to share the requests from users. In the experiment, the data producing rate for each user is 1 request per second. The performance



	Homo.	Hetero.
🟦 16000	0.75%	0.00%
🟧 28000	0.00%	0.00%
⬜ 32000	0.15%	0.00%
🟨 36000	0.18%	0.00%

Tier-1, $1/\lambda = 333$

	Homo.	Hetero.
🟦 16000	0.00%	0.00%
🟧 28000	0.00%	0.12%
⬜ 32000	0.15%	0.00%
🟨 36000	0.00%	0.00%

Tier-1, $1/\lambda = 666$

	Homo.	Hetero.
🟦 16000	0.00%	0.00%
🟧 28000	0.00%	0.00%
⬜ 32000	0.00%	0.00%
🟨 36000	0.00%	0.00%

Tier-1, $1/\lambda = 999$

(a)

	Homo.	Hetero.
🟦 16000	1.10%	3.40%
🟧 28000	0.50%	2.30%
⬜ 32000	0.82%	1.80%
🟨 36000	0.80%	1.70%

Tier-0, $1/\lambda = 333$

	Homo.	Hetero.
🟦 16000	0.30%	3.30%
🟧 28000	0.16%	1.60%
⬜ 32000	0.00%	1.30%
🟨 36000	0.16%	0.30%

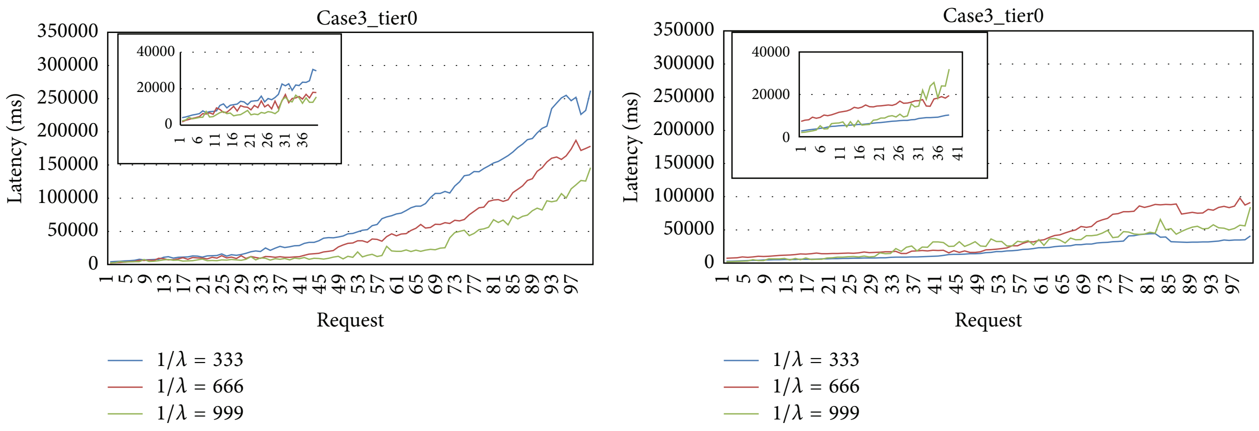
Tier-0, $1/\lambda = 666$

	Homo.	Hetero.
🟦 16000	0.30%	0.20%
🟧 28000	0.00%	0.00%
⬜ 32000	0.00%	0.10%
🟨 36000	0.00%	0.00%

Tier-0, $1/\lambda = 999$

(b)

FIGURE 7: Mean losses of tier 0 (b) and tier 1 (a) in cases of different resource distributions when 16000, 28000, 32,000, and 36,000 VRIs are duplicated to each tier-1 federation. Request intervals $1/\lambda$ are in milliseconds.



(a) 16,000 VRIs in each tier-1 federation

(b) 40,000 VRIs in each tier-1 federation

FIGURE 8: Mean latencies in cases of different resource distributions.

metric is the missing rate defined as $(T - S)/T$, where T is the number of total requests and S is the number of successful requests. Table 2 shows the experimental results under different amount of users from 100 to 300. We can see that when the number of users is low (100 users), both CSI designs can successfully process their data requests. As the number of users increases (e.g., 300 users), our proposed CSI can almost process it with only 0 : 27% missing rate; however, over 27% requests are failed to deliver data to cloud database in the CSI of single machine.

6. Conclusion

We have proposed and implemented a mobile device-centric cloud computing architecture, HiBA, based on feedback control framework. The proposed hierarchical brokering architecture is self-organized featuring scalability and hierarchical autonomy and is easy to embed management and control algorithms to develop a federation with better performance in terms of availability and latency. Mobile devices and small servers federate into cloudlets and cloudlings of hierarchical

TABLE 2: Experimental comparison of different CSI designs.

Users	Single			Load-balanced scheme		
	Total requests	Successful requests	Missing requests	Total requests	Successful requests	Missing requests
100	52395	52395	**0%**	53370	53370	**0%**
200	97457	80947	**16.91%**	104244	104244	**0%**
300	113624	81883	**27.94%**	131742	131388	**0.27%**

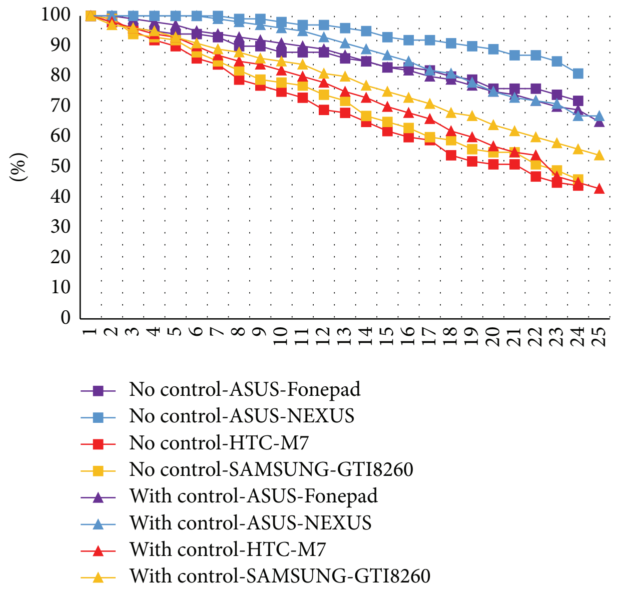


FIGURE 9: Energy-aware balancing result. x-axis: round; y-axis: remaining energy.

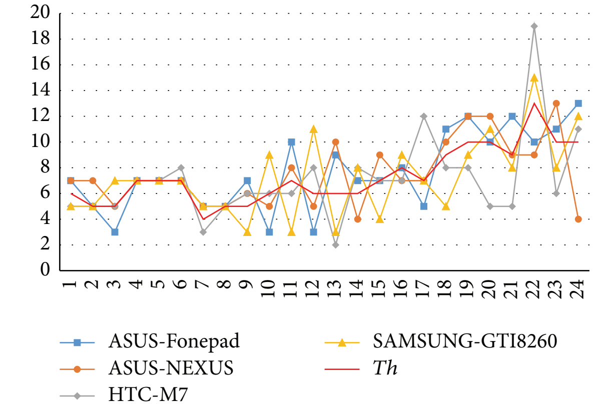


FIGURE 10: Data amount to energy proportions of individual devices (Xi) and the cloudlet federation (Th). x-axis: round; y-axis: data amount to energy proportions.

tiers such that mobile devices not only request services but also provide resources required by diverse services. The implemented HiBA architecture with feedback control framework proves improved performance when resources are adequately distributed to lower tier federations rather than being centralized in a remote cloud server. Through two case studies of cloudlet energy-aware balancing and bigger data center load balancing, we show that the HiBA is actually a development platform for mobile cloud computing and provides a feasible solution to UCN services. Future works include optimization algorithms embedding and big data analytics applications with crowd sensing are to be continued.

Competing Interests

The authors declare that they have no competing interests.

Acknowledgments

Thanks are due to the Ministry of Science and Technology (MOST, which was the National Science Council) in Taiwan for sponsoring this research under continuous funded Projects NSC101-2221-E-327-020-, NSC101-2221-E-327-022-, NSC102-2218-E-327-002-, MOST103-2221-E-327-048-, MOST105-2221-E-327-027-, and MOST105-2221-E-006-141-.

References

[1] Technicolor, "User-Centric Networking," http://usercentricnetworking.eu/about-ucn/.

[2] G. Iosifidis, L. Gao, J. Huang, and L. Tassiulas, "Incentive mechanisms for user-provided networks," *IEEE Communications Magazine*, vol. 52, no. 9, pp. 20–27, 2014.

[3] B. A. A. Nunes, M. A. S. SAntos, B. T. De OliveirA, C. B. MArgi, K. ObrAczkA, and T. Turletti, "Software-defined-networking-enabled capacity sharing in user-centric networks," *IEEE Communications Magazine*, vol. 52, no. 9, pp. 28–36, 2014.

[4] G. Aloi, M. D. Felice, V. Loscrì, P. Pace, and G. Ruggeri, "Spontaneous smartphone networks as a user-centric solution for the future internet," *IEEE Communications Magazine*, vol. 52, no. 12, pp. 26–33, 2014.

[5] F. Hao, M. Jiao, G. Min, and L. T. Yang, "A trajectory-based recruitment strategy of social sensors for participatory sensing," *IEEE Communications Magazine*, vol. 52, no. 12, pp. 41–47, 2014.

[6] M.-L. Lee, H.-Y. Chung, and F.-M. Yu, "Modeling of hierarchical fuzzy systems," *Fuzzy Sets and Systems*, vol. 138, no. 2, pp. 343–361, 2003.

[7] T. C. Lin, M.-Y. Pai, C.-L. Chen, and C.-C. Chen, "Load-balanced cloud service interface for the HiBA mobile cloud environment," in *Proceedings of the IEEE International Conference on Consumer Electronics—Taiwan (ICCE-TW '15)*, pp. 360–361, Taipei, Taiwan, June 2015.

[8] C.-L. Chen, S.-C. Chen, C. Chang, and C. Lin, "Scalable and autonomous mobile device-centric cloud for secured D2D sharing," in *Information Security Applications*, vol. 8909 of *Lecture Notes in Computer Science*, pp. 177–189, 2015.

[9] W.-T. Su, W. Liu, C.-L. Chen, and T.-P. Chen, "Cloud access control in multi-layer cloud networks," in *Proceedings of the IEEE International Conference on Consumer Electronics—Taiwan (ICCE-TW '15)*, pp. 364–365, IEEE, Taipei, Taiwan, June 2015.

[10] H. Huang, H. Yang, and E. H. Lu, "A fuzzy-rough set based ontology for hybrid recommendation," in *Proceedings of the IEEE International Conference on Consumer Electronics—Taiwan (ICCE-TW '12)*, pp. 358–359, Taipei, Taiwan, June 2015.

[11] C.-L. Chen and C.-T. Chen, "Hierarchical Brokering with feedback state observation in Mobile Device-Centric Clouds," in *Proceedings of the 7th International Conference on Ubiquitous and Future Networks (ICUFN '15)*, pp. 410–415, July 2015.

[12] C. J. S. Decusatis, A. Carranza, and C. M. Decusatis, "Communication within clouds: open standards and proprietary protocols for data center networking," *IEEE Communications Magazine*, vol. 50, no. 9, pp. 26–33, 2012.

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

[14] S. Mahadev, "Mobile computing: the next decade," in *Proceedings of the the 8th International Conference on Mobile Systems, Applications, and Services (MobiSys'10)*, pp. 1–6, San Francisco, Calif, USA, June 2010.

[15] C.-L. Chen, J.-W. Lee, W.-T. Su, M.-F. Horng, and Y.-H. Kuo, "Noise-referred energy-proportional routing with packet length adaptation for clustered sensor networks," *International Journal of Ad Hoc and Ubiquitous Computing*, vol. 3, no. 4, pp. 224–235, 2008.

Hindawi Publishing Corporation
Mobile Information Systems
Volume 2016, Article ID 9874969, 9 pages
http://dx.doi.org/10.1155/2016/9874969

Research Article

SDN Based User-Centric Framework for Heterogeneous Wireless Networks

Zhaoming Lu,[1,2] Tao Lei,[1,2] Xiangming Wen,[1,2] Luhan Wang,[1,2] and Xin Chen[1,2]

[1] *Beijing Key Laboratory of Network System Architecture and Convergence, Beijing University of Posts and Telecommunications, Beijing 100876, China*
[2] *Beijing Laboratory of Advanced Information Networks, Beijing University of Posts and Telecommunications, Beijing 100876, China*

Correspondence should be addressed to Zhaoming Lu; lzy_0372@163.com

Received 5 February 2016; Accepted 23 June 2016

Academic Editor: Young-June Choi

Due to the rapid growth of mobile data traffic, more and more basestations and access points (APs) have been densely deployed to provide users with ubiquitous network access, which make current wireless network a complex heterogeneous network (HetNet). However, traditional wireless networks are designed with network-centric approaches where different networks have different quality of service (QoS) strategies and cannot easily cooperate with each other to serve network users. Massive network infrastructures could not assure users perceived network and service quality, which is an indisputable fact. To address this issue, we design a new framework for heterogeneous wireless networks with the principle of user-centricity, refactoring the network from users' perspective to suffice their requirements and preferences. Different from network-centric approaches, the proposed framework takes advantage of Software Defined Networking (SDN) and virtualization technology, which will bring better perceived services quality for wireless network users. In the proposed user-centric framework, control plane and data plane are decoupled to manage the HetNets in a flexible and coadjutant way, and resource virtualization technology is introduced to abstract physical resources of HetNets into unified virtualized resources. Hence, ubiquitous and undifferentiated network connectivity and QoE (quality of experience) driven fine-grained resource management could be achieved for wireless network users.

1. Introduction

Wireless networks are undergoing a bold change. Increasing data traffic pours into wireless networks from wired networks, in which video streaming traffic is the main force. According to the report of Ericsson [1], mobile data traffic is expected to grow at a CAGR (Compound Average Growth Rate) of around 45 percent (2013–2019). This will result in a traffic increase of around 10 times by the end of 2019. In order to accommodate the explosive mobile data traffic growth and a large number of new applications and services demand, operators have deployed large-scale network infrastructures to provide users with ubiquitous network service. Meanwhile, different radio access networks such as cellular networks, WLAN, and wireless mesh networks coexist in the free space around us forming a complex heterogeneous network environment. As a result, huge traffic and heterogeneous characteristics make the management of wireless networks difficult. In addition, with the increase of mobile service vendors and available bandwidth of wireless networks, users' requirements for performance of wireless networks and mobile services have become more and more trenchant. Mismatch between trenchant user requirements of perceived network quality for networks and difficult network management leads to great challenges in wireless networks.

However, wireless networks are traditionally designed with network-centric approaches. Different radio access technologies (RATs) have different resource formats and quality of service (QoS) strategies, and wireless networks are managed and optimized with the goal of improving network performance. Unfortunately, best network performance does not mean best user perceived quality of network service. Coexisting HetNets and QoS driven network management make wireless networks difficult to meet trenchant user requirements for network services. To address this problem brought by network-centric approaches, user-centric framework emerges as a disruptive new communication paradigm. User-centricity means the network is designed and built with the network users in the center to satisfy their requirements and preferences, which refactors the traditional wireless network models due to its user-centricity.

In terms of the user-centricity of proposed framework, two fundamental characteristics are indispensable to assuring user perceived quality in HetNets. Firstly, user-centric framework is supposed to provide users with ubiquitous and undifferentiated network connectivity, and all kinds of wireless networks should be scheduled uniformly to serve users. Users do not need to distinguish different RATs, as unified radio resources are provided for them. When users hand off between HetNets, seamless mobility could be achieved. Secondly, assurance of users' QoE for network services serves as the primary principle of user-centric resource management. Heterogeneous resources should be allocated dynamically according to user perceived quality in real time. Therefore, the fundamental principles related to user-centric wireless networks include ubiquitous and undifferentiated network connectivity and QoE assurance by effective resource and traffic management. However, how to design an integrated user-centric framework for heterogeneous wireless networks is still an open question.

Fortunately, the emergence of SDN proposes a possible solution for user-centric design for wireless networks. Meanwhile, network resource virtualization has emerged as a powerful technique for customized resource provisioning in wireless networks. Through virtualization based SDN approaches, user-centric design could be implemented. In this paper, we propose a user-centric framework for heterogeneous wireless networks based on SDN and virtualization. User agent corresponding to each user could be generated on soft access devices, which possess ability of connection information keeping and user situation awareness. Ubiquitous and undifferentiated network connectivity for users in heterogeneous wireless networks could be achieved by dynamic mitigation of user agent, and QoE assurance for users by flexible resources and traffic management could be achieved in the light of user situation awareness of user agent.

The remainder of this paper is organized as follows. We summarize the related work in Section 2. User-centric framework for heterogeneous wireless networks is described in Section 3. Ubiquitous and undifferentiated network connectivity and QoE assurance by flexible resources and traffic management are depicted in Sections 4 and 5, respectively. Performance of proposed user-centric framework and possible overhead is analyzed in Section 6, and the paper is concluded in Section 7.

2. Related Work

Generally, user-centricity is a key aspect of user-centric framework [2]. Researchers have introduced user-centric design to wireless networks in two aspects, such as user-centric radio resource management and user-centric service performance optimization. In the aspect of user-centric radio resource management, a user-centric adaptive clustering method [3] for coordinated multipoint transmission in dense cellular networks is described, and normalized outage capacity of each mobile station is maximized. A user-centric intercell interference coordination strategy [4] is proposed for downlink small cell networks. Each user selects the coordinating base stations (BSs) based on the relative distance between the home BS and the interfering BSs, and the dominant interference for each user is effectively identified and mitigated. A user-centric downlink cooperative transmission scheme [5] with orthogonal beamforming based limited feedback is proposed, and the percentage of users with satisfactory QoS demands is significantly increased. In the aspect of user-centric service performance optimization, user-centric QoE function [6] is modeled by a sigmoid function. Users' satisfaction on wireless services could be incorporated into the scheduler, and the average number of satisfied users is maximized. A QoE-driven user-centric solution for video on demand services [7] in urban vehicular network environments is introduced, and high QoE service level is provided to vehicle passengers. A user-centric mobile cloud computing service model [8] is presented, and the increasing demands from mobile users in terms of services diversity, user experience, security, and privacy could be met. Unfortunately, there is no work focusing on the user-centric design for heterogeneous wireless networks from resource management level to service performance optimization level. As heterogeneous wireless networks are built separately and have various radio resource formats, a user-centric framework will be a tough challenging issue.

On the other hand, to make the architecture of wireless networks more flexible, flattening, and programmable, the traditional wireless networks trend to combine with the concept of SDN. With the development of soft baseband and resource virtualization [9], this research field has attracted more and more attention in recent years. An SDN-enabled architecture for converged networks [10], which builds on the decoupling of data and control functionalities in the radio access network and control and forwarding functionalities in the core network, is proposed. Efficient resource management, QoS enforcement, and flexibility and scalability for future network evolution could be achieved. The authors present software defined access (SDA) [11], which introduces a novel logical control path across radio interfaces and up to mobile devices. Unlike SDN and SDWN (Software Defined Wireless Network), SDA can be deployed without changing network elements of radio access technologies. To deal with the increasing complexity in heterogeneous mobile networks (HMNs), authors believe that SDN based control is a promising approach to solve control problems in HMNs. An SDN based control framework named SoftMobile [12] to coordinate complex radio access in HMNs is proposed. In [13], all-SDN network architecture with hierarchical network control capabilities is advocated to allow for different grades of performance and complexity in offering core network services and provide service differentiation for 5G systems. In brief, SDN can be used to solve many challenging problems in wireless networks. Advanced wireless resources management in wireless networks, such as load balancing and coordination of inter-RAT basestation/APs, could be achieved through

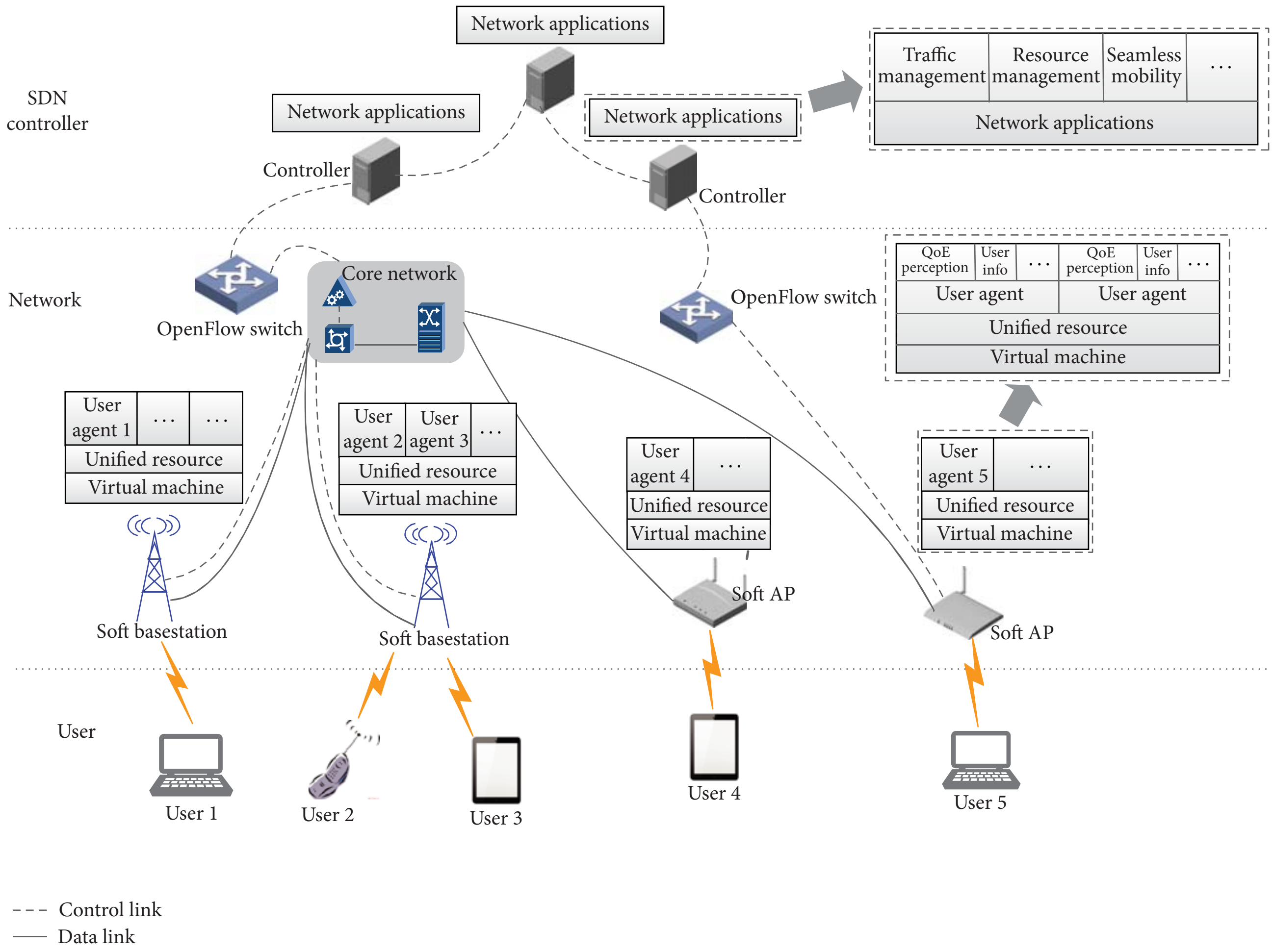


FIGURE 1: SDN based user-centric framework for wireless networks.

overall view of the controller. To operate wireless networks in a more efficient way, network resource virtualization may be a perfect method to work with SDN. Therefore, network frameworks coupling SDN with virtualization could be a pragmatic approach for user-centric design. Current wireless network virtualization researches, such as [14, 15], mainly focus on the sharing mechanism, including access infrastructure sharing and network sharing. In [16], the authors mention the virtualization of physical radio resources by adopting {base station index, time, and $frequency$} to abstract the radio resources. Authors illustrate the framework for combining SDN and wireless network virtualization and discuss the challenges for future study in [17]. However, no research work has attempted to design a user-centric framework by SDN, and several important issues should be focused on to reach this target. In this paper, we present a set of programming abstractions modeling the fundamental aspects of a wireless network, namely, user agent abstraction, user situation awareness, and resource and traffic management. The proposed abstractions hide away the implementation details of the underlying wireless technology and provide users with ubiquitous and undifferentiated network connectivity.

3. User-Centric Framework for Heterogeneous Wireless Networks

In order to accommodate the explosive mobile data traffic growth, current network operator may operate several wireless networks with different technologies simultaneously. Those heterogeneous networks are completely designed by network-centric approaches, built separately and mainly implemented by hardware which makes heterogeneous network management functions such as ubiquitous and undifferentiated network connectivity and QoE assurance for mobile users unprocurable. However, implementation of user-centric wireless networks needs a highly concentrated and flexible control plane to manage the heterogeneous underlying resources according to user's QoE. As depicted in Section 1, network resource virtualization technology and SDN will be two keys to resolve these issues. An SDN based user-centric framework for heterogeneous networks is depicted in Figure 1, and the network components of this proposed framework are composed of soft basestation/AP, virtual machine, user agent, OpenFlow switch, controller, and network applications.

Soft basestation/AP is software defined radio (SDR) based basestation/AP whose baseband part is implemented by software. It provides users programming interfaces by which user agents can obtain user state information, link state information, types of users' services, and so on. To implement the soft basestation/AP, we adopt soft baseband and soft MAC [18] technologies, so that eNodeB in LTE, basestation in WiMAX, and AP in WLAN could be realized by general purpose processor (GPP) and software radio peripheral (either universal or exclusive).

Virtual machine (VM) realizes the function of the network resource virtualization. It has the advantage of network status awareness which includes physical layer perception and network layer measurement. The information perceived in physical layer refers to signal strength, interference, spectrum usage, and so on. And the network layer information includes connectivity, throughput, bandwidth, delay, jitter, and packet loss rate. Leveraged by the information, the VM could provide a mapping from heterogeneous wireless resources to unified virtualized resource elements with uniform format and informs controller of the usage situation of these unified virtualized resources. We use a plugin mode [19] to develop resource virtualization modules for each RAT.

User agent is an agent for users operating on the soft basestation/AP. It holds the user information for a certain user who is connected to the network and calculates the QoE of network service for this user. It uses the underlying virtual resources according to the schedule instructions of controller. When a user switches from one basestation/AP to another, the corresponding user agent also migrates from original basestation/AP to the corresponding one. When the traffic management function requires traffic of one user transmitted through two different basestation/APs, a new user agent will generate in the collaboration basestation/AP, which means that there will be two user agents operating on two corresponding basestation/APs for one specific user. The user agent is implemented by maintaining a configuration table in soft basestation/AP.

OpenFlow switch is a switch supporting OpenFlow protocol. It communicates with controller via OpenFlow protocol and executes the control instructions issued by the controller, and traffic management can be carried out by dynamic routing policy.

Controller is the core of the user-centric framework. It has a global view of the underlying network and can obtain the entire virtual resources of the whole network and all the available virtual resources information from the VMs, as well as user state, link state, traffic state, and QoE information from user agents. Traffic control messages are generated by controller and sent to OpenFlow switches to manage the user's traffic. Resource control messages are also generated by controller and sent to user agents so as to delete user agent, generate user agent, and manage the virtual radio resource. In addition, the controller provides open APIs which could be used by network administrators to develop network management applications and implement user-centric network functions.

Network applications include user-centric applications operating on the controller, such as traffic management, resource management, and seamless mobility. These applications are implemented through open APIs provided by the controller. Therefore, they are easy to develop and modify without any hardware changes.

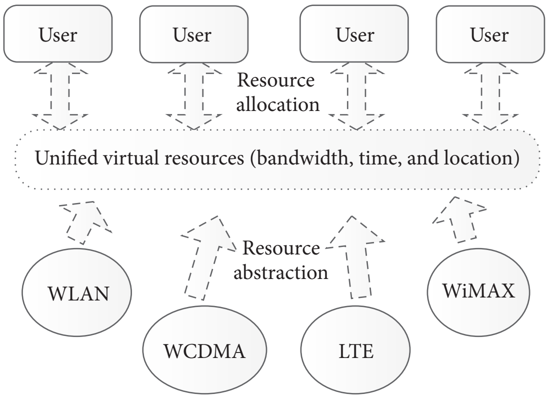


Figure 2: Resource abstraction and resource allocation in user-centric framework.

Through SDN and network resources virtualization, the proposed framework changes the network management methodology from network-centric to user-centric, which thoroughly refactors the traditional wireless heterogeneous networks to a more flexible and coadjutant network. As a result, assurance of user perceived network qualities such as ubiquitous and undifferentiated network connectivity and QoE assurance of network services could be achieved.

4. Ubiquitous and Undifferentiated Network Connectivity

User-centric framework is supposed to provide users with ubiquitous network connectivity, and all kinds of RATs should be scheduled uniformly to serve users in the form of collaboration. In the proposed user-centric framework, heterogeneous wireless resources are abstracted to unified virtualized resources. Different kinds of wireless access methods have different kinds of physical radio resources, including power, spectrum, time, space, and code. By resource abstraction, a virtual resource pool that maintains the available resources comes into being in the heterogeneous wireless networks. As shown in Figure 2, we abstract the physical radio resource as a 3D resource grid: bandwidth, time, and location [19]. This is a location based resources abstraction, and it represents, at a specific time and location, how many bandwidth resources could be used by a corresponding user. With this metaresource model, the wireless resources are supposed to be expressed in a format directly corresponding to user's requirements.

To support the ubiquitous network connectivity for users in HetNets, seamless mobility among different RATs is another significant issue that should be addressed. User-centric wireless networks is expected to give network users the ability to get undifferentiated connectivity to a certain point over the access network that best suits his or her current needs at any point in time. Seamless mobility is one fundamental function for our proposed user-centric framework. It is implemented by a network management application running on the controller and leveraged by the network resources virtualization technology. The introduction of user agent makes the connection status between user and soft basestation/AP controllable. User agent holds user information of multiple layers for a certain user which could help maintain the user's connection status. From user's perspective, an agent is a general basestation or AP that handles regular communication handshakes corresponding to this specific user. Just like virtual machines in data center can be migrated between different physical servers, user agents can migrate between different soft basestation/APs. If the user agent migrates as fast as the corresponding user's movement, a virtual persistent connection can be maintained for this user. That is, when one user switches from one soft basestation/AP to another, time delay caused by handover just includes time delay of break and reconnection in physical layer rather than time delay of multilayers such as physical layer, MAC layer, and network layer. Therefore, dynamic migration of user agent can be utilized to realize seamless mobility.

Specifically, resources virtualization and virtual machine migration technologies make unified radio resources and seamless mobility easy to implement. As previously described, heterogeneous wireless resources are abstracted into unified virtualized resource in a location based virtual resource format. Meanwhile, the controller, who has a global view of the virtual resources, can acquire user's location information, physical layer information such as signal strength, signal-noise ratio (SNR), and network layer information such as link throughput, delay, and packet loss rate from VM. Hence, the controller can perceive user's movement and predict the moving direction of mobile user. Then a variety of handover decisions algorithm which makes switching decisions based on different strategies [20] could be applied. If the handover conditions are met, the controller will migrate the user's agent from the source basestation/AP to the destination basestation/AP that the user is moving to, and seamless mobility of users could be achieved.

5. QoE Assurance by Flexible Resources and Traffic Management

The proposed user-centric framework aims at refactoring wireless mobile networks, and network management methodology is changed from network-centric to user-centric. Through migration of user agent, one virtual persistent and undifferentiated connection can be maintained for one specific user, which is the foundation of user-centric framework. Further, as user agent operating in soft basestation/AP is a virtual proxy of user and the user situation of mobile users can be aware, the controller can manage the traffic and virtualized resource aiming at best QoE for users. Hence, user-centric network functions such as flexible resource management and fine-grained traffic management for users could be achieved, and user-centric framework from resource management level to service performance optimization level could be implemented.

5.1. User Situation Awareness. In the proposed user-centric framework, user agent in soft basestation/AP calculates the QoE of network services for users and proactively sends this user's situation to the controller. By putting QoE as the optimization objective of network management, network design philosophy has been transferred to user-centric. Controller executes network operations including virtual resources management and traffic management according to a certain criterion to enhance user's QoE. This is a negative feedback procedure, and different strategies could be adopted by controller to realize QoE assurance of different granularity.

The goal of user situation awareness is to acquire the QoE information related to the specific user. Since different types of network services have different characteristics, in order to perceive user's QoE more accurately, mean objective score (MOS) is used as a uniform metric, while different wireless services are built by different QoE perception models. For example, for best effort (BE) services, we explore the role of human cognition and the psychophysics method in QoE assessment and establish a model in terms of the service information, complete time, and bandwidth to figure out QoE. For mobile video services, acceptability based quality assessment methods are adopted to perceive the QoE [21]. For different mobile video services, different feature vectors are chosen, and corresponding mapping approaches are adopted to calculate the MOS of video services.

In order to utilize the radio resource more effectively and enhance the user perceived service quality, user-centric function takes advantage of the broadcast nature of the wireless media by means of cooperative approaches such as flexible resource management and fine-grained traffic management.

5.2. Flexible Resource Management. Flexible resource management is the main user-centric function for heterogeneous radio resources, which manages heterogeneous radio resources of wireless networks by dynamic tuning the radio resource in soft basestation/AP, with QoE being the objective.

The virtualization and abstraction of heterogeneous wireless resources make the management of underlying physical resources as flexible as possible. These unified virtual resources are allocated to mobile users according to their requirements by optimization algorithms, such as auction algorithm, games theory, and water-filling algorithm. Resource management application is implemented in the controller as shown in Figure 3, which mainly consists of three components: status collection, decision making, and issuing instruction. Before making any decisions, controller will collect and update the underlying network status information, which includes the available virtual resources and user's QoE. Then, controller will execute decision making component which determines the virtual resource allocation scheme in each time slice. As soon as the allocation scheme is determined, controller will issue control instructions to user

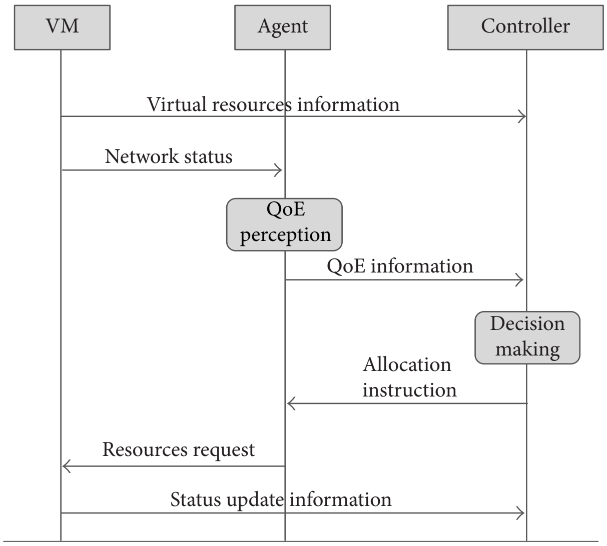


Figure 3: Flexible resources management in user-centric framework.

agent, who will take up underlying virtual resources held by VM according to the control instructions of controller.

5.3. Fine-Grained Traffic Management. Fine-grained traffic management is the user-centric function for network traffic, which manages the network traffic of HetNets by dynamically adjusting the traffic path and flexible basestation/AP selection with QoE as the objective. Traffic management is a comprehensive network application composed of flow table management, user agent generation/deletion, and data stream merging schemes. The controller can find the optimal routing path for user's traffic according to different strategies. OpenFlow switches in the network consist of one or more flow tables which perform packet lookups and forwarding. The controller communicates with the switch via the OpenFlow protocol to add, update, and delete flow tables reactively or proactively. Meanwhile, user agent receives the user data from OpenFlow switches and transfers it to user equipment through the underlying hardware infrastructure using corresponding packet format of different RATs. This mechanism makes the multicell cooperative transmission based traffic management possible. That is, when a user is in the overlapping coverage area of multiple basestation/APs, controller can issue instructions to generate multiple agents on these basestation/APs for this user, and data stream of services could be transmitted to user equipment from multiple paths. Data stream merging schemes are adopted by user equipment to combine streams transmitted by different paths. In user-centric wireless networking, network user's equipment are expected to support multiple RATs and treat the multiradio HetNets as a single network, which creates rich opportunities for intelligently combining and aggregating capacity across these RATs. That is, multiple RATs serve users in a collaborative manner to satisfy the requirements and preferences of network users.

As Figure 4 shows, eNodeB *A* is heavily loaded while AP *A* and AP *B* serve few users. UE 1, who is served by eNodeB *A* with a bad link state, is located at the overlapping area. In the user-centric design, UE 1 can support multiple RATs and establish a connection with AP *A* to offload partial traffic from eNodeB *A*. This procedure should be completely automatic and bring no interruption for network services.

6. Performance and Overhead Analysis

To prove the performance of the propose user-centric framework in heterogeneous wireless networks, a test bed is set up according to Figure 5. It is consisted of LTE (based on 3GPP release 10) and WLAN. Specifically, the LTE network consists of a core network and two soft basestations, which are implemented based on OpenAirInterface project [22]. The WLAN is implemented based on our existing SWAN experimental network [23]. Ten soft APs are deployed in this test bed, and each AP is equipped with wireless NICs working specifically in channel 6 (2.4 GHz band). The operating system of these APs is the OpenWrt "Backfire 10.03.1" release. Both basestations and APs are implemented by software on general purpose processor (GPP) platform and software radio peripheral. The network controller is a ThinkServer RD640 server equipped with a six-core Xeon E5-2620 CPU and 16 GB of RAM. iPerf is used for synthetic traffic generation.

Figure 6 shows the throughput over time of a user in the proposed framework and traditional heterogeneous network. For the traditional heterogeneous network, the throughput drops to zero for several seconds due to the user's mobility. However, in user-centric framework, users will perceive a stable connectivity in the handover process and the throughput curve remains uninterrupted with the user's mobility. As the user agent migrates with the corresponding user's movement, user can always see the consistent user agent connectivity regardless of the associated physical AP or basestation. The connectivity of users in user-centric based heterogeneous wireless networks will not break off. Therefore, the proposed framework can provide mobile users with continuous and consistent connectivity.

To demonstrate the performance of QoE assurance based on flexible resources and traffic management, user satisfaction, which denotes the ratio of the number of users who are satisfied with their QoE to the total number of users, is defined as

$$\text{USER SATISFACTION} = \frac{\text{USER NUM}|_{\text{MOS}>3.5}}{\text{TOTAL NUM}}. \quad (1)$$

In our test bed, users with MOS value more than 3.5 are regarded as users satisfied with their QoE of services. Two most major services for users in wireless networks such as video streaming services and best effort services are considered in this test. Figure 7 depicted the user satisfaction of the proposed method, cross-layer QoE-aware based method [24], and proportional fairness based method. HetNets is composed of LTE and WLAN in this test. It is shown that the proposed method always maintains a high and stabilized user satisfaction with the increasing number of users, while the user satisfaction of QoE-aware based algorithm declines

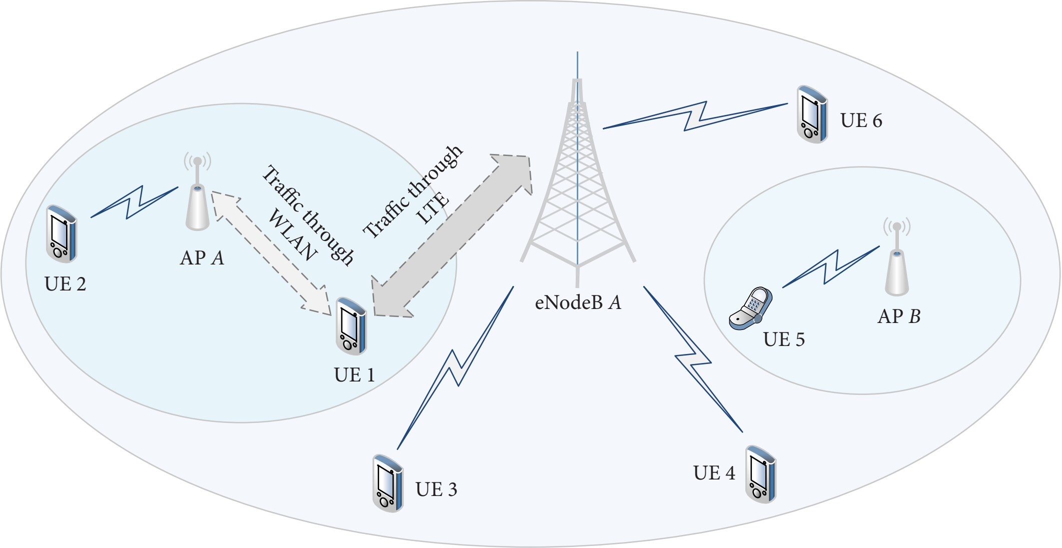


FIGURE 4: LTE cooperation with WLAN in user-centric framework.

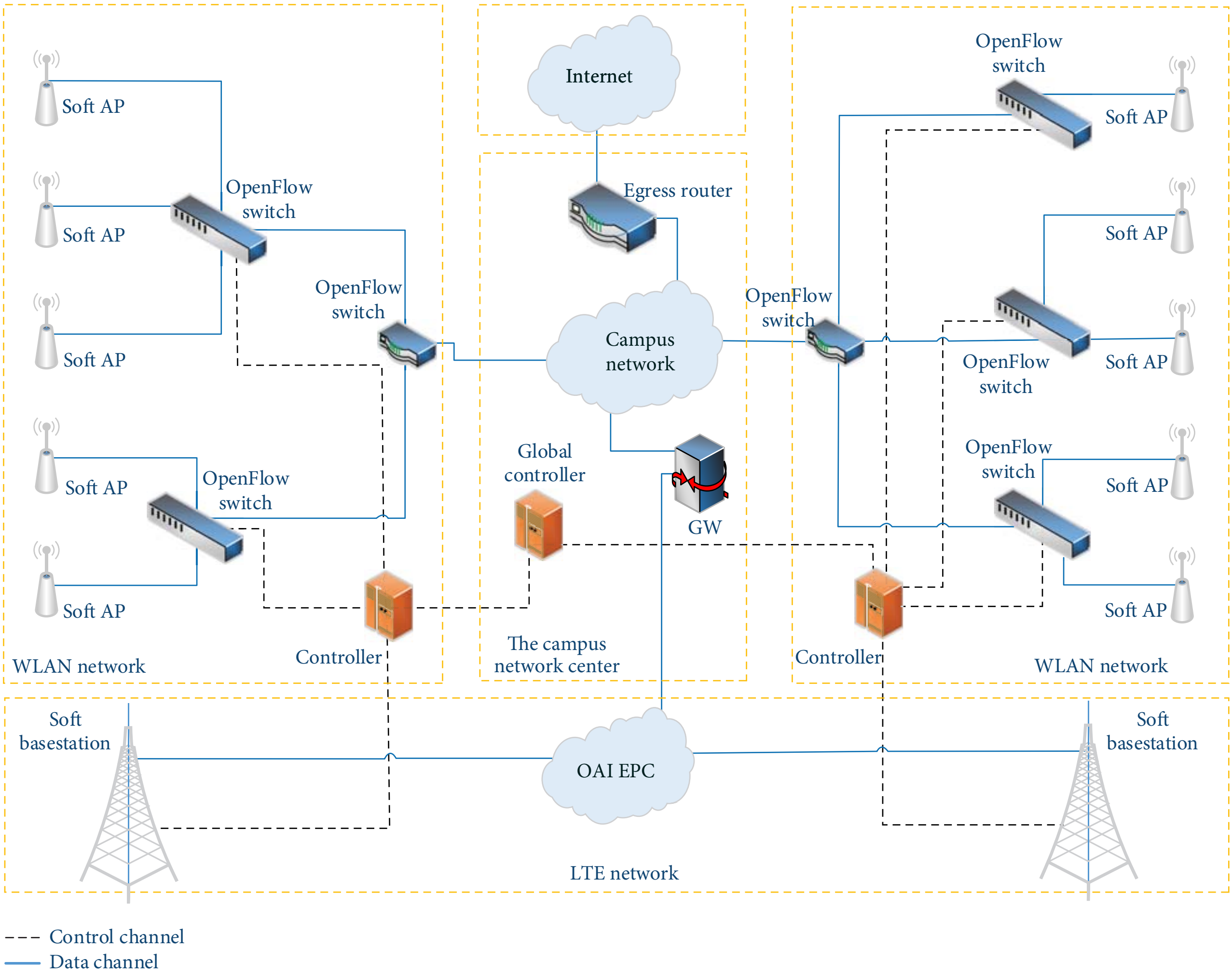


FIGURE 5: The test bed for heterogeneous wireless networks.

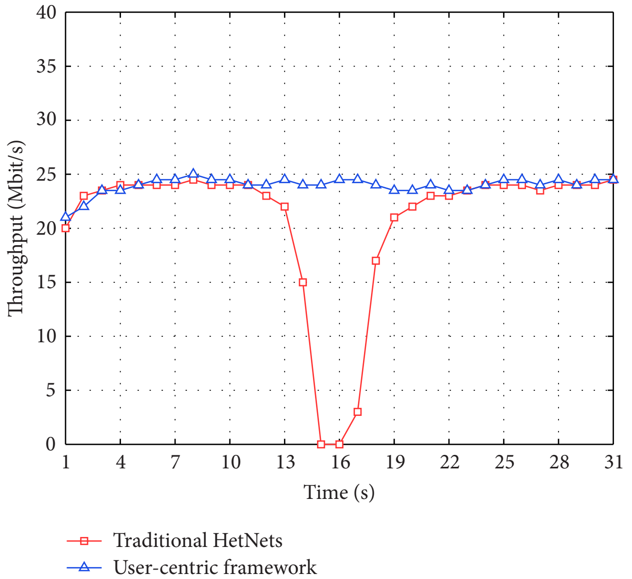


Figure 6: Seamless mobility for user-centric framework in heterogeneous wireless networks.

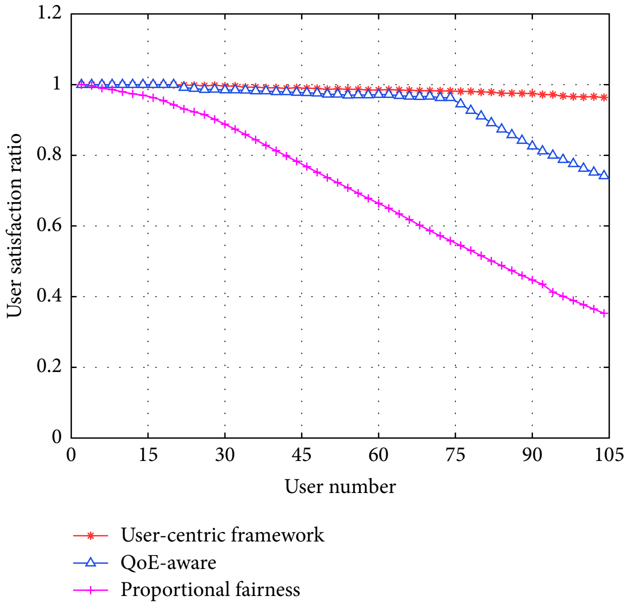


Figure 7: The user satisfaction of different resource allocation methods.

sharply when the user number exceeds a certain value when network capacity of LTE or WLAN is achieved. Besides, the curve of proportional fairness based algorithm gradually declines with the increasing number of users. Therefore, the proposed method can ensure users' QoE by flexible resources and traffic management.

Our proposed framework is built by the QoE metrics, namely, the intention to achieve more profit for users from the perspective of user-centric, while at the same time sharing the limited radio resources more effectively. Hence, algorithms and strategies proposed under this user-centric framework may be evaluated by QoE metrics in the future. Due to the user-centric framework design in wireless networks, QoE metrics could contribute a more dynamic behavior, while some costs that might be brought by SDN and resource virtualization, such as signaling overhead and control delay, are also investigated in this framework.

Signaling and Processing Overhead. With a global network awareness and management, the controller should perform dynamic network status awareness and virtualized resource management, which will generate the amount of signaling and processing overhead in controller. To solve this defect, hierarchical control plane could be adopted in this framework, by which domain controller and global controller appear to specialize in handling various tasks. Besides, efficient signaling compression and aggregation schemes could be used in this user-centric framework.

Control Delay. SDN makes a separation between the control plane and data plane in heterogeneous wireless networks. The virtualization of radio resources is accomplished on the data plane while the virtualized resources allocation is implemented on the control plane. As the wireless networks are highly time sensitive, the control signaling interaction between control plane and data plane may cause delays. Therefore, we argue that the data plane could be realized by a centralized base band processing pool as presented in CRAN, which will considerably shorten the control delay between control plane and data plane.

7. Conclusion

Network-centric framework in the dense HetNets scenario leads to a contradiction between trenchant user requirements and difficult management of the inflexible wireless networks. It is not able to provide network users with ubiquitous and undifferentiated network connectivity and QoE assurance. To address this issue, user-centric framework could be an effective solution. In this paper, we analyze the challenges and requirements for user-centric based heterogeneous wireless networks. Then, we propose a user-centric framework based on network resource virtualization technology and SDN for heterogeneous wireless networks. Ubiquitous and undifferentiated network connectivity and QoE assurance by flexible resources and traffic management are presented to implement the two fundamental characteristics of user-centricity. In addition, possible overheads of user-centric framework such as signaling overhead and control delay are also analyzed. As part of our ongoing work for the architecture of future wireless networks, we believe that this study can shed light on how we use SDN and virtualization technology to refactor future wireless network architecture and promote the research for 5G and beyond networks.

Competing Interests

The authors declare that they have no competing interests.

Acknowledgments

This work is supported by Beijing Municipal Science and Technology Commission Research Fund Project no. D151100000115002 and the WLAN achievement transformation based on SDN Project of Beijing Municipal Commission of Education (Grant no. 201501001).

References

[1] D. Gilstrap, *Ericsson Mobility Report*, Ericsson, 2013.

[2] B. A. A. Nunes, M. A. S. Santos, B. T. De Oliveira, C. B. Margi, K. Obraczka, and T. Turletti, "Software-defined-networking-enabled capacity sharing in user-centric networks," *IEEE Communications Magazine*, vol. 52, no. 9, pp. 28–36, 2014.

[3] V. Garcia, Y. Zhou, and J. Shi, "Coordinated multipoint transmission in dense cellular networks with user-centric adaptive clustering," *IEEE Transactions on Wireless Communications*, vol. 13, no. 8, pp. 4297–4308, 2014.

[4] C. Li, J. Zhang, M. Haenggi, and K. B. Letaief, "User-centric intercell interference nulling for downlink small cell networks," *IEEE Transactions on Communications*, vol. 63, no. 4, pp. 1419–1431, 2015.

[5] D. Su and C. Yang, "User-centric downlink cooperative transmission with orthogonal beamforming based limited feedback," *IEEE Transactions on Communications*, vol. 63, no. 8, pp. 2996–3007, 2015.

[6] G. Lee, H. Kim, Y. Cho, and S.-H. Lee, "QoE-aware scheduling for sigmoid optimization in wireless networks," *IEEE Communications Letters*, vol. 18, no. 11, pp. 1995–1998, 2014.

[7] C. Xu, F. Zhao, J. Guan, H. Zhang, and G.-M. Muntean, "QoE-driven user-centric VoD services in urban multihomed P2P-based vehicular networks," *IEEE Transactions on Vehicular Technology*, vol. 62, no. 5, pp. 2273–2289, 2013.

[8] D. Huang, T. Xing, and H. Wu, "Mobile cloud computing service models: a user-centric approach," *IEEE Network*, vol. 27, no. 5, pp. 6–11, 2013.

[9] Q. Yang, X. Li, H. Yao et al., "Bigstation: enabling scalable real-time signal processing in large mu-mimo systems," *Proceedings of the ACM SIGCOMM Computer Communication Review (SIGCOMM '13)*, vol. 43, no. 4, pp. 399–410, 2013.

[10] W. Tan, J. Zhang, C. Peng, B. Xia, and Y. Kou, "SDN-enabled converged networks," *IEEE Wireless Communications*, vol. 21, no. 6, pp. 79–85, 2014.

[11] V. Sagar, R. Chandramouli, and K. P. Subbalakshmi, "Software defined access for HetNets," *IEEE Communications Magazine*, vol. 54, no. 1, pp. 84–89, 2016.

[12] T. Chen, H. Zhang, X. Chen, and O. Tirkkonen, "SoftMobile: control evolution for future heterogeneous mobile networks," *IEEE Wireless Communications*, vol. 21, no. 6, pp. 70–78, 2014.

[13] V. Yazici, U. C. Kozat, and M. O. Sunay, "A new control plane for 5G network architecture with a case study on unified handoff, mobility, and routing management," *IEEE Communications Magazine*, vol. 52, no. 11, pp. 76–85, 2014.

[14] R. Kokku, R. Mahindra, H. Zhang, and S. Rangarajan, "NVS: a substrate for virtualizing wireless resources in cellular networks," *IEEE/ACM Transactions on Networking*, vol. 20, no. 5, pp. 1333–1346, 2012.

[15] C. J. Bernardos, A. De La Oliva, P. Serrano et al., "An architecture for software defined wireless networking," *IEEE Wireless Communications*, vol. 21, no. 3, pp. 52–61, 2014.

[16] A. Gudipati, D. Perry, L. E. Li, and S. Katti, "Softran: software defined radio access network," in *Proceedings of the 2nd ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking (HotSDN '13)*, pp. 25–30, August 2013.

[17] B. Cao, F. He, Y. Li, C. Wang, and W. Lang, "Software defined virtual wireless network: framework and challenges," *IEEE Network*, vol. 29, no. 4, pp. 6–12, 2015.

[18] MAC80211, https://wireless.wiki.kernel.org/en/developers/Documentation/mac80211.

[19] L. Wang, Z. Lu, X. Wen, and W. Guan, "Converged management in heterogeneous wireless networks based on resource virtualization," *Mobile Networks and Applications*, vol. 20, no. 1, pp. 53–61, 2015.

[20] A. Ahmed, L. M. Boulahia, and D. Gaïti, "Enabling vertical handover decisions in heterogeneous wireless networks: a state-of-the-art and a classification," *IEEE Communications Surveys and Tutorials*, vol. 16, no. 2, pp. 776–811, 2014.

[21] W. Song and D. W. Tjondronegoro, "Acceptability-based QoE models for mobile video," *IEEE Transactions on Multimedia*, vol. 16, no. 3, pp. 738–750, 2014.

[22] OpenAirInterface Project, http://www.openairinterface.org/.

[23] T. Lei, Z. Lu, X. Wen, X. Zhao, and L. Wang, "SWAN: an SDN based campus WLAN framework," in *Proceedings of the 4th International Conference on Wireless Communications, Vehicular Technology, Information Theory and Aerospace and Electronic Systems (VITAE '14)*, pp. 1–5, May 2014.

[24] M. Rugelj, U. Sedlar, M. Volk, J. Sterle, M. Hajdinjak, and A. Kos, "Novel cross-layer QoE-aware radio resource allocation algorithms in multiuser OFDMA systems," *IEEE Transactions on Communications*, vol. 62, no. 9, pp. 3196–3208, 2014.

Hindawi Publishing Corporation
Mobile Information Systems
Volume 2016, Article ID 6013158, 11 pages
http://dx.doi.org/10.1155/2016/6013158

Research Article

Performance Evaluation of Moving Small-Cell Network with Proactive Cache

Young Min Kwon,[1] Syed Tariq Shah,[1] JaeSheung Shin,[2] Ae-Soon Park,[2] and Min Young Chung[1]

[1] *College of Information and Communication Engineering, Sungkyunkwan University, 2066 Seobu-Ro, Jangan-Gu, Suwon, Gyeonggi-Do 16419, Republic of Korea*
[2] *Mobile Access Research Division, Electronics and Telecommunications Research Institute, 138 Gajeongno, Yuseong-gu, Daejeon 34129, Republic of Korea*

Correspondence should be addressed to Min Young Chung; mychung@skku.edu

Received 21 January 2016; Revised 20 June 2016; Accepted 21 June 2016

Academic Editor: Juan C. Cano

Due to rapid growth in mobile traffic, mobile network operators (MNOs) are considering the deployment of moving small-cells (mSCs). mSC is a user-centric network which provides voice and data services during mobility. mSC can receive and forward data traffic via wireless backhaul and sidehaul links. In addition, due to the predictive nature of users demand, mSCs can proactively cache the predicted contents in off-peak-traffic periods. Due to these characteristics, MNOs consider mSCs as a cost-efficient solution to not only enhance the system capacity but also provide guaranteed quality of service (QoS) requirements to moving user equipment (UE) in peak-traffic periods. In this paper, we conduct extensive system level simulations to analyze the performance of mSCs with varying cache size and content popularity and their effect on wireless backhaul load. The performance evaluation confirms that the QoS of moving small-cell UE (mSUE) notably improves by using mSCs together with proactive caching. We also show that the effective use of proactive cache significantly reduces the wireless backhaul load and increases the overall network capacity.

1. Introduction

Due to the increasing number of smart phone devices and data services, the users demand for mobile data traffic has also increased. Global mobile traffic will nearly increase tenfold until 2018 [1]. To accommodate this emerging demand of data traffic, mobile network operators (MNOs) have already adopted advanced communication techniques such as orthogonal frequency division multiple access (OFDMA), multiple input multiple output (MIMO), and carrier aggregation (CA). It is possible to make the spectrum efficiency reach its theoretical limit in 4G mobile network by using these technologies. However, the networks only implementing these advanced radio access and transmission technologies will not be able to accommodate the tremendous increment of mobile traffic and it may exhaust the available system capacity of 4G mobile networks. Thus, MNOs have considered heterogeneous networks (HetNets) in order to continuously improve the systems capacity by adding more base stations [2, 3].

The HetNet terminology indicates that various types of fixed small-cells (fSCs) such as pico- and femtocell coexist in a macrocell. fSCs can share the traffic overload of macrocell by providing mobile services to densely populated areas such as hotspots [4]. However, fSCs using wired backhaul have drawback in terms of signaling overheard, infrastructure cost, and mobility [5]. When many fSCs densely exist in cellular networks, frequent handovers occur between macrocell and fSCs [6]. For successful handover, both base stations of macrocell and fSCs should exchange control messages via wired backhaul comprised of several network entities [7]. Thus, dense deployment of fSCs increases signaling load in the wired backhaul. Secondly, existing fSCs require wired backhauls such as optical fiber or coaxial cable, in order to connect them to the core network. Laying these wired backhaul is not a very cost-effective solution for MNOs. Moreover, fSCs using a wired backhaul cannot consistently provide wireless broadband services to users that ride public transportation vehicles [8]. Recently, working group (WG) of 3GPP standardization has investigated the moving cell utilizing the wireless backhaul as a solution to overcome the limitations of fSCs [9].

In this paper, we introduce the concept of moving small-cell (mSC) with various transmission paths, that is, wireless backhaul, sidehaul, and caching transmission. mSCs are user-centric networks that autonomously establish connections between users and provide the voice and data services while moving [10]. mSCs communicate with their respective MBSs via wireless backhaul links. mSCs can also exchange data through wireless sidehaul links among neighboring mSCs. Due to predictable nature of users, the nodes in the network track can learn and construct the users' demand profiles in order to predict their future requests effectively. Thus, in the proposed mSC network, each mSC has a storage capability to cache the predicted contents. The proposed caching mechanism is proactive in principle and it aims to anticipate users demands. It can reduce the backhaul load by saving the scarce frequency resources. Due to these unique characteristics, mSC has several advantages over other fSCs. By supporting group handover, mSCs can reduce both signaling overhead and handover failure probability [11]. Since wireless backhaul and sidehaul links do not require any additional deployment cost, mSCs can become a cost-efficient solution to enhance the systems capacity [12]. Furthermore, the proposed proactive caching mechanism used in mSCs can not only reduce the traffic load of wireless backhaul link but also guarantee quality of service (QoS) performance in peak-traffic hours [13, 14].

The deployment of mSCs can enhance system capacity and accommodate the increasing mobile traffic with reasonable cost. Instead of deploying new fSCs, mSCs can be utilized as a cost-effective solution to solve the temporary hotspot issues. Although mSCs have many advantages in terms of traffic distribution and system capacity, their performance is limited due to cotier interference among neighboring mSCs. Since mSCs accommodate all the data traffic of wireless backhaul link, wireless sidehaul link, and proactive content cache, it is obvious that the performance of mSC is affected by ratio between data traffic delivered via these various links. Thus, we have developed and conducted extensive system level simulations to analyze the effect of mSCs with proactive caching enabled in a multitier HetNet environment.

Contributions. System level simulation is one of the most useful methodologies to analyze the performance of various network scenarios [15]. A preliminary version of this paper appears in the 8th ACM International Conference on Ubiquitous Information Management and Communication (IMCOM), 2014 [16]. In this study, we first highlight the challenges associated with mSCs deployment in multitier HetNets scenarios. Then, in order to exploit the advantages of mSCs and proactive caching, we evaluate and compare the performance of mSCs in different multitier HetNet scenarios. We show the relation between contents popularity, cache size, and operating modes and their positive effects on overall network performance.

The rest of the paper is organized as follows. Section 2 presents the previous studies related to mSCs and proactive caching. In Section 3, we introduce the proposed mSC network, its architecture, and proactive caching mechanism used. Section 4 contains the detailed performance evaluation of proposed mSC network and Section 5 provides the conclusion of this paper.

2. Related Works

Due to unprecedented growth in mobile data traffic, network densification and modification in its current architecture are inevitable. In order to maximize the reuse of available frequency spectrum, introducing HetNets is one of the key solutions. HetNets can accommodate the growing demand of data traffic by deploying more small-cells in a given area [2, 17, 18]. In [19], Dhillon and others have proposed a tractable model for a K-tiers downlink HetNet. It shows that in an ideal HetNet scenario, beside severe interference, the network densification can still significantly enhance the overall network capacity. In order to provide better and reliable network services to moving users, the use of mSCs has been proposed, studied, and evaluated in [20–24].

The authors in [20] have shown that, in a coverage limited scenario the use of coordinated and cooperative relays in public vehicles can significantly improve the network experience of on-board moving users. In [8, 21–23], Sui and others have studied performance of moving relay node (MRN), which is a type of mSCs, in cellular networks. MRNs are deployed in public transportation vehicles such as trains, trams, and buses in order to provide wireless broadband services to moving UE. Since MRN uses wireless backhaul link to connect to MBS, it can reduce the cost of wired backhaul link. In addition, by supporting group handover of all on-board UE, MRN can significantly reduce the signaling overhead and probability of handover failure. Compared to MBS, MRN is very close to its UE; therefore it can enhance the signal quality of the respective UE in access link. However, the performance of MRN mainly depends on the capacity of wireless backhaul link [21, 22]. Since the capacity of wireless backhaul link is normally limited, it is difficult to increase the overall network capacity by deploying large number of MRNs significantly.

The ability to predict user demands and recent developments in context awareness and data storage has enabled the future networks to proactively cache the popular contents in advance [25–28]. The proactive caching technique in small-cells will not only reduce the backhaul load but also guarantee the QoS requirements in peak-traffic periods. In [25], Tadrous and others have studied the concept of proactive resource allocation by utilizing the predictability of user behavior for load balancing. Authors in [26] have proposed the idea of femtocaching in fSCs with very limited backhaul bandwidth and large storage capacity. Authors in [27] have studied the asymptotic scaling laws of caching in D2D communications. In their proposed distributed caching scheme, users store the popular contents and forward them to

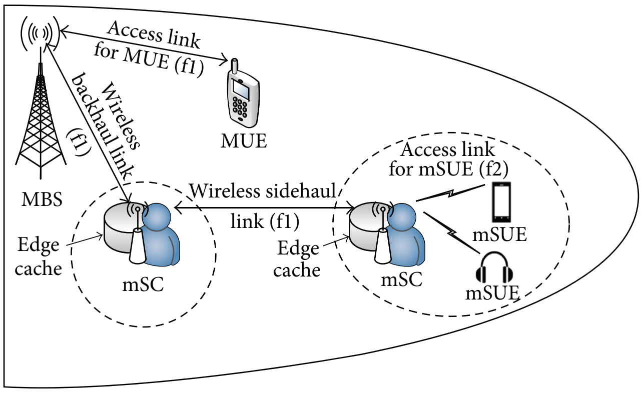


Figure 1: Moving small-cell network.

other users by using D2D communications. Bastug and others in [28] have examined two cases of proactive caching. First, in order to reduce the backhaul load, they have proposed a mechanism, which proactively caches the popular files in off-peak hours (e.g., at night) proactively. In second case, based on the social structure of the network, the proposed scheme predicts the set of potential users who can proactively cache and distribute the popular contents utilizing D2D communications.

Nonetheless, these studies on proactive caching have only considered the fSCs (pico- and femtocells) which usually have wired backhauls and do not have any backhaul bandwidth constraint. Moreover, they also rarely consider the mobility of either small-cells (picocells and femtocells) or users (D2D). These key aspects are the motivation behind this paper and the aim of this is to study the role of proactive caching in mSCs.

3. Proposed Moving Small-Cell Network with Proactive Cache

3.1. Network Architecture. The proposed mSC network consists of four network entities, MBS, macrocell UE (MUE), mSC, and moving small-cell UE (mSUE) as shown in Figure 1. MBS in mSC network provides wireless access link and backhaul link connections to its serving MUE and mSCs, respectively. Each mSC is a moving small-cell, which provides wireless broadband services to its serving mSUE in access links. To communicate between mSCs directly, mSCs can also establish wireless sidehaul connections with their neighboring mSCs. Based on measurement information, the MBS is also responsible for radio resource management of both wireless backhaul and sidehaul links of mSCs. Furthermore, in our proposed mSC network, each mSC has the ability to cache popular contents. If mSUE requests contents that are already stored in the cache of its connected mSC, the mSC directly sends the contents to its mSUE. More detail on proactive caching is given in the next section.

As discussed earlier, due to wireless backhaul and sidehaul connectivity, mPCs can be deployed on moving vehicles to provide enhanced network services to moving UE. It is obvious that, instead of deploying large number of fSC, mSCs are the cost-efficient technique to serve moving UE and increase the overall network capacity. In order to avoid severe interference between MUE and mSUE, both MBSs and mSCs in the proposed scheme use different frequency bands of 2.0 GHz and 3.5 GHz in their access links, respectively. Figure 2 shows the proposed channels and frequencies assignment scheme for wireless backhaul/sidehaul and access links of mSCs, MUE, and mSUE, respectively. In mSC network, in-band full duplex transmission may be used for wireless backhaul link. Thus, for wireless backhaul transmissions, mSCs share the same radio resources of uplink and downlink in 2 GHz frequency band with MUE. Furthermore, mSCs also perform in-band half-duplex transmission for wireless sidehaul links, where they reuse the uplink radio resources of mSC backhaul and MUE in 2 GHz frequency band. Unlike MUE, mSUE is very close to the serving mSCs; thus the transmit power of mSC is relatively lower than MBS.

3.2. Proposed Proactive Caching Scheme for mSC Network. It is mentioned earlier in this paper that preloading and proactive caching can significantly reduce the traffic load on wireless backhaul link and conserve the scarce radio resources. The key issues of proactive caching are methods to decide caching data and an efficient mechanism to transmit the selected data (preloading) [29]. This paper focuses on the second key issue of cache preloading. We assume that, based on collaborative filtering (CF) tools [30], the MBS can effectively decide the popularity of the contents such as video contents (e.g., TV series and advertisements), web contents (e.g., daily news, blogs, and digests), and software update files (e.g., software drivers and patches) [31]. These contents are usually time-insensitive and available long before their scheduled publishing time. The effect on time-sensitive contents has not been evaluated in this paper; it is because we assume that MBSs transmit the selected contents to their respective mSCs in off-peak period (e.g., night time). In other words, the cache of mSCs in our proposed scheme is only updated in low traffic hours when the traffic load on backhaul link is very low [28]. In order to continuously update the cache with time-sensitive popular contents, a full-time dedicated backhaul link is required. However, due to scarce availability of the radio resources, it is not feasible to fully dedicate certain backhaul resources only for cache management.

In order to make the preloading scheme more efficient, the MBS transmits the popular contents to mSCs in two possible modes: broadcasting and multicasting. If the content files are equally popular among all mSCs in the network, the MBS will broadcast the selected contents to all mSCs in the network. Similarly, if different content files are popular among different mSCs, the MBS will make groups of mSCs with same interest and it will multicast the desired contents to each particular group. Furthermore, in multicast mode mSCs of one group can exchange their cache contents with nearest neighboring mSC of other groups via sidehaul link. In other words, if the requested contents are available in neighboring mSCs, the MBS will provide the necessary information (mSC ID, radio resources for sidehaul, and so on) of that particular mSC in order to establish sidehaul link. In our proposed

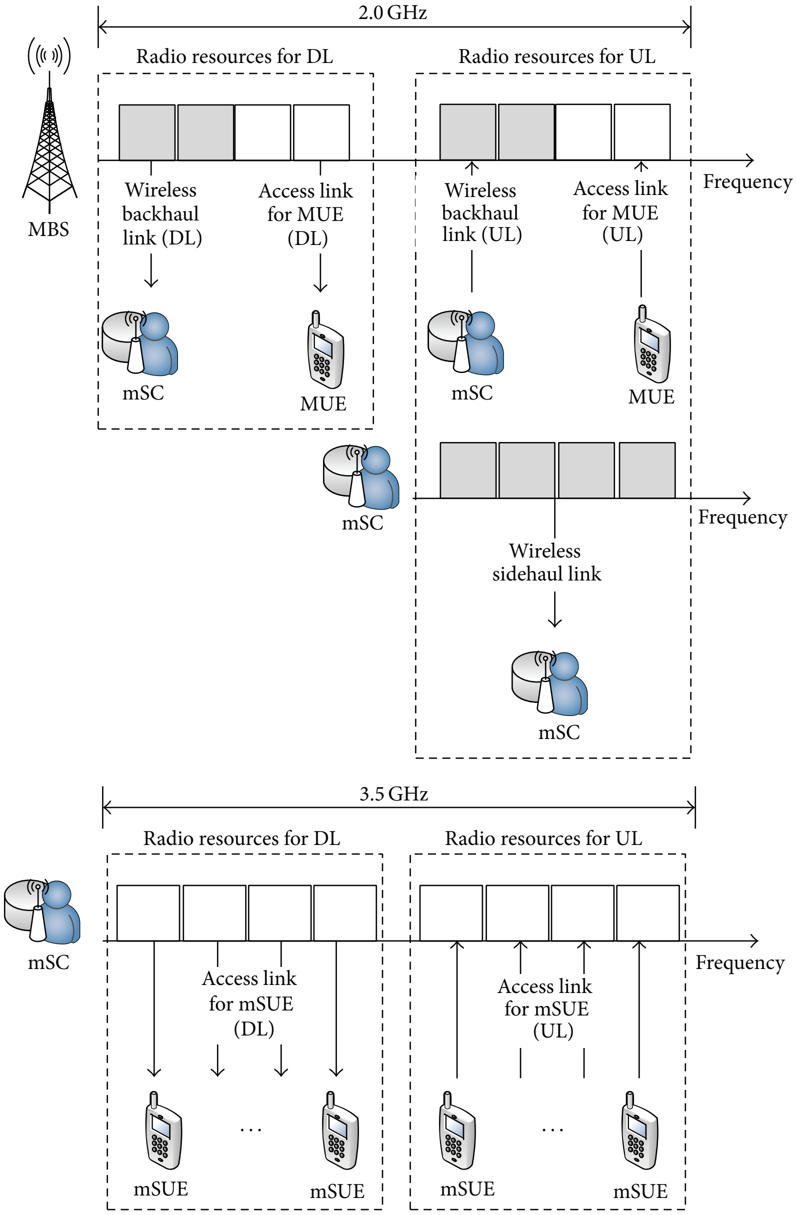


Figure 2: Channel assignment in mSC network.

network, mSCs can only establish sidehaul connection with neighboring mSCs that are located in the radius of 200 meters. Since, sidehaul links reuse the uplink frequencies of both mSCs and MUE, they can significantly reduce the backhaul traffic load. Note that the aim of our proposed scheme is to evaluate the performance of a fully loaded MSC network with active sidehaul links and proactive cache, under the constraint of limited wireless backhaul capacity. Therefore, in our proposed network model, we have considered that the number of mSCs in each macrocell and the number of mSUE pieces in each mSC are uniform and fixed. The aim of such network model is to find the upper bound of network capacity. Consequently, due to these considerations the traffic conditions of an mSC in our proposed network do not vary over time and the resource allocation is static. Figure 3 shows the proposed preloading scheme, where, during off-peak period, the backhaul bandwidth is divided into two parts, one for reactive backhaul traffic and the second for proactive broadcast/multicast caching traffic.

In our proposed mSC network, the network performance depends on three different factors: content popularity distribution, cache size of mSC, and the number of multicasting groups. In this paper, popularity distributions are obtained from ZipF (α) distribution [32]. It has been shown in [33, 34] that the global content popularity usually follows the ZipF distribution. It is also shown in [34] that a simple model for an independent request stream following a ZipF distribution is sufficient to capture certain asymptotic properties observed at proactive caches (such as web proxies). Another reason for using ZipF distribution is its simplicity;

Table 1: Simulation parameters.

Parameter	Macrocell	Moving small-cell
Cell layout	Hexagonal grid, 3 sectors per site	Uniform random distribution
Radius of cell	166 m (ISD = $3R$ = 500 m)	10 m
Number of cells	7 sites	20~100
Access link		
Carrier freq.	2 GHz	3.5 GHz
Bandwidth	10 MHz	10 MHz
Tx power	46 dBm (downlink) 23 dBm (uplink)	23 dBm (downlink) 23 dBm (uplink)
Wireless BH/SH link		
Carrier freq.	—	2 GHz
Bandwidth	—	10 MHz
Tx power	—	46 dBm (downlink of BH) 23 dBm (uplink of BH, SH)
Antenna pattern	Three-sector (2D)	Three-sector (2D) (BH) Omnidirectional (2D) (SH)
Antenna height	MBS: 25 m MUE: 1.5 m	mSC: 2 m mSUE: 1.5 m
Mobility model	MUE: random walk model	mSC: random walk model mSUE: group moving
Sidehaul connection	—	Contents and distance based connection (max distance: 200 m)
Number of UE pieces per cell	$300 - (2 \cdot M)$ M = the number of mSCs	2 per mSC

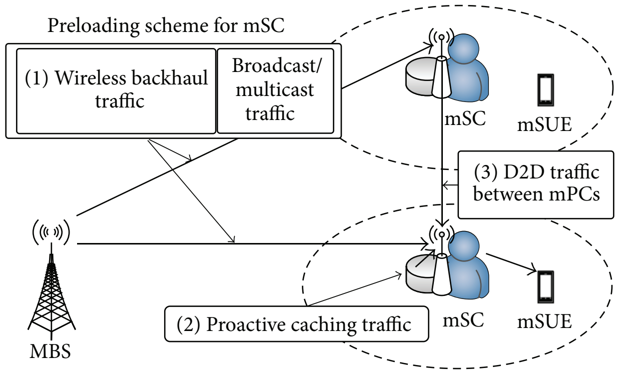

Preloading scheme for mSC
(1) Wireless backhaul traffic
Broadcast/ multicast traffic
mSC
mSUE
(3) D2D traffic between mPCs
(2) Proactive caching traffic
mSC
mSUE
MBS

Figure 3: Off-peak time proactive caching scheme for mSC network.

we believe that the complexity cost of other machine learning algorithms will overburden the MSC network which have limited computational capabilities. In ZipF distribution, α is the characterization exponent that ranges from zero to one. Moreover, it is obvious that the performance of mSC network is decidedly dependent on cache size (S). Huge cache size can significantly reduce the backhaul load and improve the QoS of mSC network. Furthermore, unlike broadcast mode, orthogonal radio resources are required for each multicast group transmission. Thus, the number of multicasting groups can significantly affect the performance of overall network.

4. Performance Evaluation

4.1. Simulation Environment. In order to evaluate the performance of our proposed mSC network with proactive cache, we conducted system level simulations. We consider a seven-macrocell network, where each cell consists of three hexagonal sectors. MBSs are located in the center of each macrocell and the intercell distance is 500 meters. MUE and mSCs are randomly deployed and then they move within macrocells. Similarly, mSUE pieces are randomly and uniformly deployed and move within the coverage area of their serving mSCs. In order to capture the real time mobility pattern of mSCs, we have used random walk mobility model [35]. According to our considered random walk model the moving cell (which can be a public transportation vehicle) travels in a random direction with random velocity and flight time. More detailed simulation parameters are given in Table 1.

In our system level simulator, we have adopted ITU UMa and WINNER path loss models for macrocells and mSCs, respectively. ITU UMa model considers urban macrocell environment [36, 37]. Pathloss equation of ITU UMa model is as follows:

$$\mathrm{PL} = 22.0\log_{10}(d) + 28.0 + 20\log_{10}(f_c), \qquad 10\,\mathrm{m} < d < d_{\mathrm{BP}},$$

$$\mathrm{PL} = 40.0\log_{10}(d) + 7.8 - 18.0\log_{10}(h_{\mathrm{BS}})$$

$$- 18.0\log_{10}\left(h_{\mathrm{UT}}\right) + 2.0\log_{10}\left(f_c\right), \qquad d_{\mathrm{BP}} < d < 5000\,\mathrm{m}, \tag{1}$$

where d is distance between transmitter and receiver. f_c is carrier frequency with range of 2 to 6 GHz. h_{BS} and h_{UT} are antenna heights of BS and UE, respectively, where d_{BP} is break point distance defined as

$$d_{\mathrm{BP}} = \frac{4h_{\mathrm{BS}}h_{\mathrm{UT}}f_c}{c}, \quad c = 3.0 \cdot 10^8\ \mathrm{m/s}. \tag{2}$$

WINNER model provides pathloss model for small-cells which has low power and small coverage area [38, 39] and its pathloss equations are

$$\begin{aligned} \mathrm{PL}_{B1\_\mathrm{total}}(d) &= \max\left(\mathrm{PL}_{\mathrm{free}}(d), \mathrm{PL}_{B1}(d)\right), \\ \mathrm{PL}_{\mathrm{free}}(d) &= 20\log_{10}(d) + 46.4 + 20\log_{10}\left(\frac{f_c}{2.0}\right), \\ \mathrm{PL}_{B1}(d) &= \left(44.9 - 6.55\log_{10}\left(h_{\mathrm{BS}}\right)\right)\log_{10}(d) \\ &\quad + 5.83\log_{10}\left(h_{\mathrm{BS}}\right) + 18.38 \\ &\quad + 23\log_{10}\left(f_c\right), \end{aligned} \tag{3}$$

where $\mathrm{PL}_{\mathrm{free}}(d)$ and $\mathrm{PL}_{B1}(d)$ mean free space pathloss and pathloss for small-cell, respectively.

In this paper, we have used the overall network capacity (C_{Total}) as a performance metric, which is total sum of macrocell capacity (C_{Macro}) and mSC capacity (C_{mSC}) in downlink. The capacity of each cell depends on the spectral efficiency and bandwidth assigned to UE. Spectral efficiency of UE can be obtained as the relationship between the signal to interference and noise ratio (SINR) and modulation and coding scheme (MCS) table [36].

Let U and M denote the numbers of MUE pieces and mSCs deployed in each macrocell, respectively. U_k denotes the number of mSUE pieces in the coverage of mSC k. The total available bandwidths in 2 GHz and 3.5 GHz frequency bands are $W_{2\,\mathrm{GHz}}$ and $W_{3.5\,\mathrm{GHz}}$, respectively. We define the macrocell capacity (C_{Macro}) as the sum of all MUE capacities. Thus, it can be calculated as

$$C_{\mathrm{Macro}} = (1-\rho)\frac{W_{2\,\mathrm{GHz}}}{U+M}\sum_{i=1}^{U}\mathrm{MCS}_{\mathrm{DL}}\left(\mathrm{SINR}_i\right), \tag{4}$$

where i means index of MUE attached to the MBS. ρ ($0 \leq \rho \leq 1$) depicts the ratio of radio resources for broadcasting/multicasting to overall radio resources for 2 GHz downlink.

Likewise, the capacity of mSC k ($C_{\mathrm{mSC},k}$) is also defined as the sum of all connected mSUE's capacities. However, the capacity of mSC depends on its transmission mode, that is, relay mode, cache mode, and mSC-to-mSC (sidehaul) mode. If mSUE requests a content file not cached in its respective or neighboring mSCs, the mSC performs relay transmission. In this case, the mSUE receives its data via wireless backhaul link and access link for mSUE. Thus, capacity of mSC k ($C_{\mathrm{mSC},k}$) in relay mode is defined as the minimum value between capacity of wireless backhaul link ($C_{\mathrm{BH},k}$) and capacity of access link for mSUE ($C_{\mathrm{access},k}$) and it can be expressed as

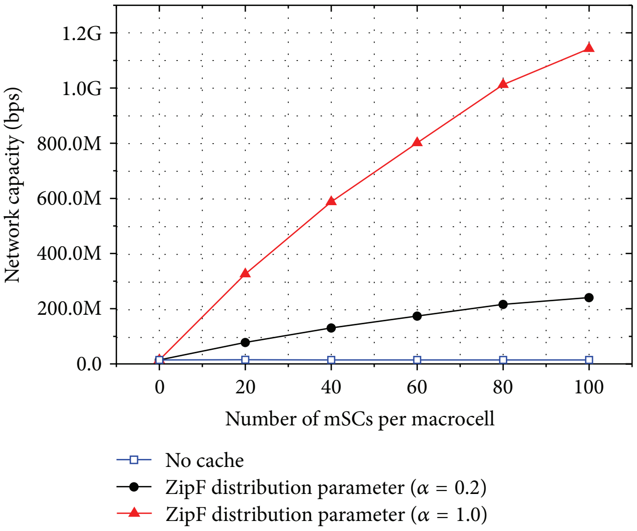


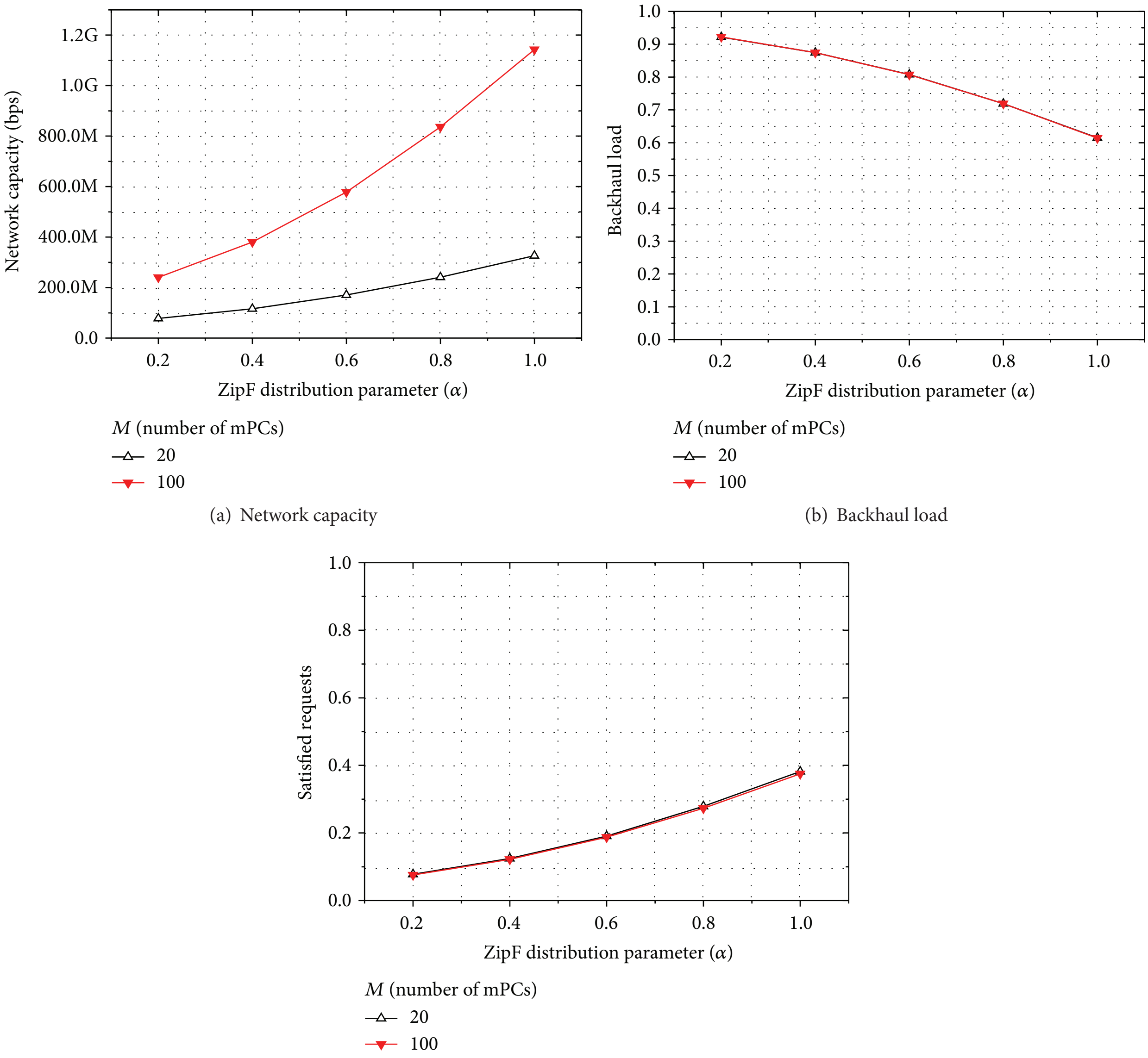
FIGURE 4: Network capacity varying numbers of mSCs operating in broadcast mode.

$$\begin{aligned} C_{\mathrm{mSC},k} &= \min\left(C_{\mathrm{BH},k}, C_{\mathrm{access},k}\right), \\ C_{\mathrm{BH},k} &= (1-\rho)\frac{W_{2\,\mathrm{GHz}}}{U+M}\mathrm{MCS}_{\mathrm{DL}}\left(\mathrm{SINR}_{\mathrm{BH},k}\right), \\ C_{\mathrm{access},k} &= \frac{W_{3.5\,\mathrm{GHz}}}{U_k}\sum_{j=1}^{U_k}\mathrm{MCS}_{\mathrm{DL}}\left(\mathrm{SINR}_j\right). \end{aligned} \tag{5}$$

Similarly, if mSUE requests a content file that is available in the cache to its serving mSC, the mSC performs cache transmission. In cache transmission mode, the mSUE directly receives its requested data from its serving mSC via access link. Thus, the capacity of mSC k operating in cache mode can be determined by the capacity of its access link for mSUE ($C_{\mathrm{access},k}$). On the other hand, mSCs operate in sidehaul transmission mode, if the contents requested by mSUE are not available in its serving mSC but are available in the cache of a neighboring mSC. The neighboring mSC delivers such data to serving mSC via wireless sidehaul link. The serving mSC forwards the received data to its respective mSUE via access link. In this case, capacity of mSC k ($C_{\mathrm{mSC},k}$) is decided as the minimum value between capacity of wireless sidehaul link ($C_{\mathrm{SH},k}$) and access link ($C_{\mathrm{access},k}$), and it can be expressed as

$$\begin{aligned} C_{\mathrm{mSC},k} &= \min\left(\delta \cdot C_{\mathrm{SH},k}, C_{\mathrm{access},k}\right), \\ \delta &= \begin{cases} 0, & \text{if sidehaul link does not exist} \\ 1, & \text{if sidehaul link exists,} \end{cases} \\ C_{\mathrm{SH},k} &= W_{2\,\mathrm{GHz}}\mathrm{MCS}_{\mathrm{UL}}\left(\mathrm{SINR}_{\mathrm{SH},k}\right). \end{aligned} \tag{6}$$

4.2. Simulation Results. Figure 4 shows the overall network capacity with varying number of mSCs operating in broadcast mode. It depicts that the overall network capacity is



(a) Network capacity

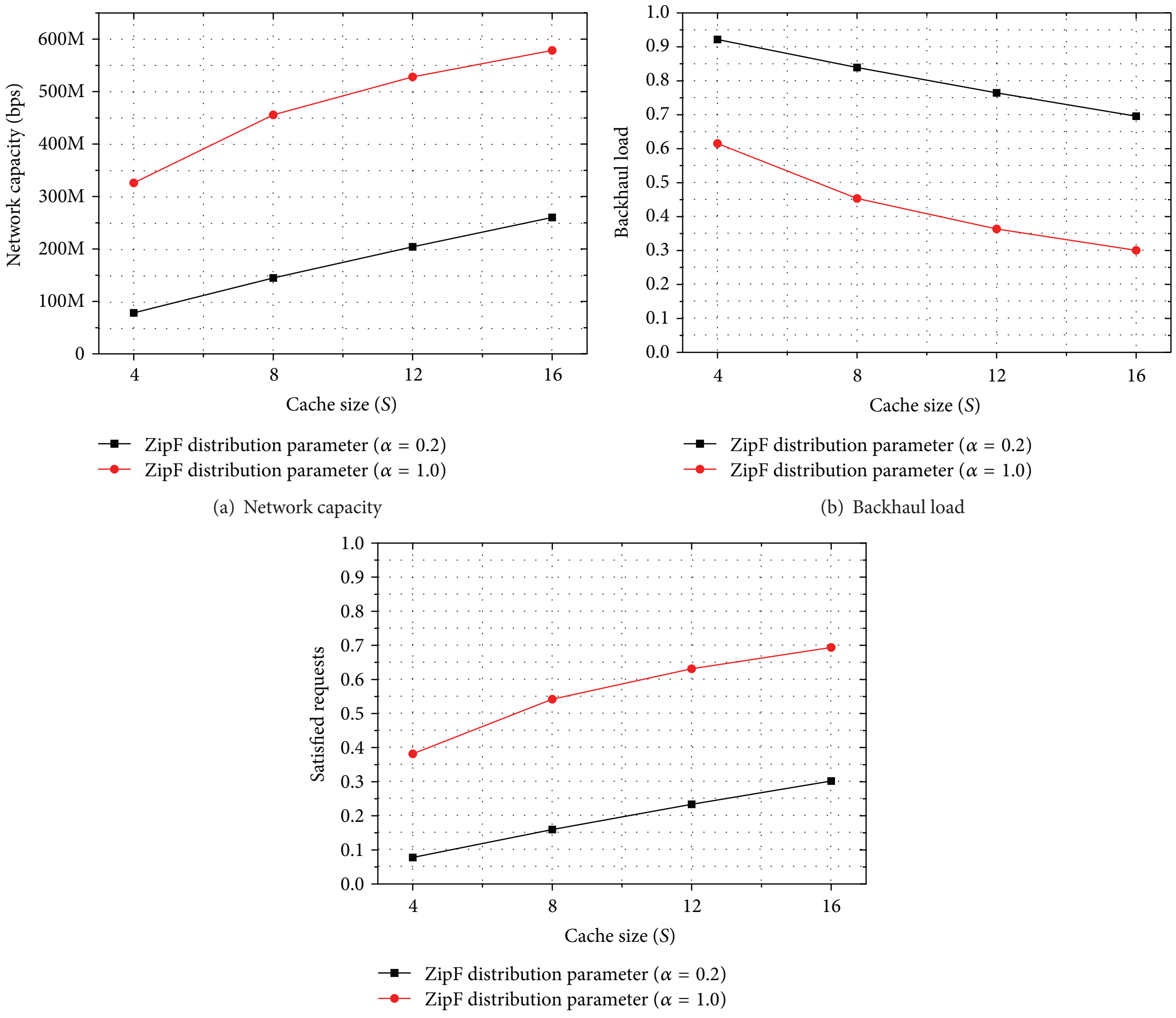
(b) Backhaul load

(c) Number of satisfied requests

Figure 5: Effect of ZipF distribution (α) on overall network performance operating at broadcasting and multicasting mode.

highly dependent on number of mSCs in the cell. It also shows that the mSC with cache scenario outperforms the no-cache scenario, because most of the contents requested by mSUE are already available in the cache of mSCs. Furthermore, as the popularity of files increases (α increases.) the overall network capacity also increases. It is because more mSUE pieces request the already cached files.

Similarly, Figure 5 depicts the effect of ZipF distribution (α) on overall network capacity, backhaul load, and number of satisfied requests. Two different mSC deployment scenarios (sparse and dense) are considered. Figure 5(a) shows that, beside the inter-mSC interference, the overall network capacity in dense deployment scenario (100 mSCs per macrocell) is significantly higher than sparse deployment scenario (20 mSCs per macrocell). The reason is that each mSC uses the same 2 GHz frequency band in access link. In dense deployment, more mSCs reuse the same frequency band in their access links. Likewise, Figure 5(b) depicts that the backhaul load significantly reduces as the file popularity increases. Furthermore, it also shows that, in both deployment scenarios, the file popularity has no major effect on backhaul load. In this work, we define backhaul load as the ratio of number of mSCs using backhaul link over total number of mSCs. Similarly, Figure 5(c) illustrates the relation between satisfied requests and file popularity. It shows that in both deployment scenarios the number of satisfied requests increases as the popularity of file increases. Here the term of satisfied requests means the ratio between numbers of



(a) Network capacity

(b) Backhaul load

(c) Number of satisfied requests

Figure 6: Effect of cache size (S) on overall network performance with varying ZipF distribution parameters.

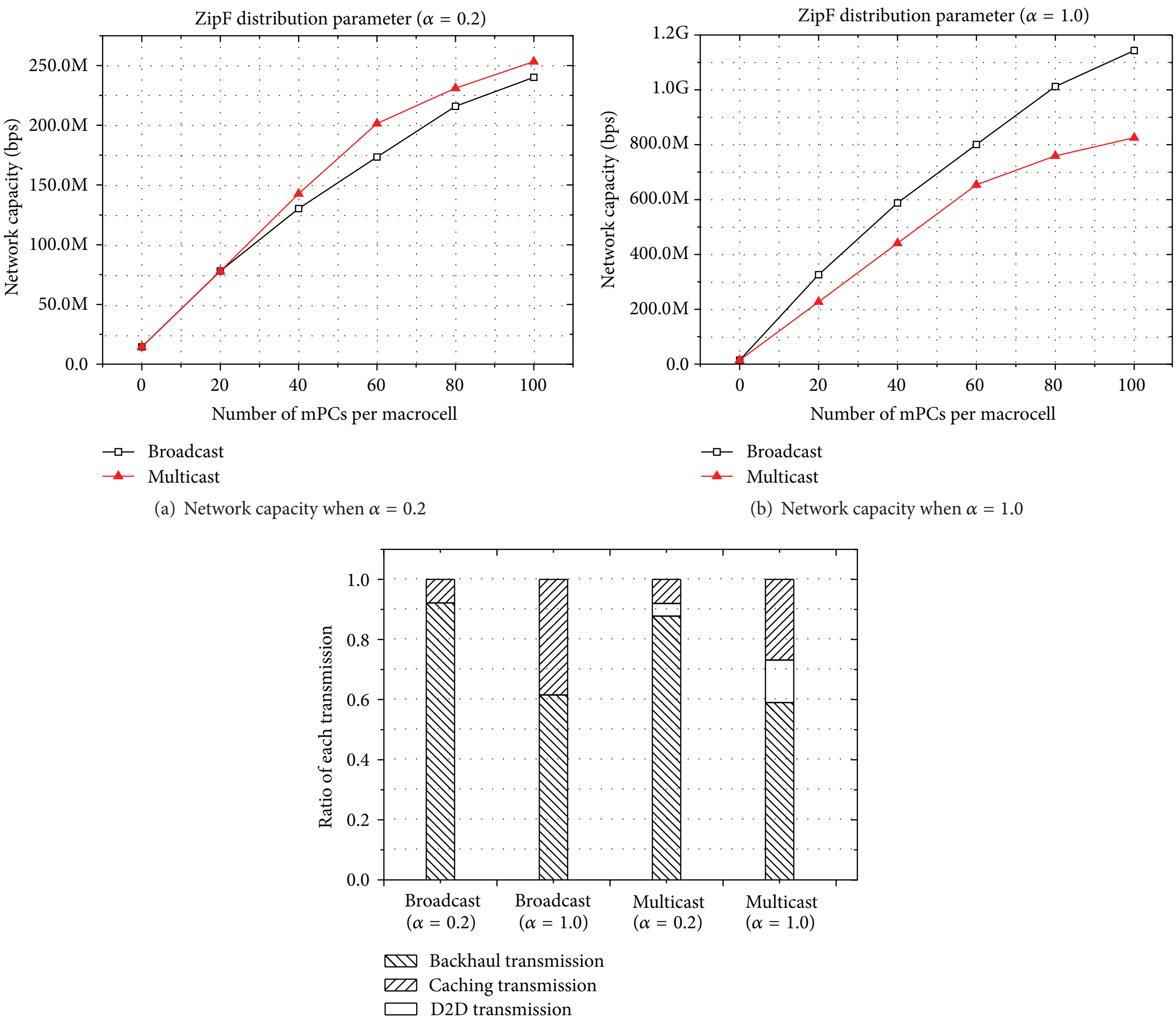
satisfied requests over total number of requests. If a user successfully receives a file size of 1 MB within 1 second after his request, we call this request as satisfied one.

The effect of cache size (S) on overall network capacity, backhaul load, and number of satisfied requests is shown in Figure 6. It is shown in Figure 6(a) that, with fixed number of mSCs (in this case 20), the overall network capacity significantly increases as the cache size increases. Since large cache size can proactively store popular contents, they can also significantly reduce the backhaul traffic load (Figure 6(b)) and increase the numbers of satisfied user requests (Figure 6(c)) in mSC network.

Figure 7 shows the effect of multicast groups on overall network capacity. It can be observed from Figures 7(a) and 7(b) that, for two different zip distribution parameters ($\alpha = 0.2$ and $\alpha = 1$), the broadcast mode outperforms the multicast mode. It is because the MBS in multicast mode uses orthogonal channels to transmit different contents to different mSC groups (in this case 2 groups), and thus it consumes more backhaul bandwidth than broadcast mode. Figure 7(c) depicts the comparison of different resource utilization of mSCs operating at broadcast and multicast modes. It can be observed that in both broadcast and multicast mode the ZipF distribution factor plays a vital role and the backhaul load reduces to 61% and 59% when it approaches to 1, respectively. Furthermore, the utilization of sidehaul link in multicast mode increases up to 14% when α approaches to 1.

5. Conclusion

In this paper, we discuss the role of mSCs in future HetNets and proposed a novel proactive caching based mSC network. We show that, by using the predictive nature of user demands, next generation networks can effectively preload their cache



(a) Network capacity when α = 0.2

(b) Network capacity when α = 1.0

(c) Ratio of different transmission modes

Figure 7: Influence of multicast groups on overall network performance with varying values of α.

with popular contents and reduce the traffic data demand in peak hours. Our extensive system level simulation results show that the proposed mSC network can significantly improve the QoS performance and overall system capacity of the network. We also show that the overall network performance is highly dependent on number of mSCs deployed, cache size, and content popularity. For future studies, we are aiming at incorporating the transmitted power control schemes in our simulator, which will effectively mitigate cross- and cotier interference in mSC networks. Another interesting line of investigation is to study various resource partitioning and scheduling schemes, which can statically or dynamically divide radio resources between macrocell and mSCs and reduce the interference and improve overall performance of the network.

Competing Interests

The authors declare that they have no competing interests.

Acknowledgments

This work was supported by Institute for Information & communications Technology Promotion (IITP) grant funded by the Korean government (MSIP) (no. R0101-15-244; Development of 5G Mobile Communication Technologies for Hyperconnected Smart Services).

References

[1] Cisco, "Global mobile data traffic forecast update, 2013–2018," White Paper, 2014.

[2] A. Damnjanovic, J. Montojo, Y. Wei et al., “A survey on 3GPP heterogeneous networks,” *IEEE Wireless Communications*, vol. 18, no. 3, pp. 10–21, 2011.

[3] S.-P. Yeh, S. Talwar, G. Wu, N. Himayat, and K. Johnsson, “Capacity and coverage enhancement in heterogeneous networks,” *IEEE Wireless Communications*, vol. 18, no. 3, pp. 32–38, 2011.

[4] D. López-Pérez, A. Valcarce, G. De La Roche, and J. Zhang, “OFDMA femtocells: a roadmap on interference avoidance,” *IEEE Communications Magazine*, vol. 47, no. 9, pp. 41–48, 2009.

[5] X. Ge, H. Cheng, M. Guizani, and T. Han, “5G wireless backhaul networks: challenges and research advances,” *IEEE Network*, vol. 28, no. 6, pp. 6–11, 2014.

[6] T. Yamamoto and S. Konishi, “Impact of small cell deployments on mobility performance in LTE-advanced systems,” in *Proceedings of the IEEE 24th International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC Workshops ’13)*, pp. 189–193, London, UK, September 2013.

[7] 3GPP, “Evolved Universal Terrestrial Radio Access (E-UTRA) and Evolved Universal Terrestrial Radio Access Network (E-UTRAN); Overall description; Stage 2,” 3rd Generation Partnership Project (3GPP), TS 36.300, v12.3.0, March 2011, http://www.3gpp.org/ftp/Specs/html-info/36300.htm.

[8] Y. Sui, J. Vihriala, A. Papadogiannis, M. Sternad, W. Yang, and T. Svensson, “Moving cells: a promising solution to boost performance for vehicular users,” *IEEE Communications Magazine*, vol. 51, no. 6, pp. 62–68, 2013.

[9] L. Chen, Y. Huang, F. Xie et al., “Mobile relay in LTE-advanced systems,” *IEEE Communications Magazine*, vol. 51, no. 11, pp. 144–151, 2013.

[10] 3GPP, “Evolved Universal Terrestrial Radio Access (E-UTRA); Study on mobile relay,” 3rd Generation Partnership Project (3GPP), TR 36.836, v2.0.2, June 2014, http://www.3gpp.org/dynareport/36836.htm.

[11] Y. Sui, Z. Ren, W. Sun, T. Svensson, and P. Fertl, “Performance study of fixed and moving relays for vehicular users with multi-cell handover under co-channel interference,” in *Proceedings of the 2nd IEEE International Conference on Connected Vehicles and Expo (ICCVE ’13)*, pp. 514–520, IEEE, Las Vegas, Nev, USA, December 2013.

[12] A. Enrique, “User-centric wireless local loop use-cases: a brief overview on assumptions and requirements,” White Paper, 2011.

[13] A. Ghosh, N. Mangalvedhe, R. Ratasuk et al., “Heterogeneous cellular networks: from theory to practice,” *IEEE Communications Magazine*, vol. 50, no. 6, pp. 54–64, 2012.

[14] E. Baştug, J.-L. Guénégo, and M. Debbah, “Proactive small cell networks,” in *Proceedings of the 20th International Conference on Telecommunications (ICT ’13)*, pp. 1–5, Casablanca, Morocco, May 2013.

[15] L. Chen, W. Chen, B. Wang, X. Zhang, H. Chen, and D. Yang, “System-level simulation methodology and platform for mobile cellular systems,” *IEEE Communications Magazine*, vol. 49, no. 7, pp. 148–155, 2011.

[16] Y. M. Kwon, J. Shin, J. S. Kim, S.-M. Oh, M. Y. Chung, and A.-S. Park, “Development of system level simulator for evaluating performance of moving personalcell network,” in *Proceedings of the 8th ACM International Conference on Ubiquitous Information Management and Communication (ICUIMC ’14)*, p. 19, Siem Reap, Cambodia, January 2014.

[17] I. Hwang, B. Song, and S. S. Soliman, “A holistic view on hyper-dense heterogeneous and small cell networks,” *IEEE Communications Magazine*, vol. 51, no. 6, pp. 20–27, 2013.

[18] J. G. Andrews, S. Buzzi, W. Choi et al., “What will 5G be?” *IEEE Journal on Selected Areas in Communications*, vol. 32, no. 6, pp. 1065–1082, 2014.

[19] H. S. Dhillon, R. K. Ganti, F. Baccelli, and J. G. Andrews, “Modeling and analysis of K-tier downlink heterogeneous cellular networks,” *IEEE Journal on Selected Areas in Communications*, vol. 30, no. 3, pp. 550–560, 2012.

[20] V. Van Phan, K. Horneman, L. Yu, and J. Vihriala, “Providing enhanced cellular coverage in public transportation with smart relay systems,” in *Proceedings of the IEEE Vehicular Networking Conference (VNC ’10)*, pp. 301–308, Jersey City, NJ, USA, December 2010.

[21] Y. Sui, A. Papadogiannis, W. Yang, and T. Svensson, “Performance comparison of fixed and moving relays under co-channel interference,” in *Proceedings of the IEEE Globecom Workshops (GC Wkshps ’12)*, pp. 574–579, IEEE, Anaheim, Calif, USA, December 2012.

[22] Y. Sui, A. Papadogiannis, and T. Svensson, “The potential of moving relays—a performance analysis,” in *Proceedings of the IEEE 75th Vehicular Technology Conference (VTC Spring ’12)*, pp. 1–5, IEEE, Yokohama, Japan, May 2012.

[23] Y. Sui, A. Papadogiannis, W. Yang, and T. Svensson, “The energy efficiency potential of moving and fixed relays for vehicular users,” in *Proceedings of the IEEE 78th Vehicular Technology Conference (VTC Fall ’13)*, pp. 1–7, IEEE, Las Vegas, Nev, USA, September 2013.

[24] IEEE 802.16 Broadband Wireless Access Working Group, “IEEE C802.16j-07/087r2 mobile relay station operation,” IEEE, Technical Report, 2007, http://www.ieee802.org/16/.

[25] J. Tadrous, A. Eryilmaz, and H. El Gamal, “Proactive content download and user demand shaping for data networks,” *IEEE/ACM Transactions on Networking*, vol. 23, no. 6, pp. 1917–1930, 2013.

[26] N. Golrezaei, K. Shanmugam, A. G. Dimakis, A. F. Molisch, and G. Caire, “Femtocaching: wireless video content delivery through distributed caching helpers,” in *Proceedings of the IEEE (INFOCOM ’12)*, pp. 1107–1115, IEEE, Orlando, Fla, USA, 2012.

[27] M. Ji, G. Caire, and A. F. Molisch, “Fundamental limits of distributed caching in D2D wireless networks,” in *Proceedings of the IEEE Information Theory Workshop (ITW ’13)*, pp. 1–5, IEEE, Sevilla, Spain, September 2013.

[28] E. Bastug, M. Bennis, and M. Debbah, “Living on the edge: the role of proactive caching in 5G wireless networks,” *IEEE Communications Magazine*, vol. 52, no. 8, pp. 82–89, 2014.

[29] A. Bar-Noy, R. E. Ladner, and T. Tamir, “Optimal delay for media-on-demand with pre-loading and pre-buffering,” *Theoretical Computer Science*, vol. 399, no. 1-2, pp. 3–11, 2008.

[30] J. Lee, M. Sun, and G. Lebanon, “A comparative study of collaborative filtering algorithms,” https://arxiv.org/abs/1205.3193.

[31] R. Vale and H. Viswanathan, “Further Efficiencies with eMBMS Preloading,” https://techzine.alcatel-lucent.com/further-efficiencies-embms-preloading.

[32] G. K. Zipf, “Relative frequency as a determinant of phonetic change,” *Harvard Studies in Classical Philology*, vol. 40, pp. 1–95, 1929.

[33] K. Shanmugam, N. Golrezaei, A. G. Dimakis, A. F. Molisch, and G. Caire, “FemtoCaching: wireless content delivery through distributed caching helpers,” *IEEE Transactions on Information Theory*, vol. 59, no. 12, pp. 8402–8413, 2013.

[34] L. Breslau, P. Cao, L. Fan, G. Phillips, and S. Shenker, “Web caching and Zipf-like distributions: evidence and implications,”

in *Proceedings of the IEEE 18th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '99)*, vol. 1, pp. 126–134, New York, NY, USA, March 1999.

[35] D. B. Johnson and D. A. Maltz, "Dynamic source routing in ad hoc wireless networks," in *Mobile Computing*, T. Imielinski and H. Korth, Eds., chapter 5, pp. 153–181, Kluwer Academic, Dordrecht, Netherlands, 1996.

[36] 3GPP, "Further advancements for E-UTRA physical layer aspects (Release 9)," 3rd Generation Partnership Project (3GPP) TR 36.814, v9.0.0, 2010.

[37] 3GPP, "Evolved Universal Terrestrial Radio Access (E-UTRA); Radio Frequency (RF) system scenarios (Release 11)," 3rd Generation Partnership Project (3GPP), TR 36.942, v11.0.0, September 2012.

[38] IST-4-027756 WINNER II D1.1.2V1.2, WINNER II Channel Models, 2007.

[39] D5.3: WINNER+ Final Channel Models, V1.0, 2010.

Hindawi Publishing Corporation
Mobile Information Systems
Volume 2016, Article ID 1353290, 12 pages
http://dx.doi.org/10.1155/2016/1353290

Research Article

SmartCop: Enabling Smart Traffic Violations Ticketing in Vehicular Named Data Networks

Syed Hassan Ahmed, Muhammad Azfar Yaqub, Safdar Hussain Bouk, and Dongkyun Kim

School of Computer Science & Engineering, Kyungpook National University, Daegu 702-701, Republic of Korea

Correspondence should be addressed to Dongkyun Kim; dongkyun@knu.ac.kr

Received 30 March 2016; Accepted 29 May 2016

Academic Editor: Yeong M. Jang

Recently, various applications for Vehicular Ad hoc Networks (VANETs) have been proposed and smart traffic violation ticketing is one of them. On the other hand, the new Information-Centric Networking (ICN) architectures have emerged and been investigated into VANETs, such as Vehicular Named Data Networking (VNDN). However, the existing applications in VANETs are not suitable for VNDN paradigm due to the dependency on a *"named content"* instead of a current *"host-centric"* approach. Thus, we need to design the emerging and new architectures for VNDN applications. In this paper, we propose a smart traffic violation ticketing (TVT) system for VNDN, named as *SmartCop*, that enables a cop vehicle (CV) to issue tickets for traffic violation(s) to the offender(s) autonomously, once they are in the transmission range of that CV. The ticket issuing delay, messaging cost, and percentage of violations detected for varying number of vehicles, violators, CVs, and vehicles speeds are estimated through simulations. In addition, we provide a road map of future research directions for enabling safe driving experience in future cars aided with VNDN technology.

1. Introduction

For the past decades, VANETs have been extensively investigated by the researchers, academia, and industries. Although initially designed to improve the road safety, VANETs can additionally offer commercial, informational, and entertainment services to the drivers and passengers, thus also increasing the revenues to the car manufacturers and various service providers. To be precise, the safety applications are mostly supported by the on-board units (OBUs) that depend on a Dedicated Short Range Communication (DSRC) protocol between vehicles (V2V) and in some cases infrastructures (V2I) as well. On the other hand, the nonsafety applications depend on the various TCP/IP protocols that have been proposed to operate on top of the amended 802.11p/Wireless Access in Vehicular Environments (WAVE) in the VANETs [1]. Furthermore, the IEEE 1609.4 multichannel architecture has also been introduced to WAVE standard that allowed efficient use of available spectrum for vehicular communications both in Europe and in USA. According to the standard, there is a 10 MHz Control Channel (CCH) and six 10 MHz Service Channels (SCHs) for exchanging safety/control messages (i.e., Beacons) and nonsafety applications' data, respectively.

In short, such advancements in vehicular communications system pursue as a potential tool to tackle the increasing number of road accidents caused by various violations been made on the roads. In this era of automation, we expect that new cars will be smart enough to proactively detect emergency situations and avoid road accidents [2]. For instance, we have seen Google, Tesla, Hyundai, BMW, and so many manufactures moving towards the autonomous cars. Figure 1 reflects the future smart vehicle. In the context of this paper, our focus will be on automating the traffic police vehicles and law enforcement departments in order to assist cops on the roads.

Conventionally, a traffic cop needs to identify a vehicle violating any traffic rule either manually or by use of electronic devices such as speed sensors and cameras. Then a cop follows the said vehicle and instructs the driver to pull over. The same cop then has to alight from his/her patrol car to

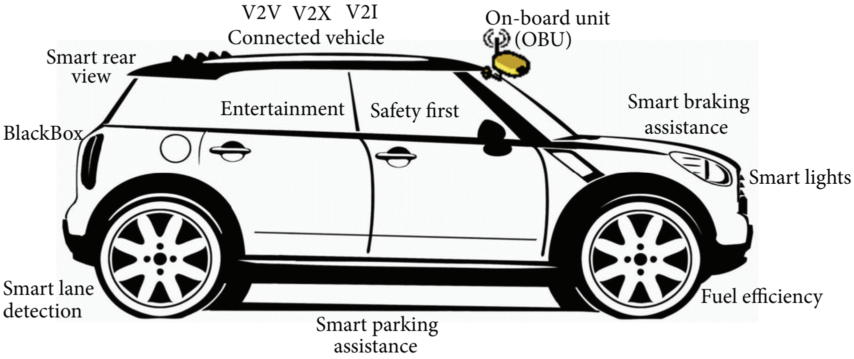


Figure 1: Components of a future smart vehicle.

manually inspect the vehicle to determine its identity and to manually inspect the offending driver's license to determine his/her identity. He/she then issues a violation ticket bearing the identity of the vehicle, the identity of the driver, nature of traffic violation, and the associated fine. The present system of issuing tickets for traffic violations has many shortcomings; for example,

(1) it is a time consuming and labor intensive process;

(2) sometimes the offending driver engages in a violent encounter with the traffic cop;

(3) in case of multiple violators, it is hard to follow up all at once;

(4) it is nearly impossible to cover all the road segments for sensing the traffic rules violations (by the means of camera, speed sensors, etc.).

The alternative way is to install speed cameras everywhere and monitor them all at once or partially while sitting back in the office. Once the installed camera detects any violator, it captures the video and image of the vehicle and later on the ticket is sent to the relevant owner by pulling out the relevant information against the number plate. However, it is impractical assumption to get all the streets and stop signs covered by the cameras. Moreover, on the long distance highways, also it would be an immature argument to install the speed cameras leaving no uncovered area behind. In addition, the maintenance cost of those cameras and sensors will compromise the cost effectiveness of the transportation departments and also the privacy concerns of the civilians will be disturbed.

Therefore, the researchers came up with an idea of equipping the vehicles with automatic traffic ticketing devices. Again, the main objective is to minimize the human errors and danger to the life of both the cop and the offender at the same time. For example, authors in [3] proposed to install a Radio Frequency Identification Device (RFID) that collects the data from in-vehicle sensors and delivers that data once the vehicle is crossing any tollbooth or cop vehicle (CV). This system aims to issue tickets autonomously in case of any violation within safe distance between cop and offender. However, the RFID systems lack meeting the current VANETs requirements, especially in terms of transmission ranges, speed variations, and authentication. Also, some additional hardware needs to be installed in vehicles other than OBUs, which are mostly used for V2V and V2I communications. One solution is to use the existing OBUs to enable the smarter ticket issuing mechanism for traffic violations given that the OBUs are equipped with the wireless communication technologies, for example, IEEE 802.11p/DSRC technologies. Although, we have a variety of solutions available for WAVE enabled OBUs in VANETs empowering the communication capabilities, all share the following common features:

(i) Each vehicle is assigned an IP address.

(ii) Specific destination addresses are used for applications to communicate.

(iii) Mostly, the candidate solutions aim to select one best path to reach the destination IP address.

However, assigning IP addresses to the mobile objects such as vehicles is not straight forward. The reason is simple; that is, IP address management requires infrastructure support, such as a central server (e.g., DHCP). Here it is worth mentioning that IP address concepts were originally introduced for wired technologies while mobility is an intrinsic feature of the VANETs, resulting in a highly dynamic network topology. Similarly, the best way to assign IPs to the mobile objects has been recognized as an open research issue [4].

Meanwhile, Named Data Network (NDN) [5] as an extension of Content-Centric Network (CCN) [6] has been applied in VANETs by several researchers, which is an emerging architecture of the future Internet projects [7]. NDN mainly shifts the communication concept from IP/host-based to the data centric in VANETs and can be referred to as Vehicular NDN (VNDN). In contrast to the IP based communications, in VNDN, a unique ID (called name) is assigned to the content instead of a host (i.e., end device), which attempts to relinquish the information from host's physical location and supports node mobility (i.e., vehicles in our case). NDN treats data or content as a first-class citizen of the network. In addition, VNDN uses simple request-reply based communication model, where a requesting node sends an "Interest" message and the provider sends back a response message with a requested data. Moreover, the recent literature

Table 1: Traffic violations in USA per year[†].

Description	Value
Average number of people per day that receive a speeding ticket	112,000
Total annual number of people who receive speeding tickets	41,000,000
Total percentage of drivers that will get a speeding ticket this year	20.6%
Average cost of a speeding ticket (including fees)	$152
Total paid in speeding tickets per year	$6,232,000,000
Average annual speeding ticket revenue per US police officer	$300,000
Percent of speeding tickets that get contested in traffic court	5%
Total number of licensed drivers in America today	196,000,000

[†] Statistics source: http://www.statisticbrain.com/driving-citation-statistics.

shows that data in NDN is more secure than the IP based communications due to intrinsic security within the data rather than the secured communication session [8]. The fact of the matter is that including the future Internet technologies into the existing ad hoc infrastructures is a potential solution. It is obvious that VNDN tends to support various nonsafety applications such as video streaming. To the best of our knowledge, apart from the nonsafety applications, we proposed smart *traffic violation ticketing* (TVT) architecture as a first step towards the applied Vehicular CCN [9].

In this paper, we extend our work to apply the latest NDN architecture in VANETs and propose a complete system that tends to aid law enforcement agencies with safer and smarter TVT system. We name our proposed scheme as "*SmartCop.*" In SmartCop, we define several packet types and their roles to support traffic violation ticketing system. Moreover, it is able to detect the offenders and issue them tickets without human interference. Unlike the existing solutions, we only rely on OBU(s). Our main objective is to enable a CV to autonomously receive all violations' information from the neighboring OV(s). The major contributions of our SmartCop are (1) to detect the violator(s) from safe distance using future Internet, (2) to issue the tickets using wireless medium regardless of vehicles' speed and moving directions, (3) to collect ticket dues automatically, thus saving the time and efforts of both the offender and the cop at the same time, and (4) to tend to leave no unmonitored areas on the roads.

The rest of the paper is organized as follows. Section 2 summarizes the recent efforts being driven to automate the traffic violation ticketing system. Section 3 describes our proposed SmartCop system, while Section 4 provides simulation results and analysis. In Section 5, we briefly discuss the current issues and challenges in VNDN. Finally, Section 6 concludes the paper.

2. Related Work

The law enforcement agencies make a good amount of revenue each year by issuing road violation tickets. Table 1 shows that 20% of the total drivers in the United States receive tickets for overspeeding each year. More or less, the same statistics will be for other road violations globally. In this section, we overview the recent advancements being made to CVs and OVs to assist government officials (i.e., cops) on the roads and reduce the traffic violations, respectively.

In [10], the authors utilize GPS to get information about the vehicle state, that is, location and speed. The vehicle is equipped with a traffic violation warning and traffic violation storage device, which is used to store the map data, traffic regulations of the current road segment, and the traffic violations made by the driver. A controller is used to control and manage the different units of the device. The GPS data is matched with the map data and traffic regulations, stored previously in the device, to determine if a violation has been made. Based on the result, either the driver is issued a warning if a possible violation is calculated or a ticket is stored in the violation memory of the device if a violation has been committed. Furthermore, an encryption mechanism is also presented to store encrypted tickets in the memory. The issued tickets along with the violations details and personal information can be viewed later on the management display.

In [11], the authors utilize a radio frequency (RF) reader to determine the identity of a vehicle and conversely the identity of the driver and then issue traffic tickets according to the applicable traffic laws. The smart ticket device is controlled by a central processing unit and the device contains radio frequency reader, wireless transceiver, memory, and communication ports. RF tags are mounted on the number plate of the vehicles and in the driving license of the driver, which contain the vehicle and driver's identification information, respectively. The RF reader of the smart ticketing device is able to read the information from these RF tags from static and mobile vehicles. The information obtained is used to issue a traffic violation ticket, containing the vehicle and driver's information, time and nature of the violation, and the respective fine. Furthermore, an extension to this idea is to install speed sensors in the smart ticketing device with which overspeeding vehicles can also be caught easily.

In [12], the authors use a series of digital cameras, still and video, to monitor a traffic location. This system is coupled to a processing system, where image processors are used to compile vehicle and scene images produced by the digital camera system; furthermore, a verification system verifies the vehicle and driver's identity from the vehicle images. A notification system then notifies the potential violation information to the law enforcement agencies. The video camera records the footage both before and after the

Table 2: Former research efforts to reduce traffic violations.

Type	Year	Objectives	Technologies
US patent [10]	2004	Traffic violation warning & storage	GPS
US patent [11]	2006	Traffic violation identification & ticketing	Radio frequency reader
US patent [12]	2011	Traffic violation identification & ticketing	Digital still & video cameras
Research article [13]	2013	Traffic violation identification, ticketing & tracking	GPRS/GSM, Google Maps
US patent [14]	2014	Traffic violation detection device	Radio frequency reader
Research article [9]	2015	Traffic violation identification & ticketing	VCCN, DSRC, V2V, V2I

detection of the violation. A buffer is used to capture the footage before the violation is detected; it stores a nonstop video footage of the preceding few seconds. In case a violation is detected, the timer is started and when the timer expires the contents of the buffer are recorded and the resulting video clip is incorporated with the evidence from the digital still images of the violation of the identified violating car and the driver.

The authors propose a ticketing and tracking system in [13], by implementing a smart on-board GPS/GPRS system attached to the vehicle. Along with that, speed of the vehicle is monitored by the on-board system and in case of any speed violations, information about the vehicle, that is, location and maximum speed, is sent to the authorized office using a GPRS message which issues a violation ticket to the driver. The speed is monitored by GPS signal and accelerometer of the car. Moreover, the authors also propose a geocasting feature, that is, using Google Map to track the vehicles current location. The shock/vibrator sensors installed in the air bags are used to identify an accident, which leads to GSM/GPRS messages being sent to nearby vehicles, hospital, and other authorities.

An automated system is proposed in [14], where the police officers are given a handheld device which automatically detects traffic violations. The device is equipped with the traffic regulations and in case the vehicle driver is violating these regulations, an audio and visual system is installed to inform the driver and the authorities. The device is used to read the RF tags installed in the vehicles' number plates. The RF tag contains the vehicle ownership data which is used to issue ticket to the concerned driver. Furthermore, the device can also be connected to an on-site printer which prints the traffic violation ticket.

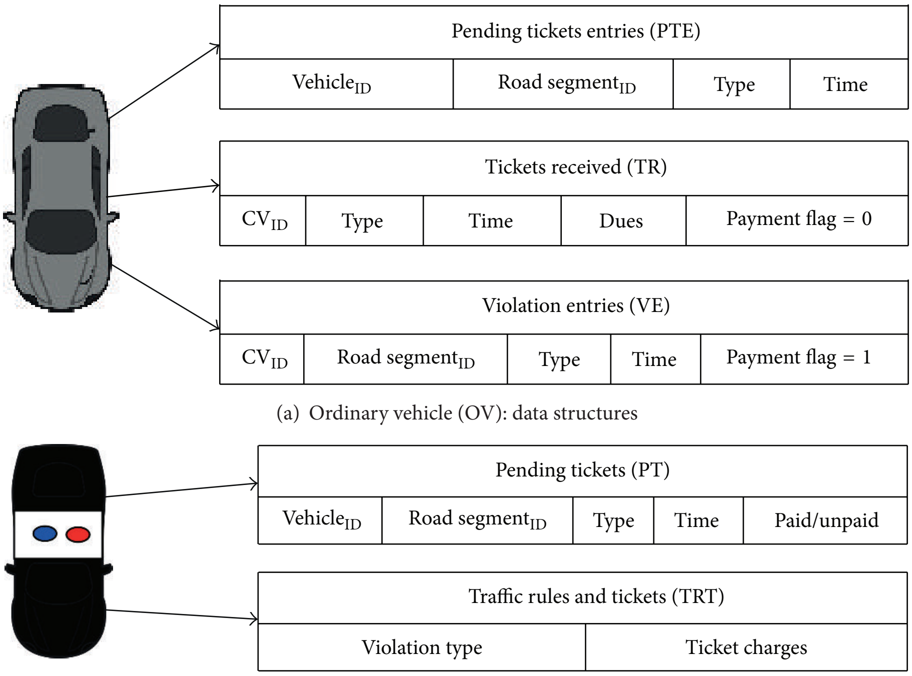
Furthermore, we proposed a unique traffic violation ticketing (TVT) system architecture in [9], where we considered the emergence of the content-centric and vehicular networking (VCCN). The main idea of the proposed architecture was to detect the offenders and issue them tickets without any human interference. However, we were precise and did not perform any experimentation. In this paper, we further extend our work and name it as a SmartCop, where extensive simulations have been performed and the architecture has been implemented over the IEEE 802.11p. Unlike the existing solutions, SmartCop only relies on on-board units (OBUs) with multiple interfaces. The proposed method contains different data structures; the ordinary vehicles (OV) contain three data structures, that is, Pending Tickets Entry (PTE), Tickets Received (TR), and Violation Entries (VE), whereas the cop vehicle (CV) contains two data structures, that is, Pending Tickets (PT) and Traffic Rules and Tickets (TRT). Also, it is able to cover the patrolled areas and more importantly the unpatrolled areas on the road where the violations by the ordinary vehicles (OV) go unnoticed. The violations by an OV in these areas are stored in the PTE. Since VNDN is a pull based communication paradigm, therefore in our architecture a CV periodically broadcasts an Interest message to have PTE(s) from its immediate neighbors. This allows the CV to issue ticket(s) in run time and avoid any manual contact with the driver. The tickets received by an OV are stored in the TR structure and upon contact with the RSU or tollbooth, the ticket's amount is deducted from the driver's bank account. For ticket payment the banking information of the drivers is accessible to the tollbooth and RSUs. Thus, the offenders are fined and charged autonomously. The record of payed tickets is stored in the VE structure of the OV.

Unfortunately, the automation of ticketing has not been investigated much as it argues to be and it can be seen from the summary depicted in Table 2.

3. SmartCop: Smart Traffic Ticketing in Vehicular NDN

3.1. VNDN: Communication Background. In VNDN, communication is a receiver-driven process based on two types of packets: the Interest, which carries the request for a content unit identified by its name. Each vehicle propagating an Interest is named a *consumer* and similarly a vehicle providing that content is called a *provider*. Conventionally, each vehicle in VNDN maintains three data structures: (i) a Content Store (CS) storing the produced and incoming contents; (ii) a routing table named Forwarding Information Base (FIB), which stores the outgoing interface(s) (in VNDN, each vehicle is expected to be equipped with multiple interfaces for communication such as 802.11, LTE, and WiMax) to forward the Interests; (iii) a Pending Interest Table (PIT), which keeps track of forwarded Interests so that received content can be stored in the CS or sent back to the consumer(s) accordingly.

3.2. Proposed SmartCop System Architecture. Along with an assumption of dividing roads into segments, we bring homogeneity in all public and private vehicles and named them as ordinary vehicles (OVs). Moreover, we named the traffic monitoring vehicles as cop vehicles (CVs). As we mentioned before, there are unmonitored areas on highway



Pending tickets entries (PTE)			
$\text{Vehicle}_{\text{ID}}$	$\text{Road segment}_{\text{ID}}$	Type	Time

Tickets received (TR)				
CV_{ID}	Type	Time	Dues	Payment flag = 0

Violation entries (VE)				
CV_{ID}	$\text{Road segment}_{\text{ID}}$	Type	Time	Payment flag = 1

(a) Ordinary vehicle (OV): data structures

Pending tickets (PT)				
$\text{Vehicle}_{\text{ID}}$	$\text{Road segment}_{\text{ID}}$	Type	Time	Paid/unpaid

Traffic rules and tickets (TRT)	
Violation type	Ticket charges

(b) Cop vehicle (CV): data structures

Figure 2: New data structures for SmartCop.

and also in an urban environment, which are collectively referred to here as an *"unpatrolled area."* It is expected that if any rule gets violated in an unpatrolled area it never gets noticed. Those violations can be of various types such as overspeeding, avoiding STOP signs, wrong lane, and parking in a no-parking zone. To cope with this, our proposed architecture enables OVs and CVs to maintain additional data structures as shown in Figures 2(a) and 2(b). Here it is worth mentioning that currently the OBUs and sensors installed in vehicles are capable of sensing violation(s) depleted by the driver. However, there might be a case where an OV violates a traffic rule and there is no nearby camera or CV to pursuit accordingly (refer to Figure 6). Therefore, we intend to manage those recorded violations in Pending Tickets Entry (PTE) table at each OV. Since, VNDN is a pull based communication paradigm, therefore in our architecture a CV periodically broadcasts an Interest message to have PTE(s) from its immediate neighbors (i.e., one-hop neighbors). This exchange of PTE enables each CV to issue ticket(s) at run time while avoiding the existing manual operations. More specifically, PTEs are shared using the same interfaces, from where the Interest was received, and upon receiving PTEs from neighboring OVs, a CV checks its Traffic Rules and Tickets (TRT) database (each CV is equipped with updated traffic rules and ticket prices table similar to Content Store in conventional CCN). Once the type of violation is matched, a corresponding CV sends back a ticket and dues to the relevant OV. Afterward, an OV stores this ticket information in its Tickets Received (TR) table. Here we assume that each driver has one central bank account or payment card registered with the department of transportation used for the payment of toll and other charges. While crossing any upcoming tollbooth, the automatic payment of the issued tickets is completed and the entry from TR moves to Violation Entries (VE) for the purpose of keeping records. Basic operations of the proposed SmartCop system and its behavior in the urban and highway environments are discussed in the following text.

3.3. Violator Detection and Ticket Issuing Process. In a SmartCop system, all OVs maintain PTE structure in its untampered blackbox. An OV becomes a violator when it has committed violations and has entry(ies) in its PTE. Cop vehicles periodically send the Interest messages to detect the violators and this Interest message is similar to the default NDN Interest message with additional PTE option (I_{PTE}). The CV stores I_{PTE} information in its PIT, which also includes the NONCE value. The NONCE value is a 32-bit long integer that is randomly generated by the originator of the Interest message. Along with that the same NONCE value is present in the Data message that is received in response, to recognize that the Data message is a response of the particular Interest.

When an OV receives I_{PTE}, it first searches its PTE. In case of no entry in PTE, it discards the Interest message. On the other hand, if the PTE is not empty, then OV sends all the PTE information in the Data message (D_{PTE}). D_{PTE} contains all the PTE information, the vehicle's ID, and the same NONCE from the Interest message. When the CV receives the D_{PTE}, it searches its PIT and if the entry is found, then it stores the PTE data in the PT. The overall flow of this process is shown in Figure 3.

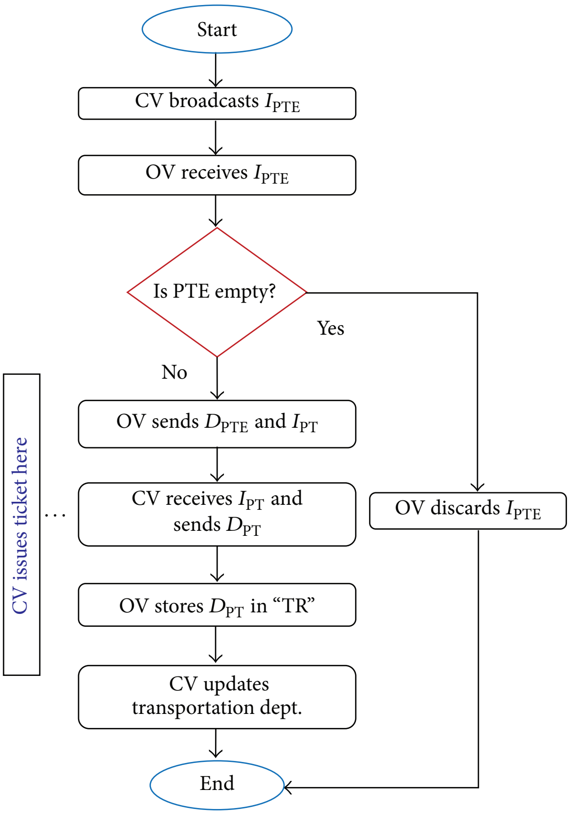

Start
CV broadcasts I_{PTE}
OV receives I_{PTE}
Is PTE empty?
Yes
No
OV sends D_{PTE} and I_{PT}
CV receives I_{PT} and sends D_{PT}
OV discards I_{PTE}
OV stores D_{PT} in "TR"
CV updates transportation dept.
End
CV issues ticket here
...

Figure 3: Violator detection and ticket issuing process.

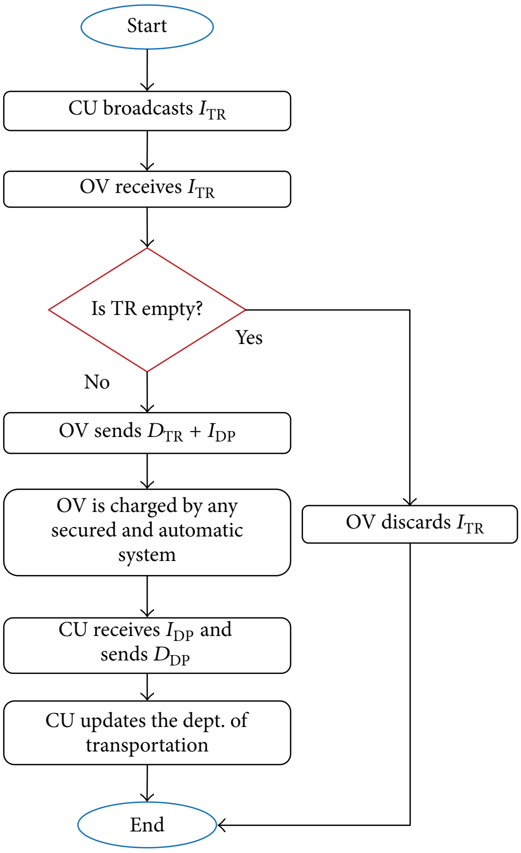

Start
CU broadcasts I_{TR}
OV receives I_{TR}
Is TR empty?
Yes
No
OV sends D_{TR} + I_{DP}
OV is charged by any secured and automatic system
OV discards I_{TR}
CU receives I_{DP} and sends D_{DP}
CU updates the dept. of transportation
End

Figure 4: Violation of fine collection process.

Immediately after sending D_{PTE}, the OV also generates the Interest message (I_{PT}) including its own ID, NONCE, and the "PT" header to request the ticket from the cop vehicle and the OV creates an entry within its PIT. On reception of I_{PT}, the CV matches the OV's ID in its PT and if CV finds entries, it sends D_{PT} containing all entries along with the violation charges that are fixed for each violation. These violation charges are referenced by the CV from the standard TRT, which is available to all the CVs in the region, and the amount is fixed by the department of transportation or law enforcement agency, which is out of the scope of our paper. When D_{PT} is received at the OV, it finds the PIT and then creates the record in its TR with *payment* flag 0. In addition to that, the same entries are discarded from the PTE. Here it is worth mentioning that the CV also sets the *paid/unpaid* flag to 0 to highlight that the fines are still pending. We name this whole message exchange between the CV and OV as a *session*. In a single session there may be possible that an OV receives multiple tickets. The rationale behind this is that the OV made multiple violations and did not come in close proximity of the CV. Therefore, the *session* is one of the SmartCop evaluation parameters in simulations that is discussed in the next section.

3.4. Fine Collection Process. The fine collection process is almost similar to the ticket issuing process; however, it involves the collection unit (CU) instead of the CV to collect fines from the OVs. The CU can be the equipment installed at the tollgate on a highway or highway exit, or it may be installed at RSU installed at any dedicated location, for example, highway and bank. Figure 4 shows the flow diagram of this step.

The CU periodically sends the Interest with TR header/option (I_{TR}). Upon reception of I_{TR}, the OV finds the entries in its TR. In case of no entries, I_{TR} is discarded. On the other hand, if there is/are entry(ies) in the TR, the OV sends D_{TR} along with its ID. When the CU receives D_{TR}, then it deducts the charges from the account or any payment card type associated with that OV ID. After successful payment of the fines, this information is sent to the CVs in the region to mark their respective entries in their PT as 1/paid (the dues payment method as well as the information dissemination to all CVs in the region is out of the scope of this work). Immediately after sending D_{TR}, the OV sends I_{DP} to receive the confirmation that whether the fine is paid or not. In case of successful payment, the CU sends D_{DP}, which indicates that the fine of the said violations fine has been successfully collected. Afterwards, the OV removes all the matching entries with those in D_{DP} from the TR and stores them in the violations record, the Violation Entries (VE) table.

3.5. SmartCop in Urban Environment. In case of urban region, we witness a lot of Road Side Units (RSUs) deployed,

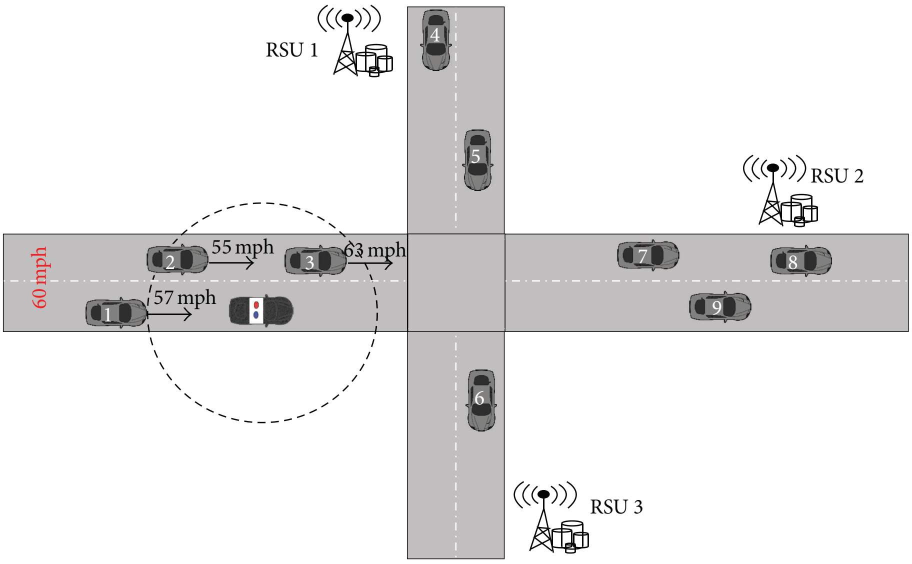


Figure 5: SmartCop: urban environment.

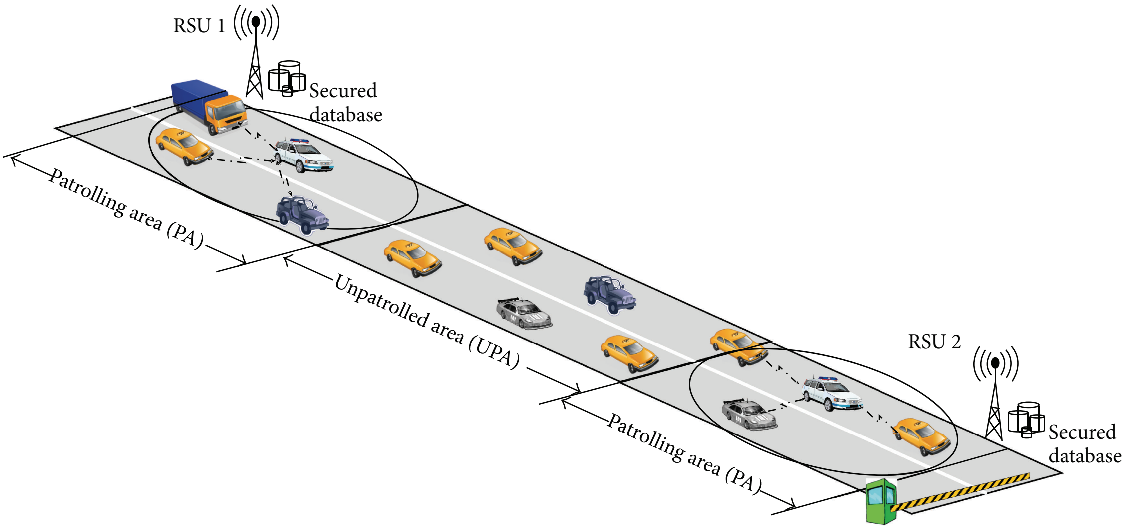


Figure 6: SmartCop: highway environment.

apparently supporting various applications. We expect those RSUs to work as ticket dues collectors due to their strong backbone connections to the wired networks. Figure 5 shows an urban scenario where an OV_3 is overspeeding and hence needs to be ticketed. Our smart traffic violation ticketing system enables the forthcoming CV to issue a relevant ticket to the vehicle OV_3 by sending an Interest packet for PTE, receiving PTE in return and updating the local law enforcement database through any available interface such as LTE or 3G (in VNDN, each vehicle is expected to be equipped with multiple interfaces for communication such as 802.11, LTE, and WiMax). For instance, the selection of the most reliable interface is out of the scope of this work. In SmartCop system, eventually an OV_3 pays the ticket dues while crossing the next RSU.

3.6. SmartCop in Highway Environment. On highways, mostly we have tollbooths as illustrated in Figure 6. Each tollbooth is equipped with at least one RSU and thus can attempt to charge the pending tickets stored in TR. However, there might be a case when a violating OV is not charged due to insufficient amount in the bank account and so on. In that case, we incorporate a binary flag in TR (i.e., 0 and 1) in the case of unpaid and paid tickets, respectively. In the former case, the transport department follows the conventional procedure that is mailing a ticket

Table 3: Simulation parameters.

Parameter	Value
MAC/PHY	IEEE 802.11p
Frequency band	5.9 GHz
Simulation duration	200 s
CVs	1–5
OVs	30–80
Violators	5–25
Number of violations/violators	Random (1–5)
Average vehicle speed	50–100 km/h

manually. In the latter case, the entry is moved to VE with a flag value of 1, thus ensuring that the payment has been completed.

4. Simulation Results and Analysis

In this section, we briefly discuss the simulation environment, the parameters, and the results of the proposed SmartCop scheme.

4.1. Simulation Environment. To evaluate the proposed SmartCop scheme, the NDN forwarding daemon architecture and IEEE 802.11p are implemented over each vehicle and simulated in the Network Simulator (NS2). Each vehicle in the simulation has the capability to communicate at the transmission range of 300 meters. Along with the default NDN structures (CS, PIT, and FIB), the structures supported by SmartCop, that is, PTE, TR, VE, PT, and TRT, are also implemented to properly evaluate its functionality. NDN's default Interest and data messages are modified to support violation ticketing operations. The highway mobility model along with the varying number of vehicles is simulated, which move at the average speed of 50 to 100 km/h. The total number of vehicles (N) is the sum of CV, OV, and the violators. Violator vehicles randomly make violations between 1 and 5 randomly during the simulation time of 200 s. Each CV sends the periodic Interest message after every 1 s to find the violators. The rest of the simulation parameters are shown in Table 3.

The SmartCop performance is the average of twenty simulation runs for each point in graphs with the confidence interval of 10%. Following is the description of the performance metrics that have been analyzed.

(i) Average cost is the total number of messages (Interest and data) that have been exchanged between the CV and the violators to successfully issue the violation ticket.

(ii) Satisfied delay is the amount of time between the Interest and the data messages received by a violator from the CV to successfully get the ticket(s) for violation(s).

(iii) Total delay is the amount of time when a violator committed the violation and received the ticket for that violation.

(iv) Number of sessions is the message exchange between a CV and the violator to get the violation ticket.

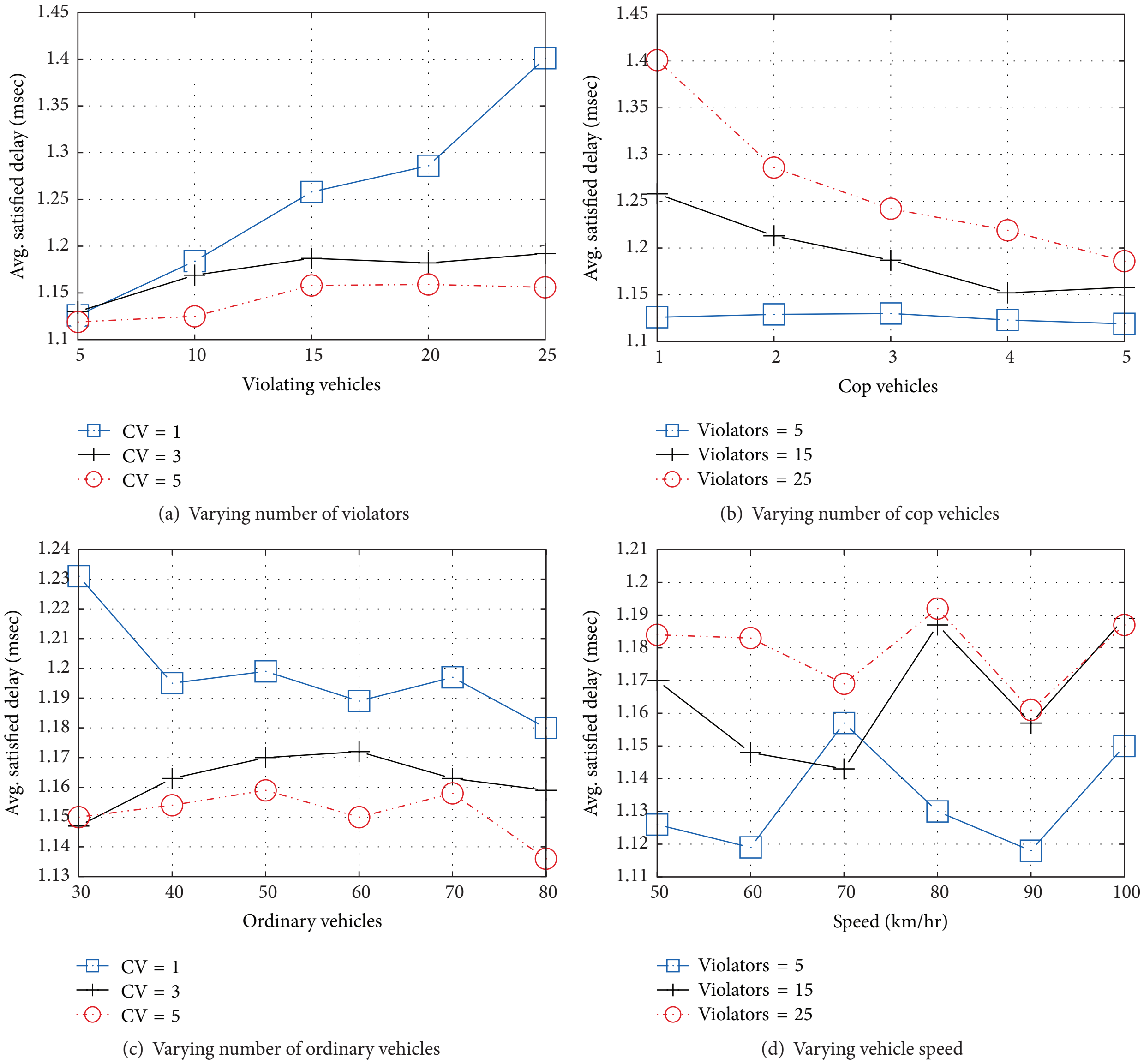
(v) Tickets satisfied is the ratio of tickets received and the total number of violations during the simulation period.

4.2. Results and Analysis. In this section, we briefly discuss the simulation results of the proposed SmartCop scheme.

Figure 7 shows the average satisfied delay for varying number of violators (Figure 7(a)), CVs (Figure 7(b)), OVs (Figure 7(c)), and the vehicle's speed (Figure 7(d)). The average satisfied delay is the duration between the I_{PTE} and D_{PT} received by a violator to successfully get the ticket for a traffic violation. To simply state, it is the violation ticket *session* delay during which the CV and the violator exchange messages for the violation ticket. It is evident from the figure that the higher the number of violators, the longer the delay. The rationale of this phenomenon is that, in the presence of a large number of violators, the message exchange will increase the traffic and the PTE, PT, TR, and other structures' search delay will be larger than results in a large violation satisfaction delay. The opposite is the case with the number of CVs. In case of more CVs, the violation ticket messaging overhead is distributed among the CVs that issue tickets with less delay; refer to Figures 7(a) and 7(b). On the other hand, the number of ordinary vehicles and the speed of the vehicle have not that much impact on the average satisfied delay because the NDN traffic on the ordinary vehicles does not access the PTE, PT, TR, and SmartCop related data structures. Therefore, the maximum difference in the satisfied delay is 0.03 ms for varying number of OVs and CV = 1 in Figure 7(c) and less than 0.035 ms for varying speed as evident from Figure 7(d). This concludes that the number of CVs and the violators in the area have the major impact on the satisfied delay.

Next, we analyzed the average total delay, which is the total time between the instance when a violation was committed (or entry was created in the PTE) and the ticket that was issued to the vehicle (or the entry was created in the TR for the respective PTE entry). It is obvious from Figures 8(a), 8(b), and 8(c) that average total delay is indirectly proportional to the number of cop vehicles because the tickets are only issued by the CVs. In case of less number of CVs, the violations will be pending the PTE for a longer time until the violator enters the communication range of the CV. The opposite is the case for the large number of the CVs in the area. Another factor that has the huge effect on the average total delay is the node's speed; refer to Figure 8(d). A vehicle that drives at a faster speed may quickly come in the communication range of the ticket issuing point and happens to have a short delay.

The other parameter that we analyzed through simulations is the messaging cost to satisfy all the violations during a simulation run. Messaging cost is the total number of messages (Interest and data) exchanged between the violator and the CV to issue the ticket. Figure 9 shows the average cost for varying the above discussed parameters. It is obvious that the cost is directly proportional to the number of



(a) Varying number of violators

(b) Varying number of cop vehicles

(c) Varying number of ordinary vehicles

(d) Varying vehicle speed

Figure 7: Average satisfied delay.

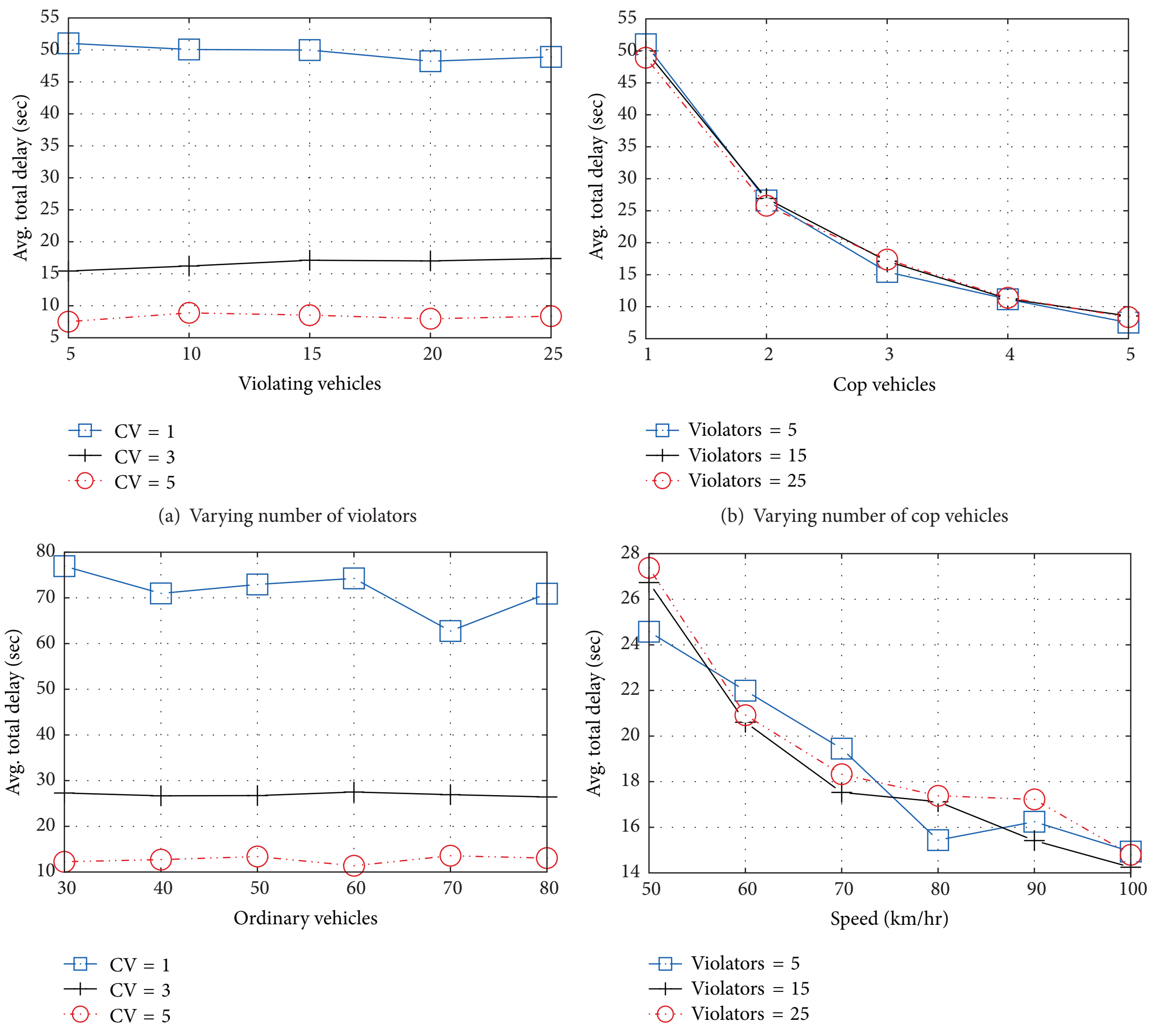
violators and the number of CVs. If the number of violators increases, then it requires more numbers of messages to issue tickets. Similarly, the larger the number of CVs in the area, the more tickets are issued to the violators that increases the messaging cost and it is obvious from the figure. Additionally, it can easily be analyzed from the results that the number of ordinary vehicles and the vehicle's speed have no significant effect on the messaging cost; refer to Figures 9(c) and 9(d).

5. Open Issues in VNDN and SmartCop System

In this section, we provide readers with the open issues connected to SmartCop system and VNDN, needing the attention from researchers working for a secure driving experience and other application domains in VANETs.

5.1. Naming in VNDN and SmartCop. Content naming is the most important issue in future networks where the focus of communication is *content* but not the IP/TCP based device addresses. Therefore, we have various naming schemes for conventional CCN, NDN, and VNDN. Some of them are categorized as hierarchical, flat, human readable, hash-based, attribute-based, and so on [15]. However, it is difficult to determine the best suitable scheme for VNDN, especially when we are trying to communicate a highly sensitive data between vehicles on roads such as in SmartCop system. Similarly, we need to design a hybrid naming scheme for different violation types and their relevant entries to be included in an *"Interest"* packet, which will be broadcast by a CV to each OV in its transmission range.

5.2. Content Distribution. For instance, we have assumed that every road segment will be covered with one CV or none.



(a) Varying number of violators

(b) Varying number of cop vehicles

(c) Varying number of ordinary vehicles

(d) Varying vehicle speed

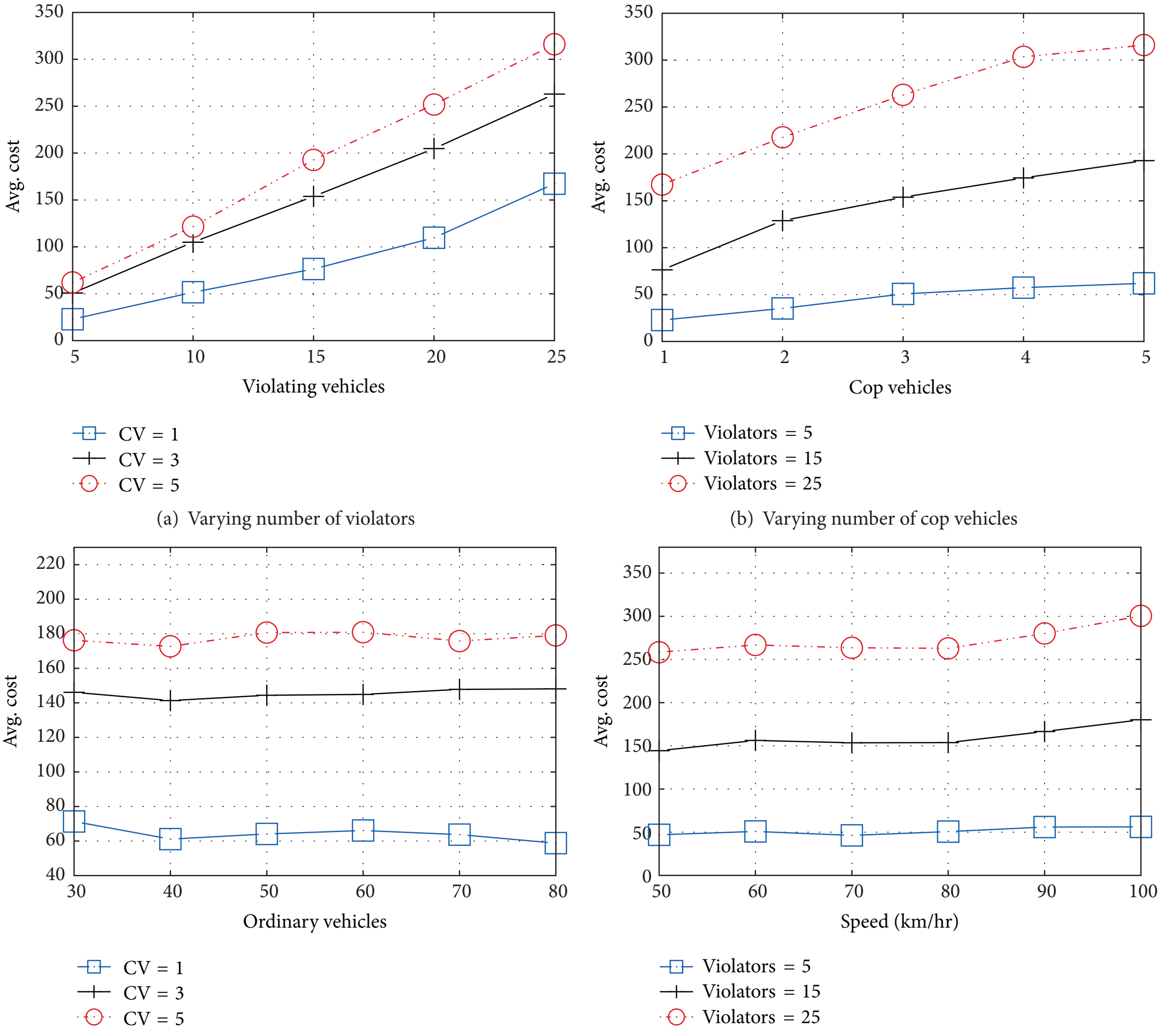
Figure 8: Average total delay.

However, there may be a case where two or more CVs come across; in that case we need to address the selection process of a CV for exchanging PTE by any OV. On the other hand, the identification of redundant data of PTE received by any CV is a significant challenge to be addressed.

5.3. Autonomous Ticket Issuance. Using our SmartCop system in a highly dynamic environment such as VANETs in urban scenario is a challenging task. There is the possibility of multiple offenders/violators in the immediate transmission range of a CV. Therefore, issuing a violation ticket to multiple OVs requires a highly cooperative and fast synchronization mechanism. Moreover, managing the PT entries in the CV's local memory should be addressed, respectively.

5.4. Interest Packets Flooding. Due to the broadcast nature of the wireless medium, conventionally, Interest packets are flooded within the network. Since SmartCop system applies only to the immediate neighbors of any CV, while preparing the test-bed or experimental environment, a controlled Interest flooding technique needs to be implemented. For instance, one can use the hop count flag to limit the Interest flooding.

5.5. Security and Privacy Issues. Although our SmartCop is an initial step towards the smart ticketing in future vehicles. It is also very important to address the security issues at the different levels of communications, especially in the presence of the wireless medium. Those include the authenticity of the *content* being sent to any CV and also of any *Interest* packet sent by a CV itself. Furthermore, issuing a ticket is very sensitive and private step, so it is required to make sure that no other vehicle can access and open the history of neighbor vehicle.



(a) Varying number of violators

(b) Varying number of cop vehicles

(c) Varying number of ordinary vehicles

(d) Varying vehicle speed

Figure 9: Average Interest-Data messaging cost to successfully issue violation tickets during simulation.

6. Conclusion

In this paper, we present an architecture for a smart and efficient traffic violation ticketing system for vehicles with future Internet technologies such as NDN. Our architecture will enable traffic law officials to identify drivers and violating vehicles without chasing and putting lives in danger. In order to achieve this, we apply basic VNDN operations into our SmartCop system, where a cop vehicle periodically broadcasts an Interest packet for violation entries saved by every ordinary vehicle in its local memory (PTE). This exchange of PTE enables a cop vehicle to issue a relevant ticket to the offender. Later on, the offenders' vehicle, when connected to any road side unit, pays the charged ticket autonomously. As a result, all the manual operations and delays caused by human errors are skipped. In the end, we also enlist the future work directions for improving and implementing our proposed SmartCop system into real test-bed environments and simulations. The simulations show that the ticket issuing delay and its messaging cost depend upon the number of violators, vehicles, and speed of the vehicles on the road.

Competing Interests

There are no competing interests regarding the publication of this paper.

Acknowledgments

This research was supported by Kyungpook National University Bokhyeon Research Fund, 2015.

References

[1] X. Liu, Z. Li, P. Yang, and Y. Dong, "Information-centric mobile ad hoc networks and content routing: a survey," *Ad Hoc Networks*, 2016.

[2] B. Das, S. Misra, and U. Roy, "Coalition formation for cooperative service-based message sharing in vehicular ad hoc networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 27, no. 1, pp. 144–156, 2016.

[3] N. Ratnakar, "Smart Traffic Ticket Device," U.S. Patent Application 10/907, 271, filed March, 2005.

[4] N. Vaidya, "Open problems in mobile ad hoc networking," in *IEEE Local Area Networks*, p. 516, 2001.

[5] M. Amadeo, C. Campolo, and A. Molinaro, "Information-centric networking for connected vehicles: a survey and future perspectives," *IEEE Communications Magazine*, vol. 54, no. 2, pp. 98–104, 2016.

[6] S. H. Ahmed, S. H. Bouk, and D. Kim, "RUFS: RobUst forwarder selection in vehicular content-centric networks," *IEEE Communications Letters*, vol. 19, no. 9, pp. 1616–1619, 2015.

[7] M. Amadeo, C. Campolo, and A. Molinaro, "Enhancing content-centric networking for vehicular environments," *The Elsevier Journal of Computer Networks*, no. 16, pp. 3222–3234, 2013.

[8] Z. Yan, S. Zeadally, S. Zhang, R. Guo, and Y.-J. Park, "Distributed mobility management in named data networking," *Wireless Communications and Mobile Computing*, 2015.

[9] S. H. Ahmed, M. A. Yaqub, S. H. Bouk, and D. Kim, "Towards content-centric traffic ticketing in VANETs: an application perspective," in *Proceedings of the 7th International Conference on Ubiquitous and Future Networks (ICUFN '15)*, pp. 237–239, Sapporo, Japan, July 2015.

[10] T. Yamaki and T. Nishizaka, "Traffic violation warning and traffic violation storage equipment," U.S. Patent No. 6,720,889, April 2004.

[11] N. Ratnakar, "Smart traffic ticket device," US Patent Application 20060214783 A1, 2006.

[12] B. E. Higgins, "Automated traffic violation monitoring and reporting system with combined video and still-image data," U.S. Patent No. 7,986,339. 26 Jul. 2011.

[13] S. Tarapiah, S. Atalla, and R. AbuHania, "Smart on-board transportation management system using GPS/GSM/GPRS technologies to reduce traffic violation in developing countries," *International Journal of Digital Information and Wireless Communications*, vol. 3, no. 4, pp. 430–439, 2013.

[14] A. Harbi, S. H. S. Harmad, and D. A. M. Al-Fayez, "System for detecting and identifying traffic law violators and issuing citations," U.S. Patent No. 8,633,815. 21 Jan. 2014.

[15] S. H. Bouk, S. H. Ahmed, and D. Kim, "Hierarchical and hash based naming with Compact Trie name management scheme for Vehicular Content Centric Networks," *Computer Communications*, vol. 71, pp. 73–83, 2015.