Efficient Access Method for Multi-access Edge Servers in Dynamic Map Systems

Kohei Hosono (kohei.hosono@nislab.doshisha.ac.jp)
Doshisha University: Doshisha Daigaku https://orcid.org/0000-0002-5810-2937
Akihiko Maki
Fujitsu Ltd.
Yosuke Watanabe
Nagoya University: Nagoya Daigaku
Hiroaki Takada
Nagoya University: Nagoya Daigaku
Kenya Sato
Doshisha University: Doshisha Daigaku

Research Article

Keywords: Connected Car, Dynamic Map Systems, Multi-access Edge Computing

Posted Date: July 26th, 2021

DOI: https://doi.org/10.21203/rs.3.rs-593110/v1

Version of Record: A version of this preprint was published at International Journal of Intelligent Transportation Systems Research on January 24th, 2022. See the published version at https://doi.org/10.1007/s13177-021-00289-8.

Noname manuscript No.
(will be inserted by the editor)

Efficient Access Method for Multi-access Edge Servers in Dynamic Map Systems

Kohei Hosono · Akihiko Maki · Yosuke Watanabe · Hiroaki Takada · Kenya Sato

Received: date / Accepted: date

Abstract Research and development on connected cars equipped with communication functions is being conducted, and dynamic maps are being researched and developed as an information and communication platform for cooperative automatic driving. There is a concern about scalability when dynamic maps are constructed on a cloud server to aggregate a wide range of vehicle information. The problem can be alleviated by deploying edge servers that divide the geographic area where vehicles travel and manage each of them. The IP addresses of the edge servers need to be resolved on the basis of the location information of the moving vehicles. Using TCP improves reliability but reduces efficiency because the vehicles move. In this study, we developed a novel method for accessing edge servers that achieves higher reliability and efficiency by adopting UDP using anycast for transmission from vehicle to edge server and implementing a retransmission function. The effectiveness of this access method was verified by using a vehicle driving simulation.

Keywords Connected Car · Dynamic Map Systems · Multi-access Edge Computing

K. Hosono
Graduate School of Science and Engineering, Doshisha University, Kyotanabe, Kyoto 6100321, Japan.
E-mail: kohei.hosono@nislab.doshisha.ac.jp

A. Maki
Fujitsu Ltd., Kanagawa 2158588, Japan.

Y. Watanabe · H. Takada
The Institute of Innovation for Future Society, Nagoya University, Nagoya, Aichi, 4648601, Japan.

K. Sato
Graduate School of Science and Engineering, Doshisha University, Kyotanabe, Kyoto 6100321, Japan.
Mobility Research Center, Doshisha University, Kyoto 6100321, Japan.

1 Introduction

With the aim of realizing a safe and secure mobility society, efforts are being made to develop automobile traffic systems that integrate automobiles and information communications, such as safe driving support systems and automated vehicles[1–3]. In particular, the sophistication of sensors installed in vehicles has enabled detailed recognition of the surrounding driving environment, which has accelerated the research and development of advanced safe driving support systems that warn the driver and automatically avoid danger, as well as automated driving systems that enable autonomous driving[4, 5].

However, the range of objects that can be recognized by sensors mounted on vehicles is limited, and although these sensors can detect objects within the range of visibility, they cannot detect objects beyond it[6, 7]. For example, it is difficult to respond to an imminent collision at an intersection that has poor visibility or a sudden jump out from behind an object. To cope with these situations, research is being conducted on cooperative automatic driving, which aims to improve safety by using wireless communication technology to exchange information between vehicles in motion or between vehicles and roadside equipment[8–10].

Various applications of cooperative automatic driving are being considered, such as for warning of a risk of collision at an intersection, providing information on traffic congestion, road surface conditions, and signal status, as well as support for merging on expressways[11]. Maps are expected to play a role not only in conventional route search to a destination but also in providing highly accurate spatial features for self-positioning estimation of each automated vehicle and as basic information for linking sensor information shared

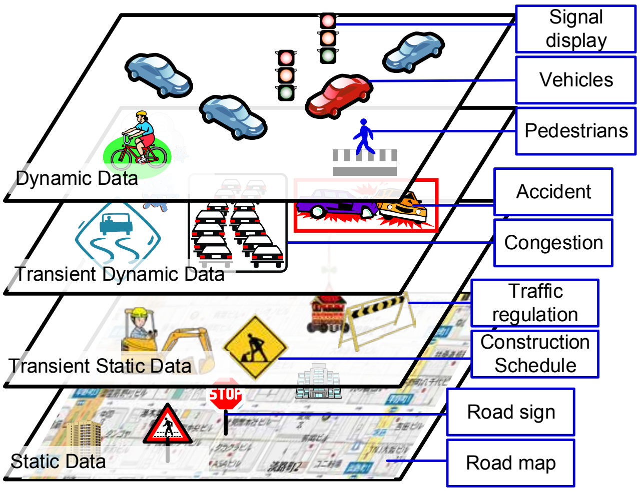

Fig. 1 Dynamic map overview

among vehicles. Currently, each piece of information is managed and processed separately for each application. An information and communication platform that enables dynamic information obtained from sensors to be superimposed on a high-precision road map is called a "dynamic map," and it is a necessary information infrastructure that supports advanced transportation services such as automated driving[12,13]. As shown in Fig. 1, the generally recognized structure of a dynamic map consists of a static road map on which dynamic information is layered in accordance with the frequency of information updates.

Dynamic maps provide the necessary functions for data management, such as collection, integration, and retrieval, for high-precision road maps and sensor information, making it possible to develop and execute applications that provide transportation services more flexibly and easily. The dynamic information of a dynamic map is generally managed by sensor information transmitted from vehicles in 100-millisecond cycles[14]. Applications that have a high impact on traffic safety, such as safe driving support and cooperative automatic driving, operate on the basis of this information, so information processing with low latency is required. Normally, a dynamic map system assumes the use of a cloud server on the Internet, but if a huge amount of data from vehicles is concentrated on the server, it will cause a processing load and communication delay, so scalability will become a problem. Many applications that operate on dynamic maps have a high impact on traffic safety, such as intersection collision risk warning and merge control. Therefore, these applications need to communicate and process data with vehicles with low latency. Thus, the processing load should be distributed and communication delay reduced by deploying an edge server between cloud servers and vehicles[15–17].

Data sent from vehicles is processed sequentially by geographically distributed edge servers, and the processing results are notified directly to the vehicles, which is expected to improve the real-time performance. However, as shown in Fig. 2, since dynamic maps handle applications that have a high impact on traffic safety, edge servers not only need to process vehicle data in real time but also need to aggregate all vehicle data in a target area for the applications handled by each edge server. ETSI is considering Multi-access Edge Computing (MEC), where edge servers are placed around mobile phone base stations[18,19]. In this paper, we propose an access method for a dynamic map system that can aggregate and process the data sent from a vehicle to an edge server through a core network while keeping the data in front of the Internet[20]. We also proposes a method that enables efficient communication through a dynamic map system consisting of edge servers and vehicles.

2 Proposed Methods

Our dynamic map system reduces the load on a network by distributing the processing of data collection and distribution between a vehicle and the nearest edge server and enables real-time processing, as shown in Fig. 3. When this system is implemented on the cloud, tens of millions of cars will be connected, constantly communicating and processing. To exchange support information for safe driving control, a single cloud needs to receive and process data from each vehicle at a rate of about 100 milliseconds per cycle, which is not appropriate from the perspective of network load, server processing load, and real-time performance. In this study, we consider a dynamic map system with edge servers located around mobile phone base stations that receive and send data from moving vehicles. Vehicles send data obtained from onboard sensors to edge servers. The data sent from the vehicles include the vehicle ID, vehicle position, speed, direction of travel, time stamp, etc. The vehicle position is obtained by using vehicle position information obtained by GPS or scan matching[21–23]. In this paper, each vehicle and edge server are referred to as a node, and each node has a unique ID, like the ITS systems that are being standardized in Europe[24]. Each node sends and receives data, and applications are executed. The dynamic map system can be used as a common infrastructure for linking sensor information and map data, and it facilitates the construction of applications.

In the dynamic map system in this paper, IP is used as the communication method between the vehicle and the edge server. And 5G, Wi-Fi, C-V2X, DSRC, etc. are

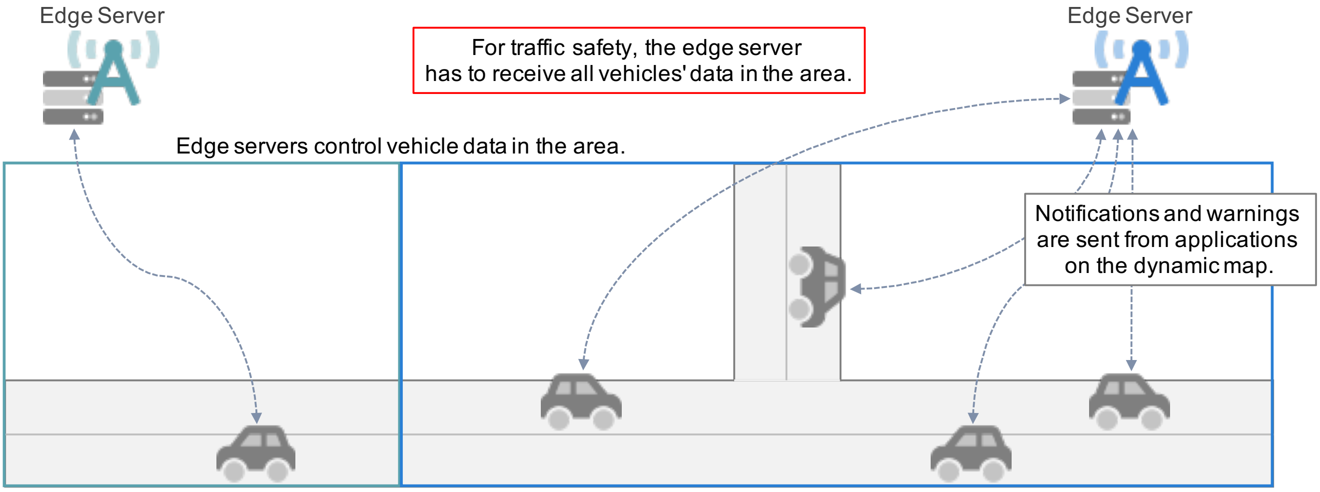

Fig. 2 Dynamic map with edge servers

assumed to be used as the communication method between the vehicle and the edge server[25–27]. In the case of using the dynamic map system only on the cloud, when sending data from the vehicle, the IP address of the vehicle is used as the source and the IP address of the cloud as the destination. When an application is executed on the dynamic map system in the cloud, the IP address of the vehicle can be used to notify the vehicle of the processing result. However, in a configuration using edge servers, the IP address of the edge server cannot be uniquely set as the destination when sending data from the vehicle because the vehicle switches between multiple edge servers while driving. One solution to this problem is to link the IP addresses of neighboring edge servers corresponding to the location information and keep this information in each vehicle in advance. In this case, however, it is necessary for the vehicle to have information on all edge servers, and the process of updating the information held by each vehicle becomes complicated. In addition, since each vehicle transmits data on the basis of the information of its own edge server, each edge server on which the dynamic map operates cannot know whether it has received all the vehicle data in the target range. In addition, it is not always possible to connect to the target edge server due to wireless communication conditions. In addition, the range managed by the edge servers changes depending on the road conditions. Therefore, we propose an access method for our dynamic map system that assigns an edge server on the basis of the location information of vehicles and uses UDP anycast addresses[28].

Our dynamic map system allows vehicles to send data to edge servers using UDP with an anycast address as the destination. Each edge server that receives data is assigned the same anycast address, which allows a vehicle to send data without being aware of the destination server while it continues to move. However, although the use of anycast addresses allows vehicles to be unaware of the destination server, they do not know which edge server will receive the data sent from each vehicle. In addition, since the applications operated by the dynamic map system include those that have a high impact on traffic safety, each edge server needs to aggregate all vehicle data within an application's operating area. Each edge server determines which edge server will receive data from a vehicle on the basis of the vehicle location information in the received data and transfers the data to the surrounding edge servers as necessary. In addition to a payload consisting of sensor data, a unique header is added to the UDP data sent by the vehicle[29]. This header is used to manage the vehicle data in the dynamic map system and includes a station ID (SID) to identify the vehicle and a transfer flag to recognize the transfer between edge servers. The UDP packets sent by a vehicle are shown in Table 1. When an edge server receives the data from a vehicle, it determines the necessity of the transfer on the basis of the SID, and if the transfer is necessary, it specifies the edge server to be the destination using unicast and sends a TCP packet.

Table 1 The format of UDP packet

Data category	Size
IP header	40 Byte
UDP header	8 Byte
Custom header	80 Byte
Payload	1,030 Byte

Instead of the vehicle designating the edge server to which the data is to be sent, the edge server closest to

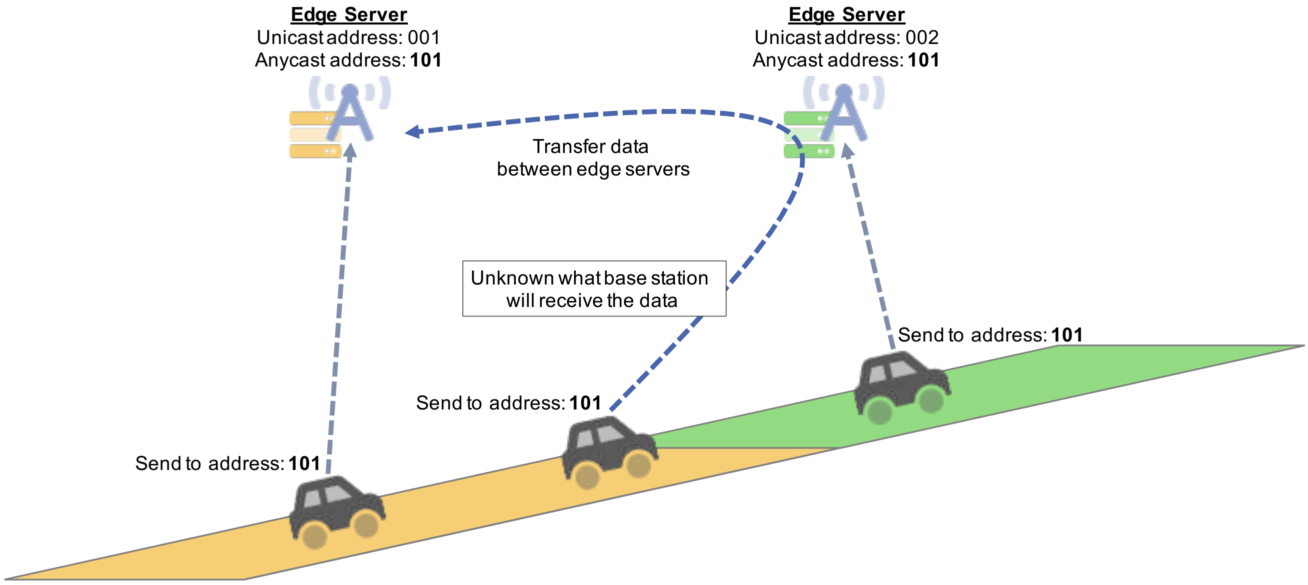

Fig. 3 Dynamic map using anycast address

the vehicle receives the data and, if necessary, transfers the data using high-speed dedicated lines maintained between each edge server, thereby enabling more efficient exchange of vehicle data[30]. In addition, we are considering placing each edge server in the vicinity of a cell phone base station, and the operating area of the applications running on the dynamic map system on each edge server is expected to basically overlap with the reception area of the base station, thus reducing the need for data transfer and improving real-time performance. This is expected to reduce the need for transmission and improve real-time performance.

UDP is a connectionless protocol, so it is faster than TCP, but it is less reliable, and packet loss is a concern[31]. In addition, since the communication between a vehicle and edge server is wireless, it is less reliable than wired communication[32]. To solve this problem, the edge server sends back ACK data to confirm the reception of the data sent by the vehicle. As shown in Fig. 4, the vehicle continues to retransmit sensor data periodically until it receives ACK from the edge server and stops retransmitting data when it receives it. Data from the vehicle is sent to an anycast address, and the edge server sends back ACK to the sender's unicast address. The edge server forwards the data between the edge servers as necessary, using the Unicast address of the vehicle as the source of the data as the destination, and sends an ACK, as shown in Fig. 5. Since the ACK data from the edge server is also sent using UDP, there is no guarantee that the ACK data will reach the vehicle. Therefore, each edge server filters the data from the vehicle and discards the data if it is the same data due to retransmission.

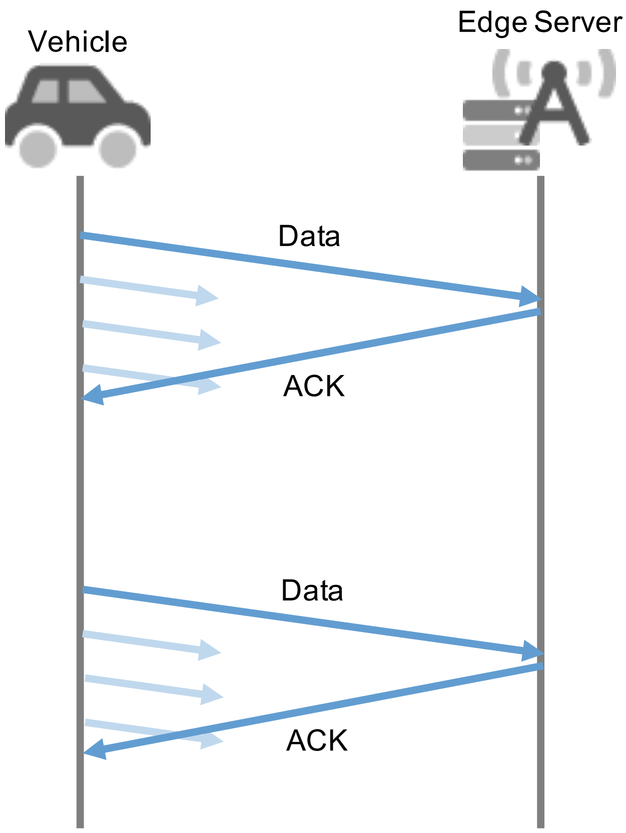

Fig. 4 Resend function using ACK

Data transmission from a vehicle to an edge server uses an anycast address, and for data transmission from an edge server to a vehicle, the IP address of the vehicle is determined from the packets sent by the vehicle, and the unicast address is used. In this case, there is no

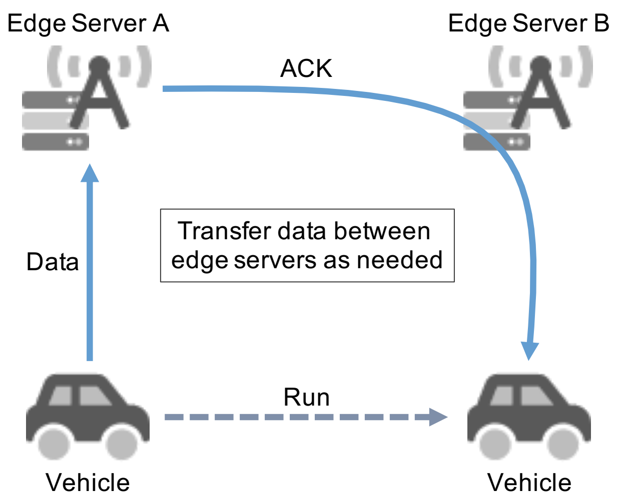

Fig. 5 Transfer data using ACK

problem if the edge server has received the data from the vehicle beforehand, but it is not possible to notify information such as on the location of emergency vehicles before receiving the data from the vehicle. In addition, when unicast is used, communication occurs for each vehicle, which increases the communication cost. The information notified to vehicles from applications in dynamic map systems is not only individual information for each vehicle but also the same information in many cases. Therefore, we use the multicast address as a method of simultaneous notification from edge servers to vehicles[33]. This allows an edge server to send data without knowing the unicast address of each vehicle. In addition, each vehicle can determine the necessity of discarding data by judging the SID of the destination stored in the header of the data sent by the edge server.

3 Experiments

To evaluate the access method, we prepared an edge server and a PC as a vehicle and constructed a dynamic map system. Using dynamic map system, we implemented an access method using anycast and a retransmission function using ACK. To evaluate the effectiveness of three access methods introduced below, we simulated the dynamic map system by driving multiple vehicles. The access methods used and the evaluation environment are described in this section.

Table 2 The format of TCP packet

Data category	Size
IP header	40 Byte
UDP header	20 Byte
Custom header	80 Byte
Payload	1,030 Byte

3.1 Access Methods

3.1.1 UDP access method using anycast & no retransmission by ACK

The data sent from the vehicle was a UDP packet destined to a unique anycast address assigned to the edge server. Each edge server determined the edge server that needed the data on the basis of the SID and vehicle location information stored in the header of the data received from the vehicle. If the data needed to be forwarded, it sent a TCP packet with the unicast address of the destination edge server as the destination. The vehicle sent the sensor data, including the current position information, to the edge server at intervals of 100 milliseconds, without retransmission. Each edge server executed an application in the dynamic map system on the basis of the aggregated vehicle data and notified the vehicle of the processing results.

3.1.2 UDP access method using anycast & with retransmission by ACK

As in section 3.1.1, the vehicle sent data to the edge server using the assigned anycast address as the destination. The edge server received the data and forwarded the data as necessary on the basis of the SID and vehicle location information stored in the header. The edge server that received the data with its own destination sent back ACK data to the sender vehicle. The vehicle retransmitted the data until it received the ACK data.

3.1.3 TCP access method using unicast

The vehicle sent a TCP packet with the unicast address of the edge server as the destination. In this case, the vehicle was assumed to have the unicast address of the edge server corresponding to its own location information. The TCP packets sent by the vehicle are shown in Table 2.

3.2 Scenarios for Simulation

To evaluate the effectiveness of the access method of the dynamic map system, we conducted a simulation of ve-

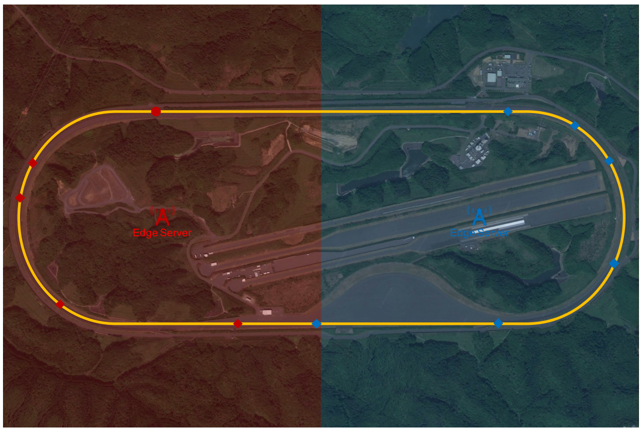

Fig. 6 Simulation at Shirosato Test Center

hicle driving. Vehicles traveled around a road with two edge servers and sent data to them. To evaluate the effectiveness of the access method without depending on the application on the dynamic map system, we simulated vehicles driving through the Shirosato Test Center of the Japan Automobile Research Institute[34]. The high-speed track of the Shirosato Test Center is 5500-m long, and vehicles can drive around it. Two edge servers were installed at the track, and each handled over half of the track. The vehicles traveled at a speed between 40 and 60 km/h with a smooth speed change. The data sent from the vehicles was received by the nearest edge server on the basis of the radio wave conditions, etc. The data was not necessarily sent directly to the appropriate edge server but was forwarded as necessary. The scene during the simulation is shown in Fig. 6. Each vehicle is indicated by a marker of the color of the edge server that received the transmitted data. The vehicles sent 1158 bytes of data, consisting of an Ether header, IP header, UDP header, proprietary header, and payload. In order to evaluate the impact of an increase in the number of vehicles operated by the edge server, we construct a PC as a load vehicle and perform a driving simulation.

3.3 Dynamic Map System

Table 3 and 4 show the configuration of the edge server and the PC used as the vehicle for the dynamic map system we constructed. The structure of the dynamic map system we constructed is shown in Fig. 7. The vehicle is expected to communicate with the dynamic map using multiple wireless communication methods, such as 5G and Wi-Fi, and we previously conducted demonstration experiments with dynamic maps using LTE and Wi-Fi. However, wireless communication depends not only on system factors such as the access

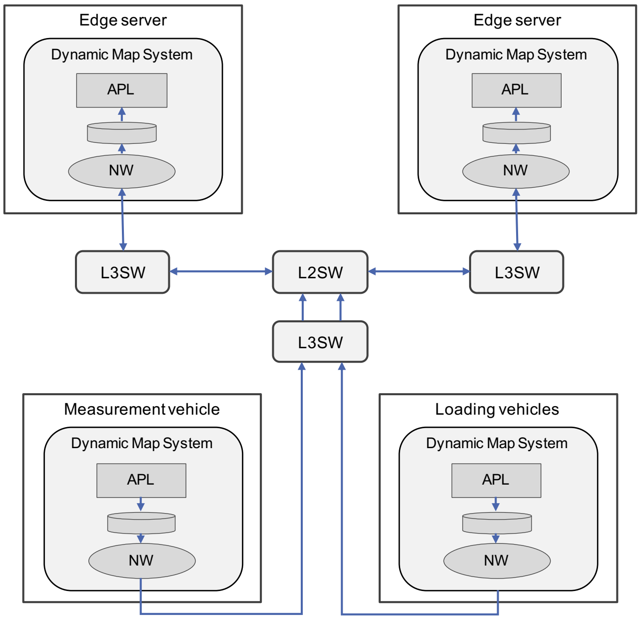

Fig. 7 Dynamic map system configuration

Table 3 Vehicle configuration

OS	Ubuntu 16.04
CPU	4 Core 8 thread (1.90GHz)
Memory	16 GB
SSD	256 GB
Network	Up to 1 Gbps

Table 4 Edge servers and load vehicles configuration

OS	Ubuntu 16.04
CPU	8 Core 16 thread (3.60GHz)
Memory	16 GB
SSD	256 GB
Network	Up to 1 Gbps

method proposed in this paper but also on communication conditions[35]. In this paper, we propose the novel method for accessing edge servers for moving vehicles in used for any networks. We are constructing a dynamic map system to manage vehicles in an integrated manner by flexibly using such technologies and have proposed a method for accessing edge servers for moving vehicles. To clarify the contribution of the access method to the effectiveness of the dynamic map system, we constructed and evaluated a communication environment using Ethernet, which reduces the uncertainty of wireless communication.

To verify the effectiveness of the access method, we measured the packet loss rate, throughput, and latency. These were measured using log data from our dynamic map system and Wireshark, the most widely used network protocol analyzer in the world[36]. We also use

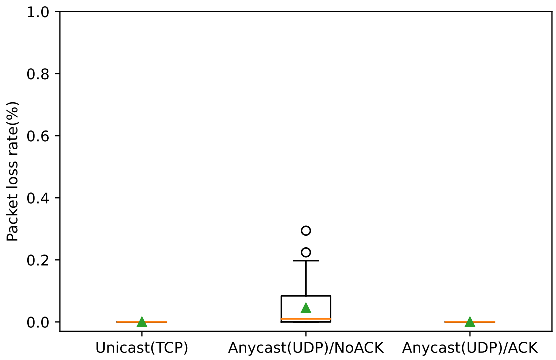

Fig. 8 Packet loss rate by three access methods

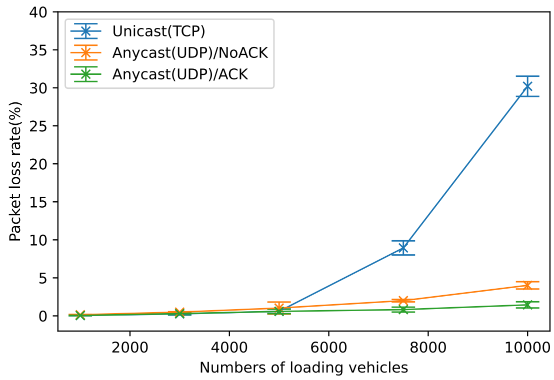

Fig. 9 Effect of number of vehicles on packet loss rate

the Google Maps Platform to visualize the location of the vehicles and the edge servers[37].

4 Results

The results in terms of packet loss rate, throughput, and latency for the UDP access method using anycast (with/without retransmission by ACK) and the TCP access method using unicast are discussed. For each scenario, a 5-minute simulation was performed, and the number of trials was 20 for each.

4.1 Packet Loss

We measured the packet loss rate, which is the ratio of data sent from a vehicle that reaches the edge server that needs the data. The evaluation results are shown in Fig. 8. In the case of the UDP access method with ACK retransmission and the TCP access method using unicast, the packet loss rate was 0% in all trials. In the case of the UDP access method without ACK retransmission, the average packet loss rate was 0.05%, and the 95% confidence interval was $0.033\% \leq \mu \leq 0.058\%$. This is because UDP is a connectionless protocol, and packet loss may occur when the retransmission function is not available in the upper layer protocol.

In addition to the measurement vehicles, we also prepared several load vehicles to run around the same course. The simulations were performed with scenarios using 10, 100, 1000, and 10,000 load vehicles, and all the load vehicles were assumed to travel at varying speeds and send data to the edge server in the same way as the measurement vehicles. The simulation results for each scenario are shown in Fig. 9. In the case of the UDP access method without ACK retransmission, the packet loss rate increased as the number of vehicles increased. In the case of the TCP access method using unicast, the packet loss rate remained low up to a certain number of vehicles but increased rapidly beyond that number. The UDP access method with the ACK retransmission function also maintained a low packet loss rate up to a certain number of vehicles, but the packet loss rate increased rapidly after that. The reason the UDP access method was able to maintain a low packet loss rate even with a larger number of vehicles than the TCP access method is that the UDP access method achieved a higher throughput.

In a dynamic map system using UDP, if there is no ACK retransmission function, data arrival cannot be guaranteed, and packet loss occurs. However, by implementing the ACK retransmission function, we were able to guarantee the arrival of data and achieve the same level of packet loss rate as TCP while using UDP. In dynamic maps, which have a significant impact on traffic safety, the data sent from vehicles must be transmitted to the edge server in real time. In our dynamic map system, it is possible to switch between using the retransmission function with ACK or not using it. Therefore, we believe that the UDP communication method using anycast addresses is effective because it is possible to switch between both cases according to the requirements of the dynamic map applications.

4.2 Throughput

We measured the throughput of a vehicle that sent the data and the throughput of the edge server that received the data. Fig. 10 shows the throughput of the sending vehicle and the receiving edge server when there was only one vehicle. The throughput of the sending vehicle was $18.1Kbps$ on average for both access methods, and the throughput of the receiving server was $18.0Kbps$ on average for both access methods.

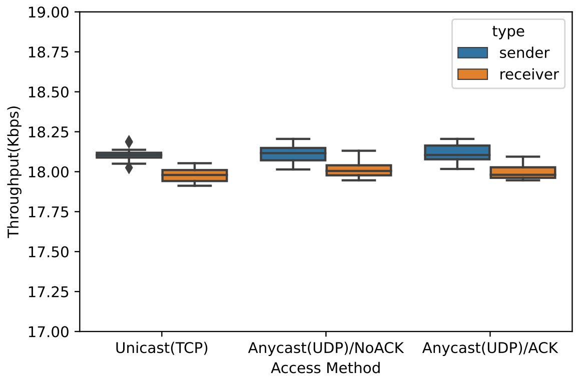

Fig. 10 Throughput by three access methods

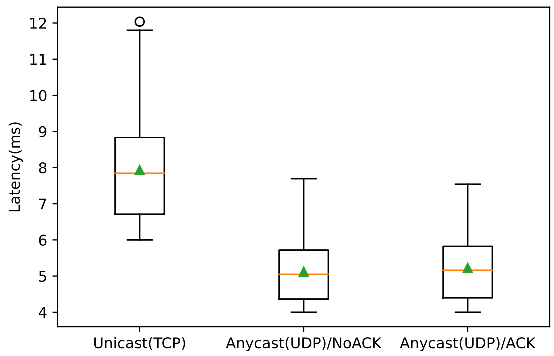

Fig. 12 Latency by three access methods

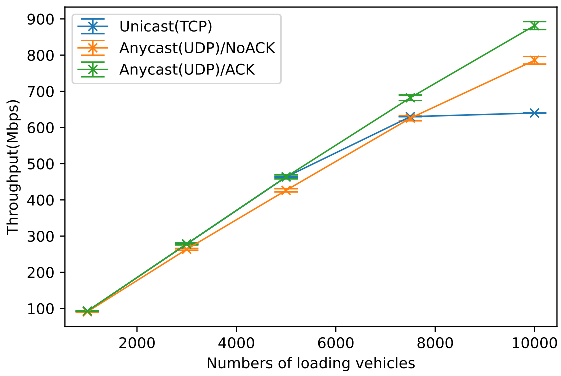

Fig. 11 Effect of number of vehicles on throughput

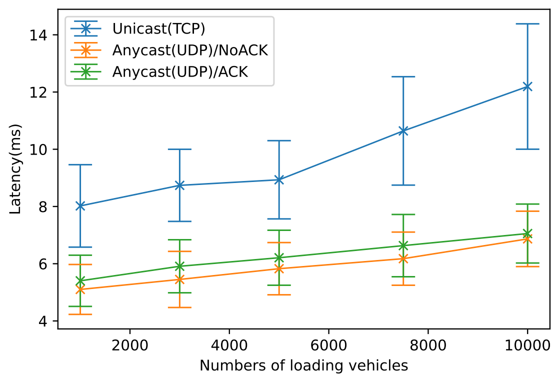

Fig. 13 Effect of number of vehicles on latency

As in section 4.1, Fig. 11 shows the results of the evaluated throughput of the receiving server when the number of load vehicles was varied. $1Gbps$, the bandwidth used in this experiment, was the upper limit for the UDP access method, and a high throughput was achieved. For the TCP access method, the upper limit was about $650Mbps$, and packets above that were discarded. This is thought to be because the throughput of TCP decreases with RTT. For dynamic maps that send and receive huge amounts of data, the higher the throughput, the more effective the UDP access method using anycast is at achieving a higher throughput than the TCP access method. In this experiment, we used Ethernet to verify the effectiveness of the access method. However, since actual vehicles communicate wirelessly and RTT is likely to decrease, a method that can achieve high throughput independent of RTT is effective.

4.3 Latency

We measured the latency for the data sent from a vehicle to reach the dynamic map application on the edge server that needs the data. The evaluation results are shown in Fig. 12. The average latency for the UDP communication method without ACK retransmission was $6.0ms$ with a 95% confidence interval of $5.8ms \leq \mu \leq 6.2ms$. In the case of UDP with ACK retransmission, the average latency was $6.1ms$ with a 95% confidence interval of $5.8ms \leq \mu \leq 6.3ms$. In the case of TCP with unicast, the average latency was $9.8ms$ with a 95% confidence interval of $9.5ms \leq \mu \leq 10.0ms$. The reason for the higher latency in the case of TCP with unicast was due to the 3-way handshake and retransmission control of TCP.

As in section 4.1, the results of a latency evaluation for different numbers of load vehicles are shown in Fig. 13. For the UDP access method, the latency results did not change much when the number of load vehicles was changed. For the TCP access method using unicast, the latency increased as the number of vehicles increased, which may be due to the influence of TCP's retransmission control and congestion control.

In a dynamic map system, real-time performance is very important, and the shorter the latency in sending and receiving data, the better. In our dynamic map system, UDP communication using anycast addresses enables us to send and receive data with very low latency compared with TCP communication. In addition, the ACK retransmission function can improve the reliability of data, so we believe that the UDP access method using anycast addresses is effective.

From these evaluation results, we believe that our proposed UDP access method using anycast is effective in dynamic map systems. In this paper, we constructed a dynamic map system using Ethernet to evaluate the effectiveness of our access method. In the future, it will be important to conduct demonstration experiments using wireless communication in a situation close to an actual environment. In addition, the dynamic map needs to cover many roads in the future, and since it is necessary to assign the same anycast address to each edge server for access using anycast addresses, it is necessary to conduct evaluations on a larger scale.

5 Conclusion

Cooperative automatic driving is actively studied to improve safety by exchanging information between vehicles and roadside equipment. Dynamic map systems are being studied; they are an information and communication platform for superimposing dynamic information such as sensor data on high-precision road maps, managing them in an integrated manner and executing applications. A vehicle always sends sensor data to the dynamic map system, and the applications on the dynamic map system need to process the aggregated data in real time and notify the results to the vehicle. Therefore, it is expected that dynamic map systems can be built on geographically distributed edge servers, which can efficiently communicate and process the data.

In a dynamic map system that deals with applications that have a large impact on traffic safety, the data sent from a vehicle must be sent to the appropriate edge server. In addition, high real-time performance is required. We propose a UDP access method using anycast addresses to access data from a vehicle to an edge server, and we conducted a simulation to evaluate the effectiveness of the method. The proposed UDP access method with anycast addresses showed superiority in throughput and latency compared with a TCP access method with unicast addresses and achieved the same reliability as TCP by implementing an ACK retransmission function.

Dynamic map systems for safe driving support and automatic driving require low-latency communication, and the amount of data communicated between vehicles and servers is expected to increase in the future. In this paper, we evaluated the effectiveness of our access method by allocating edge servers geographically on roads where vehicles travel and constructing a vehicle driving simulation environment. The effectiveness of the access method between the vehicle and the edge server was demonstrated by constructing a dynamic map system using this access method.

Acknowledgements This work was supported by JSPS KAKENHI Grant Number JP20H00589.

References

1. Guillaume Bresson, Zayed Alsayed, Li Yu, and Sébastien Glaser. Simultaneous localization and mapping: A survey of current trends in autonomous driving. *IEEE Transactions on Intelligent Vehicles*, 2(3):194–220, 2017.
2. Suda Yoshihiro and Aoki Keiji. Current activities and some issues on the development of automated driving. *Johokanri*, 57(11):809–817, 2015.
3. Klaus Bengler, Klaus Dietmayer, Berthold Farber, Markus Maurer, Christoph Stiller, and Hermann Winner. Three decades of driver assistance systems: Review and future perspectives. *IEEE Intelligent Transportation Systems Magazine*, 6(4):6–22, 2014.
4. Sayanan Sivaraman and Mohan Manubhai Trivedi. Looking at vehicles on the road: A survey of vision-based vehicle detection, tracking, and behavior analysis. *IEEE Transactions on Intelligent Transportation Systems*, 14(4):1773–1795, 2013.
5. Vicente Milanés, Steven E. Shladover, John Spring, Christopher Nowakowski, Hiroshi Kawazoe, and Masahide Nakamura. Cooperative adaptive cruise control in real traffic situations. *IEEE Transactions on Intelligent Transportation Systems*, 15(1):296–305, 2014.
6. C.T. Chen and Y.S. Chen. Real-time approaching vehicle detection in blind-spot area. In *2009 12th International IEEE Conference on Intelligent Transportation Systems*, pages 1–6, 2009.
7. Bin-Feng Lin, Yi-Ming Chan, Li-Chen Fu, Pei-Yung Hsiao, Li-An Chuang, Shin-Shinh Huang, and Min-Fang Lo. Integrating appearance and edge features for sedan vehicle detection in the blind-spot area. *IEEE Transactions on Intelligent Transportation Systems*, 13(2):737–747, 2012.
8. Jian Wang, Yameng Shao, Yuming Ge, and Rundong Yu. A survey of vehicle to everything (v2x) testing. *Sensors*, 19(2), 2019.
9. Sampo Kuutti, Saber Fallah, Konstantinos Katsaros, Mehrdad Dianati, Francis Mccullough, and Alexandros Mouzakitis. A survey of the state-of-the-art localization techniques and their potentials for autonomous vehicle applications. *IEEE Internet of Things Journal*, 5(2):829–846, 2018.
10. Joshua E. Siegel, Dylan C. Erb, and Sanjay E. Sarma. A survey of the connected vehicle landscape—architectures, enabling technologies, applications, and development areas. *IEEE Transactions on Intelligent Transportation Systems*, 19(8):2391–2406, 2018.
11. Amir Mukhtar, Likun Xia, and Tong Boon Tang. Vehicle detection techniques for collision avoidance systems: A review. *IEEE Transactions on Intelligent Transportation Systems*, 16(5):2318–2338, 2015.

12. Eiter Thomas, Füreder Herbert, Kasslatter Fritz, Parreira Josiane, Xavier, and Schneider Patrik. Towards a semantically enriched local dynamic map. *International Journal of Intelligent Transportation Systems Research*, 17(1):32–48, 2019.
13. Watanabe Yousuke, Sato Kenya, and Takada Hiroaki. Dynamicmap 2.0: A traffic data management platform leveraging clouds, edges and embedded systems. *International Journal of Intelligent Transportation Systems Research*, 18(1):77–89, 2020.
14. *Guidelines for Experiments on Inter-Vehicle Communication Systems Using the 5.8 GHz Band*, 2017.
15. Suzuki Naoya, Sasaki Kengo, and Makido Satoshi. A prediction handover method between edge servers on infrastructure-based vehicle control system. Technical Report 6, Toyota Cenral R&D Labs., Inc., mar 2018.
16. Yi Liu, Huimin Yu, Shengli Xie, and Yan Zhang. Deep reinforcement learning for offloading and resource allocation in vehicle edge computing and networks. *IEEE Transactions on Vehicular Technology*, 68(11):11158–11168, 2019.
17. Fuhui Zhou, Rose Qingyang Hu, Zan Li, and Yuhao Wang. Mobile edge computing in unmanned aerial vehicle networks. *IEEE Wireless Communications*, 27(1):140–146, 2020.
18. ETSI. *Mobile-Edge Computing (MEC); Service Scenarios*, 2015.
19. ETSI. *Multi-access Edge Computing (MEC); Study on MEC Support for V2X Use Cases*, 2018.
20. Imtiaz Parvez, Ali Rahmati, Ismail Guvenc, Arif I. Sarwat, and Huaiyu Dai. A survey on low latency towards 5g: Ran, core network and caching solutions. *IEEE Communications Surveys Tutorials*, 20(4):3098–3130, 2018.
21. Zhengbing He, Liang Zheng, Peng Chen, and Wei Guan. Mapping to cells: A simple method to extract traffic dynamics from probe vehicle data. *Computer-Aided Civil and Infrastructure Engineering*, 32(3):252–267, 2017.
22. Mohamed Maher Atia, Allaa R. Hilal, Clive Stellings, Eric Hartwell, Jason Toonstra, William B. Miners, and Otman A. Basir. A low-cost lane-determination system using gnss/imu fusion and hmm-based multistage map matching. *IEEE Transactions on Intelligent Transportation Systems*, 18(11):3027–3037, 2017.
23. Andi Zang, Zichen Li, David Doria, and Goce Trajcevski. Accurate vehicle self-localization in high definition map dataset. In *Proceedings of the 1st ACM SIGSPATIAL Workshop on High-Precision Maps and Intelligent Applications for Autonomous Vehicles*, AutonomousGIS '17, 2017.
24. Mobility Open Blockchain Initiative (MOBI). *VID Business White Paper*, 2021.
25. Gaurang Naik, Biplav Choudhury, and Jung-Min Park. Ieee 802.11bd 5g nr v2x: Evolution of radio access technologies for v2x communications. *IEEE Access*, 7:70169–70184, 2019.
26. Waqar Anwar, Norman Franchi, and Gerhard Fettweis. Physical layer evaluation of v2x communications technologies: 5g nr-v2x, lte-v2x, ieee 802.11bd, and ieee 802.11p. In *2019 IEEE 90th Vehicular Technology Conference (VTC2019-Fall)*, pages 1–7, 2019.
27. Valerian Mannoni, Vincent Berg, Stefania Sesia, and Eric Perraud. A comparison of the v2x communication systems: Its-g5 and c-v2x. In *2019 IEEE 89th Vehicular Technology Conference (VTC2019-Spring)*, pages 1–5, 2019.
28. C. Partridge, T. Mendez, and W. Milliken. *Host Anycasting Service*, 1993.
29. T. Herbert, L. Yong, and O. Zia. *Generic UDP Encapsulation*, 2016.
30. Raza Salman, Wang Shangguang, Ahmed Manzoor, and Anwar Muhammad, Rizwan. A survey on vehicular edge computing: Architecture, applications, technical issues, and future directions. *Wireless Communications and Mobile Computing*, 2019(1):1–19, 2019.
31. Shahrudin Awang Nor, Raaid Alubady, and Wisam Abduladeem Kamil. Simulated performance of tcp, sctp, dccp and udp protocols over 4g network. *Procedia Computer Science*, 111:2–7, 2017.
32. Rahul Malhotra, Vikas Gupta, and Dr R. K. Bansal. Simulation & performance analysis of wired and wireless computer networks. *Global Journal of Computer Science and Technology*, 2011.
33. R. Hinden and S. Deering. *IPv6 Multicast Address Assignments*, 1998.
34. Shirosato test center. Japan Automobile Research Institute.
35. Tongyi Huang, Wu Yang, Jun Wu, Jin Ma, Xiaofei Zhang, and Daoyin Zhang. A survey on green 6g network: Architecture and technologies. *IEEE Access*, 7:175758–175768, 2019.
36. Wireshark. Wireshark Foundation.
37. Google maps platform. Google Developers.