A Study of Mobile Edge Computing System Architecture for Connected Car Media Services on Highway

Sangyub Lee[1,2], Jaekyu Lee[1,2] and Hyeonjoong Cho[2,*]
[1]Embedded SW Research R&D Center, Korea Electronics Technology Institute
Seongnam, South Korea
[2]Department of Computer and Information Science, Korea University
Sejong, South Korea
[e-mail: syublee@keti.re.kr, jae4850@keti.re.kr, raycho@korea.ac.kr]
*Corresponding author: Hyeonjoong Cho

Received February 27, 2018; revised June 26, 2018; accepted August 20, 2018; published December 31, 2018

Abstract

The new mobile edge network architecture has been required for an increasing amount of traffic, quality requirements, advanced driver assistance system for autonomous driving and new cloud computing demands on highway. This article proposes a hierarchical cloud computing architecture to enhance performance by using adaptive data load distribution for buses that play the role of edge computing server. A vehicular dynamic cloud is based on wireless architecture including Wireless Local Area Network and Long Term Evolution Advanced communication is used for data transmission between moving buses and cars. The main advantages of the proposed architecture include both a reduction of data loading for top layer cloud server and effective data distribution on traffic jam highway where moving vehicles require video on demand (VOD) services from server. Through the description of real environment based on NS-2 network simulation, we conducted experiments to validate the proposed new architecture. Moreover, we show the feasibility and effectiveness for the connected car media service on highway.

Keywords: Mobile edge computing system, media service, bus, vehicular network system, cloud service.

A preliminary version of this paper was presented at ICONI 2017, and was selected as an outstanding paper. This research was supported by a research grant from the IT R&D program of MKE/KIAT, the Korean government [R0004937, Development of Lightweight VOD system for the premium class bus].

http://doi.org/10.3837/tiis.2018.12.004 **ISSN : 1976-7277**

1. Introduction

Cloud computing has become synonymous with hosted services over the wireless communications. In reference [1-2], network distribution for mobile edge computing system is an end-to-end concept that encompasses network functions, radio accesses, and clouds for enabling decentric services. This powerful idea has been suggested based on ubiquitous and relatively low cost and high speed networks and virtualized, parallel and distributed computing and databases [3-5]. These days, cloud computing paradigm has been shifted to exploitation of excess computing capacity. In recent researches [6-10], vehicular cloud concept is introduced that leverages the excess on board resources of participating cars.

What distinguishes vehicles from nodes in a conventional cloud is the dynamic availability of resources. Clearly, some vehicles required downloading the data from the cloud server where they want to see the video streaming and web based information data. According to traffic condition, vehicles spend substantial amounts of time on the road and may be involved in dynamically changing situations: heavy traffic on road, some vehicles cannot access the VOD service from cloud server and waste the time and money in order to download data. With this in mind, we proposed the architecture based on bus as a Mobile Edge Computing Server (MECS) on highway to refer to a bridge server system that supports media files to vehicles existed in communication coverage directly. Cisco [11-13] recently delivered the vision of MECS to enable applications on billions of connected devices, already connected in the Internet of Things. Both cloud and MECS provide data, computation, storage, and application services to end-users. The distinguishing MECS characteristics include its proximity to end users, its dense geographical distribution, and its support for mobility. Services can be hosted at end devices such as set-top-boxes or access points. MECS provides low latency, location awareness. In addition, MECS improves quality of service for streaming and real-time application, and supports heterogeneity. In our view, existing for computing based architectures can be modelled by a simple level architecture where each vehicle attached to the bus as mobile edge computing device, and mobile edge computing devices could be interconnected, and each of them is linked to the cloud [14]. It is defined that the MECS model for the data communication between the bus and cars are on the high way and this environment shows more developers to bring their own applications and connectivity interfaces at the edge of the network. This proposed architecture of MECS allows application to run as close as possible to sensed actionable and massive data as possible.

The major contributions of this paper are as follows:

1. The state-of-the-art models of MECS using bus on highway are shown.
2. A new MECS architecture for VOD service and vehicular status information are proposed. These services were considered as next trend of in-vehicle entertainment system and autonomous car.
3. In traffic jam scenario, the proposed network simulation is examined and the load distribution performance of the proposed architecture is evaluated simply.
4. Through NS-2 real environment based simulation, it is found that the proposed architecture is needed on highway. It showed that network performance of proposed method is superior to ordinary communication system.

This paper is organized as follows. Section 1 provides motivation and an overview of related work Section 2 presents the proposed vehicular environment on highway and describes the architecture based on MECS using Bus. The performance analysis and a case study for proposed architecture are described in Section 3. In Section 4, our mechanism is evaluated simple simulation. In addition, we executed on detailed network simulation considering real communication environment using NS-2 in Section 5. Finally, we conclude this article in Section 6. As shown in **Fig. 1.**, people want to watch the video streaming or media contents on highway when caugh in congestion. Web based media services are accessible easily using mobile terminals. But, there is huge fee to users and data traffic jam to server. Thus we considered that how users can save their cost for the communication usage and how the data traffic or load can be reduced on data center cloud. Recently some buses are adopted network server to distribute media contents for passengers as described in **Fig. 3**. Thus, we proposed the architecture based on buses which include network system and define the MECS which plays on a role of bridge server. It supports media files to vehicles on highway.

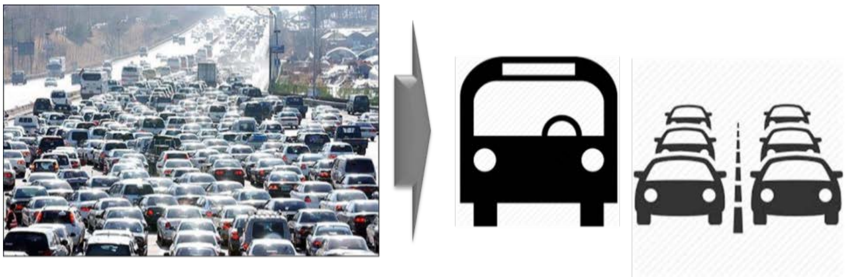

Fig. 1. Environment condition for the proposed mechanism

2. MECS Architecture for Connected Car Media Services

2.1 General Vehicular Cloud Model

The conventional vehicular cloud computing models are classified into three general service types: Network as a Service (NaaS), Storage as Service (STaaS) and Cooperation as Service (CaaS). The most of vehicles spend substantial amount of time on the road and may be involved on a daily basis in various dynamically changing situations, ranging from normal traffic to congestion, to accidents, to other similar traffic-related events. In general case, as described in **Fig. 2**, all vehicles were connected with data center cloud directly. Recently consumers want to receive multiple data flows from Data Center Cloud (DCC) and simultaneously send the own status data to the cloud. However, for given the road traffic condition, it is sometimes hard to communicate with DCC due to network congestion in same area.

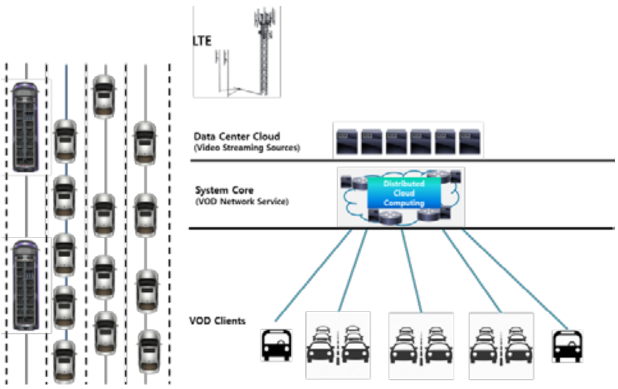

Fig. 2. General Vehicular Cloud Model

2.2 Connected Car Media Service Model

This section proposes a new architecture of a MECS for vehicles on highway. Prior to a detailed description of the proposed architecture, we explain the background of this research environment. In Korea, buses have to run legally at the left sideline on highway. As growing demand of media service and safety driving for passengers and drivers, most vehicles have to be equipped with communications system unit such as LTE module or Wi-Fi network device. Recently some buses are adopted network server system for media service described in **Fig. 3**.

Fig. 3. Network Server in Bus

This paper proposes a system architecture where a bus and cars play roles of MECS and client nodes, respectively. The proposed media services has own unique mechanism differently compared with traditional static cloud service model. In our scenario, if the moving car near the bus on the highway requests the video file stored in bus media server, bus transmits the data to the car directly without cloud server connection shown in **Fig. 4**. At that time, a car approaches within the communication range that enables the bus to transmit the media data by Wi-Fi, the user requests for the streaming data to the bus. After judging the connection status between the car and the bus, the bus sends the requested data to the car. If the car diverges from communication range, the car takes over the file from the next bus. In addition, the system core determines its order which bus is selected to support media file to moving vehicles in car. Bus Information Management (BIM) is enabled to share the connection history between system core and buses.

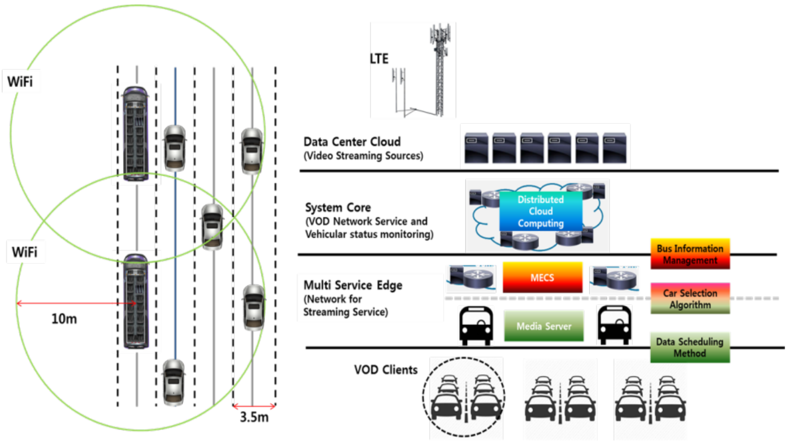

Fig. 4. Proposed Media Service Model

3. System Mechanism

MECS framwork is applied to implement system architecture concept as described in **Fig. 5**. It separates the multi service edge part and system core. Especially, vehicular status information data are collected by the system core using long term evolutioin (LTE) communications continuously. It will make the connection path which bus is appropriate to set up download link for cars. Its concept will resolve the main traffic issues in vehicular networks, intermittent connectivity, collisions and high packet loss rate by infrastructure communications system. **Fig. 5** shows the overall framework of the proposed system architecture which consists of three main components: Data Scheduling Method (DSM), Car Selection Algorithm (CSA) and Bus Information Management (BIM). Each component in the cloud service is in charge of managing accessed moving vehicles on highway with resource pool and then allocating or reallocating resources with respect to a given capacity in buses

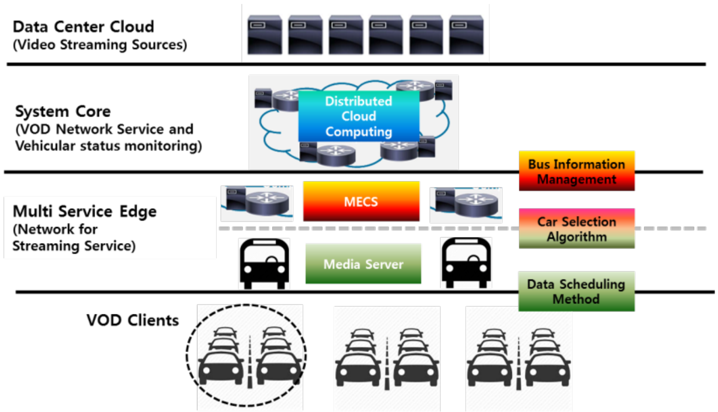

Fig. 5. MECS system architecture

3.1 System architecture

DSM manages the requested data load status of moving vehicles along with bus. It is depended on whether which car is located in within the transmitted power range or not and how far the data load required for the accessed car. According to the capacity of bus, DSM considers the load density for the limited bus capacity. Then, bus sends own status information for traffic capacity to cars located within communication range. It indicates which car will be available on data receiving and keeps communication connection. BIM shares this information with other buses from the CSA. It will be able to prevent the loss of connectivity that defines drop of service data or exceed resource throughput between cars and buses.

3.2 Data Scheduling Method (DSM)

We define the data load expressed as a load density. The area $L \in R^2$ is bounded. There are N_{bus} within this area. N_{bus} is indicates the number of buses. We focus on the downlink case, where there are N_{bus} over a circular area with radius R. The set of buses are defined as B. Symbol i represents the index of N_{bus} within B and u represents location within L. The users arrived at location u following the Poisson process with a rate $\lambda(u)$. The traffic size is independently distributed with mean $1/\mu(u)$. The traffic load density is defined by $\gamma(u) = \lambda(u)/\mu(u)$. We define $C_i(x_i, u)$ as the data rate of a user location at u served by target bus, x_i stands for location of target bus. For simplicity, we use Shannon capacity to model the transmission capacity; $C_i(x_i,u)=\log_2(1+SINR_i(u))$ Where SINR is the received signal to interference plus noise ratio at location u for the signal from target bus. Since we assume that the entire band of the buses is orthogonal, no interference is considered. $SINR_i(u)=SNR_i(u)=P_iG_i(u))/\sigma^2$, where P_i denotes the transmission power of the target bus, $G_i(u)$ denotes the channel gain from the target bus to the location u. σ^2 is noise power. Thus, load density is defined by

$$\delta_i(x_i,u)=\frac{\lambda(u)}{\mu(u)c_i(x_i,u)} \tag{1}$$

Actually, the load of target bus is the result of user association. Then we assume that a user located at u simply selects the target bus $i(u)$ using the following rule given by:

$$i(u)=\operatorname{argmax} c_j(u), \forall u \in L, j \in B \tag{2}$$

This rule means user prefers to choose the target bus that has maximum data rate. This defines a spatial coverage partition $CP=\{A_i\}$, where A_i denotes the coverage area of target bus

$$A_i=\{u \in L \mid i=\operatorname{argmax} c_j(u)\}, \forall i \in B \tag{3}$$

Then the load of target bus is defined by

$$\rho_i=\int_{A_i}\delta_i(x_i,u)du=\int_{A_i}\frac{\lambda(u)}{\mu(u)c_i(x_i,u)}du, \forall i \in B \tag{4}$$

Target bus is stable only when $\rho_i<1$, if $\rho_i>1$, the traffic load will exceed the service capability of target bus and some vehicles will be dropped.

3.3 Car Selection Algorithm (CSA)

Depending on the load density for target bus, it observes the requested data of vehicles which are moving along with the target bus. If the target bus's load density is below one at the moment and vehicular speed changing rate between bus and car is small, bus selects the car in order time of request data. And then bus gives the ordering tag to the connected car. It is necessary that checking the distance between target bus and users consistently. According to the tag, the link adaptation and data exchange will be sustained between them sequentially. Under the scenario in **Fig. 6**, where data file can be partitioned into several packets. Likewise each packet is given tag. If the car connected with target bus is out of communication range, MECS sends the out ranged car's information to Base station (BS). It supports the continuous data download regardless of position which car is accessed to bus. Described mechanism is represented in **Fig. 6**.

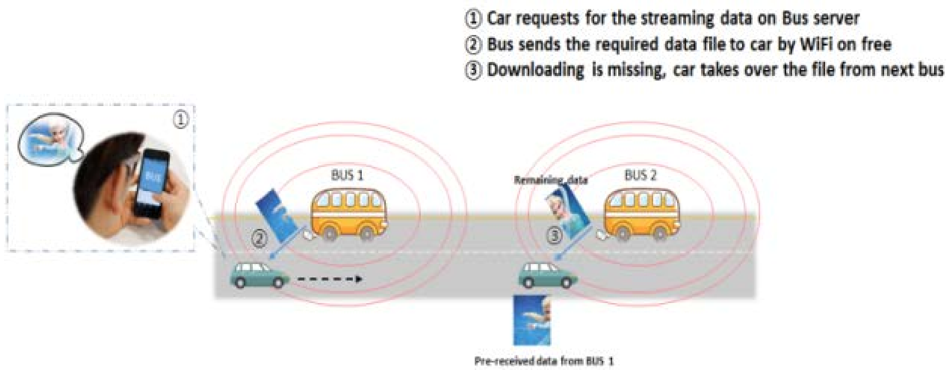

Fig. 6. CSA scenario

3.4 Bus Information Management (BIM)

In case, if the marked B car receiving the data from bus1 is going away from connected bus. BIM is used that which car might be accessed to specific bus to sustain the network connection. It evaluates the data load and checks the status of data link for cars as VOD clients before ceasing the connection from targeting bus. As described in **Fig. 7**, System core marked BS and bus remember data record of exchanging media contents and their time stamp. BS notifies the record information to another the bus2 since the marked B car is approaching the bus2 as shown. Thus for the point of view in BIM, it calculates the cumulated downloading numbers and service latency. If the marked B car is not connecting the bus2, then the BS only keep the tag how far the marked B car received media file. It reduced the data traffic and frame overhead effectively for BS. Then the marked B car connected with the bus2 again, the bus2 requests for the record tag to the BS directly and the bus2 searches for that file in their storage. Then, it checks the streamed time stamp for media file that the marked B car wants to get into their media device and send that file just before cutting their connection.

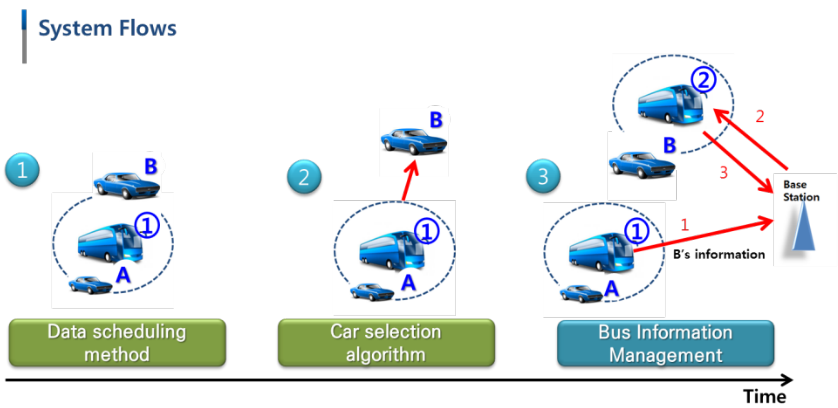

Fig. 7. System flows

Service latency for a download means its response time for the date exchange paths between cars, buses and BSs. Let B_{vi}^{t} and S_{vi}^{t} be the total amount of data and file size, respectively, that are downloaded from bus and system core in time slot. Note that, $B_{vi}^{t} \leq S_{vi}^{t}$ which is indicated that it might be stored media files in bus. We define the bus routing variable VB_{vi}^{t} which indicates the downlink route through bus to vehicles. Similarly, system core routing variable VS_{vi}^{t} is defined as the downlink route from cloud to vehicles. For given pathes, its corresponding value is set as non-zero, $VB_{vi}^{t} > \gamma$ and $VS_{vi}^{t} > \gamma$ where $\gamma \in (0,1]$ denotes the proportion of the data which traversed through the concerned path. Clearly, it is implies that any data loss is not occurred in this simuation.

$$\sum_{vi,bi} VB_{vi}^{t} = 1, \quad \sum_{vi,bi,S} VS_{vi}^{t} = 1 \tag{5}$$

Let δ_{bv} is the delays in unit data transmission from bus to cars. Similary, δ_{sb} is the denoted from system core to bus. For media contents which are served by the system core as a cloud server, the transmission latency at time t, is computed as:

$$\delta_{B}^{t} = \delta_{bv} \sum_{vi} S_{vi}^{t} - B_{vi}^{t} \tag{6}$$

$$\delta_{S}^{t} = (\delta_{bv} + \delta_{sb}) \sum_{vi} S_{vi}^{t} \tag{7}$$

Where, δ_{B}^{t} is transmission delay with BIM and δ_{S}^{t} is transmission delay for conventional cloud.

Therefore, the mean transmission latency is computed as:

$$\delta_{B}^{t} = \frac{\delta_{bv} \sum_{vi} B_{vi}^{t}}{\sum_{vi} B_{vi}^{t}} \tag{8}$$

$$\delta_{S}^{t} = \frac{\delta_{bv} \sum_{vi} B_{vi}^{t} + \delta_{sb} \sum_{vi} S_{vi}^{t}}{\sum_{vi} B_{vi}^{t}} \tag{9}$$

For the service latency, in general, it is defined as a transmission delay adds processing delay which is assumed random variable affected on vehiculuar environment. The change of rate for connection status between system core and vehicles will be bigger than communication status between buses and cars. where , $\varphi_{B} < \varphi_{S}$ represent the per byte processing latency at the MECS and cloud computing tiers, apprehensibly.

Thus, mean service latency is computed as:

$$\Delta\delta_{B}^{t} = \frac{\delta_{bv} \sum_{vi} B_{vi}^{t}}{\sum_{vi} B_{vi}^{t}} + \varphi_{B} \tag{10}$$

$$\Delta\delta_{S}^{t} = \frac{\delta_{bv} \sum_{vi} B_{vi}^{t} + \delta_{sb} \sum_{vi} S_{vi}^{t}}{\sum_{vi} B_{vi}^{t}} + \varphi_{S} \tag{11}$$

Unlikely, viewed from the side of mean the service latency, transmission delay has to be measured in worst case. Since the value of the longest delay can effect on the system performance. We tested latency delay in simuation and the result is shown in **Fig. 11**. And an experimental analysis is computed as :

$$\max(\delta_B^t) = \delta_{bv} \sum_{j=t-\tau}^{t} \phi_{bv}^j \sum_{vi} VB_{vi}^t (S_{vi}^t - B_{vi}^t) \quad (12)$$

$$\max(\delta_S^t) = (\delta_{bv} + \delta_{sb}) \sum_{j=t-\tau}^{t} \phi_{sb}^j \sum_{vi} VS_{vi}^t (S_{vi}^t) \quad (13)$$

where, ϕ_{bv}^j is the weigt-factor associated with the data which is requred for analysis, and the magnitude of ϕ_{sb}^jdecreases with the increase in the age of the data. Clearly, for j=t, $\phi^j = 1$ whereas, as j $\rightarrow 0, \phi^j \rightarrow 0$.

4. Evaluation of the Proposed Architecture

To evaluate the proposed MECS system architecture, it is compared with ordinary LTE air service to support individual media service for cars as the client nodes. In case, Wi-Fi service applied to MECS concept serves the data connection on free between cars and buses. For proposed architecture, BIM, dropped car has to be shared the status of the information of requested media file between buses. The efficiency of proposed architecture affects MESC cloud computing performance significantly. The main aim of the proposed architecture is to reduce the data load for system core and service delay to contents request nodes. The simulation of the proposed system mechanism was performed with a certain number of cars. For simplification, the following parameters are defined for the simulation in outline:

- Media File size : 50Mbytes
- Wi-Fi antenna configuration : omnidirectional beam pattern
- Location distribution and moving speed of buses and cars : uniform
- Traffic distribution of vehicles : Poisson
- Wi-Fi transmission rate: 10Mbps
- LTE transmission rate: 10Mbps
- Number of cars: 10~20 users
- Number of buses : 1~2 MECSs
- Number of line on highway : 4

4.1 Load density performance analysis

For the MECS point of view, it represents the load density performance among buses. Totally, the system core observes assigned data load for each bus. To evaluate the data traffic load and cost side, as shown in **Fig. 8**, green line represents that the VOD clients receive the contents file by only ordinary LTE. It shows that the cost charge for the data service and the data traffic load to data center cloud are very high. Relatively, through the proposed system mechanism, it reduced the data traffic and the cost of wireless network use. It shows that proposed mechanism improves the system performance and effectiveness. It can be expected that proposed architecture makes the data load be low and remains the network connection efficiently.

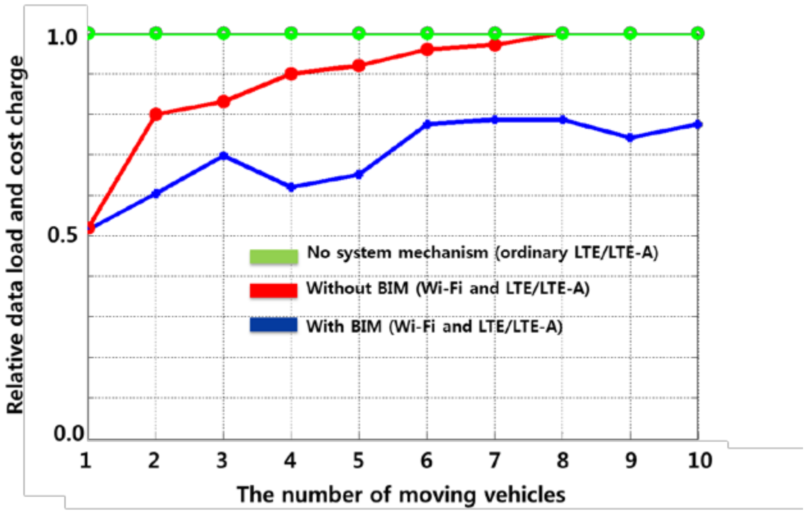

Fig. 8. Load density performance for target bus

4.2 Download number and service latency analysis

For the link adaptation and download, through adoption of the proposed system mechanism, we show that BIM for MECS has advantage for data downloading number. Sharing the vehicle's location information is useful to keep the communication link as seen in **Fig. 9**. It is superior to improve download success rate in vehicular environment.

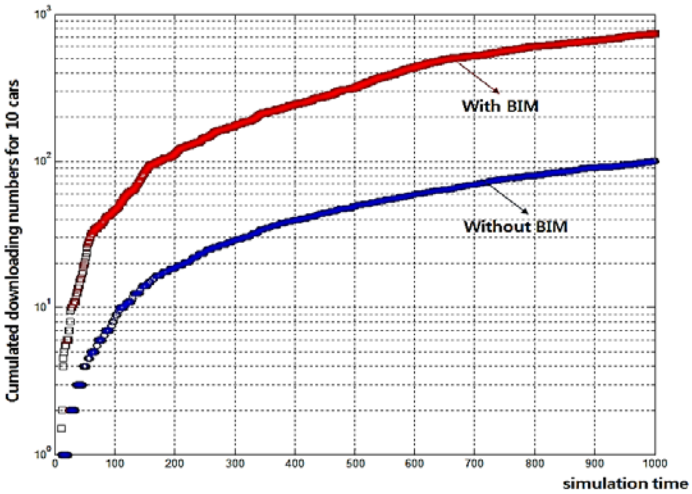

Fig. 9. Cumulated downloading number

For analyzing the mean delay for transmission and service latency, they are plotted in **Fig. 10** against the variable number of moving vehicles. With BIM, as similar as shown in **Fig. 10**, transmission and mean service delay are diminished compared to conventional cloud architecture. From the view of maximum service delay, the result of maximum delay is shown in **Fig. 11**.

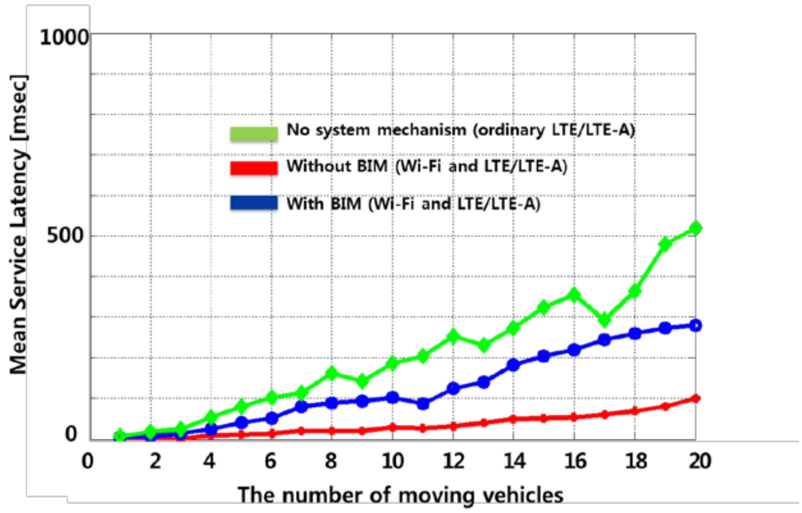

Fig. 10. Mean service latency

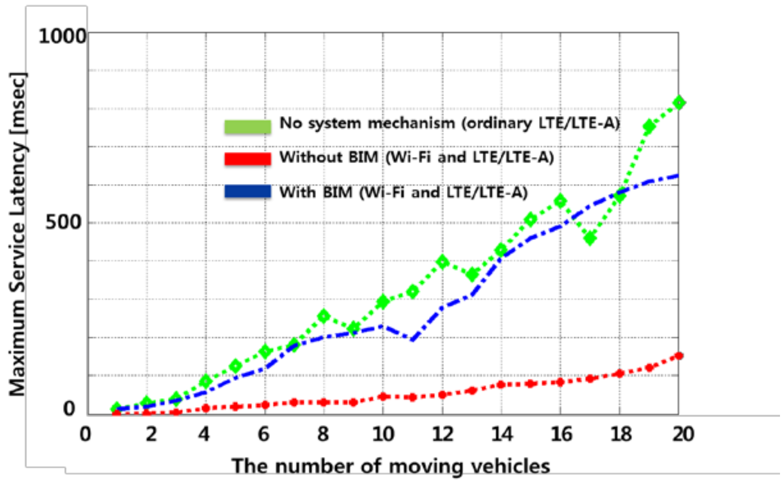

Fig. 11. Maximum service latency

5. The Performance Evaluation on the Network Simulator

We used ns-2 (Version 2.35) as a network simulator in order to evaluate the network performance of proposed system. we consider that Ns-2 simulator might reflect on real environment on highway. It is assumed that every node(buses and cars) in the network participate in mobile edge computing system, when we used BIM. Also, we assumed that every participating car continuously wants to get the multimedia application using mobile edge computing system, such as video contents, which had been exchaged between system core and cars. Their conditions are shown in **Table 1**.

Table 1. The key parameters for simulation in ns-2

Parameters	Value
Radio propagation model	Two-ray ground reflection
Routing protocol	Static
Transport protocol	UDP/NULL
Application agent (Traffic model)	CBR (constant bit rate)
Frame generating interval in sources	0.01s
Frame size	512 bytes (datalink layer)
Number of simulation flows	4/8 flows (to each cars)
Wifi / LTEbandwidth	10/5 Mbps
Moving option	Directional
Simulation time	500s
The time starting to move (in simulation)	80 sec

We set two simulation scenarios. The common option is that there are an ordinaly LTE system and two buses as a media server. The first scenario adopts 4 cars and another is 8 cars. Under the suggeseted scenarios, we made the 2 and 4 cars moving along with bus involved in new bus. It means that they can not communicate with origin bus, respectively. The described scenarioes showned in Fig. 12. and Fig. 13.

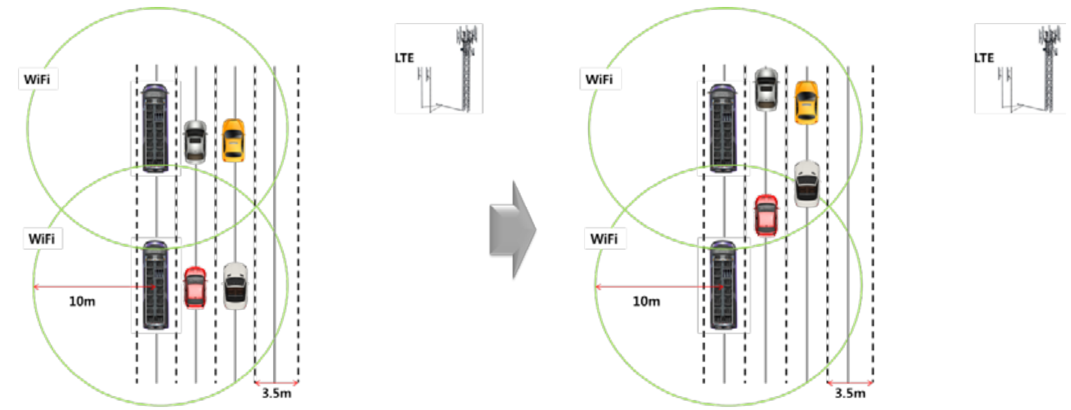

Fig. 12. Simulation of 2 cars scenario

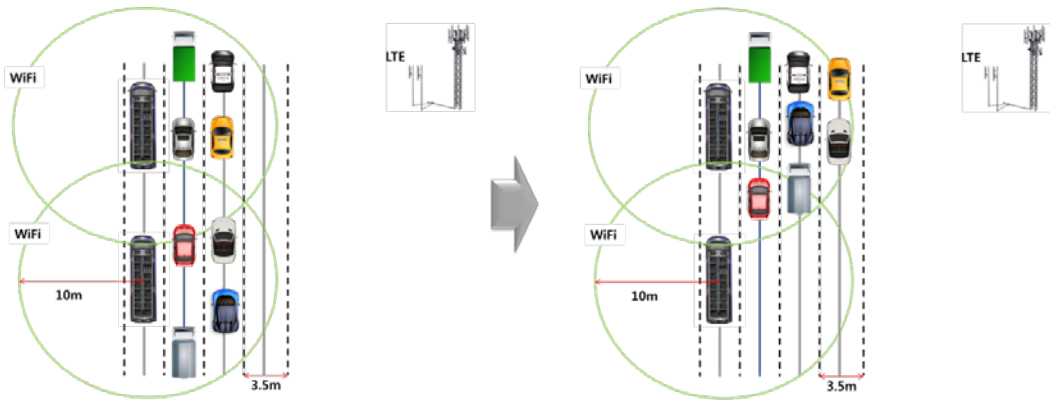

Fig. 13. Simulation of 4 cars scenario

In order to evaluate the performance for BIM, we simulated 3 situations: IDEAL, with BIM, and without BIM. At the IDEAL situation, to make the ideal data delivery, the network topology is not changed, because every node kept the same speed with bus. It means that there is no one to be moved out of the range of communication between cars and bus. Therefore IDEAL situation has the best performance. And we decided that the one produces the closer result with IDEAL situation to be better method. We have simulated two scenarios with three situations, as described in.

5.1 Performance Analysis

The simulation results are shown **Fig. 14**. Every graph as mentioned above executed under 2 cars scenario and 4 cars scenario. The x-axis presents three situations: IDEAL, with BIM and without BIM. It is noticed that the situations which have the closer results with IDEAL's is the better than the other. **Fig. 14(A)**. represents the average delay of packet per car which called as a client node. As shown in this figure, BIM system has the better performance than the other. Especially, scinario of car 8 without BIM is the worst case, because there are a lot of loss packet of the origin bus and it used a lot of streaming packets of LTE.

Fig. 14(B). shows average received packet per node during overall simulation time, where the lower is better at the same scinario. As presented in **Fig. 14(B)**, the BIM system has better performance than the other at the both scinarios.

Fig. 14(C) shows the average time for downloading a movie per a node. The every cars in the network want to get the same movie. Therefore the shorter time for getting a movie is better performance in network aspect. The BIM system has better performance than without system.

Fig. 14(D) shows the number of overall usage of LTE packets. The LTE packet costs than WiFi. So, the lower number of packets means the better for mobile edge users. The IDEAL has nothing in LTE packet, because there is no car to move. The BIM system has 30 and 50 LTE packets in two scinarios. But this results are greatly small than without BIM's.

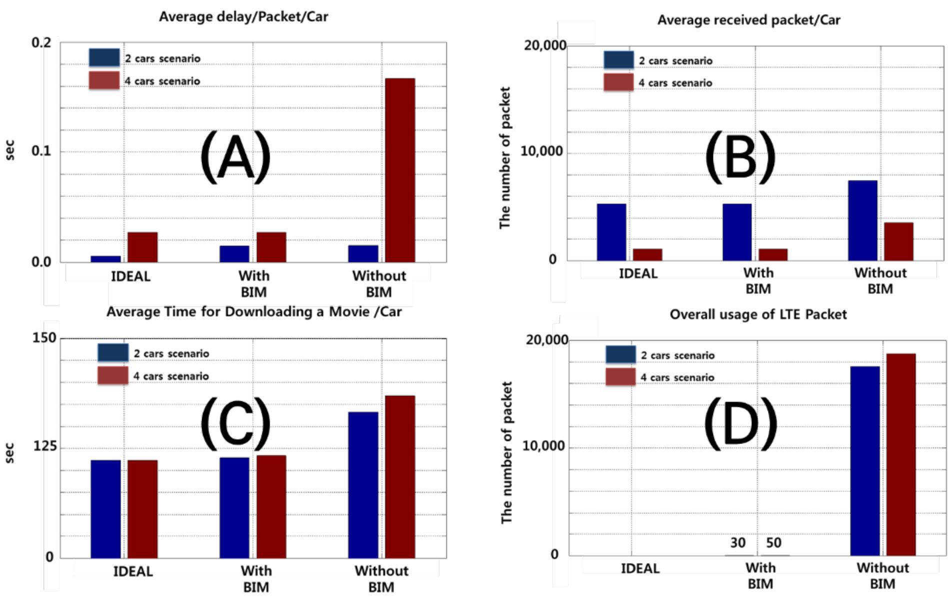

Fig. 14. Simulation resuls

Fig. 15 shows average load density per movie per a node. It has different x-axis component compared any figures. No system mechanism means that nodes use full ordinary LTE communication only. It has the similar result compared with **Fig. 8**. The IDEAL situation has very similar performance at the both scenarios. However, the others have different pattern because the first is the increasing data packet for a movie and the second is increasing downloading time for a movie. The BIM system shows greatly better performance

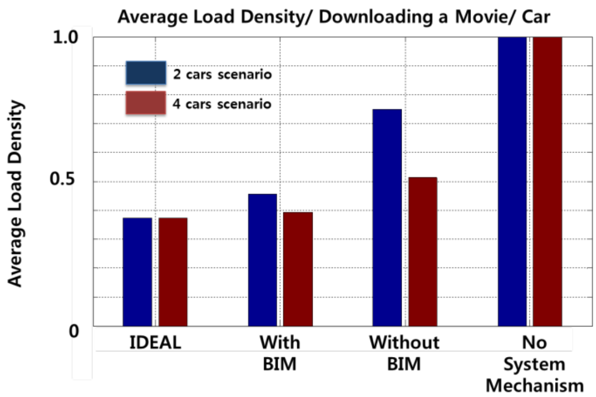

Fig. 15. Average load density per downloading a movie per car

6. Conclusion

The new system architecture is different from conventional static cloud computing and vehicular cloud environment. The proposed architecture consists of three layer architecture utilizes the advantages of VOD service. We have found that when MECS is applied, data downloading has it over usage of ordinary LTE network communication. Since the car as a VOD client is connected with VOD server on bus using Wi-Fi, the scheme reduces the data traffic load and cost charge intuitionally. Especially, through the NS-2 simulation for network communication environment, it is found again that proposed architecture is corresponded to mobile edge computing system and superior to ordinary LTE/LTE-A system for VOD service on highway. In presences, we are considering many preventable obstacles, such as other cars that reflect radio occurrence, the Doppler Effect in real environment. Therefore, we will research about more elaborated BIM system considering radio propagation affect.

References

[1] Syed Husain, Andreas Kunz, Athul Prasad, Konstantinos Samdanis and JaeSeung Song, “Mobile edge computing with network resource slicing for Internet-of-Things,” in *Proc. of 2018 IEEE 4th World Forum on Internet of Things*, pp1-6, 2018.

[2] Jorg Swetina, Guang Lu, Philip Jacobs, Francois Ennesser and Jaeseung Song, “Toward a standardized common M2M service layer platform: Introduction to oneM2M,” *IEEE Wireless Communications*, vol.21, Issue 3, pp. 20-26, 2014. Article(CrossRefLink)

[3] Kyunglag Kwon, Hansaem Park, Sungwoo Jung, Jeungmin Lee and In Jeong Chung, “Dynamic Scheduling Method for Cooperative Resource Sharing in Mobile Cloud Computing Environments,” *KSII Transactions on Internet and Information Systems*, Vol 3, No. 6, 2011.

[4] Shukun Liu and Weijia Jia, “An Adaptive Virtual Machine Location Selection Mechanism in Distributed Cloud,” *KSII Transactions on Internet and Information Systems*, Vol 9, No. 12, 2015.

[5] Yanzhe Che, Qiang Yang and Chunming Wu, "Towards a hierarchical global naming framework in network virtualization," *KSII Transactions on Internet and Information Systems*, Vol 7, No. 5, 2013.
[6] M.Abuelela and S. Olariu, "Tasking VANET to the clouds," in *Proc. of Proceeding of ACM mobile multicast*, 2010.
[7] 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 Transactions on Parallel and Distributed System*, Vol 23, No. 11, pp. 2067-2080, 2012. Article(CrossRefLink)
[8] M. Eltoweissy, S. Olariu and M. Younis, "Towards autonomous vehicular clouds," in *Proc. of Proceeding of AdHocNets*, 2010.
[9] S. Olariu, I. Khalil and M. Abuelela, "Taking VANET to the Clouds," *International Journal of Pervasive Computing and Communications*, Vol 7, No. 1, pp. 7-21, 2011. Article(CrossRefLink)
[10] S. Olariu, M. Eltoweissy and M. Younis., "Towards autonomous vehicular clouds," *ICST Transactions on Mobile Communications and Computing*, Vol 11, No. 9, pp. 1-11, 2011.
[11] Bonomi F, "Connected vehicles, the internet of things, and fog computing," in *Proc. of The Eighth ACM International Workshop on Vehicular Inter-networking VANET*, Las Vegas, USA, 2011.
[12] Bahtovski, Aleksandar, and Marjan Gusev, "Cloudlet Challenges," in *Proc. of Proceeding of Engineering 69*, pp. 704-711, 2014. Article(CrossRefLink)
[13] S Stojmenovic I, "Machine-to-machine communications with in-network data aggregation processing and actuation for large scale cyber-physical systems," *IEEE Internet of Things Journal*, vol. 1, issue 1, 2014.
[14] S. Nesargi and R. Prakash, "Distributed wireless channel allocation in networks with mobile base stations," *Vehicular Technology, IEEE Transactions on*, vol. 51, no. 6, pp. 1407 – 1421, 2002. Article(CrossRefLink)
[15] S. Olariu, T. Hristov and G. Yan, "The next paradigm shift: from vehicular networks to vehicular clouds," *Mobile Ad Hoc Networking: Cutting Edge Directions*, John Wiley & Sons, Inc.

Sangyub Lee received BS degree and the MS degree in electronic and electrical engineering from the Yonsei University, South Korea, in 2005. He is a candidate of the PhD degree in Computer and Information Science from Korea University. He is a senior researcher at the Korea Electronics Technology Institute. He was with Samsung Electro-Mechanics as a senior engineer at the Central R&D center. His research focuses on automotive network systems and various embedded platform systems.

Jaekyu Lee received BS degree and the MS degree in electronic and electrical engineering from the Chonbuk National University, South Korea, in 2012. He is a candidate of the PhD degree in Computer and Information Science from Korea University. He is a researcher at the Korea Electronics Technology Institute. His research focuses on automotive software and various embedded platform systems.

Hyeonjoong Cho received BS degree in electronic engineering from Kyungpook National University in 1996 and the MS degree in electronic and electrical engineering from the Pohang University of Science and Technology, South Korea, in 1988. He received the PhD degree in computer engineering from Virgina Polytechnic Institute and State University, Virginia Tech, in 2006. He is an associate professor in the Department of Computer and Information Science, Korea University. He is a member of the IEEE. His research focuses on real-time systems on various platforms including signal/multiprocessor, sensor networks, etc.