Design on Publish/Subscribe Message Dissemination for Vehicular Networks with Mobile Edge Computing

Lu Hou*, Lei Lei†, Kan Zheng*
*Intelligent Computing and Communication Lab,
Key Lab of Universal Wireless Communications, Ministry of Education,
Beijing University of Posts and Telecommunications, Beijing, China, 100876.
†State Key Laboratory of Rail Traffic Control and Safety, Beijing Jiaotong University, Beijing 100044, China.
Email: houlu8674@bupt.edu.cn

***Abstract*—The safety messages in vehicular networks require to be transmitted as fast as possible. On the other hand, the high mobility of vehicles poses an other major challenge on efficient message disseminations. In this paper, we present a design on publish/subscribe message dissemination (DPSMD) for vehicular networks with Mobile Edge Computing (MEC). Our design consists of message brokers that provide message exchange services, message handlers that excavate the potential value of messages and message hubs that bridge different parts of the system. MEC can be made use of in DPSMD to provide low-latency message services for vehicle users, while wide-area message disseminations are also well supported. The detailed descriptions is then given, including topic designs, subscription, publishing processes and message validating. Finally, a testbed is built up to evaluate the performance of the DPSMD in terms of the average time of message distributions.**

I. Introduction

Vehicular networks have gained a lot of attractions in recent years. By deploying computing and communication modules in vehicles, they can be "smart" and able to exchange data with each other, humans or infrastructures. Based on that, several vehicular applications emerge to provide good driving experiences. Some of the applications are even life-critical, for example, the collision detection and avoidance [1]. These applications pose a major challenge on the latency of message dissemination. On the other hand, huge amount of data is generated in vehicular networks all the time. The producers of data may be very far away from consumers who are interested in those data. For example, the parking lots can produce parking information that are desired by drivers who are about to park their vehicles there. Thus, an efficient design for message disseminations is in demand urgently.

Traditional message services are usually deployed centralized in the cloud by using publish/subscribe pattern [2]. It is difficult to enable real-time message disseminations since messages need to travel through the core networks and the Internet before reaching servers [3]. On the other hand, handling tremendous messages generated by vehicles causes significant overhead and decreases the efficiency of cloud resource utilization. To solve the problem, Mobile Edge Computing (MEC) is suitable in vehicular networks to formulate a distributed computing paradigm that extends the cloud services to the edge of the network, whereas it focuses on the edge with a more general point of view and does not assigns a defined layer of the network architecture for edge servers [4]. The MEC server enables vehicle-related services and applications running as virtual machines (VM) and operating at the edge of mobile network in a flexible and efficient way.

In this paper, we discuss the challenges and present a design on publish/subscribe message disseminations (DPSMD) with MEC. The DPSMD mainly provides two categories of services. One is the instant message exchange between vehicles, pedestrians and infrastructures. The other is to well support third-party vehicular services and applications. We deploy message hubs both in the MEC and cloud servers to support message disseminations among different entities. Message brokers served as the basic components in DPSMD that multiple brokers are mounted on a message hub to provide message distribution services. DPSMD also enable service providers to deploy their applications by implementing vehicle message handlers in their own ways. At last, we build up a testbed to evaluate the performance of DPSMD under transmission time metric.

The rest of the paper is organized as follows. Section II discusses related works. The detailed designs of DPSMD is presented in Section III. Section IV shows the process of message disseminations in DPSMD. We illustrated the performance of DPSMD in Section V. Section VI concludes the paper.

II. Related Works

Many applications in vehicular networks are life-critical, making time be given the highest priority of consideration [5]. The emerging of MEC guarantees response time for these applications, because it provides services at the edge of vehicular networks that is very close to vehicle users [6]. Addressing on this issue, [7] provides a MEC-based architecture for vehicular networks that relies on cellular networks. Network slicing is deployed in this architecture to create a "V2X-Automotive"

978-1-5386-3920-7/17/$31.00 ©2017 IEEE

TABLE I
RESERVED TOPICS AND FUNCTIONS

Topic Name	Functions	Subscriber (Consumer)	Publisher (Producer)
$reserved/cross-hub	For CMH to receive messages that needs to be disseminated across EMHs	CMH	brokers or VMHs
$reserved/publish-{VMH ID}	For brokers to publish messages to VMHs via EMHs	VMHs	brokers
$reserved/subscription-{EMH ID}-{Broker ID}	For VMHs or CMH to publish messages to brokers via EMHs	brokers	CMH or VMHs
$reserved/subscription-info	For CMH to get informed of subscription information	CMH and EMHs	brokers
$reserved/update-vehicle-info	For brokers to upload vehicle information to CMH	CMH	brokers

slice for better services. Another research in [8] provides a multi-layer architecture for vehicular networks. The local cloud in the paper is built up with MEC. By using software-defined networks technologies, the architecture can provide efficient solutions for vehicular services.

Another challenge is the efficient message disseminations in vehicular networks where network topology can be changed frequently and unpredictably. Researchers in [9] considered vehicles as service providers and presented a data dissemination model using resources in vehicles. In general, publish/subscribe paradigm provides decoupling in time, space and synchronization between communicating entities, making it most suitable for vehicular networks. Therefore, many researchers have tried to deploy publish/subscribe pattern in vehicular networks. [10] provides an information dissemination framework in publish/subscribe pattern for vehicular networks. This pattern may also bring conveniences in typical scenarios. The authors in [11] put forward a publish/subscribe framework for charging management of electric vehicles. The charging stations publish state information while electric vehicles can make optimal chooses. Publish/subscribe pattern can also be helpful for service discovery in vehicular networks [12]. However, the latency requirements are hard to be satisfied, especially for those cloud-based architectures. To the best of our knowledge, no existing literatures have given solutions on message disseminations with MEC for the special requirements of vehicular networks.

III. DETAILED DESCRIPTIONS FOR DPSMD

MEC makes servers being deployed at the edge of the networks, i.e., near the RSUs. These MEC servers can be regarded as independent network elements complementing the existing vehicular networks. An isolated MEC server is connected to multiple RSUs via dedicated reliable fiber links. Because no core networks or Internet is delayed between RSUs with MEC servers, the response speed of MEC servers can be extremely fast. Also, MEC servers can connect and share information with cloud servers via Internet backbone to work in a cooperative manner. The latency of this backbone channel is apparently very high.

Based on the MEC framework mentioned above, DPSMD is designed with several major components, i.e.:

- Central Message Hub: The Central Message Hub (CMH) is beyond all the other components. The key functionalities of CMH is the coordination of Edge Message Hubs (EMHs) and other modules. Besides, it is responsible for message disseminations across multiple EMHs. An Location Register (LR) is deployed in CMH. It maintains the locations and affiliations of vehicles. This information shows which broker the vehicle is connecting with and is crucial for CMH to route messages towards correct EMHs. CMH is deployed on cloud, which is far away from the vehicle users. Therefore, the latency it takes to transmit messages through CMH is high.
- Edge Message Hub: The EMH is the key component in each edge network. It mainly maintains a message queue where messages are received from producers and waited to be consumed by consumers. Therefore, messages can be distributed among different brokers or Vehicle Message Handlers (VMHs). EMHs are deployed in MEC servers, making them able to provide fast services for vehicle users. The CMH also serves as the potential producer and consumer of EMHs. EMHs only provide a transparent channel for messages to be propagated that they never care about the content of the messages.
- Vehicle Message Handler: For some kind of vehicular applications, service providers need to excavate the potential value of messages. The VMH is competent to do such things. One EMH connects with several VMHs, each of which can serve several vehicular applications. VMHs receive messages from EMHs and provide value-added services. They have to publish back responses to EMHs in some situations. VMHs can also offer high-level applications, for example, Complex Event Process (CEP) or big data processing. They can even provide application programming interfaces (APIs) for more developers to integrate their applications in DPSMD.
- Message Broker: Each EMH connects with several message brokers. These brokers provide instant message exchange for vehicle users. Brokers can be deployed besides

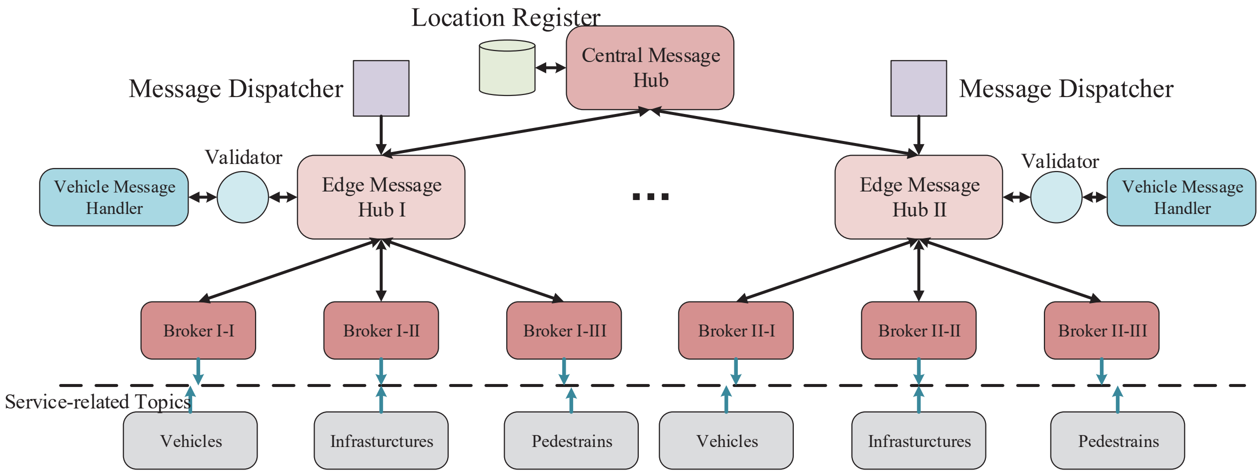

Fig. 1. The illustration of DPSMD

RSUs so as vehicles communicate with RSUs, they can exchange messages with each other through brokers. Therefore, brokers serves as the message "switchboard" and play as the bridges between vehicle users with the system.

Both CMH and EMHs have message dispatchers. Thus, message brokers or VMHs can produce data into message queues in EMHs or CMH, or consume messages from them. During this process, message dispatchers are responsible for distributing messages in the queue to the proper consumers. The relations between consumers and producers are established via topics which is similar with publish/subscribe pattern. In this way, message brokers and VMHs that are connecting with EMHs serve as the producers and consumers of the messages in the queue. On the other hand, they can also be considered as the publishers or subscribers.

A. Topics

The topics are ties between different entities in DPSMD. All topics can be divided into two categories, i.e., service-related topics and reserved topics. The service-related topics are defined by service providers. These topics can be subscribed or published on arbitrarily by vehicles, pedestrians or infrastructures whoever are interested in them. Similarly, the messages that are delivered over these topics are also service-related. DPSMD only offers transparent and reliable transmission for these messages.

Another category of topic, i.e. reserved topics, are predefined in DPSMD mainly to coordinate EMHs, CMH, brokers and VMHs. These topics are mainly used for different components to exchange important information among each other, for example, to exchange subscription information among all the EMHs. All reserved topics are summarized in Table I, and specific descriptions are given below:

1) $reserved/cross-hub:: is used for CMH to receive messages that need to span over multi-EMHs. It is subscribed only by CMH, while it can be published on by any brokers or VMHs via EMHs.

2) $reserved/publish-{VMH ID}:: Brokers can use this topic to publish messages to VMHs if these messages are required to be handled by them. Multiple VMHs are allowed so a VMH ID is needed to distinguish different VMH.

3) $reserved/subscription-{EMH ID}-{broker ID}:: is designed for brokers to receive messages from VMHs or CMH. Each broker is mounted on one EMH, thus, EMH ID and broker ID are both needed in this topic.

4) $reserved/subscription-info:: can help share subscription information among EMHs. Brokers use this topic to inform CMH of new subscriptions.

5) $reserved/update-vehicle-info:: This topic is subscribed only by CMH and published on by brokers. Locations of vehicles are updated in LR when CMH receives message from this topic.

During the bootstrap process of the architecture, all brokers need to subscribe "$reserved/subscription-{EMH ID}-{broker ID}". In the same time, each VMH needs to subscribe topic "$reserved/publish-{VMH ID}". Finally, CMH has to monitor messages on topic "$reserved/cross-hub", "$reserved/subscription-info" and "$reserved/update-vehicle-info". In this way, the system can work properly.

B. Subscription

Vehicles, pedestrians and infrastructures are free to subscribe any topics they are interested in even if the publishers of certain topics are far away. The subscription information can be shared among all the EMHs, enabling the message disseminations in different regions. Assuming a vehicle subscribes a topic to the broker-I-I, the broker-I-I then updates the topic cache and save this subscription information. In the same time, it publishes this information to the EMH-I through topic "$reserved/subscription-info" which is subscribed by CMH. Then, CMH stores the specific subscription information in LR

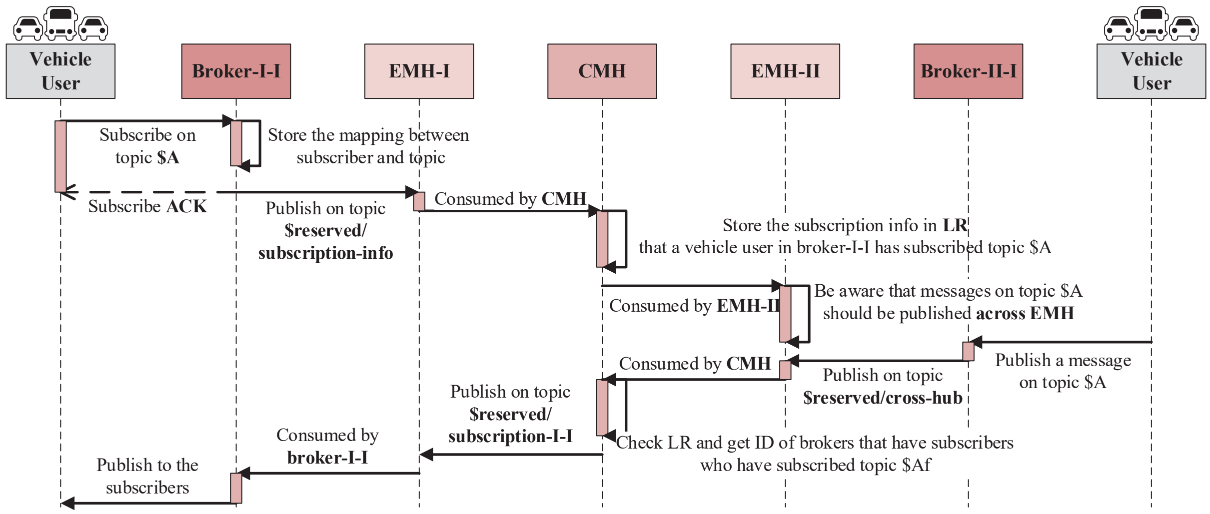

Fig. 2. The subscription flow

and keeps on forwarding the information to all other EMHs. All the EMHs then update their topic caches to store the information that a client in other broker has subscribed a topic. They have no needs to save the specific information because the CMH has done such things. When any message is published on this topic from other EMHs, they only need to produce the message to CMH via "$reserved/cross-hub". And the CMH is responsible for distributing it to correct EMHs. This subscription flow is illustrated in Fig. 2.

The vehicles move fast, leading to frequent changing of brokers they connect. Therefore, the subscription information in old broker is useless, and new broker needs to be aware of such information. This can be realized when vehicles upload their location information via topic "$reserved/update-vehicle-info". When CMH receives the location information of vehicles, it starts to update the broker ID that the vehicles are connecting, and change the subscriptions to the new broker. CMH also publishes two messages. One is for the old broker on topic "$reserved/subscription-{old EMH ID}-{old broker ID}" to inform it the cease of the subscription. The other is to notify the new broker to store the subscription information via topic "$reserved/subscription-{new EMH ID}-{new broker ID}". In this way, vehicles can subscribe the same topic without sending subscription request again when they reach service areas of other brokers.

IV. Message Disseminations and Validating

With the pre-designed architecture and topics, service-related messages can be disseminated inside a broker, to a handler or across different regions efficiently. On the basis of that, services for vehicular networks can be put into effect. On the other hand, with the support of MEC, the message exchange inside a EMH can be very fast. Therefore, low-latency services can also be supported in DPSMD.

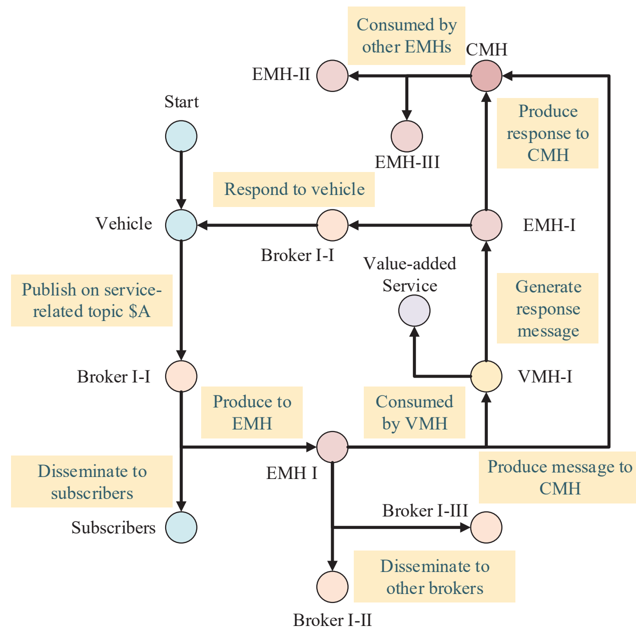

Fig. 3. An example of publishing process

A. Publishing

When a vehicle user publishes a message on a certain service-related topic, the broker firstly disseminates this message to subscribers it connects, as the example shows in Fig. 3. Then, the broker encloses this message inside a new message and produces it to the EMH via topic "$reserved/publish-{VMH ID}". The payload of this new message is actually the whole original message, while the topic of the new message is changed to the reserved topic name. Some meta data is also required to be included in the message, for example, the broker ID and the EMH ID. After the EMH receives the message, it checks the subscription table to see whether any subscriber from other EMHs has subscribed this topic. If so, the message dispatcher forwards this message to the CMH over topic "$reserved/cross-hub". After received messages from this topic, CMH queries the LR to get all the broker IDs that has subscribers who have subscribed the topic inside the message. Then, CMH publishes the messages to all the EMHs via topic "$reserved/subscription-EHM number-broker number". EMHs then forward the messages to all the target brokers by message dispatcher. The brokers analyze the messages and get the actual topic and data. They finally disseminate the real messages to the subscribers. In this way, messages can be disseminated over a wide region in a short time.

On the other hand, EMHs may forward the messages to the VMHs when VMHs subscribed the topic "$reserved/publish-{VMH ID}". After receiving the messages, VMHs can process them as the service providers design. For example, traffic management applications can collect traffic information, analyze and make guidances for the travels of vehicle users. If any response is necessary after message processing, VMHs generate response messages and publish them back via topic "$reserved/subscription-{EMH ID}-{broker ID}". The payload of this response message is also a whole message. In convenience of the requesters, the topic of responses can be defined based on the original topic. For example, to inform the acceptance or rejection of messages from topic "$service/advertisement", the VMHs can publish the response messages on either "$service/advertisement/accept" or "$service/advertisement/reject". These responses are then consumed by proper brokers, and disseminated to the subscribers who need them.

The response messages are probably required by subscribers from other EMHs. Therefore, the EMHs may also need to publish the response messages on topic "$reserved/cross-hub" to the CMH. These messages can be treated as normal messages, thus, CMH can dispatch them to other EMHs ordinarily.

B. Message Validating

In DPSMD, service-related messages can be made use of in two ways, i.e., exchanged among entities or processed by VMHs. We only provide transparent transmission for messages. Service providers can deploy their applications via implementing VMHs. However, to ensure validity and avoid exceptions, message validators are mandatory. Service providers need to define certain validators according to their applications. Message validating need to address on two aspects. One is to verify the type, integrity and validity of the message fields, the other is to impose restrictions on some special values. Different message formats have their dedicated schema definitions, for example, if Javascript Serialized Object (JSON) is used as the message format, the service providers could use JSON Schema[1] to build up a integrated validator.

When messages pass the validators, they reach the VMHs where value-added services are deployed. Service providers

TABLE II
EMULATION PARAMETERS

Name	Value	Name	Value
brokers	2	EMHs	2
CMHs	1	Subscribers	[5, 10, 20]
Publishers	[10, ..., 100]	CPU cores	2
Memory	4G		

can consume the data as they want in this module. Therefore, different providers may have different implementations of the VMHs. Message validators can give protection of the system from malformed or vicious messages.

After handling the messages, handlers need to give proper responses in some scenarios. These responses are firstly required to pass the validators. This is necessary because the architecture must make sure that response messages are also in correct forms to avoid system errors. Next, the response messages are published on topic "$reserved/subscription-EMH ID-Broker ID" to EMH, as previously stated.

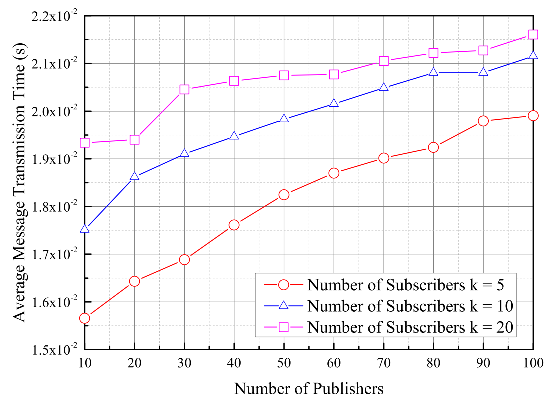

Fig. 4. The average transmission time of messages

V. PERFORMANCE EVALUATION AND ANALYSIS

A. Testbed

In order to evaluate the performance of DPSMD, we implement an example testbed that includes several brokers, EMHs, VMHs and a CMH. The Message Queue Telemetry Transport (MQTT) is chosen as the message protocol. Each broker is an instance of an MQTT server that is developed by Node.js. We use open-source Apache Kafka[2] as our EMHs and the CMH. Kafka maintains a message queue and works in produce/consume pattern which is similar with normal message protocols. Furthermore, we implement a specific VMH aiming at providing management of vehicle information. The VMH accepts vehicle data in JSON format, and responds with differences between last two commits of the information. Our testbed are deployed on a Virtual Machines (VM) located

[1]http://json-schema.org

[2]https://kafka.apache.org

at Northern Virginia (NV) of America from Amazon Web Services (AWS).

We use Python to emulate vehicles connecting and using services from our platform. The emulated vehicles publish messages that include some basic information. These data is forwarded to the EMHs and distributed to different subscribers. Up to 100 vehicles are emulated, each of which publishes 100 message packets every $50ms$ after it connects to the brokers. We count the message transmission time of each message and get the average of it. The detailed parameters are listed in Table II. The results are shown in the following content.

B. Evaluation Results and Analysis

All vehicles are emulated similarly, thus, we compute the average transmission times (ATTs) by averaging them among all vehicles in Fig. 4. It clearly shows the trend of ATTs and the performance of DPSMD. With the increase of number of publishers, the ATTs becomes larger. This shows the workload of servers. More publishers means more messages to be delivered. As can be seen, the maximum ATT is $21.5ms$ when 100 publishers send messages at the same time. As for subscribers, we can see same trend. When more subscribers have subscribed the topic, brokers need to deliver messages for more destinations. This causes the rise of ATTs.

VI. Conclusion

In this paper, we discussed a design on message disseminations in vehicular networks. Messages are distributed mainly in publish/subscribe pattern. The brokers and EMHs that are deployed at the edge of the networks can provide low-latency message exchange services, while the CMH ensures message disseminations in a wide area. Different service providers can put their services or applications into effect in DPSMD by implementing customized message validators and VMHs. DPSMD can guarantee low latencies for message exchange inside EMH due to the use of MEC. Furthermore, efficient message disseminations are also ensured in DPSMD.

VII. Acknowledgement

This work is funded in part by National Science Foundation of China (No. 61331009) and China Unicom Project.

References

[1] S. Datta, J. Haerri, C. Bonnet and R. Costa, “Vehicles as connected resources: opportunities and challenges for the future,” *IEEE Vehicular Technology Magazine*, vol. 12, no. 2, pp. 26-35, June 2017.

[2] K. Zheng, Q. Zheng, P. Chatzimisios, W. Xiang and Y. Zhou, “Heterogeneous vehicular networking: a survey on architecture, challenges, and solutions,” *IEEE Communications Surveys & Tutorials*, vol. 17, no. 4, Fourth quarter 2015, pp. 2377-2396.

[3] K. Zheng, H. Meng, P. Chatzimisios, L. Lei and X. Shen, “An SMDP-based resource allocation in vehicular cloud computing systems,” *IEEE Transactions on Industrial Electronics*, vol. 62, no. 12, Dec. 2015, pp. 7920-7928.

[4] “Mobile-Edge Computing Introductory Technical White Paper,” Sept. 2014.

[5] K. Zheng, F. Liu, L. Lei, C. Lin and Y. Jiang, “Stochastic performance analysis of a wireless finite-state Markov channel,” *IEEE Transactions on Wireless Communications*, vol. 12, no. 2, Feb. 2013, pp. 782-793.

[6] K. Zhang, Y. Mao, S. Leng, Y. He and Y. Zhang, “Predictive offloading in cloud-driven vehicles: using mobile-edge computing for a promising network paradigm,” *IEEE Vehicular Technology Magazine*, vol. 12, no. 2, pp. 36-44, June 2017.

[7] L. Li, Y. Li and R. Hou, “A novel mobile edge computing-based architecture for future cellular vehicular networks,” *IEEE Wireless Communications and Networking Conference*, Mar. 2017.

[8] K. Zheng, L. Hou, H. Meng, Q. Zheng, N. Lu and L. Lei, “Soft-defined heterogeneous vehicular network: architecture and challenges,” *IEEE Network*, vol. 30, no. 4, pp. 72-80, Aug. 2016.

[9] M. U. Farooq, M. Pasha and K. R. Khan, “A data dissemination model for Cloud enabled VANETs using In-Vehicular resources,” *IEEE International Conference on Computing for Sustainable Global Development*, Mar. 2014, pp. 458-462.

[10] T. Mishra, D. Garg and M. Gore, “A Publish/Subscribe Communication Infrastructure for VANET Applications,” *IEEE Workshops of International Conference on Advanced Information Networking and Applications*, Mar. 2011, pp. 442-446.

[11] Y. Cao, Y. Miao, G. Min, T. Wang, Z. Zhao and H. Song, “Vehicular-Publish/Subscribe (V-P/S) communication enabled On-The-Move EV charging management,” *IEEE Communications Magazine*, vol. 54, no. 12, pp. 84-92, Dec. 16.

[12] S. Noguchi, S. Matsuura and K. Fujikawa, “Performance analysis of mobile publish-subscribe service discovery on IPv6 over geonetworking,” *International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing*, July 2012, pp. 535-540.