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

Kohei Hosono¹ · Akihiko Maki² · Yousuke Watanabe³ · Hiroaki Takada³ · Kenya Sato^{1,4}

Received: 8 June 2021 / Revised: 4 November 2021 / Accepted: 3 December 2021 / Published online: 24 January 2022
© The Author(s) 2022

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

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 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

✉ Kohei Hosono
kohei.hosono@nislabs.doshisha.ac.jp

¹ Graduate School of Science and Engineering, Doshisha University, Kyotanabe, Kyoto 6100321, Japan

² Fujitsu Ltd, Kanagawa 2158588, Japan

³ The Institute of Innovation for Future Society, Nagoya University, Nagoya, Aichi 4648601, Japan

⁴ Mobility Research Center, Doshisha University, Kyoto 6100321, Japan

Fig. 1 Dynamic map overview

control and construction information, requires an update frequency of about one hour. Finally, static data, such as lane and structure information, is expected to be updated at a frequency of about one month. 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

Fig. 2 Dynamic map with edge servers

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, of high-precision road maps and sensor information, making it possible to develop and execute applications that provide transportation services more flexibly and easily. Sensor and signal information transmitted from vehicles, shown as dynamic data in Fig. 1, are generally required to be updated at a frequency of 100 ms in a dynamic map [14]. Transient dynamic data, which involves accident and traffic information, requires an update frequency of about one minute, while transient static data, which involves traffic

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

Fig. 3 Dynamic map using anycast address

while keeping the data in front of the Internet. Parvez et al.'s research investigates network architectures that use core networks to achieve low-latency communication when MEC is used in robot control and manufacturing [20]. However, those architectures assume that end devices are fixed or move within a very narrow range; thus, they do not account for vehicles with a wide range and high speed of movement, as in this study. Accordingly, the dynamic map system needs to aggregate the data sent from vehicles to appropriate edge servers. We thus propose a dynamic map system based on MEC, in which each edge server has jurisdiction over a geographically segmented area. To facilitate this system, we propose an access method in which edge servers are linked through a core network in order to operate dynamic maps more efficiently for vehicles moving at high speed.

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 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

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

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.

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

We define two types of areas covered by edge servers. The receiving area is the area under the jurisdiction of the dynamic map running on each edge server. The jurisdictional

area is the geographical area under the jurisdiction of the dynamic map running on each edge server. Each edge server aggregates the vehicle data within its jurisdictional area, regardless of the reception area. The reason for this is that the reception area of each edge server may change due to the influence of radio wave conditions. In addition, the reception area of each edge server may have overlapping ranges. In this paper, we propose a method to aggregate the data of vehicles in a dynamic map, which can be used for applications that have a high impact on traffic safety. The proposed method can determine the edge server to send the vehicle data based on the vehicle location. In order to operate in a large-scale environment in the future, we are conducting a demonstration experiment in a natural environment using a dynamic map system based on this method. Here, we assume that the jurisdictional area of the edge server is different from the receiving area or that the receiving areas overlap. We confirm that the data sent from the vehicle is correctly received by the dynamic map on the target edge server.

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.

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 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

Fig. 4 Resend function using ACK

Fig. 5 Transfer data using ACK

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 Related Works

Usually, an end device sends data to an edge server by specifying the IP unicast address of the destination edge server. Adika's research proposes the use of anycast addresses as an access method from end devices to edge servers in multi-access edge computing [34]. In this research, however, the objective is to send data to the nearest edge server, and it is thus unknown which edge server will receive the data sent from a vehicle. A dynamic map requires aggregating the vehicle data in the area under the jurisdiction of each edge server. In addition, because vehicles keep moving at high speed across the areas of different edge servers, it is necessary to resolve the edge server to which the data is sent. Hence, in this paper, we propose a dynamic map based on an access method that uses anycast addresses to link fast-moving vehicles and edge servers.

Mobile IP is a protocol that allows devices to connect with a unique IP address regardless of the network to which they are connected [35, 36]. Mobile IP operates at the network layer and provides mobility transparency by maintaining sessions even when nodes move. However, in dynamic maps for use in cooperative automatic driving, applications require a vehicles location information, and it is not necessary to guarantee location transparency, which is a feature of Mobile IP. Therefore, by managing the location information in the application layer rather than in the network layer, the proposed system can use the existing network layer mechanism instead of the complicated Mobile IP mechanism. Specifically, a custom header is introduced between the UDP header and the payload. By managing the location information at the application layer, a flexible dynamic map system can be implemented for various network environments. Furthermore, the use of Mobile IP requires communication through a home agent, which entails a redundant triangular path and thus causes issues such as reduced communication efficiency and a single point of failure. For these reasons, in this paper, we propose an efficient access method between a vehicle and an edge server by using IP anycast addresses, which can use the existing network layer mechanism instead of Mobile IP. In addition, Mobile IPv6 and its extensions allow knowing location information and avoiding redundant triangular transmissions. Nevertheless, Mobile IPv6 cannot be applied to the use cases such as merging arbitration because the edge server needs to be determined by accurate location information based on GPS. Most mobile operators do not support Mobile IPv6 from the perspective of efficient mobility support (i.e., the current mobile networks provide mobility support in L2 instead of L3). Therefore, as

mentioned in the previous comment, we propose an access method for dynamic map systems that adopt various available communication methods.

4 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.

4.1 Access Methods

4.1.1 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.

4.1.2 UDP Access Method Using Anycast & with Retransmission by ACK

As in Section 4.1.1, 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.

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

4.1.3 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.

4.2 Scenarios for Simulation

To evaluate the effectiveness of the access method of the dynamic map system, we conducted a simulation of vehicle 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 Shiroato Test Center of the Japan Automobile Research Institute [37]. The high-speed track of the Shiroato 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.

4.3 Dynamic Map System

Tables 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

Fig. 6 Simulation at Shirosato Test Center

Fig. 7 Dynamic map system configuration

Table 3 Vehicle configuration

OS	Ubuntu 16.04
CPU	Intel Core i7-8665U 4Core 8thread (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	Intel Core i9-9900K 8Core 16thread (3.60GHz)
Memory	16 GB
SSD	256 GB
Network	Up to 1 Gbps

communication depends not only on system factors such as the access method proposed in this paper but also on communication conditions [38]. 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 [39]. We also use the Google Maps Platform to visualize the location of the vehicles and the edge servers [40].

5 Results

We evaluated the UDP access method using anycast addresses, the retransmission function using ACK, and the data transfer function between edge servers, in terms of the packet loss rate, throughput, and latency of each system. For each scenario, we conducted 20 trials of a 5-minute simulation.

Fig. 8 Packet loss rate by three access methods **Fig. 9** Throughput by three access methods

5.1 AccEss Method Using Anycast Address

We compared our proposed UDP access method using anycast addresses with the conventional TCP access method using unicast addresses. First, we explain the evaluation results obtained with one vehicle. As shown in Fig. 8, the packet loss rate was 0 % for all trials with the TCP access method. For the UDP access method, the average packet loss rate μ was 0.05 % and the 95 % confidence interval by the chi-squared distribution was $0.033 \% \leq \mu \leq 0.058 \%$. The reason for this result is that UDP is a connectionless protocol, so packet loss may occur when there is no retransmission function in the upper protocol.

Next, we measured the throughput on the vehicle side, which sends the data, and on the edge server side, which receives it. As shown in Fig. 9, the sending throughput averaged 18.1Kbps for both the TCP access method and the UDP access method, and the receiving throughput averaged 18.0Kbps for both access methods.

Finally, we measured the latency as the time it took for the data sent from the vehicle to reach the dynamic map

Fig. 10 Latency by three access methods

Fig. 12 Effect of number of vehicles on throughput

Fig. 11 Effect of number of vehicles on packet loss rate

Fig. 13 Effect of number of vehicles on latency

application on the edge server that needed the data. As shown in Fig. 10, for the TCP access method, the average latency was $7.9ms$ with a 95 % confidence interval of $7.7ms \leq \mu \leq 8.2ms$. For the UDP access method, the average latency was $5.1ms$ with a 95 % confidence interval of $4.9ms \leq \mu \leq 5.3ms$. The latency of the TCP access method using the unicast address was large because of the influence of TCPs three-way handshake and retransmission control, whereas the proposed UDP access method using the anycast address enabled communication with low latency.

This experiment was performed not only with the measurement vehicle but also with simulated load vehicles running around the same course. Specifically, we ran the simulation with 1000, 3000, 5000, 7500, 10000, and 12000 load vehicles. We assumed that all the load vehicles ran at varying speeds and sent data to the edge server in the same way as the measurement vehicle.

As shown in Fig. 11, with the TCP access method using the unicast address, the packet loss rate remained low up to

a certain number of vehicles but then increased rapidly. This may be due to the fact that our system reached its throughput limit because of the use of TCP. In contrast, with the UDP access method using the anycast address, the packet loss rate remained lower than with the TCP access method even as the number of vehicles increased. However, some packet loss occurred even with a small number of vehicles because of the use of UDP.

Figure 12 shows the throughput of the server at the receiving end when the number of vehicles was varied. With the proposed UDP access method, the throughput increased as the number of vehicles increased, with an upper limit of about $950Mbps$, beyond which packets were discarded. As for the TCP access method, the upper limit was about $650Mbps$, and packets beyond that were also discarded. This may be due to the fact that the throughput of TCP decreases with the RTT. For dynamic maps that send and receive huge amounts of data, higher throughput would make the UDP access method more effective. In this experiment, we used Ethernet to verify the effectiveness of the access method.

However, actual vehicles are expected to communicate wirelessly, and the RTT is likely to decrease; thus, the proposed method would be effective because it can achieve high throughput independently of the RTT.

As shown in Fig. 13, the latency with the UDP access method using the anycast address did not change significantly as the number of vehicles increased, whereas the latency of the TCP access method using the unicast address did increase with the number of vehicles. This increasing tendency with the TCP access method was probably due to the influence of TCPs retransmission control and congestion control. In a dynamic map system, real-time performance is a very important issue, and the shorter the latency in sending and receiving data, the better. Accordingly, the proposed UDP access method has the benefit of providing data transmission with very low latency as compared to TCP communication.

5.2 ACK Retransmission

As described in Section 5.1, our proposed UDP access method using anycast addresses can be used in dynamic map systems to efficiently manage a large number of vehicles. However, UDP communication is less reliable and more prone to packet loss as compared to TCP communication. In our experiment, as shown in Fig. 8, the UDP access method caused slightly more packet loss than the TCP access method. Accordingly, we implemented the ACK retransmission function described in Section 2 to improve the reliability of the proposed UDP access method, and we compared the performance with and without ACK retransmission.

First, we explain the evaluation results obtained with only one measurement vehicle. As shown in Fig. 8, the average packet loss rate was 0.05 % without the ACK retransmission function, but it was reduced to 0 % in all trials by implementing the retransmission function. As shown in Fig. 9, the throughputs of the sending vehicle and the receiving server averaged 18.1 and 18.0 Kbps, respectively, both with and without ACK retransmission. Furthermore, the average latency was 6.0 ms both with and without ACK retransmission, as shown in Fig. 10.

This experiment was also performed with simulated load vehicles (1000, 3000, 5000, 7500, 10000, and 12000) running around the same course. The load vehicles ran at different speeds and sent data to the edge server in the same way as the measurement vehicle.

As shown in Fig. 11, the packet loss rate with the ACK retransmission function remained lower than that without the retransmission function even when the number of vehicles increased. When the number exceeded approximately 10000, a small amount of packet loss occurred even with the retransmission function, but this was because the sent

Fig. 14 Packet loss rate by data transfer between edge servers

packets exceeded 1 Gbps, which was the bandwidth of the dynamic map system used in this experiment.

Figure 12 shows that the throughput of the server at the receiving end increased as the number of vehicles increased, with or without the ACK retransmission function. The throughput was capped at about 950 Mbps, and packets beyond that were discarded.

Finally, as shown in Fig. 13, there was almost no difference in latency between the cases with and without the ACK retransmission function. The slightly larger latency with ACK retransmission was probably due to the delay caused by retransmission when packet loss occurred.

5.3 Data Transfer Between Edge Servers

As described in Section 2, when edge servers are deployed to construct a dynamic map, each edge server needs to aggregate the data of all vehicles in the area where it operates. However, in the proposed system with a UDP access method using the anycast address, the data sent by a vehicle is not always received by the target edge server. Therefore, we implemented a function to transfer data between edge servers as needed. Basically, the operating area is determined according to the reception range of each edge server, so the number of vehicles that require data transfer is expected to be small, but it may depend on the communication situation. Accordingly, we evaluated the performance of the system with and without data transfer between edge servers. We ran the same simulation as above and compared the performance between the case without data transfer, in which the data sent from a vehicle was always received by the target edge server, and the case with data transfer, in which the data was always received by a non-target edge server and then transferred.

Again, we first describe the evaluation results obtained with only one measurement vehicle. As shown in Fig. 14, the packet loss rate was 0 % in all trials regardless of whether data was transferred between the edge servers. The latency,

Fig. 15 Latency by data transfer between edge servers

shown in Fig. 15, averaged 4.9ms without data transfer, with a 95 % confidence interval of $4.8\text{ms} \leq \mu \leq 5.0\text{ms}$; with data transfer between the edge servers, the mean latency was 5.6ms with a 95 % confidence interval of $5.5\text{ms} \leq \mu \leq 5.7\text{ms}$.

This experiment was also performed with simulated load vehicles (1000, 3000, 5000, 7500, 10000, and 12000) running around the same course. The load vehicles ran at different speeds and sent data to the edge server in the same way as the measurement vehicle.

As shown in Fig. 16, the packet loss rate remained similar regardless of the data transfer between the edge servers even when the number of vehicles increased. This result is similar to the result described in Section 5.2 for the proposed system with the ACK retransmission function. The packet loss rate increased when the number of vehicles exceeded about 10000, but this again occurred when the sent packets exceeded 1Gbps, the bandwidth of the dynamic map system.

As shown in Fig. 17, the latency was slightly higher when there was data transfer between edge servers than when there was no data transfer. However, this difference did not increase as the number of vehicles increased, and the latency was still lower than that of the conventional TCP access method using unicast addresses.

6 Discussion

In this paper, to evaluate the effectiveness of the proposed system, we conducted a multi-vehicle driving simulation with a dynamic map system. In our experiments, the UDP access method using anycast addresses achieved lower latency and higher throughput than the conventional TCP access method using unicast addresses. A dynamic map system requires low latency because its real-time operation is a very important issue. In addition, for a dynamic map system that manages a large number of vehicles, it is better to have high throughput in the same environment.

Fig. 16 Effect of number of vehicles on packet loss rate by data transfer between edge servers

Fig. 17 Effect of number of vehicles on latency by data transfer between edge servers

Accordingly, we showed that the proposed UDP access method is effective for dynamic map systems.

However, this UDP access method cannot guarantee the arrival of data because it uses the UDP protocol, and packet loss may occur. Hence, we implemented a retransmission function using ACK to guarantee the arrival of data while using UDP. The simulation results showed that the packet loss rate was as low as that of the conventional access method. As a result, by applying the UDP access method using anycast addresses and the ACK retransmission function, we have developed a dynamic map system that provides low-latency, high-capacity communication while maintaining the same high reliability as the conventional access method. In dynamic maps with a significant impact on traffic safety, data sent from vehicles must be transmitted to edge servers in real time. In our dynamic map system, it is possible to switch the ACK retransmission function on or off. As a result, we believe that the proposed UDP access method is effective because it can

be configured according to the requirements of a dynamic map application.

As described in Section 2, in a dynamic map system with edge servers, the data sent by a vehicle is not always received by the desired edge server. Accordingly, for our dynamic map system, we implemented a function to transfer data between edge servers. The latency for this data transfer is a little higher than the latency when the data sent from a vehicle is received directly by the target edge server. However, this overhead was experimentally found to be very small compared to the latency of the conventional access method. In addition, the data transfer between the edge servers improved the packet loss rate and did not degrade the throughput, thus demonstrating that the proposed system is effective.

In each simulation, the number of vehicles was increased to evaluate the scalability. As a result, we found that the proposed system could handle about 9500 vehicles with low latency and high reliability, whereas the conventional access system could handle only about 5500 vehicles. This was because the bottleneck in each system is the limit of processing by physical resources. If we have more such resources, we can manage more vehicles; however, the latency of the proposed system is lower than that of the conventional access method even with more physical resources. In addition, the proposed system requires fewer resources to manage a certain number of vehicles. Overall, these results suggest that our access system is more effective than the conventional access system.

In this paper, to evaluate the specific effectiveness of the access method, we constructed a dynamic map system using wired communication to mitigate the uncertainty of wireless communication. In the future, we plan to conduct a demonstration experiment using wireless technology, which will be closer to the actual application environment. Dynamic maps will need to cover a huge number of roads in the future, and it will be necessary to assign the same anycast address to each edge server in order to access the map via anycast addresses. Therefore, we will also need to discuss the need for a provider to assign these addresses.

7 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.

Declarations

Conflicts of interest The authors declare that they have no conflict of interest.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Bresson, G., Alsayed, Z., Yu, L., Glaser, S.: Simultaneous localization and mapping: A survey of current trends in autonomous driving. *IEEE Transactions on Intelligent Vehicles* **2**(3), 194–220 (2017)

2. Yoshihiro, S., Keiji, A.: Current activities and some issues on the development of automated driving. *Johokanri* **57**(11), 809–817 (2015)
3. Bengler, K., Dietmayer, K., Farber, B., Maurer, M., Stiller, C., Winner, H.: Three decades of driver assistance systems: Review and future perspectives. *IEEE Intelligent Transportation Systems Magazine* **6**(4), 6–22 (2014)
4. Sivaraman, S., Trivedi, M.M.: 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. Milanés, V., Shladover, S.E., Spring, J., Nowakowski, C., Kawazoe, H., Nakamura, M.: Cooperative adaptive cruise control in real traffic situations. *IEEE Transactions on Intelligent Transportation Systems* **15**(1), 296–305 (2014)
6. Chen, C.T., Chen, Y.S.: Real-time approaching vehicle detection in blind-spot area. In: 2009 12th International IEEE Conference on Intelligent Transportation Systems, pp. 1–6 (2009)
7. Lin, B.-F., Chan, Y.-M., Fu, L.-C., Hsiao, P.-Y., Chuang, L.-A., Huang, S.-S., Lo, M.-F.: 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. Wang, J., Shao, Y., Ge, Y., Yu, R.: A survey of vehicle to everything (v2x) testing. *Sensors* **19**(2) (2019)
9. Kuutti, S., Fallah, S., Katsaros, K., Dianati, M., McCullough, F., Mouzakitis, A.: 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. Siegel, J.E., Erb, D.C., Sarma, S.E.: 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. Mukhtar, A., Xia, L., Tang, T.B.: Vehicle detection techniques for collision avoidance systems: A review. *IEEE Transactions on Intelligent Transportation Systems* **16**(5), 2318–2338 (2015)
12. Thomas, E., Fèderer, H., Kasslatter, F., Parreira, J.X., Patrik, S.: Towards a semantically enriched local dynamic map. *International Journal of Intelligent Transportation Systems Research* **17**(1), 32–48 (2019)
13. Yousuke, W., Kenya, S., Hiroaki, T.: 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, N., Sasaki, K., Makido, S.: A prediction handover method between edge servers on infrastructure-based vehicle control system. Technical Report 6, Toyota Cenral R&D Labs., Inc. (2018)
16. Liu, Y., Yu, H., Xie, S., Zhang, Y.: 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. Zhou, F., Hu, R.Q., Li, Z., Wang, Y.: 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. Parvez, I., Rahmati, A., Guvenc, I., Sarwat, A.I., Dai, H.: A survey on low latency towards 5g: Ran, core network and caching solutions. *IEEE Communications Surveys Tutorials* **20**(4), 3098–3130 (2018)
21. He, Z., Zheng, L., Chen, P., Guan, W.: 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. Atia, M.M., Hilal, A.R., Stellings, C., Hartwell, E., Toonstra, J., Miners, W.B., Basir, O.A.: 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. Zang, A., Li, Z., Doria, D., Trajcevski, G.: 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, Autonomous-GIS '17 (2017)
24. Mobility Open Blockchain Initiative (MOBI). VID Business White Paper (2021)
25. Naik, G., Choudhury, B., Park, J.-M.: IEEE 802.11bd 5g nr v2x: Evolution of radio access technologies for v2x communications. *IEEE Access* **7**, 70169–70184 (2019)
26. Anwar, W., Franchi, N., Fettweis, G.: 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), pp. 1–7 (2019)
27. Mannoni, V., Berg, V., Sesia, S., Perraud, E.: A comparison of the v2x communication systems: Its-g5 and c-v2x. In: 2019 IEEE 89th Vehicular Technology Conference (VTC2019-Spring), pp. 1–5 (2019)
28. Milliken, W., Mendez, T., Partridge, C.: Host Anycasting Service. RFC 1546 (1993)
29. Herbert, T., Yong, L., Zia, O.: Generic UDP Encapsulation (2016)
30. Salman, R., Shangguang, W., Manzoor, A., Rizwan, A.M.: A survey on vehicular edge computing: Architecture, applications, technical issues, and future directions. *Wireless Communications and Mobile Computing* **2019**(1), 1–19 (2019)
31. Nor, S.A., Alubady, R., Kamil, W.A.: Simulated performance of tcp, sctp, dccp and udp protocols over 4g network. *Procedia Computer Science* **111**, 2–7 (2017)
32. Malhotra, R., Gupta, V., Bansal, R.K.: Simulation & performance analysis of wired and wireless computer networks. *Global J. Comput. Sci. Technol.* (2011)
33. Deering, S.E., Hinden, B.: IPv6 Multicast Address Assignments. RFC 2375 (1998)
34. Sulaeman, A.: A Highly-Available Multiple Region Multi-access Edge Computing Platform with Traffic Failover. Master's thesis, Aalto University. School of Science (2020)
35. Perkins, C.E.: IP Mobility Support for IPv4. RFC 3344 (2002)
36. Perkins, C.E.: IP Mobility Support. RFC 2002 (1996)
37. Shirosato test center. Japan Automobile Research Institute
38. Huang, T., Yang, W., Wu, J., Ma, J., Zhang, X., Zhang, D.: A survey on green 6g network: Architecture and technologies. *IEEE Access* **7**, 175758–175768 (2019)
39. Wireshark. Wireshark Foundation
40. Google maps platform. Google Developers

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Kohei Hosono graduated from the Department of Information Systems Design, Faculty of Science and Engineering, Doshisha University in 2020. In the same year, he entered the master's program in Information Engineering at the Graduate School of Science and Engineering, Doshisha University. He is engaged in research on a communication platform for cooperative automated vehicles using edge computing.

Akihiko Maki is working in the Software and Service Business Division, Network Solutions Business Group, Fujitsu Limited. He received his M.S. in Electronic and Information Engineering from Tokyo University of Agriculture and Technology in 2002, and joined the current company.

Yousuke Watanabe received M.E. and Dr.E. degrees in University of Tsukuba in 2003 and 2006. In 2014, he joined Institute of Innovation for Future Society, Nagoya University, as a designated associate professor. His research interests include data stream processing and information integration. He is a member of the Database Society of Japan, IEICE, and ACM.

Hiroaki Takada is a professor at Institutes of Innovation for Future Society, Nagoya University. He is also a professor and the Executive Director of the Center for Embedded Computing Systems (NCES), the Graduate School of Informatics, Nagoya University. He received his Ph.D. degree in Information Science from University of Tokyo in 1996. He was a Research Associate at University of Tokyo from 1989 to 1997, and was a Lecturer and then an Associate Professor at Toyohashi University of Technology from 1997 to 2003. His research interests include real-time operating systems, real-time scheduling theory, embedded system design, and automotive embedded systems. He is a fellow of IPSJ and JSSST, and is a member of ACM, IEEE, IEICE, and JSAE.

Kenya Sato is a professor of Doshisha University, Kyoto, Japan. He also currently leads the Mobility Research Center of the university, and serves as a designated professor of Nagoya University. He received the BE and ME degree from Osaka University in 1984 and 1986. He joined Sumitomo Electric Industries in 1986. He was a visiting researcher at the Computer Science Department, Stanford University from 1991 to 1994. He received the Ph.D. degree from Nara Institute of Science and Technology in 2000. After working as Chief Technologist at AMI-C, Inc. in Michigan, U.S., he joined Doshisha University in 2004. Dr. Sato is a member of ACM, SAE and Information Processing Society of Japan.