

Implementation and Evaluation of Load Balancing Mechanism With Multiple Edge Server Cooperation for Dynamic Map System

Kohei Hosono^{ID}, *Member, IEEE*, Akihiko Maki^{ID}, *Member, IEEE*, Yosuke Watanabe^{ID}, *Member, IEEE*, Hiroaki Takada^{ID}, *Member, IEEE*, and Kenya Sato^{ID}, *Member, IEEE*

Abstract—In recent years, research has been conducted on dynamic map systems, which are information and communication platforms that manage vehicle sensor information and run applications. However, there is a concern about scalability in the dynamic map system, which operates on a cloud on the Internet, when the number of vehicles that transmit and receive sensor information increases. Therefore, it is considered that the application load can be distributed by distributing multiple edge servers geographically and having the information managed in the cloud by the edge servers. However, the edge server that controls the vehicle information and the edge server that receives the data do not always match depending on the actual radio wave conditions. Therefore, some applications are difficult to manage, such as intersection collision danger warning and merging arbitration when edge servers are assigned in the same way as base stations that receive data from vehicles. Such an application should receive all vehicle data for the target road area. Therefore, we divide the area on the road where a vehicle travels as “lane section ID” and assign an edge server to each lane section ID on the basis of that area. In addition, we implemented a dynamic map system that connects vehicles and edge servers by linking multiple edge servers. This enables the edge server to aggregate vehicle data without being affected by radio traffic conditions. We evaluated the scalability of the dynamic map system and verified the effectiveness of the load balancing mechanism using multiple edge servers.

Index Terms—ITS, dynamic map, edge computing, load balancing.

I. INTRODUCTION

CURRENTLY, advanced safe driving support systems and automatic driving systems are being researched and developed that recognize the driving environment by using sensors mounted on the vehicle and that automatically warn drivers and avoid dangers [1]–[3]. However, the on-board sensor can detect objects in the visible range but not in

Manuscript received 25 March 2020; revised 31 August 2020, 13 November 2020, 29 January 2021, and 7 February 2021; accepted 11 March 2021. Date of publication 30 March 2021; date of current version 8 July 2022. This work was supported in part by JSPS KAKENHI under Grant JP20H00589. The Associate Editor for this article was M. Brackstone. (*Corresponding author: Kohei Hosono.*)

Kohei Hosono is with the Computer and Information Science, Graduate School of Science and Engineering, Doshisha University, Kyoto 610-0321, Japan (e-mail: kohei.hosono@nislabs.doshisha.ac.jp).

Akihiko Maki is with Fujitsu Ltd., Kanagawa 215-8588, Japan.

Yosuke Watanabe and Hiroaki Takada are with the Institute of Innovation for Future Society, Nagoya University, Nagoya 464-8601, Japan.

Kenya Sato is with the Mobility Research Center, Doshisha University, Kyoto 610-0321, Japan.

Digital Object Identifier 10.1109/TITS.2021.3067909

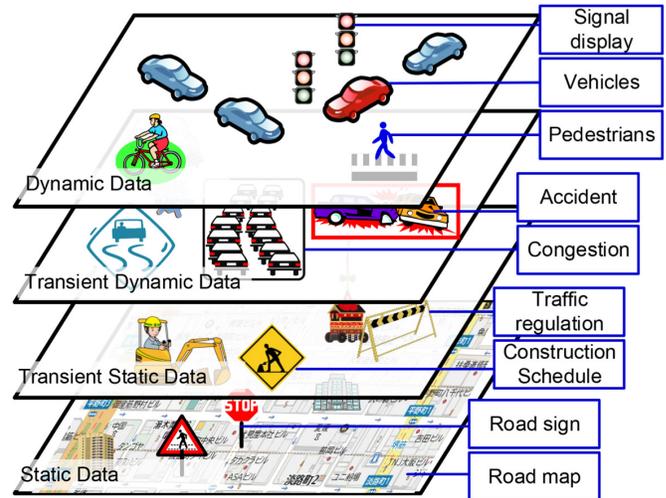


Fig. 1. Overview of dynamic map.

the invisible range. For example, in-vehicle sensors have difficulty coping with a collision at an intersection with poor visibility or something suddenly appearing out of a shadow.

To cope with these situations, cooperative ITS (Intelligent Transport Systems) are intended to improve safety by exchanging information between running vehicles or between vehicles and roadside units by using wireless communication technology [4]–[6]. Various applications of collaborative ITS is being studied, such as intersection collision danger warning, information on traffic congestion, road surface conditions, and signal status, as well as merging support on expressways [7], [8]. Currently, data transmitted from vehicles is managed and processed separately for each application. However, by managing the data as a dynamic map [9]–[12] and integrating all information on the basis of the base map information, the application becomes possible to execute efficiently [13]. Figure 1 shows the structure of the dynamic map.

The dynamic map system in this study has a structure in which dynamic information is hierarchized on a static road map in accordance with the update frequency of information. In general, dynamic information handled by a dynamic map is transmitted from a vehicle within 100 milliseconds, and sensor information is managed. An application is required that realizes safe driving support and cooperative automatic driving

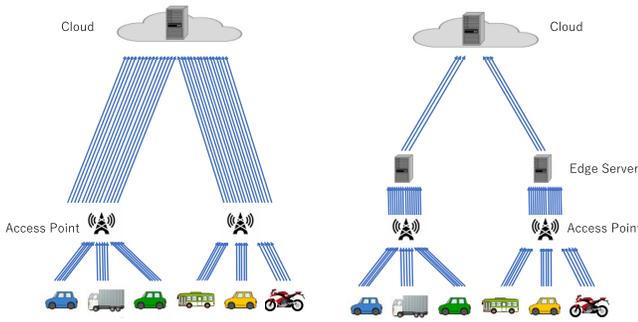


Fig. 2. Placement and operation of edge server.

operations on the basis of the dynamic map, so information needs to be processed with low delay [14].

Usually, the dynamic map system is used in a cloud on the Internet [15]. However, scalability is a problem if data from a huge number of vehicles is aggregated in the cloud, causing processing load and communication delay. Therefore, locating the edge server between the cloud and the vehicle is expected to distribute the processing load and reduce communication delays [16]–[22]. The connection method between the edge server and the vehicle and the management method of the vehicle are also challenges because the edge server needs to continuously provide information to a vehicle moving at high speed. To reduce the cloud scalability problem, which occurs as the number of vehicles increases, we implement an edge server in a mobile phone base station, for example, to achieve load balancing. Figure 2 compares the network load between the dynamic map system using the cloud and the dynamic map system using the edge server.

Although there have been several suggestions about where to place the edge server [17], [23], [24], such as around mobile phone base stations and at the entrance to the Internet, the model proposed by ETSI is the most promising for connected cars [25]–[27]. In this study, we follow that model and place the edge server at a location corresponding to a mobile phone base station that sends and receives data to and from the mobile vehicle. The vehicle sends data to the edge server. Here, the data transmitted from the vehicle is vehicle ID, vehicle position, speed, traveling direction, time stamp, etc. [14], [28], and the vehicle uses vehicle position information obtained by locating the vehicle position by GPS (global positioning system) [29] or scan matching [30]. A dynamic map system with edge servers manages global information in the cloud and local information that requires real-time processing in the edge server. This dynamic map system has a three-layer structure consisting of a cloud, an edge server, and a vehicle. Figure 3 shows the configuration of the dynamic map system.

We call cloud, edge, and vehicle nodes. In the three-layered dynamic map system, the network load can be reduced and the real-time performance can be improved by distributing the data collection and distribution processing between the vehicle and the nearest edge server. In the dynamic map system in this research, each node consists of four layers: OS (operating system)/hardware, communication unit,

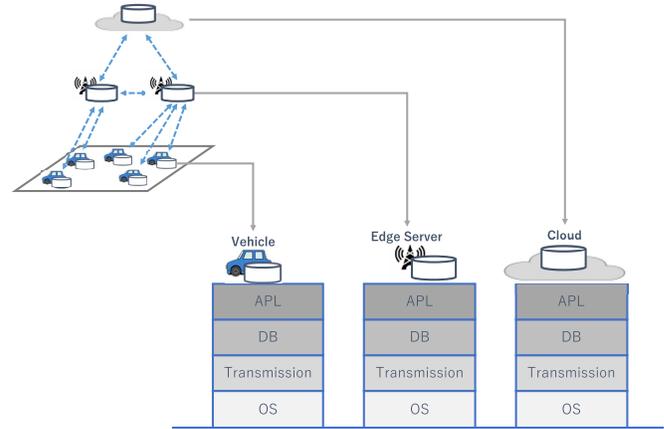


Fig. 3. Architecture of dynamic map system.

DB (database) system, and application. Each node has a unique ID [31]–[33] as well as the cooperative ITS system [34] that is being standardized in Europe. Each node communicates through the communication unit, and the transmitted and received data is processed by the application of each node. We construct a dynamic map system that can be used as a common base for linking sensor information and map data, making applications easy to build.

II. RELATED WORKS

Dynamic Map is one of the key technological elements in realizing automated driving by communicating between vehicles and servers. Also, the scalability of dynamic maps, which aggregate data from a huge number of vehicles, is a concern, and edge servers are expected to be placed between the cloud and vehicles to distribute the processing load and reduce communication delays. ETSI [25] is considering placing these edge servers around cell phone base stations.

When a vehicle transmits data to an edge server, the edge server needs to implement an application [35]–[37] that depends on a specific position (for example, an intersection or a junction of a road) such as merging and arbitration. At this time, the edge server needs to possess all the vehicle data required by that application. However, in a related study [25], vehicle data outside the area under the jurisdiction of the edge server could not be known, and the data had to be aggregated in the cloud. In addition, the edge server does not always receive the data sent from the vehicle at a specific location due to the layout of the surrounding buildings and the signal status of the mobile phone base station [38]–[40]. Under these circumstances, even if edge servers are deployed, they are unlikely to improve efficiency through distributed processing. To construct a dynamic map system by arranging edge servers, vehicles and edge servers need to be connected geographically. Therefore, we construct a dynamic map system that geographically connects edge servers on the basis of vehicle location information by using lane section IDs and evaluate its effectiveness.

Efficient vehicle-to-server communication methods using edge servers have also been studied by Xie *et al.* [21] and

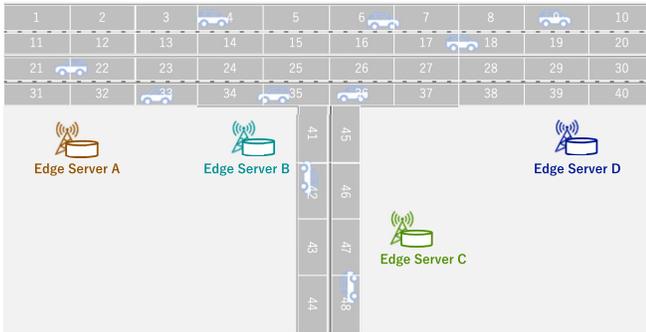


Fig. 4. Lane section ID.

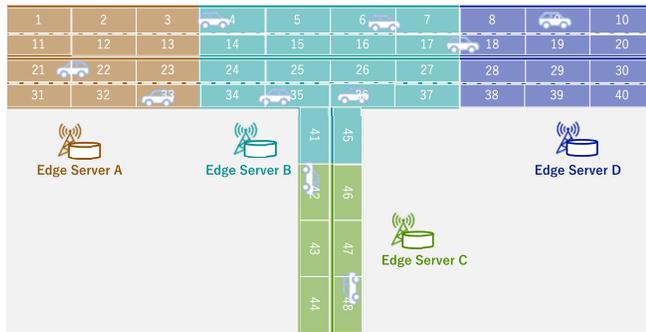


Fig. 5. Edge server arrangement based on lane section ID.

Li *et al.* [22]. These studies propose a Vehicle-to-Edge system using an edge server. However, these studies do not discuss the method of aggregating all vehicle data in the target area to the edge server. For dynamic maps that deal with applications that have a high impact on traffic safety, it is necessary to consider a method in which geographically distributed edge servers aggregate the data of each vehicle. Therefore, in this study, we construct a dynamic map system that geographically attaches edge servers based on the location information of vehicles using lane section IDs, and we evaluate its effectiveness.

III. EDGE SERVER ASSIGNMENT BASED ON LANE SECTION ID

A. Lane Section ID

In this study, when a vehicle sends data to an edge server, the edge server compares vehicle position information measured by GPS or scan matching with detailed map data containing lane level information. This is compared with the lane section ID in which the vehicle runs, and the information is added to the data and transmitted to the edge server. This lane section ID divides the lane of the road on which the vehicle runs into sections of a specific length and assigns a unique ID to each section. Figure 4 shows an example of a vehicle lane section ID, so the length and number of sections can be freely changed and the application can run efficiently.

The edge server that manages the information of the divided geographical area (set of lane section IDs) is called the jurisdiction edge server. Figure 5 shows an example of the relationship between the lane section ID and the jurisdiction

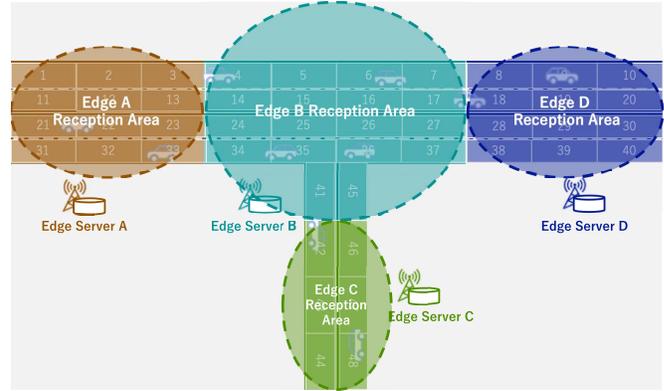


Fig. 6. Reception area for lane section ID.

edge server. Each edge server has jurisdiction over a lane in the corresponding color range and must aggregate the data sent by the vehicles in that range and process it, for example, in an application. As described in a related study [25], the jurisdictional area of the edge server is basically the same as the allocation area of the nearby mobile phone base station. This enables the edge server to identify the vehicle's jurisdictional edge server by checking the lane section ID of the data sent by the vehicle.

The edge server receives the data sent by the vehicle, but the receiving edge server is determined in accordance with the allocation of the mobile phone base station. Also, the jurisdictional area of the edge server is determined to be equivalent to the allocated area of this mobile phone base station. Therefore, as shown in Figure 6, the jurisdictional area of the edge server and the receiving area are basically the same.

The data sent by the vehicles in the reception area of Edge Server A is received by Edge Server A in accordance with the assignment of the mobile phone base station. In addition, the data sent by vehicles in the jurisdictional area of Edge Server B must be executed by an application implemented on Edge Server B. The reception area and the jurisdictional area are configured to be basically the same. The reason for defining each separately, however, is that this area may be unintentionally different or may require vehicle data that is outside the jurisdictional area of a particular edge server.

B. Load Balancing Mechanism by Vehicle Data Transfer

For example, the edge server receives information from vehicles around the intersection, judges the driving conditions of the vehicles, and gives driving instructions. This requires that the data of the vehicles around the intersection be aggregated into an edge server that runs the application. Since the control area of the edge server is allocated equally to the allocated area of the mobile phone base stations around the edge server, the boundary of the edge server's jurisdictional area may be located within the intersection, as shown in Figure 7. At this time, if the intersection mediation application is run on Edge Server A, Edge Server A may need data on vehicles in the jurisdictional areas of Edge Servers B and C. Therefore, the

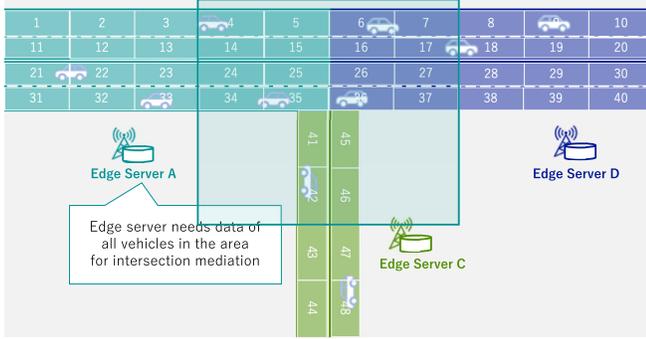


Fig. 7. Deviation between control area and required range.

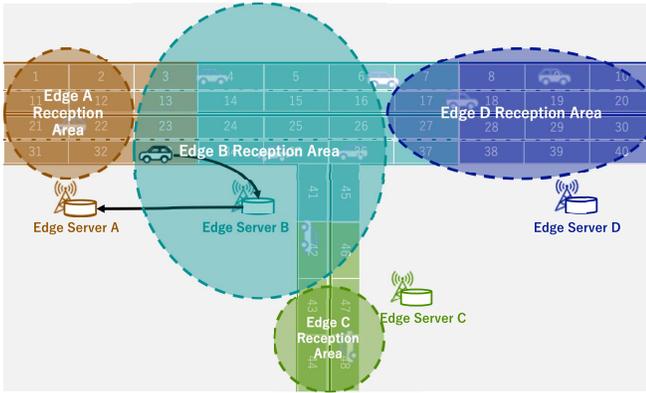


Fig. 8. Lane section ID transfer.

system in the related study [25] needs to aggregate vehicle data in the cloud and run applications.

However, in actual wireless communication such as mobile telephone networks, the edge server associated with the intended access point or base station is not always received in accordance with the radio wave conditions. In addition, the same edge server does not always receive data transmitted from a vehicle when using multiple communication means such as a mobile phone network and a wireless LAN. Even if the area under the jurisdiction of the edge server and the receiving area are allocated in the same way, as shown in Figure 6, Edge Server B may unintentionally receive the data in the area under the jurisdiction of Edge Server A due to the effects of radio wave conditions and other factors, as shown in Figure 8. In such cases, the related research system cannot run applications on the edge servers, and the data must be consolidated and the applications must be run in the cloud. However, this may cause communication and processing delays.

Therefore, in this study, we use the lane section IDs to achieve the cooperation function between edge servers. The edge server has a relationship with the lane section ID that oversees each edge server. Therefore, the edge server inspects the lane section ID in the data sent from the vehicle, and if the inspecting edge server itself is the edge server that controls the lane section ID, the inspecting edge server will be the upper application, and if another edge server controls the lane section ID, the inspecting edge server transfers the data to the relevant

edge server. At this time, the edge server receiving the data determines the necessity of transfer on the basis of the lane section ID in the communication unit. Therefore, data other than the data to be processed by the receiving edge server application is processed by the edge server application after transfer. This makes it possible to process the data at the edge server instead of the cloud when vehicle data outside the area under the jurisdiction of one edge server is needed.

In addition, since there are many vehicles under the jurisdiction of a particular edge server and scalability problems due to congestion are possible, we have been investigating a load balancing function between edge servers. Although there are currently no standard specifications for the length of the lane section IDs or the number of edge servers, this paper proposes a load balancing mechanism, and the length of the road and the lane section IDs can be configured in any way. Therefore, we study the effect of the number of lane sections on scalability.

We have implemented a load distribution mechanism that distributes processing by transferring the edge server on the basis of the lane section ID in the data transmitted from the vehicle. This load distribution mechanism enables vehicles to transmit data to the jurisdiction edge server without being affected by radio wave conditions or means of communication.

IV. COMMUNICATION METHOD

In this system, IP (Internet protocol) is used for communication between the vehicle, the edge server, and the cloud. In a dynamic map system that uses only the cloud, when transmitting data, the vehicle uses the IP addresses of the vehicle and the cloud as the source and the destination, respectively. The cloud can use the IP address of the vehicle when sending data back to the vehicle. However, in the configuration using edge computing, the cloud does not directly receive the data from the vehicle, so the IP address of the destination vehicle cannot be specified. Therefore, in the communication unit of the edge server in this system, when the edge server receives data from the vehicle, the edge server notifies the cloud of the ID of the vehicle that transmitted the data, so that the cloud receives the data of which vehicle can be determined from which edge server. The cloud can use this aggregated ID to enable communication from the cloud, edge servers, and vehicles to specific moving vehicles. The edge server notifies the cloud only when the vehicle moves and the destination edge server is switched, thus reducing the communication load compared with the case where vehicle data is sent directly to the cloud periodically. Figure 9 shows how the edge server reduces the communication load.

In addition, since the moving vehicle travels while switching the reception area of multiple edge servers, the IP address of the edge server that is the destination when transmitting data from the vehicle cannot be set uniquely. One solution is a method in which the IP address information of the nearby edge server corresponding to the location information is provided in advance in each vehicle. However, for that purpose, the vehicle needs to have information of all edge servers, so the process of updating the information held by each vehicle becomes complicated. Furthermore, the data transmitted from the vehicle is

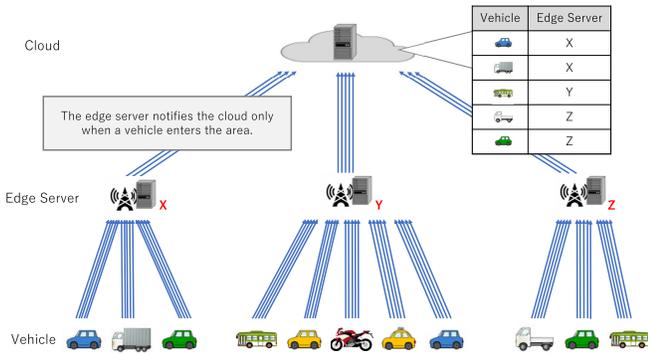


Fig. 9. Reduction of communication load by edge server.

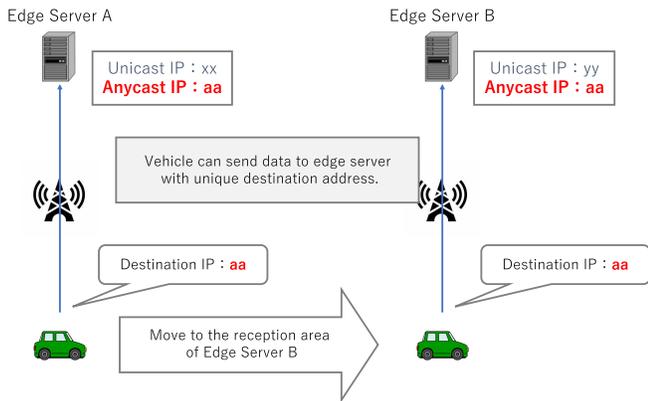


Fig. 10. Anycast in communication for vehicles.

not always received by the target edge server depending on the wireless communication situation. Another solution is a method in which a vehicle makes an inquiry to a server that manages location information and the IP address information of a nearby edge server and resolves the IP address of the edge server corresponding to the current location. However, the vehicle needs to resolve the IP address of the edge server converted into the location information before sending the data, so the network load increases.

To solve this problem, we use the anycast address [41] as the destination IP address when sending data from the vehicle to the edge server. This makes it possible to send data directly to a nearby edge server using a specific IP address, regardless of where the vehicle is or of the network configuration of the cloud or edge server. Figure 10 shows communication from the vehicle to the edge server using anycast.

On the other hand, when communicating from the edge server to the vehicle, the edge server needs to know the IP address of the vehicle that is the destination. Therefore, we use multicast communication [42] for communication from the edge server to the vehicle. Figure 11 shows the multicast communication from the edge server to the vehicle.

Each vehicle can determine whether it is received or discarded by looking at the destination vehicle ID in the data received by the communication unit. This allows the edge server to send data without knowing the unicast address of the vehicle under its control. In addition, the edge server can reduce communication congestion in the network by notifying

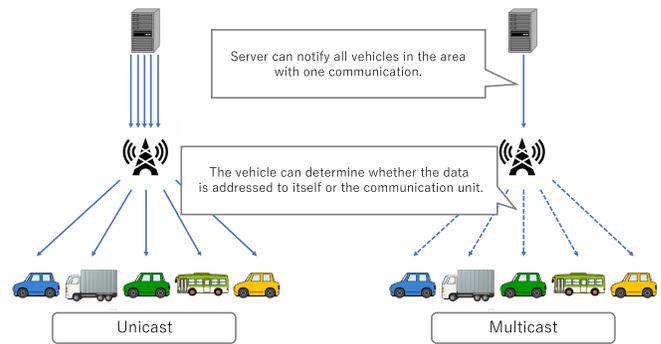


Fig. 11. Multicast in communication for vehicles.

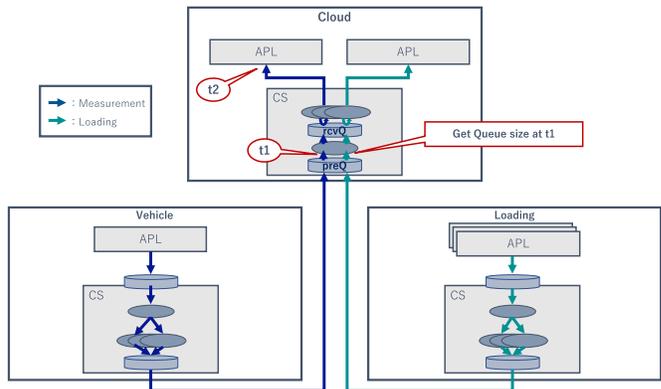


Fig. 12. Data flow of cloud computing.

multiple vehicles with the same message if the message is the same.

V. EVALUATION SYSTEM

To evaluate the scalability of this system, a dynamic map system has been constructed with the cloud, edge servers and vehicle computers. In the cloud-based dynamic map system without using lane section IDs in related research [25], the vehicle acquires sensor information and other information using the vehicle application and sends it to the cloud through the communication unit. The data is temporarily stored in the cloud queue, and if it is not a transfer target, it is stored again in another queue. At that time, the data is not transferred in the dynamic map system by the cloud. After that, the data is sent to the application in the cloud for processing. Figure 12 shows the data flow of the dynamic map system using the cloud.

In a dynamic map system in which an edge server is assigned on the basis of the lane section ID, the vehicle acquires sensor information and other information using the vehicle application and sends it to the edge server through the communication unit. The edge server temporarily stores the received data in a queue, and if the received data is not a transfer target, the edge server stores the data again in another queue. After that, the data is sent to the application in the edge server and processed. Figure 13 shows the data flow of the dynamic map system using the edge server.

If the received data is the transfer target, the edge server transfers the data between the edge servers. After that, the data

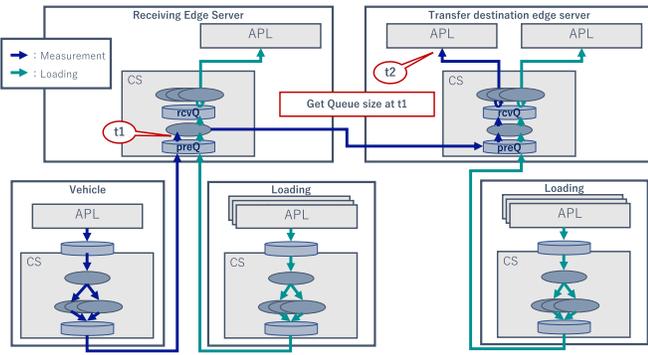


Fig. 13. Data flow of edge computing.

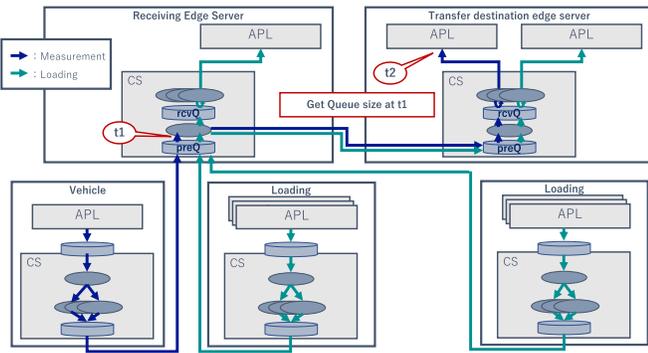


Fig. 14. Data flow of lane section ID transfer.

TABLE I
CLOUD AND EDGE SERVER CONFIGURATION

OS	Ubuntu16.04
CPU	8 Core 16 Thread (3.60GHz)
Memory	16GB
SSD	256GB
Communication Method	Wired(up to 1 Gbps)

TABLE II
CLOUD AND EDGE SERVER BENCHMARK

Server	Benchmark
Cloud	22950
Edge Server	16050

is stored in the queue in the destination edge server and processed by the application. Figure 14 shows the data flow of lane section ID transfer in the dynamic map system.

Table I shows the specifications of the PC used as the cloud and the edge server used for the evaluation. Since edge servers generally perform worse than the cloud, the central processing unit (CPU) was resource-limited [43] and the performance was adjusted. The cloud and edge servers were measured using the CPU benchmark [44], and the results are shown in Table II. Table III shows the specifications of the vehicle PCs to be measured and the parameters for sending the data. Table IV shows the specifications of the PC's for transmitting data with the number of vehicles under load and the parameters for transmitting the data. Although the resend and simultaneous transmission features can be used to improve

TABLE III

VEHICLE CONFIGURATION AND TRANSMISSION PARAMETERS

OS	Ubuntu16.04
CPU	4 Core 8 Thread (1.90GHz)
Memory	16GB
SSD	256GB
Communication Method	Wired(up to 1 Gbps)
Resend	No
Synchronous Transmission	No
Transmission Interval	1000 ms
Measurement Interval	1000 ms
Transmission Time	150s

TABLE IV

LOADING CONFIGURATION AND TRANSMISSION PARAMETERS

OS	Ubuntu16.04
CPU	8 Core 16 Thread (3.60GHz)
Memory	16GB
SSD	256GB
Communication Method	Wired(up to 1 Gbps)
Resend	No
Synchronous Transmission	No
Transmission Interval	100 ms
Measurement Interval	100 ms
Transmission Time	100s

the reliability of communication, we do not use them in this study because we discuss the coordination features between edge servers. Also, since this study investigates the load balancing function through cooperation between edge servers, we built the system using a more reliable wired connection. Although we have been experimenting with a dynamic map platform using wireless communication, we do not consider a dynamic map platform using wireless communication, because we propose cooperation between edge servers in this study. The data transmission interval of the vehicles was typically set at a 100 ms interval, as described in the introduction, and the simulation time was set at 100 s, which is sufficient for the vehicles to pass through the intersection. Also, the data transmitted by the vehicle is 1158 bytes, including payloads such as location and speed.

The vehicle to be measured transmits data every second for 150 seconds. The load vehicle transmits data for 100 seconds at 100 millisecond intervals while the measurement vehicle transmits data. We measure the processing delay time from the time (t1) when the data from the measurement vehicle is received to the time (t2) when the processing is completed by the application in the cloud and edge servers to be evaluated. We also measures the queue size that stores the data. We study the effectiveness of the proposed system by using the simulation environment we built.

VI. RESULTS

A. Scalability of Edge Server Allocation System Based on Lane Section ID

Figure 15 shows the maximum processing delay time when the number of load vehicles is changed. In the case of the

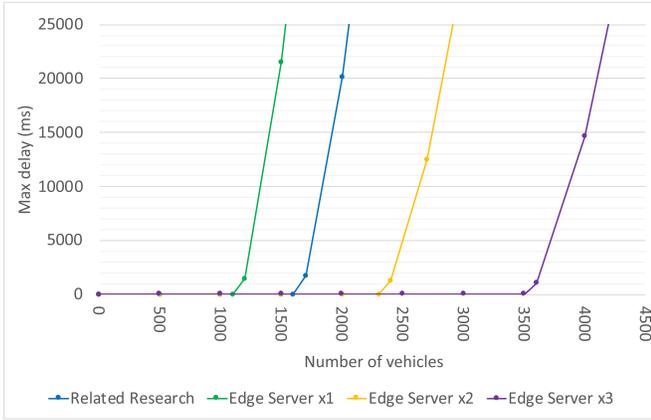


Fig. 15. Maximum delay time per vehicle.

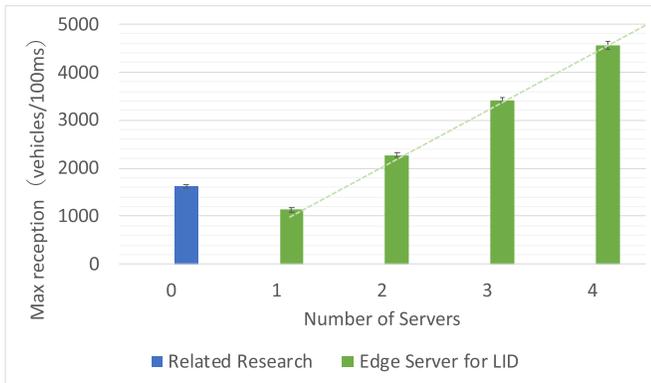


Fig. 16. Scalability by number of edge servers.

cloud-based dynamic map system, when communication was performed with fewer than 1600 vehicles, no problematic processing delay occurred. However, when communication was performed with more than 1600 vehicles, the queue size increased, and processing delay occurred. When communication was performed with 1700 and 2000 vehicles, the maximum processing delay times were about 1800 and 200,000 ms, respectively.

In the dynamic map system with edge servers, the number of edge servers was varied and evaluated. Processing delays did not occur up to about 1100 vehicles when there was one edge server, 2300 vehicles when there were two edge servers, and 3500 vehicles when there were three edge servers.

The scalability of the dynamic map system with the cloud and the dynamic map system with edge servers is shown in Figure 16. The maximum number of vehicles that can be processed in each system without problematic processing delays was evaluated. The cloud-based dynamic mapping system was able to process about 1600 vehicles without delay. Using edge servers, we were able to process about 1100 vehicles per edge server without delay and increase the number of vehicles that could be processed by increasing the number of servers. Since the lack of CPU power causes processing delays when the number of vehicles is increased, a dynamic map system with five or more edge servers can be scaled by increasing the number of edge servers.

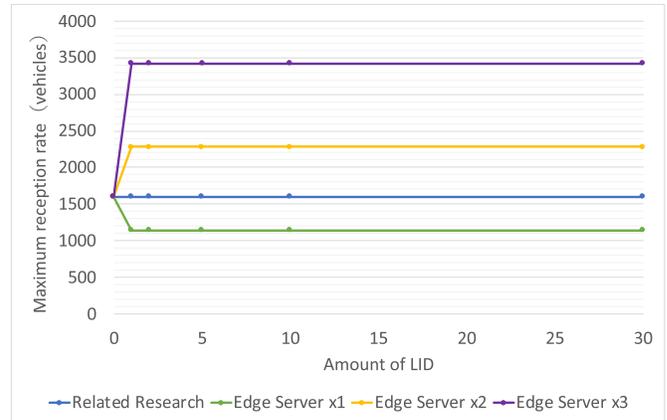


Fig. 17. Impact of lane section ID on scalability.

B. Impact of Lane Section ID on Scalability

We showed in Section VI-A that building a dynamic map system using edge servers can improve scalability. When building a dynamic map system using edge servers, the data sent by the vehicle must be processed by the edge server corresponding to the vehicle location. A vehicle can basically travel anywhere the roads are in good condition. Therefore, processing delays will be caused by requiring the vehicle to maintain the relationship between the location information and the corresponding edge server, and referring to it every time the vehicle sends data at 100 ms intervals. In addition, the response table can be changed dynamically so that the system can respond flexibly to accidents and traffic jams. Therefore, this study constructs a system that sends data sent from the vehicle to the corresponding edge server by using the lane section ID. In other words, in a dynamic map system that does not use lane section IDs, as in the related study [25], the edge server does not know whether the received data was sent by vehicles in its area of jurisdiction. Therefore, the edge server needs to query the cloud that has overall jurisdiction.

Figure 17 shows the maximum number of vehicles that can be processed by each system without delay, varying the number of lane section IDs that divide the area under the jurisdiction of each edge. If the lane section ID is zero, the data sent by the vehicle is received by the edge server at the nearest mobile phone base station. In that case, as explained in Section III-B, the data may need to be received by a different edge server from that at the nearest mobile phone base station to take into account the merging mediation and signal conditions. The data is then aggregated in the cloud, where it is processed. Therefore, even if the number of edge servers is increased, processing delays occur in the cloud when the number of vehicles increases, and the limit is about 1600 vehicles. However, if we build a dynamic map system with assigned lane sections, it can scale with the number of edge servers, as described in Section VI-A. It is also shown that the scalability is constant even when the number of lane section IDs under the jurisdiction of each edge server is different because the lane section IDs can be set to any length of the road and be changed. Although there is a concern that

increasing the number of lane section IDs will increase the communication overhead, the number of vehicles that can be processed by increasing the number of sections per edge server to 10,000 did not decrease. Since the lane section relates the vehicles to the edge servers and achieves cooperation between the edge servers, it is not a problem if an edge server can cover about 10,000 sections.

With one edge server, only about 1,600 vehicles could be processed without delay when the lane section ID was used, whereas only about 1,100 vehicles could be processed without delay when the lane section ID was used. This is because in the absence of management by lane section IDs, the transmitted data would be aggregated and processed in the cloud, which would effectively be the scalability of the cloud.

VII. EFFICIENCY OF THE LOAD BALANCING FUNCTION BY THE LANE SECTION ID TRANSFER FUNCTION

In this study, we used the lane section ID to map vehicles to edge servers. When an edge server is used to build a dynamic mapping system, the data may be needed by a different edge server than the one that received the data from the vehicle, as described in Section III-B. It is also assumed that edge servers may be assigned to locations other than base stations. In such a case, we evaluate the effectiveness of load balancing by the transfer function to transfer data between edge servers on the basis of the lane section ID.

In this evaluation, we construct a dynamic map system with two edge servers (Edge Servers A and B). As described in Section VI-A, we can process about 1600 vehicles per edge server without delay. Therefore, if each edge server processes 1600 vehicles, the scalability of the dynamic map system is about 3200 vehicles. However, suppose that out of the 1600 vehicles that should be received by Edge Server B due to the signal conditions, 100 of them send data to Edge Server A. At that time, Edge Server A can determine that the received data is to be forwarded to Edge Server B because we use the lane section ID to associate the vehicle with the edge server in this study. However, if the edge server judges whether data is to be transferred or not by an application that implements running arbitration and so on, even if Edge Server B has enough room to process the data, processing delays occur on Edge Server A and transfer to Edge Server B will also be delayed. Therefore, as described in Section V, the edge server sends only its own necessary data to the application after determining whether the received data is to be transferred or not at the communication unit. By doing so, we evaluate the impact of sending and receiving data at the edge server and application processing independently.

A. Edge Server A: 1600 Vehicles, Edge Server B: 1600 Vehicles

The results of the evaluation for sending data from 1600 vehicles each to both edge servers are shown in Figure 18. The horizontal axis represents the time, and the range between the green vertical lines represents 100 seconds when the data was sent from the load vehicle out of the



Fig. 18. Latency for lane section ID transfer (edge server A: 1600 vehicles, edge server B: 1600 Vehicles).



Fig. 19. Latency for lane section ID transfer (edge server A: 1700 vehicles, edge server B: 1600 Vehicles).

150 seconds that the data was sent from the vehicle. The vertical axis on the left represents the delay time, which is the difference between the time the edge server received the data sent by the vehicle and the time the processing was completed at Edge Server B. The vertical axis on the right side represents the queue size value, and as described in Section V, the edge server stores the received data once in the preQ and then sends it to the rcvQ of the target edge server after determining whether it is to be transferred. Then the application processing is performed in order. In other words, an increase in preQ results in communication delays, and an increase in rcvQ results in processing delays. The results did not show a significant increase in queue size or processing delay for either edge server.

B. Edge Server A: 1700 Vehicles, Edge Server B: 1600 Vehicles

Figure 19 shows the evaluation results for sending data from 1700 and 1600 vehicles to Edge Servers A and B, respectively. The rcvQ of Edge Server A increased significantly, causing processing delays in Edge Server A. This is because the application processing on Edge Server A could not keep up with the data stored in rcvQ.

C. Edge Server A: 1600 Vehicles, Edge Server B: 1700 Vehicles

As in Section VII-B, the data of 1700 and 1600 vehicles was sent to Edge Servers A and B, respectively. At this time,



Fig. 20. Latency for lane section ID transfer (edge server A: 1600 vehicles, edge server B: 1700 vehicles (100 of them were transferred from edge server A)).

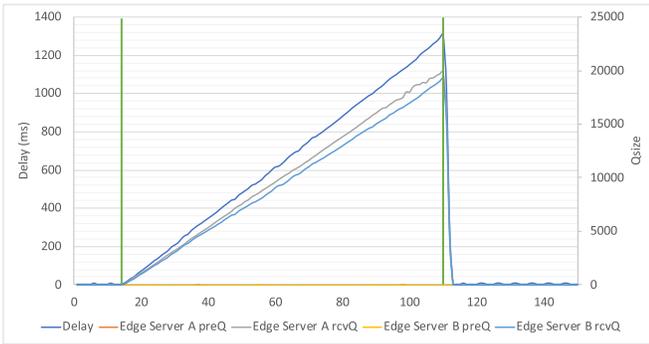


Fig. 21. Latency for lane section ID transfer (edge server A: 1700 vehicles, edge server B: 1700 vehicles (100 of them were transferred from edge server A)).

100 of the 1,700 vehicles sent to Edge Server A were targeted for transmission to Edge Server B. The results of the evaluation are shown in Figure 20. The rcvQ of Edge Server B was increased significantly, and the maximum processing delay of 1300 ms occurred on Edge Server B. This result shows that even if Edge Server A receives vehicle data that exceeds the processing limit, it can be scaled if the number of vehicles under its jurisdiction is not exceeded.

D. Edge Server A: 1700 Vehicles, Edge Server B: 1700 Vehicles

Figure 21 shows the results of the evaluation of sending data from 1800 and 1600 vehicles to Edge Servers A and B, respectively. At this time, 100 of the 1800 vehicles received by Edge Server A are targeted for transmission to Edge Server B, and only about 1700 vehicles are actually processed by each edge server. The number of rcvQ on each edge server increased significantly, causing processing delays on each edge server. In addition, the queues are stored similarly at each edge server, indicating that the transfer function by lane section IDs allows for distributed processing.

VIII. DISCUSSION

In the case of the dynamic map system using a single cloud, when there were up to about 1600 vehicles, communication and processing could be performed without any processing

delay that would be a problem. However, when there were more than about 1700 vehicles, a large processing delay occurred. The cause of the large processing delay is that the data received by the cloud cannot be processed by the application, and the data is stored in the queue and accumulated.

We show that in the case of edge server assignment by lane section ID, about 1100 vehicles can be covered per edge server and can be scaled by increasing the number of edge servers. In this study, we performed simulation evaluations of up to four edge servers, but in practice, many edge servers are expected to be deployed and to be able to handle a larger number of vehicles.

In addition, dynamic mapping systems such as those in the related study [25] do not have a way to relate vehicles to edge servers, such as lane section IDs, so all vehicle data must be aggregated to the cloud, for example, when there is a mobile phone base station's reception range boundary around an intersection. In this case, scalability is a concern. Therefore, in this study, we use the lane section ID to transfer data between edge servers to achieve the load balancing function.

In the evaluation of the effectiveness of the load balancing function through the lane section ID transfer function, regardless of the number of vehicles sent to Edge Server A, if about 1600 vehicles were sent to Edge Server B, the measured vehicles would be a problem. The communication and processing could be done without such a processing delay. However, if more than 1700 vehicles were sent to Edge Server A, significant processing delays were experienced. In this system, even though Edge Server B received data from about 3200 vehicles, it was able to process this data with low latency by using the lane section ID transfer function. In other words, even if a large amount of vehicle data is received by a particular edge server due to radio frequency conditions or other factors, there is no processing delay because the lane section ID transfer function transfers the vehicle data to each edge server and the data is processed by the edge server that received it. Therefore, the lane section ID transfer feature is enabled.

This study evaluates the scalability of the dynamic map system compared with a cloud-based dynamic map system without lane section IDs and shows the effectiveness of the dynamic map system with assigned edge servers. In addition, as shown in a related study [25], a dynamic map system that does not use lane section IDs does not enable coordination between edge servers and may consolidate data in the cloud. Therefore, in this study, we constructed a distributed processing function by transferring data between edge servers using lane section IDs. The effectiveness of this feature was demonstrated by investigating the processing delays and their causes on each edge server by varying the number of vehicles by using two edge servers.

In the dynamic map system for realizing safe driving support and automatic driving, the vehicle and the edge server need to communicate and process data with low delay. Also, since the amount of data communicated between the vehicle and the server will increase in the future, the larger the allowable number of vehicles, the better. The efficiency of the proposed system was improved significantly compared with the cloud-based dynamic map system as there were twice

as many vehicles that did not cause problematic processing delays as in the case of the dynamic map system with multiple edge servers. In addition, the amount of data that each edge server bears is smaller than that of a cloud-based dynamic map system, and communication reliability is improved. In the dynamic map system where edge servers are assigned on the basis of the lane section of the vehicle, communication can be performed with lower delay than in the dynamic map system using the cloud, and the dynamic map system using the cloud is improved in terms of smooth traffic flow.

In addition, the frequency of communication is important to grasp the real-time position information of a vehicle using communication. However, if the communication frequency is low, the error of the position information increases, and the effect on the application that uses the real-time position information is large. Therefore, evaluation of transmission interval control that optimizes the transmission interval of each vehicle by comparing the position information in the data transmitted by each vehicle to the edge with the dynamic map system will be a future topic. In addition, since there are various applications running on the edge server, such as driving support and merging and mediation, and each application requires different data, we would like to study the priority processing function as a future prospect.

IX. CONCLUSION

In recent years, research has begun on cooperative autonomous driving aiming at safety and efficiency by sharing information of sensors mounted on autonomous vehicles via wireless communication. Furthermore, a dynamic map system, which is an information communication platform for managing shared sensor information and executing applications, is being studied. However, there is a concern about scalability in the dynamic map system, which operates on the cloud on the Internet, when the number of vehicles that transmit and receive sensor information increases. Therefore, it is considered that applications can be executed efficiently by distributing multiple edge servers geographically and managing information managed in the cloud by the edge servers. However, depending on the actual radio wave conditions, the edge server that controls the vehicle information and the edge server that receives the data do not always match. Therefore, some applications are difficult to manage, such as intersection collision danger warning and merging arbitration. We divided the area on the road where the vehicle travels as “lane section ID” and assigned an edge server to each lane section ID on the basis of that. Furthermore, we implemented a dynamic map system that links vehicles and edge servers by linking multiple edge servers.

Also, for the vehicle and the edge server to communicate efficiently, the vehicle used anycast for communication to the edge server, and the edge server used multicast for communication to the vehicle. We evaluated the scalability of the processing delay on the basis of the number of vehicles to be managed and were able to process more vehicles with lower latency than a cloud-based dynamic map system. In addition, the lane section ID transfer feature allows data to be transferred between edge servers to distribute processing,

even if a large amount of data is received at a particular edge server due to radio conditions. The scalability was improved because of the ability to process vehicles with low delay, and the effectiveness of the load distribution mechanism using lane section IDs was demonstrated.

ACKNOWLEDGMENT

The authors would like to thank members of Dynamic Map 2.0 Consortium for advice.

REFERENCES

- [1] A. Geiger, P. Lenz, and R. Urtasun, “Are we ready for autonomous driving? The KITTI vision benchmark suite,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Providence, RI, USA, Jun. 2012, pp. 3354–3361.
- [2] J. Baber, J. Kolodko, T. Noel, M. Parent, and L. Vlacic, “Cooperative autonomous driving: Intelligent vehicles sharing city roads,” *IEEE Robot. Autom. Mag.*, vol. 12, no. 1, pp. 44–49, Mar. 2005.
- [3] L. Hobert, A. Festag, I. Llatser, L. Altomare, F. Visintainer, and A. Kovacs, “Enhancements of V2X communication in support of cooperative autonomous driving,” *IEEE Commun. Mag.*, vol. 53, no. 12, pp. 64–70, Dec. 2015.
- [4] J. Zhang, F.-Y. Wang, K. Wang, W.-H. Lin, X. Xu, and C. Chen, “Data-driven intelligent transportation systems: A survey,” *IEEE Trans. Intell. Transp. Syst.*, vol. 12, no. 4, pp. 1624–1639, Dec. 2011.
- [5] S. Bowles and H. Gintis, *A Cooperative Species: Human Reciprocity and Its Evolution*. Princeton, NJ, USA: Princeton Univ. Press, 2013.
- [6] *Intelligent Transport System (ITS); V2X Communications; Multimedia Content Dissemination (MCD) Basic Service Specification; Release 2*, document TS 103 152, V2.1.1, ETSI, 2019.
- [7] G. Dimitrakopoulos and P. Demestichas, “Intelligent transportation systems,” *IEEE Veh. Technol. Mag.*, vol. 5, no. 1, pp. 77–84, Mar. 2010.
- [8] J. Lee and B. Park, “Development and evaluation of a cooperative vehicle intersection control algorithm under the connected vehicles environment,” *IEEE Trans. Intell. Transp. Syst.*, vol. 13, no. 1, pp. 81–90, Mar. 2012.
- [9] J. J. Leonard, H. F. Durrant-Whyte, and I. J. Cox, “Dynamic map building for an autonomous mobile robot,” *Int. J. Robot. Res.*, vol. 11, no. 4, pp. 286–298, Aug. 1992.
- [10] H. Shimada, A. Yamaguchi, H. Takada, and K. Sato, “Implementation and evaluation of local dynamic map in safety driving systems,” *J. Transp. Technol.*, vol. 5, no. 2, pp. 103–112, 2015.
- [11] N. Varga, L. Bokor, and H.-J. Fischer, “LDM-based dynamic network discovery and selection for IPv6 mobility management optimization in C-ITS environments,” in *Proc. Int. Conf. Models Technol. Intell. Transp. Syst. (MT-ITS)*, Budapest, Hungary, Jun. 2015, pp. 483–490.
- [12] *Dynamic Map 2.0 Consortium*. Accessed: Mar. 23, 2021. [Online]. Available: <http://www.nces.i.nagoya-u.ac.jp/dm2/>
- [13] K. Sato, Y. Watanabe, and H. Takada, “Dynamic map as common application platform for dynamic geographic information management,” *J. Inst. Electron., Inf. Commun. Eng.*, vol. 101, no. 1, pp. 85–90, 2018.
- [14] S. E. Shladover and T. M. Kuhn, “Traffic probe data processing for full-scale deployment of vehicle-infrastructure integration,” *Transp. Res. Rec., J. Transp. Res. Board*, vol. 2086, no. 1, pp. 115–123, Jan. 2008.
- [15] P. Jaworski, T. Edwards, J. Moore, and K. Burnham, “Cloud computing concept for intelligent transportation systems,” in *Proc. 14th Int. IEEE Conf. Intell. Transp. Syst. (ITSC)*, Washington, DC, USA, Oct. 2011, pp. 391–936.
- [16] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, “A survey on mobile edge computing: The communication perspective,” *IEEE Commun. Surveys Tuts.*, vol. 19, no. 4, pp. 2322–2358, 4th Quart., 2017.
- [17] D. Sabella, A. Vaillant, P. Kuure, U. Rauschenbach, and F. Giust, “Mobile-edge computing architecture: The role of MEC in the Internet of Things,” *IEEE Consum. Electron. Mag.*, vol. 5, no. 4, pp. 84–91, Oct. 2016.
- [18] K. Zhang, Y. Mao, S. Leng, Y. He, and Y. Zhang, “Mobile-edge computing for vehicular networks: A promising network paradigm with predictive off-loading,” *IEEE Veh. Technol. Mag.*, vol. 12, no. 2, pp. 36–44, Jun. 2017.
- [19] J. Zhang and K. B. Letaief, “Mobile edge intelligence and computing for the Internet of vehicles,” *Proc. IEEE*, vol. 108, no. 2, pp. 246–261, Feb. 2020.

- [20] Z. Pengyuan, Z. Wenxiao, B. Camille, H. Pan, and K. Jussi, "ARVE: Augmented reality applications in vehicle to edge networks," in *Proc. Workshop Mobile Edge Commun. (SIGCOMM)*, Aug. 2018, pp. 25–30.
- [21] R. Xie, Q. Tang, Q. Wang, X. Liu, F. R. Yu, and T. Huang, "Collaborative vehicular edge computing networks: Architecture design and research challenges," *IEEE Access*, vol. 7, pp. 178942–178952, Dec. 2019.
- [22] M. Li, P. Si, and Y. Zhang, "Delay-tolerant data traffic to software-defined vehicular networks with mobile edge computing in smart city," *IEEE Trans. Veh. Technol.*, vol. 67, no. 10, pp. 9073–9086, Oct. 2018.
- [23] J. Martin-Perez, L. Cominardi, C. J. Bernardos, A. de la Oliva, and A. Azcorra, "Modeling mobile edge computing deployments for low latency multimedia services," *IEEE Trans. Broadcast.*, vol. 65, no. 2, pp. 464–474, Jun. 2019.
- [24] E. Ahmed *et al.*, "Bringing computation closer toward the user network: Is edge computing the solution?" *IEEE Commun. Mag.*, vol. 55, no. 11, pp. 138–144, Nov. 2017.
- [25] *Multi-Access Edge Computing (MEC): Study on MEC Support for V2X Use Cases*, document GR MEC 022 V2.1.1, ETSI, 2018.
- [26] P. Corcoran and S. K. Datta, "Mobile-edge computing and the Internet of Things for consumers: Extending cloud computing and services to the edge of the network," *IEEE Consum. Electron. Mag.*, vol. 5, no. 4, pp. 73–74, Oct. 2016.
- [27] S. Husain, A. Kunz, A. Prasad, K. Samdanis, and J. Song, "An overview of standardization efforts for enabling vehicular-to-everything services," in *Proc. IEEE Conf. Standards Commun. Netw. (CSCN)*, Helsinki, Finland, Sep. 2017, pp. 109–114.
- [28] C. Nanthawichit, T. Nakatsuji, and H. Suzuki, "Application of probe-vehicle data for real-time traffic-state estimation and short-term travel-time prediction on a freeway," *Transp. Res. Rec., J. Transp. Res. Board*, vol. 1855, no. 1, pp. 49–59, Jan. 2003.
- [29] P. Misra and P. Enge, *Global Positioning System: Signals, Measurements and Performance*, 2nd ed. Ganga-Jamuna Press, 2006.
- [30] K. Takagi, K. Morikawa, T. Ogawa, and M. Saburi, "Road environment recognition using on-vehicle LIDAR," in *Proc. IEEE Intell. Vehicles Symp.*, Tokyo, Japan, Jun. 2006, pp. 120–125.
- [31] *Intelligent Transport System (ITS); Users and Applications Requirements; Part 2: Applications and Facilities Layer Common Data Dictionary*, document TS 102 894-2, V1.2.1, ETSI, 2014.
- [32] *Intelligent Transport System (ITS); Vehicular Communications; Basic Set of Applications; Part 2: Specification of Cooperative Awareness Basic Service*, document EN 302 637-2, V1.4.0, ETSI, 2018.
- [33] *Intelligent Transport System (ITS); Vehicular Communications; Basic Set of Applications; Part 3: Specifications of Decentralized Environmental Notification Basic Service*, document EN 302 637-3, V1.2.1, ETSI, 2014.
- [34] S. Katsuki, "Trends on ITS standardization activities in ISO/TC204," *JARI Res. J.*, no. 6909, pp. 1–4, Jan. 2017.
- [35] J. Knaup *et al.*, "How to support drivers in intersection-related scenarios—Functions of INTERSAFE-2 demonstrator vehicles," in *Proc. IEEE 5th Int. Conf. Intell. Commun. Process.*, Cluj-Napoca, Romania, Aug. 2009, pp. 387–392.
- [36] L. Hoehmann and A. Kummert, "Mobility support for wireless sensor networks simulations for road intersection safety applications," in *Proc. 52nd IEEE Int. Midwest Symp. Circuits Syst.*, Cancún, Mexico, Aug. 2009, pp. 260–263.
- [37] C. Oh, S. G. Ritchie, and S.-T. Jeng, "Anonymous vehicle reidentification using heterogeneous detection systems," *IEEE Trans. Intell. Transp. Syst.*, vol. 8, no. 3, pp. 460–469, Sep. 2007.
- [38] S. Kokalj-Filipovic, L. Greenstein, B. Cheng, and M. Gruteser, "Methods for extracting V2V propagation models from imperfect RSSI field data," in *Proc. IEEE 82nd Veh. Technol. Conf. (VTC-Fall)*, Boston, MA, USA, Sep. 2015, pp. 1–5.
- [39] M. Jerbi, P. Marlier, and S. M. Senouci, "Experimental assessment of V2V and I2V communications," in *Proc. IEEE Int. Conf. Mobile Adhoc Sensor Syst.*, Pisa, Italy, Oct. 2007, pp. 1–6.
- [40] K. Cao, A. Kinoshita, Y. Takaki, C. Ohta, and H. Tamaki, "Efficient urban broadcast protocol for V2V communications with relay control," in *Proc. IEEE Veh. Netw. Conf.*, Boston, MA, USA, Dec. 2013, pp. 24–30.
- [41] D. Johnson, *Reserved IPv6 Subnet Anycast Address*, document RFC 2526, 1999.
- [42] R. Hinden, *IPv6 Multicast Address Assignments*, document RFC 2375, 1998.
- [43] Paul Menage. (2004). *CGROUPS*. [Online]. Available: <https://www.kernel.org/doc/Documentation/cgroup-v1/cgroups.txt>
- [44] A. Kopytov, "Sysbench manual," MySQL AB, Cupertino, CA, USA, Tech. Rep., 2012. [Online]. Available: <https://mysql.com/wp-content/uploads/2014/10/sysbench-manual.pdf>



Kohei Hosono (Member, IEEE) received the degree from the Department of Information Systems Design, Faculty of Science and Engineering, Doshisha University in 2020. He is currently pursuing the master's degree in information engineering with the Graduate School of Science and Engineering, Doshisha University. He is also engaged in research on a communication platform for cooperative automated vehicles using edge computing.



Akihiko Maki (Member, IEEE) received the M.S. degree in electronic and information engineering from the Tokyo University of Agriculture and Technology in 2002. He is currently working with the Software and Service Business Division, Network Solutions Business Group, Fujitsu Ltd.



Yosuke Watanabe (Member, IEEE) received the M.E. and Dr.Eng. degrees from the University of Tsukuba in 2003 and 2006, respectively. In 2014, he joined the 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 currently a member of the Database Society of Japan, IEICE, and ACM.



Hiroaki Takada (Member, IEEE) received the Ph.D. degree in information science from the University of Tokyo in 1996. From 1989 to 1997, he was a Research Associate with the University of Tokyo. From 1997 to 2003, he was a Lecturer and then an Associate Professor with the Toyohashi University of Technology. He is currently a Professor with the 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), Graduate School of Informatics, Nagoya University. His research interests include real-time operating systems, real-time scheduling theory, embedded system design, and automotive embedded systems. He is also a fellow of IPSJ and JSSST. He is also a member of ACM, IEICE, and JSAE.



Kenya Sato (Member, IEEE) received the B.E. and M.E. degrees from Osaka University in 1984 and 1986, respectively, and the Ph.D. degree from the Nara Institute of Science and Technology in 2000. He joined Sumitomo Electric Industries in 1986. He was a Visiting Researcher with the Department of Computer Science, Stanford University from 1991 to 1994. After working as a Chief Technologist at AMI-C Inc., Michigan, USA, he joined Doshisha University in 2004. He is currently a Professor with Doshisha University, Kyoto, Japan. He also leads the Mobility Research Center, Doshisha University, and serves as a Designated Professor for Nagoya University. He is also a member of ACM, SAE, and the Information Processing Society of Japan.