FOG VEHICULAR COMPUTING

Augmentation of Fog Computing Using Vehicular Cloud Computing

Fog computing has emerged as a promising solution for accommodating the surge of mobile traffic and reducing latency, both known to be inherent problems of cloud computing. Fog services, including computation, storage, and networking, are hosted in the vicinity of end users (edge of the network), and, as a result, reliable access is provisioned to delay-sensitive mobile applications. However, in some cases, the fog computing capacity is overwhelmed by the growing number of demands from patrons, particularly during peak hours, and this can subsequently result in acute performance degradation. In this article, we address this problem by proposing a new concept called *fog vehicular computing* (*FVC*) to augment the computation and storage power of fog computing. We also design a comprehensive architecture for FVC and present a number of salient applications. The result of implementation clearly shows the effectiveness of the proposed architecture. Finally, some open issues and envisioned directions are discussed for future research in the context of FVC.

Mehdi Sookhak, F. Richard Yu, Ying He, Hamid Talebian,
Nader Sohrabi Safa, Nan Zhao, Muhammad Khurram Khan, and Neeraj Kumar

Digital Object Identifier 10.1109/MVT.2017.2667499
Date of publication: 27 July 2017

The Internet of Things

The emergence of the Internet of Things (IoT) phenomenon is revolutionizing the era of computing. IoT refers to a large network of interconnected mobile intelligent objects (called *things*) distributed over a geographic region that tend to carry out various tasks such as monitoring, actuating, sensing, and measuring. The things are everyday physical devices such as refrigerators, vehicles, biochip transponders, industrial machines, medical instruments, and traffic lights. Each object has a unique identifier, is connected to the Internet, and gathers data and interacts with other devices in the network. The appealing concept of the IoT is that it offers a multitude of remarkable, advantageous opportunities, which so far have not been possible. A wide range of innovative applications such as health care, transportation, aviation, social networking, and traffic control can be established based on the IoT concepts. The IoT can be useful for real-time process tracking, better remote control over systems, real-time analytics, and decision making [1], [2].

During their course of operation, however, billions of connected objects collect and exchange sizable amounts of data, resulting in significant growth of network traffic. The velocity by which data are generated is often far faster than these objects' communication speed. Because the mobile objects are typically constrained by limited network connectivity, one of the primary challenges in the IoT is to deliver reliable service for latency-sensitive applications [3], [4].

Although the conventional cloud computing paradigm is known to be a robust solution for delivering a diverse range of services to mobile applications, in practice, the long network distance between mobile devices and remote data centers hinders provision of real-time service to the application, resulting in a high delay sensitivity. Examples of such applications are video streaming, online gaming, and augmented reality [5], [6].

To address the aforementioned problem, Cisco created the fog computing model as a complementary paradigm to the conventional cloud computing model. The services on the fog are hosted on the set-top box, and access points in the vicinity of end users eliminate the dispensable network hops, thus minimizing the response time for mobile applications. In addition, fog computing can alleviate the traffic congestion in the Internet backbone because the huge amount of traffic originating from end devices is locally handled by fog servers in the vicinity of these devices. As analogs to cloud computing, the fog services are computing, networking, and storage [3].

In certain scenarios, the fog resource capacity is severely swamped by the immense number of demands from smart objects. For instance, in a multistory mega shopping center, particularly during peak hours, thousands of patrons can use the fog services provided, such as video streaming and video gaming. The large number of connected clients incurs high computation overhead and acute bandwidth congestion, affecting fog-service reliability. This circumstance also deters the mobile applications' performance, which is contradictory to the aim of the fog paradigm [7], [8]. On the other hand, a huge pool of smart vehicles are often left unexploited in the parking lot of shopping centers for extended periods of time. These vehicles can be envisioned as auxiliary computing resources that serve myriad users [9].

In this article, we propose the use of the innovative approach of FVC to enhance the scalability of a fog computing infrastructure. FVC takes advantage of a dynamic group of vehicles to boost the computational power and decrease the latency of fog computing. The idea of FVC offers striking advantages for both fog-service providers and vehicle owners. For example, the fog-service provider is relieved of the substantial up-front expenditure to scale up the fog infrastructure. On the other side, vehicle owners are entitled to receive a variety of attractive incentives such as free parking, free Wi-Fi, or free shopping vouchers in exchange for providing their vehicle's computing resource to the fog provider.

The main contributions of this article 1) present the new concept of FVC to overcome the existing issues of fog computing, 2) describe the first architecture for FVC and explain its components in detail, and 3) implement the proposed architecture and compare it to fog computing to prove its effectiveness.

Fog Vehicular Computing

Fog is a cutting-edge paradigm of computing that extends conventional cloud computing to the edge of the network. In the fog model, the data processing and analytics take place in the proximity of end devices, where huge quantities of data are generated. Locating the services as closely as possible to end devices eliminates the unnecessary network hops and expedites mass data transmission originated from IoT gadgets. From a security perspective, the fog strategy mitigates the risk of data exposure to adversaries across unduly large distances between end devices and cloud data centers, thus helping to protect the secrecy of sensitive data. Deploying a fog architecture provides a more reliable service to end users. In particular, the fog model is imperative for assorted types of services characterized by latency sensitivity. In general, the fog computing model is characterized by low latency, geographical distribution of devices, mobility, and a large number of nodes [10].

Although the fog computing environment has been designed to provide reliable service to real-time applications, in certain scenarios, the fog experiences an excessive number of demands that are often far beyond its capacity; consequently, sporadic performance issues arise. For example, in a sizable multistory shopping mall where a large number of patrons are hosted on a daily basis, a fog service is established to provide a variety of services to customers and visitors while they spend time there. During weekends, public holidays, and specifically during peak hours, thousands of customers are roaming the shopping mall and using the provided services. However, the limited computation resource in the fog inhibits this use.

The simplest approach to overcome this problem is to scale up the computing infrastructure to accommodate extra demands. However, applying such a policy is subject to a substantial amount of up-front expenditure. Moreover, because the demands are not constant over time, the overprovisioned computing resources may be useless during off-peak periods. On the other hand, present and future vehicles are well-equipped with powerful onboard computers for the sake of a safe, convenient, and pleasant driving experience. These onboard computers have notable capabilities, such as global positioning systems (GPSs), cameras, bandwidths, data recorders, sensors, and actuators. Aside from managing the vehicle's functionality, the computers also help with better navigation.

A pool of smart vehicles parked at a shopping mall for a long period of time has tremendous computing power that remains unexploited under normal conditions. This large array of intelligent vehicles can be envisaged as supplementary computing, storage, and networking resources and also leveraged to serve the enormous number of requests that are impossible to regularly handle with fog resources [11]. While they are parked, the vehicles are presumed to be plugged into electricity and network outlets. Furthermore, the owners of the vehicles are offered a variety of attractive incentives described in the "Internet of Things" section in return for lending the computing resources of their vehicles to the fog-service provider. To avoid any congestion in the entire parking area, only a certain zone is dedicated to the vehicular cloud. The maximum capacity of the FVC zone is determined according to the predicted need of the computational resources. Further analysis and modeling are required to predict the actual number of vehicles that are parked in the specific parking facility at any specific time. Due to the dynamic nature of the vehicular cloud, resorting to the computing resource of vehicles does not result in overprovisioning, in contrast to the regular fog case. This is because the number of patrons and, consequently, their demands are intuitively commensurate with the number of parked vehicles. The more demands that are received from users, the more vehicles there are to serve. Moreover, we predict a transparent and seamless picture of fog services available to typical users of the fog in shopping centers. In other words, users are unaware that their computational tasks are processed by the fog's own infrastructure or delegated to the supplementary vehicular cloud. Figure 1 depicts an infrastructure of FVC to support IoT applications.

Table 1 shows the importance of the proposed architecture by comparing the fog, FVC, and vehicular cloud computing (VCC) based on the following attributes.

- Computing power indicates that the two systems have the identical computing power if they can run and execute the same program and produce the same results at the same time.
- Lifetime refers to the duration during which all components of the system are working properly.
- Decision making is at the heart of the system and decides its reaction to different situations, i.e., addressing the received tasks by assigning the available resources.

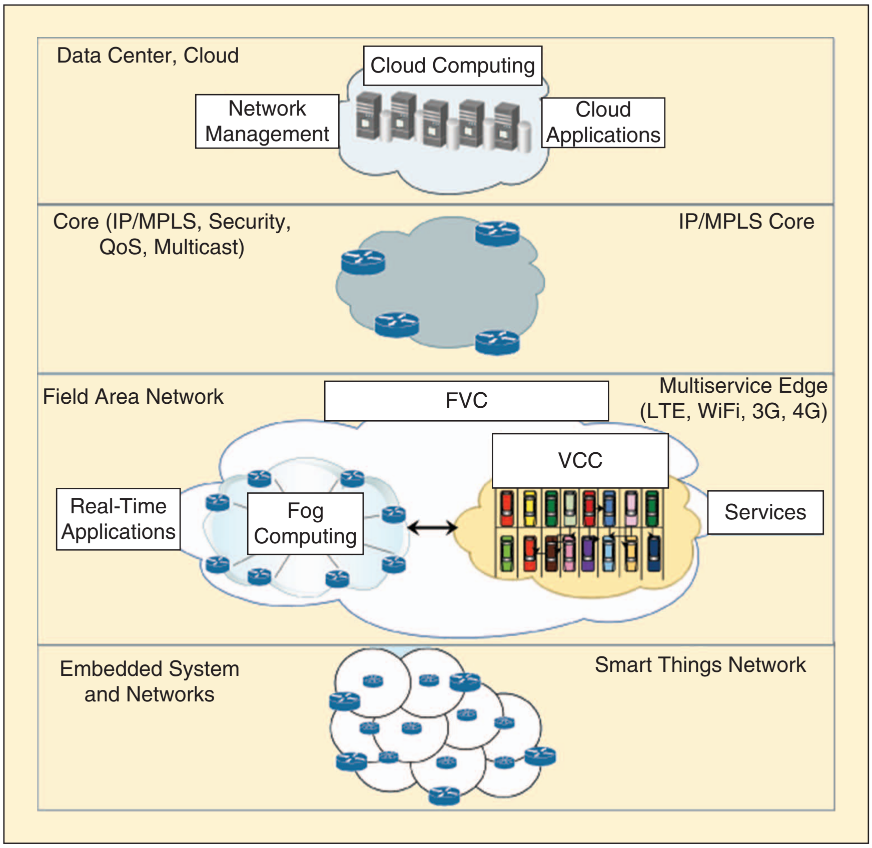

FIGURE 1 The infrastructure of the FVC for the IoT. IP: Internet protocol; MPLS: multiprotocol label switching; QoS: quality of service; LTE: long-term evolution; WiFi: wireless fidelity; 3G: third generation; 4G: fourth generation.

TABLE 1 A comparison of FVC, fog, and VCC.

Features	FVC	Fog	VCC
Computing power	High	Moderate	High
Lifetime	High	Moderate	High
Decision making	Distributed and local	Local	Distributed
Implementation cost	Moderate	Low	Moderate
Storage capacity	High	Low	High
Latency	Low	Moderate, depending on task	Moderate

- Implementation cost refers to the total cost for implementing the entire system.
- Storage capacity indicates the existing resources that are available for temporary or long-term data storage.
- Latency refers to the time delay between assigning tasks and receiving results.

Use Case: A Smart Modern Shopping Center

This use case refers to a fog computing service deployed in a large shopping center. The service is established on the basis of underlying network switches and set-top boxes, along with an array of parked vehicles available as a supplementary computing resource to serve the myriad customers inside the mall. The smart modern shopping center (SMSC) deals with this group of smart vehicles, and each vehicle has a processing unit, memory, storage, and network connection. The objectives for SMSC are as follows.

- A reliable service for fog users would be available, particularly with regard to latency-sensitive applications such as video streaming.
- Online gaming, an entertainment service by which customers can enjoy multiplayer online games, would be enabled.
- Safety instructions would be provided, such as those needed in an emergency evacuation from a fire or other disaster. Such a system would guide trapped individuals with mobile devices to the nearest and safest escape route.
- The exact position of a vehicle in a vast parking area could be pinpointed, a hassle-free option for locating a car that also provides a visual route to reach the vehicle.
- A diverse array of shopping information would be delivered in a timely fashion. The information can be very simple, e.g., the location of a store that offers a specific product such as apparel, dining, or entertainment. Customers can also access many other applications, including the latest store promotion, a comparison of prices to prepare a shopping list, and direct contact with customer services.
- Access camera surveillance systems for security and safety would be provided. Fog computing can provide a copious amount of resources to archive and process the collected footage. Specifically, when it is necessary to carry out an automated real-time video stream analysis including face detection, object counting, object tracking, and vehicle recognition, FVC capabilities can be leveraged to cope with performance issues.

FVC Architecture

Scarborough Research and Arbitron [12] conducted a survey on teenage mall shoppers and found that 68% of teens spend more than two hours at malls and 95% of adult shoppers spend one hour shopping at malls. This shows that a plentiful number of customers visit the malls every day, leaving their cars in the parking lots. Here, we present the novel architecture for FVC (based on the parked-vehicle resources in shopping centers) and describe its components in detail. Figure 2 shows the layered design of the FVC software framework and its architectural components. The FVC architecture consists of three main layers: application and services, policy management, and abstraction.

Application and Services

The first layer of the FVC architecture, application and services, provides a variety of real-time applications for end users based on collected data from the deployed sensors in the inertial navigation system, parking environment, shopping center building, and inside vehicles (i.e., environmental, health, activity, and vehicle recognition). This layer also offers the following new services to the end users.

1) *Information as a service and entertainment as a service:* These provide useful information about events and emergency circumstances for customers who are connected to the FVC. The center can also offer entertainment such as online games and commercial movies. The aim of these services is to increase the welfare and safety of customers.
2) *Network as a service*: Customers who have an Internet connection can offer this facility to the other customers who do not have an Internet connection for as long as they need it. This connection can be provided through the mobile devices or the stationary infrastructure on the roads. Many such resources can be useful for the clients, especially in emergency

situations. Such essential resources, which can be accessible for those who are interested as long as they have Internet access, can be advertised at the center when the owner agrees to do so.

3) *Storage as a service* (*STaaS*): The concept of STaaS refers to the situation in which the FVC has plenty of free onboard storage capability and this storage is present for clients who need additional storage for different purposes, e.g., running the applications that require high-storage resources, making a temporary backup, or using person-to-person applications over an extended period of time.

4) *Computation as a service*: The U.S. Department of Transportation recently released information showing that most of the registered vehicles (about 256 million) in the United States are parked for several hours per day in parking lots, garages, and centers [13]. These vehicles have enormous unused computational resources and present an opportunity to be exploited in the VFC structure as a new service for those clients who wish to augment the computation resources of their mobile devices to perform huge computational tasks.

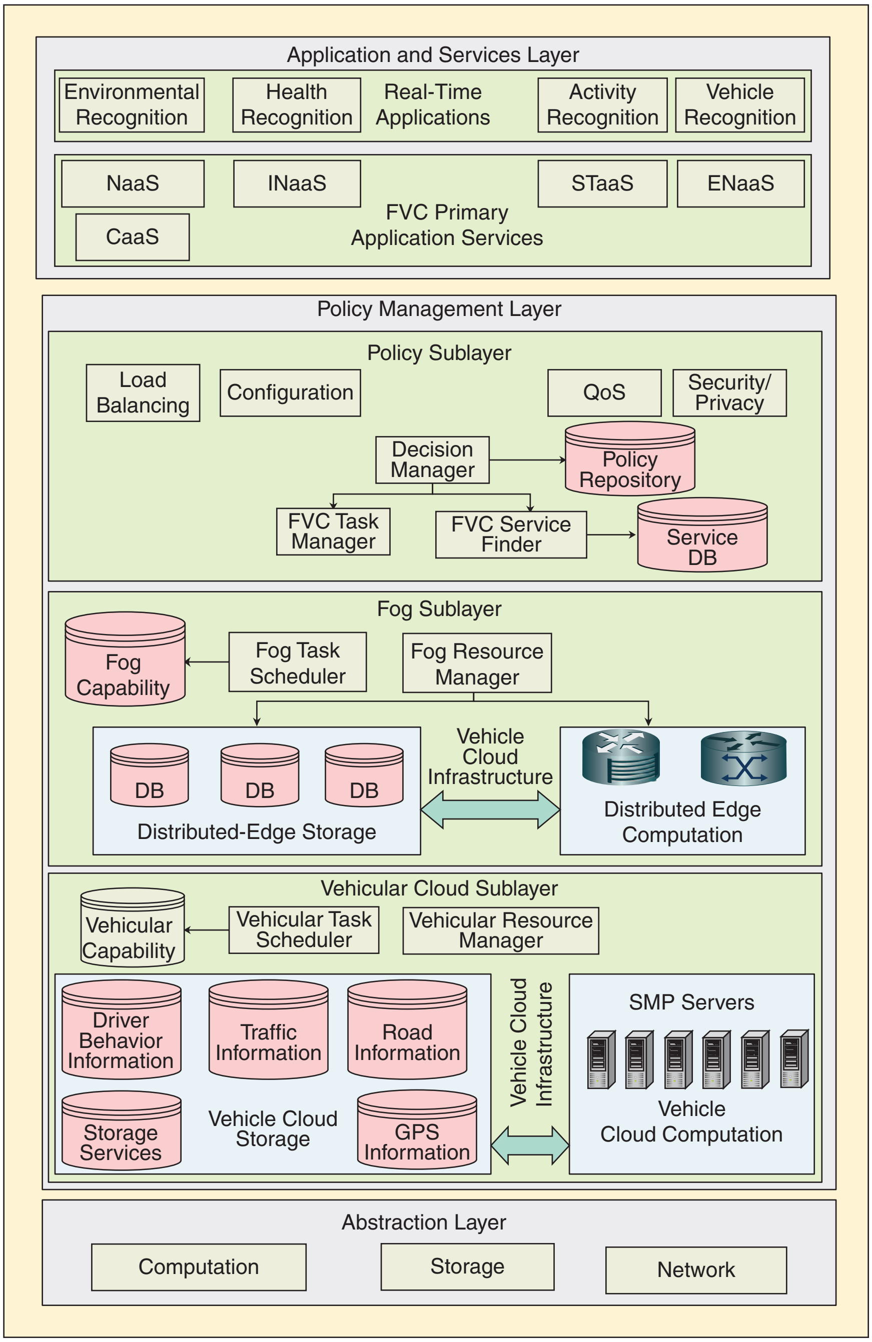

FIGURE 2 The general architecture of the FVC. NaaS: network as a service; INaaS: information as a service; CaaS: computation as a service; ENaaS: entertainment as a service.

Policy Management

The policy management layer at the heart of the FVC system is responsible for managing the life cycle of tasks by allocating to them the appropriate computation and storage resources. This layer, also responsible for dealing with basic issues such as monitoring the system state dynamically, consists of three sublayers: policy, fog, and vehicular cloud.

Policy

Policy is a centralized layer of the FVC architecture that is interconnected with both fog and vehicular cloud sublayers to assign tasks and resources dynamically. In other words, all of the clients' services must be checked by this sublayer and delivered to the fog or vehicular sublayers based on particular policies and their situations. The following are the main components of the policy sublayer.

- *Load balancing*: This refers to the threshold for a maximum number of vehicles, clients, central processing unit (CPU) load, and connections. The number of required vehicles can be established on the basis of the number of users and submitted tasks to the FVC.

IN GENERAL, THE FOG COMPUTING MODEL IS CHARACTERIZED BY LOW LATENCY, GEOGRAPHICAL DISTRIBUTION OF DEVICES, MOBILITY, AND A LARGE NUMBER OF NODES.

- *Quality of service*: This is used to set the criteria related to computing, network, and storage such as rate of delay, computation cost, and communication cost.
- *Configuration*: This refers to the necessary configuration of different devices and services that are supported or presented by the FVC.
- *Repository*: This is a database (DB) for storing the set of policies and rules used by the decision manager entity for handling security, performance, and network requirements during the decision-making step.
- *Security and privacy*: These are responsible for providing a secure environment for users by using different techniques, including determining user access control to requested applications and services, enhancing privacy for data isolation, and preserving the confidentiality and integrity of the outsourced data.
- *Service DB*: This DB presents the list of active services that are provided by fog or vehicular nodes.
- *Decision manager*: A main component of the policy manager layer, the decision manager is in charge of making decisions with regard to data collected from other components. The decision manager consists of two parts:
  1) *FVC service finder*: This module analyzes the set of active services in the service DB to identify the best service to satisfy the client's request.
  2) *FVC task manager*: When a task is received by this module, the FVC task manager decides to assign the task to the fog or vehicular section based on the predicted time needed to perform the task in each of the parts. In the next section, we explain more about the decision-making process in this component.

Fog

Located within the vicinity of the clients, fog is the second sublayer of policy management. Fog is able to serve a limited number of user needs based on its resources. The main components of the fog sublayer are as follows:

- *Fog capability DB*: This repository consists of unassigned fog nodes and their capabilities of performing the different services.
- *Fog task scheduler*: This component checks the list of unassigned fog nodes to find the appropriate nodes that can satisfy the requested service from the available nodes in the fog clusters.
- *Fog service manager*: This component updates the list of available fog nodes or clusters when a task is assigned to them. The fog task scheduler must also check the policy repository to identify network configuration and service policies before assigning the requested services to the unallocated fog nodes.

Vehicular Cloud

The vehicular cloud, the final sublayer of the policy management layer architecture, contains many resources for augmenting the fog sublayer by supporting the services that require huge computation. This sublayer consists of the following components:

- *Vehicular capability DB*: This repository stores a list of existing vehicular clusters and their capabilities.
- *Vehicular task scheduler*: This component is responsible for assigning the computational task to the available clusters or vehicles in a cluster.
- *Vehicular resource manager*: The component is responsible for identifying and managing any modification of vehicular resources dynamically, the vehicular resource manager also locates the network configuration from the policy repository and updates the service DB.

Abstraction

The abstraction layer is responsible for concealing the heterogeneous platform of the FVC and revealing a monotonic interface for monitoring, provisioning, and managing the physical resources, such as memory, CPU, and network. This layer can also be used to control different operating systems, hypervisors, and services on physical machines. Moreover, the abstraction layer has adequate capabilities to carry out the visualization technique for supporting multitenancy and execute multiple operating systems and services on physical machines for cultivating resource use.

Protecting the security and privacy is a crucial role of the abstraction layer. Different methodologies can be used in this layer to ensure the data integrity, confidentiality, and resource isolation for diverse clients. Homomorphic secret sharing, zero knowledge proof, and attribute-based cryptography are some methods that can safeguard the confidentiality, integrity, and access control in the FVC architecture [14], [15].

The Decision-Making Process in FVC

The process of decision making has an important role in the FVC architecture and determines the best way to fulfill the requested services efficiently based on the available resources, information about the demanded service, and its parameters. Figure 3 illustrates a general decision-making process in FVC. This process is carried out in three steps, outlined in the following sections.

Decision Manager

The main part of the decision-making process in the FVC architecture is the decision manager, which is responsible for computing the completion time and assigning the task to the required sublayer. Once a service request is received by the decision manager, the FVC service finder searches the list of active services in the service DB to satisfy the client request. If the available services are unable to perform the task, the decision manager must assign the task to the fog or VCC sublayers. However, before delegating the task, the decision manager must calculate the completion time of the task in each sublayer. To achieve this goal, the decision manager asks the VCC and fog sublayers to provide the computational complexity of the task execution. This complexity is presented by aN^b, where a and b are two constant integers and N indicates the problem size. Moreover, the fog resource manager and VCC resource manager must frequently update the service DB about their workloads, available services, and computation speeds by using mega-floating-point operations per second from a simple benchmark. As a result, the decision manager determines the completion time of a service with a particular problem size in each of the sublayers by estimating the 1) computation cost of the task by using the received information (last status of resources and the service complexity) and 2) communication cost of sending and receiving data. In the following text, we describe the task scheduling process in each of these sublayers.

Fog Sublayer

Fog computing has a decentralized architecture, which means it is usually used for distributed applications with the aim of decreasing the latency. To achieve this goal in the proposed architecture, the decision manager delegates the task to the fog sublayer when its completion time is less than the VCC sublayer. This type of task, not very complex, often requires less computation cost. After offloading the task by the fog sublayer, the task scheduler component of fog, which is responsible for allocating tasks to the fog clusters, searches the nodes and capabilities DB to identify a list of unassigned nodes that can satisfy the task. The task scheduler transfers the list of nodes to the resource manager component to provision the task in the fog. In addition to updating the list of fog nodes and capabilities, the resource manager must assign the task to the nearest nodes to fulfill the low-latency object. However, selecting the closest nodes is not always optimal from the point of view of the FVC provider because of the unbalanced resource use issue that occurs due to the overloading of some nodes. Moreover, the fog layer may include a variety of nodes with heterogeneous hardware specifications and energy prices, which incurs different computation cost. As a result, it is necessary to investigate the task scheduling optimization for fog computing to obtain the most efficient result, a complex process that is beyond the scope of this article.

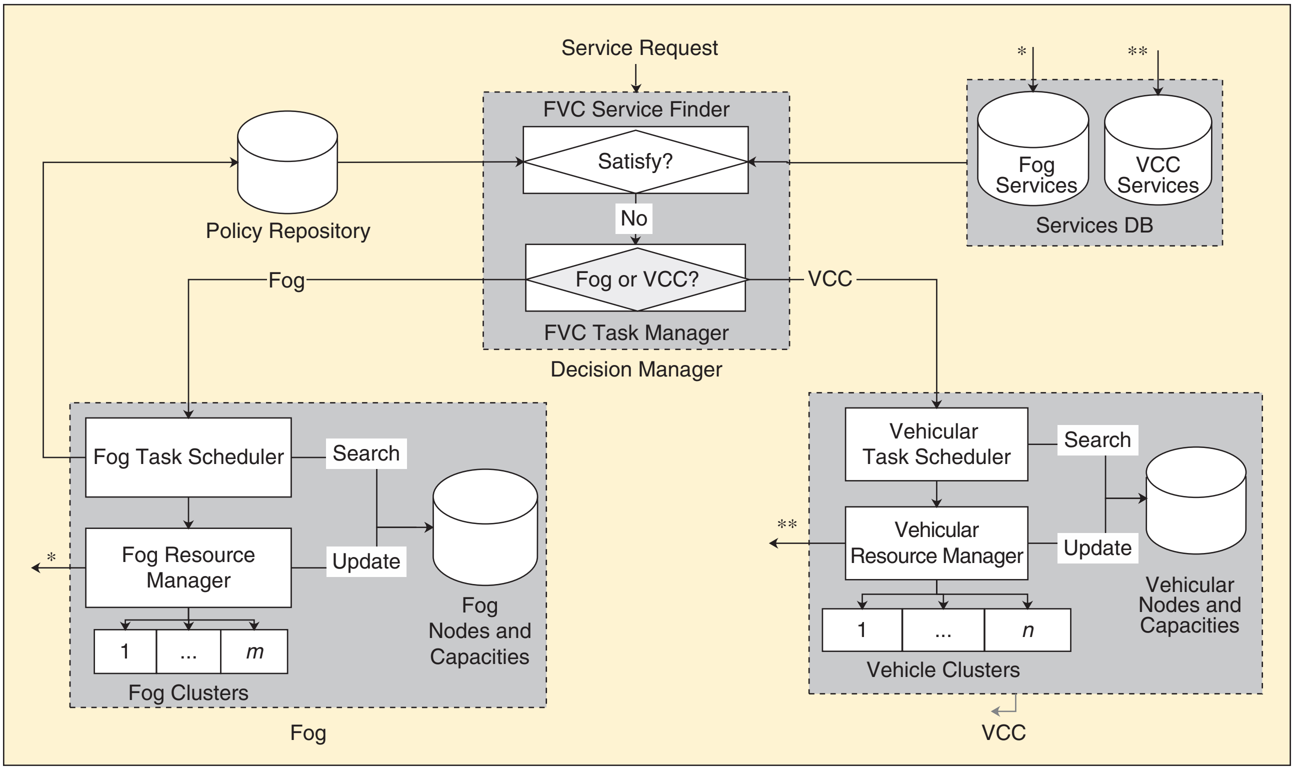

FIGURE 3 The decision-making process in FVC. The arrow with one star (*→) connects the fog resource manager to the fog services; and the arrow with two stars (**→) links the vehicular resource manager to the VCC services.

VCC Sublayer

Today, a huge number of parked vehicles are geodistributed in different indoor or outdoor parking areas, such as street parking, shopping mall parking, and office parking. The collaboration of these parked vehicles provides plenty of idle computational resources that can be used by the FVC architecture to deal with complex tasks that necessitate large computational competence. The first component of the VCC sublayer is the task scheduler, which is responsible for allocating the tasks to individual clusters or even to individual vehicles. If assigning the task to the VCC sublayer translates to quicker completion time, the vehicular task scheduler can seek out the vehicular nodes and capabilities DB to identify a list of available resources and then allocate the task. However, this DB must always be updated by the resource manager because the VCC is dynamic in nature in terms of computational resources. In other words, the preparation of the list of available resources depends on predicting the availability of vehicles as computational resources in the parking lots. One of the main components of the VCC sublayer is the resource manager, which is responsible for allocating the physical resources. Moreover, the resource manager has to control the clusters by

- following the available resources
- identifying new resources by arriving new vehicles
- migrating tasks from a vehicle before leaving the parking lot
- performing load balancing between the existing clusters.

The resource manager must also update the fog-services DB after allocating or releasing the resources.

Table 2 *The system configuration.*

Parameter	*Value*
Topology	LAN, fully connected
Number of tasks	1–5
Number of VCC nodes	15
Number of fog nodes	15
Bandwidth	1,024 Mb/s

LAN: local area network.

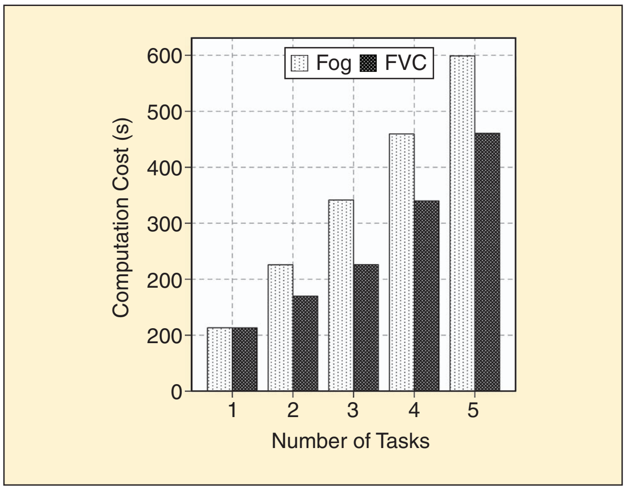

Figure 4 The comparison of the computation cost of different numbers of tasks in fog and FVC.

Results and Discussion

To investigate the effectiveness of the proposed method, we carried out a simple experiment using Openstack to implement a fog and VCC. We used Java on desktop computers with an Intel Core i5-2450 M processor at 2.5-GHz and 6-GB of random access memory. We assumed the number of vehicles in the parking lot to be 15 at the beginning of the simulation. When drivers decide to arrive at the parking lot (e.g., at shopping center parking), there is an opportunity for them to use the free accommodation by parking their vehicles in the VCC zone for at least two hours. This gives us permission to use the unused resources of the vehicles. As a result, we assumed the constant arrival rates and departure rates for the vehicles. Table 2 shows the required parameters to configure the system. We also defined a task to compare the performance of fog and FVC, in which each user is able to select a file with a size of 10 Mb and can ask the server to compute the algebraic signature of this file, which consists of the following steps: 1) dividing the file into 12,500 blocks, 2) computing the algebraic signature of each block with length 256 b, and 3) integrating the signatures to generate the signature of the file. The algebraic signature is computed along with the groundwork for defining multiplication by using the Galois theorem $GF(2^g)$ as a polynomial multiplication modulo, where g can be 16 b (half word) or 32 b (full word) [15].

We evaluated the performance of the fog and FVC on the basis of the following metrics:

1) *computation cost*: the time required to compute the algebraic signature of a file, including file division, signature generation, and signature integration steps
2) *communication cost of the fog sublayer*: the amount of data that transfers between the decision manager and fog sublayer
3) *communication cost of the VCC sublayer*: the amount of data that transfers between the decision manager and VCC sublayer.

In the first scenario, we checked the effectiveness of the proposed architecture by considering a situation in which a number of visitors at a shopping center are using the existing fog or FVC of the center to perform their tasks (e.g., generating the algebraic signature of a file). Figure 4 clearly shows that by increasing the number of tasks, the time required to perform the computation in the FVC is less than that in the fog computing. This is because the FVC is able to delegate

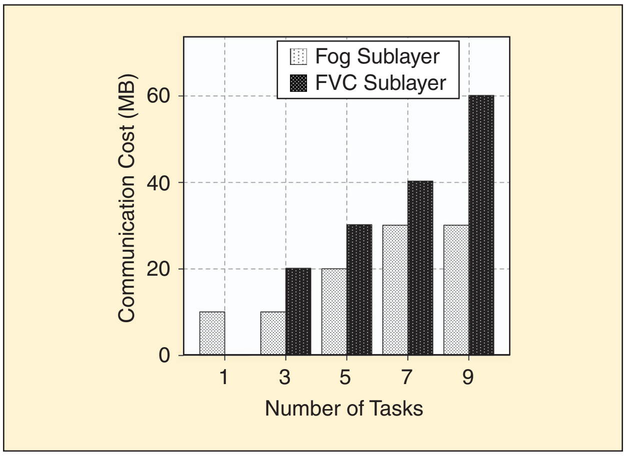

FIGURE 5 The comparison of the communication cost of fog and VCC sublayers.

the task to unused computational resources of parked vehicles in the shopping center as a VCC sublayer to decrease the latency and perform more tasks in comparison to fog computing.

In the next scenario, we checked the data flow in the FVC architecture based on the communication cost of fog and VCC sublayers, which has a direct effect on task completion time. Figure 5 shows that by increasing the number of tasks, the communication cost of the VCC sublayer increases more than the communication cost of the fog sublayer. This is because the decision manager assigned most of the tasks to the VCC sublayer due to better completion time.

Conclusions and Future Work

In accordance with the resource restrictions of fog computing, only a limited number of clients are able to use fog computing simultaneously. To alleviate this problem, we presented FVC as a new concept in which many unused resources of vehicles can be leveraged to augment fog computing resources. We also depicted the cross-layer architecture for FVC and elucidated its constitutive components along with their role in the construction of FVC. We explained a decision-making process as a very important procedure of this architecture and showed how the different types of services are distributed among vehicles of fog nodes. In the future, we plan to implement this architecture in a real environment and compare it using state-of-the-art methods to show its efficiency.

Task scheduling is one of the most important requirements to improve the efficiency of FVC. The main issue in providing an applicable task schedule is considering the role of the decision manager and its interconnection with the fog sublayer and VCC sublayer, making this a complex problem for the future. Another direction for future work is to focus on security and

IN THE FUTURE, WE PLAN TO IMPLEMENT THIS ARCHITECTURE IN A REAL ENVIRONMENT AND COMPARE IT USING STATE-OF-THE-ART METHODS TO SHOW ITS EFFICIENCY.

privacy as a main concern of users, especially when a huge number of computational tasks are to be delegated to the FVC and several users share the same set of resources in fog and vehicle nodes. Presenting a secure data-access control for FVC by using attribute-based data encryption and proxy re-encryption must be considered in future work.

Acknowledgment

This work was supported in part by the Natural Sciences and Engineering Research Council of Canada.

Author Information

Mehdi Sookhak (m.sookhak@ieee.org) received his Ph.D. degree in computer science from the University of Malaya (UM), Kuala Lumpur, in 2015. From 2012 to 2015, he was an active researcher in the Center for Mobile Cloud Computing Research and a faculty member in the Computer Science and Information Technology Department, UM. He is currently a postdoctoral fellow at Carleton University of Ottawa, Canada, and is funded by the Canadian Natural Sciences and Engineering Research Council. His areas of interest include cryptography and information security, mobile cloud computing, fog computing, vehicular cloud computing, data storage security, access control, and distributed systems.

F. Richard Yu (richard.yu@carleton.ca) received his Ph.D. degree in electrical engineering from the University of British Columbia, Canada, in 2003. From 2002 to 2006, he was with Ericsson in Lund, Sweden, and a start-up in San Diego, California. He joined Carleton University of Ottawa, Canada, in 2007, where he is currently a professor. He received the IEEE Outstanding Service Award in 2016; the IEEE Outstanding Leadership Award in 2013; the Carleton Research Achievement Award in 2012; the Ontario Early Researcher Award (formerly Premiers Research Excellence Award) in 2011; the Excellent Contribution Award at the IEEE/International Federation for Information Processing (IFIP) TrustCom in 2010; the Leadership Opportunity Fund Award from the Canada Foundation of Innovation in 2009; and Best Paper Awards at the IEEE International Conference on Communications in 2014, Globecom in 2012, IEEE/IFIP TrustCom in 2009, and the International Conference on Networking in 2005. His research interests include cross-layer/cross-system design, connected vehicles, security, and green information and computer technology. He is a Senior Member of the IEEE.

Ying He (heying@mail.dlut.edu.cn) received her B.S. degree from Dalian Ocean University, China, and her M.S. degree from Dalian University of Technology (DUT) in communication and information systems, in 2007 and 2012, respectively. She is currently pursuing a Ph.D. degree at the School of Information and Communication Engineering, DUT. A visiting student at Carleton University, her current research interests include big data, connected vehicles, and machine learning.

Hamid Talebian (talebia@siswa.um.edu.my) received his B.S. and M.S. degrees in software engineering from Islamic Azad University, Iran, in 2003 and 2007, respectively, and his Ph.D. degree from the University of Malaya (UM), Kuala Lumpur, in 2013. He then joined the Mobile Cloud Computing Research Group, UM. His research interests include cloud computing security, digital forensics, steganography and watermarking, evolutionary computation, multiobjective optimization, computational intelligence, and bioinspired metaheuristics.

Nader Sohrabi Safa (N.Sohrabi-Safa@warwick.ac.uk) received his Ph.D. degree from the University of Malaya, Kuala Lumpur, in 2015. He is a research fellow at the Cyber Security Centre of Warwick Manufacturing Group, University of Warwick (UW), Coventry, United Kingdom. He completed his postdoctoral research at the Centre for Research in Information and Cyber Security, School of Information and Computer Technology, Nelson Mandela Metropolitan University, Port Elizabeth, South Africa. He currently works for the Privacy, Ethics, Trust, Accountability, Security, Internet of Things in the United Kingdom and protects data in industry. He also collaborates in the Global Research Priority, UW.

Nan Zhao (zhaonan@dlut.edu.cn) received his B.S. degree in electronics and information engineering in 2005, his M.E. degree in signal and information processing in 2007, and his Ph.D. degree in information and communication engineering in 2011, all from the Harbin Institute of Technology, China. He is an associate professor in the School of Information and Communication Engineering at Dalian University of Technology (DUT), China. From 2011 to 2013, he engaged in postdoctoral research at DUT. His recent research interests include interference alignment, cognitive radio, wireless power transfer, and optical communications. He has published approximately 70 papers in refereed journals and international conferences and is a Member of the IEEE.

Muhammad Khurram Khan (mkuhurram@ksu.edu.sa) is an associate professor and research and development manager at the Center of Excellence in Information Assurance at King Saud University, Saudi Arabia. He has published hundreds of research papers and is the recipient of numerous awards. His research interests include biometrics, multimedia security, digital data hiding, and authentication protocols.

Neeraj Kumar (neeraj.khumar@thapar.edu) received his M.Tech. degree from Kurukshetra University in Haryana, India, and his Ph.D. degree in computer science and engineering in 2009 from Shri Mata Vaishno Devi University in Katra, Jammu, and Kashmir. He is an associate professor in the Department of Computer Science and Engineering at Thapar University in Patiala, India. He was a postdoctoral research fellow at Coventry University in the United Kingdom and has published more than 100 research papers in leading journals and conference proceedings. His research is supported by the University Grants Commission and Tata Consultancy Services.

References

[1] F. Bonomi, R. Milito, P. Natarajan, and J. Zhu, "Big data and internet of things: A roadmap for smart environments," in *Fog Computing: A Platform for Internet of Things and Analytics,* N. Bessis, C. Dobre, Eds. Cham, Switzerland: Springer International Publishing, 2014, pp. 169–186.

[2] M. Sookhak, H. Talebian, E. Ahmed, A. Gani, and M. K. Khan, "A review on remote data auditing in single cloud server: Taxonomy and open issues," *J. Network Comput. Appl.*, vol. 43, pp. 121–141, Aug. 2014.

[3] L. M. Vaquero and L. Rodero-Merino, "Finding your way in the fog: Towards a comprehensive definition of fog computing," *ACM SIGCOMM Comput. Comm. Rev.*, vol. 44, no. 5, pp. 27–32, Oct. 2014.

[4] M. Aazam and E. N. Huh, "Fog computing: The cloud-IoT/IoE middleware paradigm," *IEEE Potentials*, vol. 35, no. 3, pp. 40–44, May 2016.

[5] S. Yi, C. Li, and Q. Li, "A survey of fog computing: Concepts, applications and issues," in *Proc. 2015 Workshop on Mobile Big Data: Mobidata obi*, New York, 2015, pp. 37–42.

[6] I. Stojmenovic, "Fog computing: A cloud to the ground support for smart things and machine-to-machine networks," in *Proc. Australasian Telecomm. Networks and Appl. Conf. (ATNAC)*, Melbourne, Australia, Nov. 2014, pp. 117–122.

[7] D. Zeng, L. Gu, S. Guo, Z. Cheng, and S. Yu, "Joint optimization of task scheduling and image placement in fog computing supported software-defined embedded system," *IEEE Trans. Comput.*, vol. 65, no. 12, pp. 3702–3712, Dec. 2016.

[8] X.-Q. Pham and E.-N. Huh, "Towards task scheduling in a cloud-fog computing system," in *Proc. 18th Asia-Pacific Network Operations and Management Symposium (APNOMS)*, Kanazawa, Japan, Oct. 2016. pp. 1–4.

[9] M. Whaiduzzaman, M. Sookhak, A. Gani, and R. Buyya, "A survey on vehicular cloud computing," *J. Network Comput. Appl.*, vol. 40, pp. 325–344, Apr. 2014.

[10] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the Internet of Things," in *Proc First Ed. MCC Workshop on Mobile Cloud Computing, MCC '12*, New York, 2012, pp. 13–16.

[11] S. Arif, S. Olariu, J. Wang, G. Yan, W. Yang, and I. Khalil, "Datacenter at the airport: Reasoning about time-dependent parking lot occupancy," *IEEE Trans. Parallel Distrib. Syst.*, vol. 23, no. 11, pp. 2067–2080, Nov. 2012.

[12] Scarborough Research/Arbitron Inc. (2009). Teen mall shopping attitudes and usage survey. [Online]. Available: http://www.scarborough.com/reports/teen-mall-shopping-insights.html

[13] U.S. Department of Transportation (2008). Intelligent transportation systems for planned special events: A cross-cutting study. [Online]. Available: https://ntl.bts.gov/lib/30000/30400/30470/14436.pdf

[14] M. Sookhak, A. Gani, H. Talebian, A. Akhunzada, S. U. Khan, R. Buyya, and A. Y. Zomaya, "Remote data auditing in cloud computing environments," *ACM Comput. Surveys*, vol. 47, no. 4, pp. 1–34, July 2015.

[15] M. Sookhak, A. Gani, M. K. Khan, and R. Buyya, "Dynamic remote data auditing for securing big data storage in cloud computing," *Inform. Sci.*, vol. 380, no. 20, pp. 101–116, 2015.

VT