ISSN 1816-6075 (Print), 1818-0523 (Online)
Journal of System and Management Sciences
Vol. 10 (2020) No. 3, pp. 58-72
DOI:10.33168/JSMS.2020.0305

Designing and Implementing a Reproducible Log System Using Basic Safety Message in a V2V Environment

Sangkon Han [1] and Jung-In Choi [2]

[1] Department of Computer Science Engineering, Pusan National University, Busan, Republic of Korea

2 Software Education Center, Pusan National University, Busan, Republic of Korea sangkon@pusan.ac.kr; jungin.choi@pusan.ac.kr

Abstract. In autonomous driving, vehicles can share information that cannot otherwise be detected via sensors or is not included in the driver's perspective. Vehicle-to-everything(V2X) has several advantages in enhancing the stability of autonomous driving; however, it is not known why artificial intelligence makes specific decisions. Therefore, when an accident occurs in an autonomous driving environment, it may be difficult to determine the cause of the accident. This study proposes a method to reproduce a specific state by using a log system optimized for the V2V component of the V2X technology. The proposed method applies a command and query responsibility system pattern to the existing log system to safely store and utilize basic safety messages used in the V2V.

Keywords: log system, vehicle- to-vehicle (V2V), command and query responsibility segregation pattern, CQRS

1. Introduction

The five key technologies required for autonomous driving are cameras, radar/light-detection-and-ranging (LiDAR), sensors, computing system, and V2X technology (ITF, 2015). Cameras, radar/LiDAR, and sensors are responsible for detecting objects. Computing technology evaluates the situation and controls the system based on the detected objects. V2X communicates with other vehicles and the vehicle infrastructure and is responsible for exchanging or sharing information on vehicles and traffic conditions. With the use of the V2X technology, objects or situations outside the visual range of cameras, radar/LiDAR, and sensors can also be detected (Chen & Englund, 2016). Owing to these reasons, the demand for V2X is increasing.

V2X has several advantages in enhancing the safety of autonomous driving. Autonomous vehicles undergo the following processes: object or situation recognition; evaluation of the detected objects or situations; and appropriate system control based the evaluations. The cognitive steps include recognizing the surrounding environment using environmental recognition sensors such as cameras and radar/LiDAR. However, the most important task is to accurately detect the factors that may affect driving stability. There are situations where objects in the vicinity of the vehicle, such as on the side or rear, are not accurately detected owing to the physical limitations of the radar/LiDAR and sensors (Lee et al., 2020). Consequently, there is an insufficiency of data for the computing system to evaluate and determine the precise response to control the system (Jayaweera et al., 2019). V2X technology can be used to address this issue by supplementing the data obtained by the physical sensors (Cha et al., 2018; Kim et al., 2013), through the utilization of situational information obtained via other methods. Currently, 5G and other technologies are being actively investigated to enable faster communication (Kim et al., 2013).

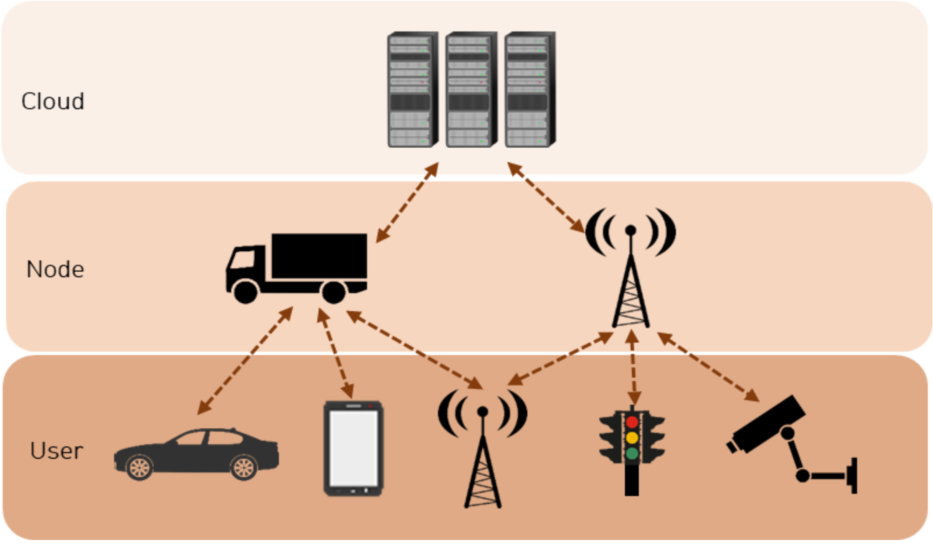


Fig. 1: Each Layer of the V2X communication.

V2X includes the following communication protocols: vehicle-to-vehicle (V2V), vehicle-to-pedestrian (V2P), vehicle to the roadside unit (RSU), vehicle-to-network (V2N), and vehicle-to-infrastructure (V2I). [Figure 1] depicts the entities that comprise each layer of the V2X communication service. It is expected that various layers such as vehicle-to-cloud (V2C) and vehicle-to-device (V2D) will be added as the traffic increases over time. Therefore, to better prepare for the future scenario when autonomous vehicle movement is expected to drastically increase, the stability of communication at high speeds is paramount, as the safety of the driver and passengers will be directly related to the safety of communication via V2X.

Due to the complexities of an autonomous driving environment, tracing the events that lead to an accident can be challenging (Codevilla et al., 2019). As sensors detect objects and the system recognizes them and controls the responses, the vehicle is in constant communication with the entities participating in the V2X environment. In addition, because state information received from individual objects in the autonomous driving system is recorded in real time, there may be too much data to analyze for a simple inference of the cause of an accident (Codevilla et al., 2019). However, determining the cause of an accident may be possible by reproducing the events that ultimately lead to it by using the source information of the relevant objects. The key information in such cases is therefore the source information of all the individual devices involved with the incident and the communication messages generated by them for mutual communication in a V2X environment.

In this study, we propose a stable and reproducible system by utilizing the basic safety message (BSM), a communication message standard used in V2V.

2. Related Work

2.1. Cooperative Intelligent Transportation System(C-ITS)

C-ITS is an open platform-based next-generation Intelligent Transportation System (ITS) that exchanges and shares traffic information through a two-way communication channel, such as V2V and V2I, to promote safety and environmental benefits.

V2V technology can be used in C-ITSs to reduce vehicular collisions by 80%, which in turn is expected to lead to a reduction in traffic congestion and increased road capacity.

Traditional ITS is a one-way information management center in the transportation infrastructure. C-ITS, on the contrary, is a safety center that continuously shares data between the vehicles, as well as between the vehicles and infrastructure, to quickly and proactively respond to and prevent emergencies.

[Figure 2] is a schematic of the C-ITS. [Figure 1] illustrates the two-way communication, thereby enabling real-time data transfer to and from all the elements around the vehicle. This not only increases safety by preventing traffic accidents, but also reduces carbon dioxide and improves fuel efficiency through cluster driving.

C-ITS is also known as the Internet-of-things (IoT) in the field of vehicles and transportation in that it delivers a variety of services in conjunction with all areas of connectivity, including V2P, V2N, V2V and V2I.

2.2. Vehicle-to-Vehicle(V2V)

Safety systems in inter-vehicle communication periodically send and receive BSMs between vehicles via Dedicated Short-range Radio Communication (DSRC). BSM includes data elements such as speed, direction, status, and the current coordinates of the vehicle.

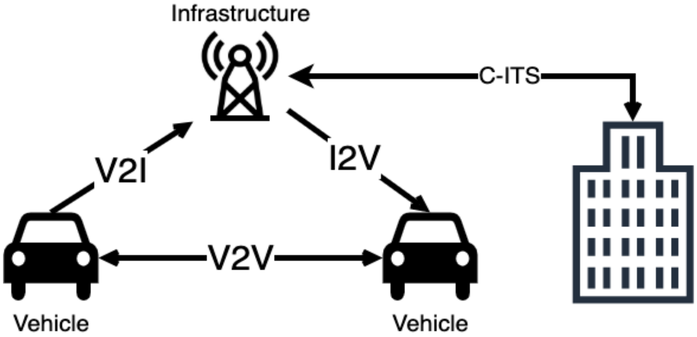


Fig. 2: A Cooperative Intelligent Transportation system (C-ITS).

Although V2V systems provide numerous advantages, there are two essential requirements to enable these functions. First, for vehicles manufactured by different manufacturers to transmit and receive signals in a uniform manner, all vehicles must use the same communication method. Therefore, a standard of communication technology is essential. The National Highway Traffic Safety Administration (NHTSA) defined Wireless Access in Vehicle Environment (WAVE) in IEEE, and J2735 and J2945 were defined by the Society of Automotive Engineers (SAE) as basic V2V communication standards. WAVE defined the DSRC communication protocol for use in V2V systems, while SAE J2735 and SAE J2945 were defined as the data elements and system requirements for use in V2V systems. Second, predicting collision possibilities based on V2V communication requires that signals to be received from 'all' surrounding vehicles. Due to the nature of V2V-based safety systems that manually detect the situation through signal reception, vehicles that do not transmit signals cannot predict the possibility of collision. This is because actual sensors or vehicles that can only be recognized by the driver are unaware of the vehicles in the V2V system. For this reason, NHTSA is demanding the legislation of Federal Motor Vehicle Safety Standard (FMVSS) No. 150 to enable all vehicle or equipment manufacturers to implement V2V features in the future. In addition, a procedure is required to verify the presence of a V2V OBU in the vehicle, before it can be made mandatory to install such V2V systems.

2.3. Basic Safety Message (BSM)

A BSM is a communication message used by V2V. This message contains the driving status information of the vehicle's GPS coordinates, direction of movement, and speed. V2V sends the BSMs to the surrounding cars and receives the ones sent by the vehicles. These messages are then used to determine the location and direction of movement of the vehicles, to send impulses in blind spots at intersections, as well as of collisions that may occur during lane changes and overtaking. Therefore, BSM serves a V2V safety service. The SAE J2735 standard BSM contains the position and motion information of the vehicle and is normally broadcast to the surrounding vehicles at intervals of 100ms.

The structure of the BSM can be divided into Part I and Part II. Part I data is recommended to be included in all messages and has 14 data elements and set rules. The data elements in Part I utilized in this paper are latitude, longitude, elevation, transmission, and speed, which are used to check the condition of the vehicle at the time the message was recorded. Part II data is optional and included as needed. [Table 1] details the structure of the 14 data elements included in the BSM Part I data as well as the three data elements included in the BSM Part II data.

Table.1: Basic Safety Message Structure

Section	Element	Size
PART 1	Message Count	1 byte
	Temporary ID	4 byte
	Dsecond	2 byte
	Latitude	4 byte
	Longitude	4 byte
	Elevation	2 byte
	Positional Accuracy	4 byte
	Transmission & Speed	2 byte
	Heading	2 byte
	Steering Wheel Angle	1 byte
	Accelerations	7 byte
	Brake System Status	2 byte
	Vehicle Size	3 byte
PART 2	Vehicle Safety Status	Optional
	Special Vehicle Extensions	
	Supplemental Vehicle Extensions	

2.4. System Log

Logging refers to the recording of tracking information of all movements or events generated by operating systems, software, and various services. The log acts similarly to a black box that records the flight history of an aircraft. For example, it provides information to determine what work the intruder in the system did or through which route it was accessed. Furthermore, security managers are based on these logs and combine various information to detect system anomalies.

A separate log is generated for various applications or services used in individual systems. In addition to the operating systems(OS), most of these applications run on databases, web servers, and so on, and write separate logs; these logs are usually recorded in a chronological order.

The reasons for recording a log are as follows: first, for analyzing system errors; second, for tracing accidents, because the time when the system error occurred can be identified as the status of the system is recorded chronologically; third, for preventing errors in the application or system.

2.5. Vehicle accident prevention research using V2X

Research related to vehicle safety proceeds in three main areas. The first research area is to focus on driver safety. Johnson et al. studied the safety of drivers using sensors attached to smartphones. The Kashevnik et al. study uses a camera attached to a vehicle to inform the driver of safety-related equipment. Secondly, there are studies that focus on vehicle safety by using a network such as V2X. In the Xin et al. study, V2X technology was applied to reduce forward collisions. A study by Khaliq et al. suggested that a vehicle accident can be transmitted using V2X technology to enable quick accident response when a vehicle accident occurs. The third is a study to investigate vehicle accidents and prevent similar accidents among studies related to vehicle accidents. These studies reproduce accidents and judge the situation at the time of the accident. A representative study is the study of Martinez et al. , which proposed a study on a method that can be reproduced using graphics when an accident occurs in a vehicle. Jeong et al. suggests a vehicle monitoring techniques. This paper is one of the third research areas and proposes a method to reproduce the vehicle state in real time using V2X technology.

3. Problems and Solutions in Applying Log to V2V

3.1. Disadvantages of Utilizing Existing Logs in V2V

Several shortcomings exist in the existing logs in V2V must be addressed using the proposed log system to enhance the driving and driver safety. The first disadvantage of traditional log systems in a V2V environment is that the format of the log is inconsistent. The V2V protocol has fixed message specifications and standardized communication methods for communication(Kim et al., 2017). However, the specifications of the log generated by the application or system are not fixed (Xu et al., 2019). This is because the log in which an individual application or service is created differs in the data covered by that application or service and in the items that they consider important (Adamov. 2014).

Second, most logs are stored in a file. Thus, systems with large amounts of communication messages, such as V2V, negatively impact the entire system because of the larger file sizes of the logs. This is because, if the log file is large, a significant amount of memory and I/O are needed to process the file, and the resources may be insufficient to process BSM messages in real time in V2V. In some cases, a large amount of logs is loaded into the internal system I/O, by which the entire system is affected. In some cases, a large amount of log gets loaded into the internal system I/O, thereby resulting in system not functioning properly (Ousterhout and Douglis, 1989).

Furthermore, the log stores the status value, not the time when it occurred 0. The log therefore does not record the cause underlying the change in the status value. To infer the cause using the log, the entire data must be examined in a chronological order. However, in V2V, the data logs transmitted from external vehicles must also be examined. It is therefore difficult to investigate using the existing log system in V2V because both the message and relevance of other vehicles must be considered (Kobayashi et al., 2017).

3.2. Command and Query Responsibility Segregation (CQRS)

To compensate for the aforementioned shortcomings, it is necessary to redefine the logs such that they can be utilized by the corresponding log system and to ensure efficient storage of large amounts of data, and at the same time, extract only those data that are needed at a particular point. To compensate for the existing shortcomings, the proposed log system utilizes the CQRS (Command and Query Responsibility Segregation) architecture pattern.

CQRS is the separation of write (Command) and read (Query) operations from the system. In CQRS, a command refers to a change of state, and a query refers to an operation that returns a state. In other words, CQRS is a pattern that separates writing from reading in the system. [Figure 3] is a simple representation of the data flow of an application with a CQRS pattern, in which the data are processed by separating the command and query from the application.

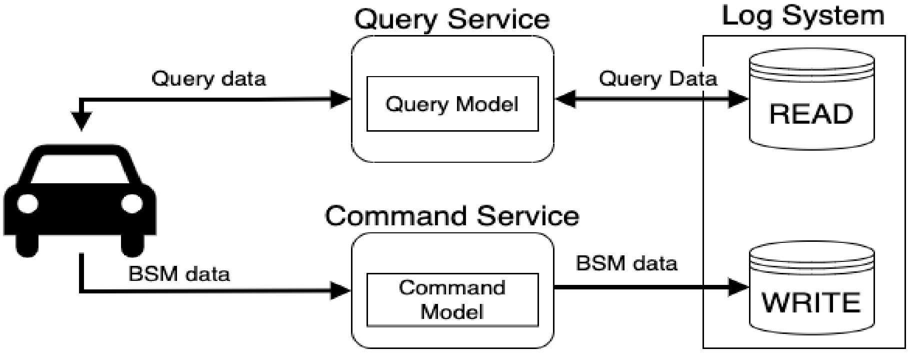


Fig. 2: Existing CQRS Pattern.

Monolithic architecture refers to a structure in which all the components are clustered on a single server. A monolithic architecture that is small in scale has a fast building and deployment time of applications and ease of maintenance. However, it is not agile enough to respond to state changes because it suffers from I/O performance degradation and has internally complex forms of dependencies when functionalities are added for storing and analyzing the data. The most critical issue is that errors in certain components can propagate throughout the system. The CQRS architecture pattern can compensate for the shortcomings of I/O performance degradation and functionality dependencies present in monolithic architectures (Esposito, 2015).

CQRS separates writing and reading, which can prevent I/O performance degradation compared to that in conventional systems. In addition, this separation allows us to add functionalities without any dependencies on the system (Esposito, 2015).

4. V2V-based Reproducible Log System

The system proposed in this study was designed to compensate for the shortcomings of the existing log system. It utilizes NoSQL for storage and BSMs using CQRS patterns. [Figure 4] presents the proposed log system based on V2V.

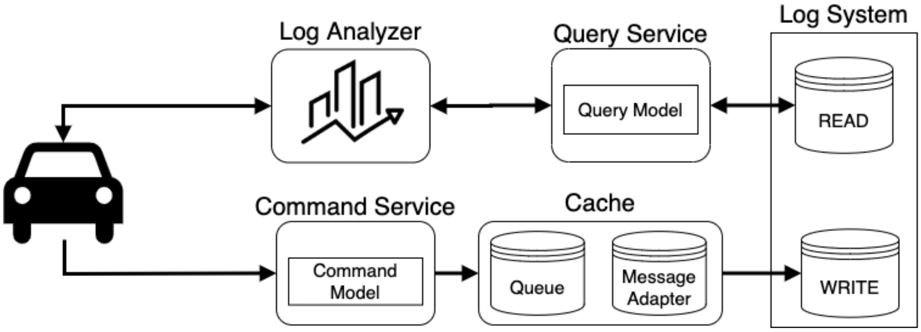


Fig. 3: Proposed BSM Log System.

One of the key features of our V2V-based reproducible log system is that it utilizes CQRS patterns, which has two functions: storage of BSMs in the log system, and the reading and analysis of the stored logs. Second, our system uses BSM transformers to distinguish the BSMs (from the current vehicle) from those transmitted from other vehicles. Therefore, BSMs stored in the log system can be stored separately, and useful information can be provided to the driver in real-time. In addition, BSMs received from other vehicles and those generated from the current vehicle are used to facilitate the process of analysis in cases of accident.

4.1. Log System Using CQRS

With the application of the CQRS, the log system can be operated in urban areas or the areas where the traffic volumes are significant. In addition, if the system is overloaded, the ability to record and analyze external and internal logs in real time can be limited (Han and Choi, 2020). Using CQRS, the challenges encountered when processing BSMs transmitted in real time are effectively addressed; the read and write operations are separately performed in CQRS, and as such, the log system can be operated in real time (Erb et al., 2018), for example, at 100ms from places with a considerable amount of traffic. Even when new functions are added, changes in the log system can be easily implemented as the read and write operations are separately performed in the CQRS architecture.

4.2. Queue-based Lightweight Cache

Most logs identify special events or problems at a particular point in time. Messages received from other vehicles in the V2V environment are identified, non-visible information is obtained and used to identify the current status of the messages. BSMs received from multiple vehicles in the V2V environment should be stored using the log system. Simultaneously, in a V2V environment, the BSMs also serves to communicate to other vehicles. That is, in a V2V environment, BSMs can be distinguished from being received from outside and transmitted from inside to the outside. In the case of the log system, the message must be stored separately for easy analysis in case of an accident.

The function of separating BSMs is thus essential. If the stored messages are also to be processed, a separate storage device is required, so that the reading performance is not affected. This requires the use of a write cache to store the messages in advance, and the BSM Adapter's processing may result in load-bearing, so the BSM can be used to maintain the read performance of the log system by utilizing a queue-based lightweight cache.

The log system is characteristically used to identify the message for future accident handling. However, the message should be analyzed in real time for the driver's safety and the user should be informed of the result. Therefore, the log system must provide a means to safely store BSMs, and the log system presented in this paper provides that function using a queue-based lightweight cache.

4.3. Analysis and Reproduction

The stored BSM can be used to provide safety information to the driver in real time (Kim et al., 2017). Blind-spot-warning (BSW), for example, helps the driver detect objects in the vehicle's blind spots. To detect BSW, the status can be inferred using the latitude, longitude, elevation, and heading items in Part I Data of BSM. Any essential information for the driver can be displayed on the vehicle's dashboard. Furthermore, sound and vehicle notification systems can be used to actively inform the driver of the information. This analysis uses a combination of the components of existing BSM data.

It should be possible to reproduce the driving based on the BSM stored in the log system. [Figure 5] presents the data of a BSM sample provided by the US

Department of Transportation, displayed on the map using Python's potassium. BSM data was processed using Python for reproduction. The data to be used in the process was changed to BSM format from the comma-separated-values (CSV) data provided by the U.S Department of Transportation (Johnson and Trivedi, 2011).

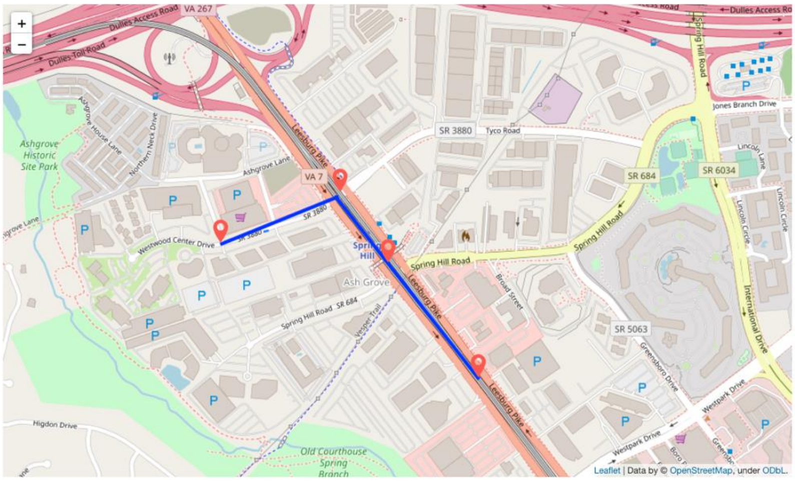
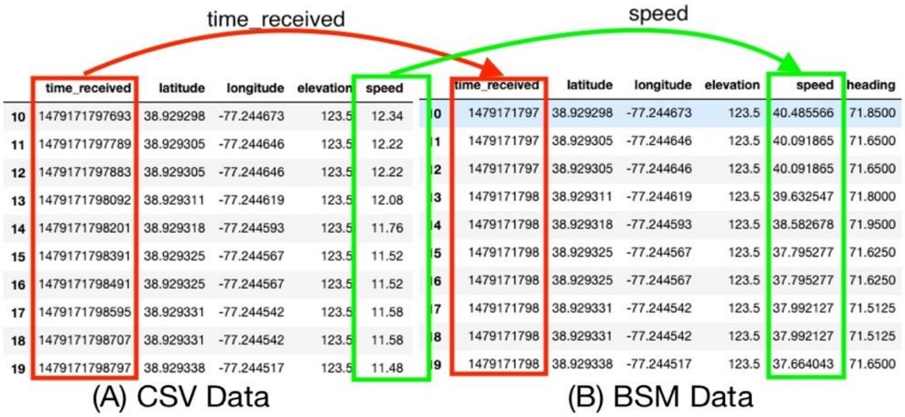


Fig. 4: Reproduced using BSM of U.S. Department of Transportation.

[Figure 6] is part of the change from the original data (A) to the BSM data structure (B) provided by the US Department of Transportation. The time reserved value of the original data was divided by 1000 and saved in seconds. The speed was changed from meters per second to feet per second. It should be changed from meter to foot, or from feet to meter. The data to be displayed on the map are paired with latitude and longitude.


time_received
speed

	time_received	latitude	longitude	elevation	speed
10	1479171797693	38.929298	-77.244673	123.5	12.34
11	1479171797789	38.929305	-77.244646	123.5	12.22
12	1479171797883	38.929305	-77.244646	123.5	12.22
13	1479171798092	38.929311	-77.244619	123.5	12.08
14	1479171798201	38.929318	-77.244593	123.5	11.76
15	1479171798391	38.929325	-77.244567	123.5	11.52
16	1479171798491	38.929325	-77.244567	123.5	11.52
17	1479171798595	38.929331	-77.244542	123.5	11.58
18	1479171798707	38.929331	-77.244542	123.5	11.58
19	1479171798797	38.929338	-77.244517	123.5	11.48

(A) CSV Data

	time_received	latitude	longitude	elevation	speed	heading
0	1479171797	38.929298	-77.244673	123.5	40.485566	71.8500
1	1479171797	38.929305	-77.244646	123.5	40.091865	71.6500
2	1479171797	38.929305	-77.244646	123.5	40.091865	71.6500
3	1479171798	38.929311	-77.244619	123.5	39.632547	71.8000
4	1479171798	38.929318	-77.244593	123.5	38.582678	71.9500
5	1479171798	38.929325	-77.244567	123.5	37.795277	71.6250
6	1479171798	38.929325	-77.244567	123.5	37.795277	71.6250
7	1479171798	38.929331	-77.244542	123.5	37.992127	71.5125
8	1479171798	38.929331	-77.244542	123.5	37.992127	71.5125
9	1479171798	38.929338	-77.244517	123.5	37.664043	71.6500

(B) BSM Data

Fig. 5: BSM conversion.

4.4. Performance analysis

Table 2 shows the experimental environment for the performance analysis of the log system. Performance test was conducted by implementing a log system using the CQRS pattern and a log system using a general DBMS. As a test tool, cProfile, a basic profiling tool provided by Python, was used. All tests were conducted in the same environment. The test system was run on Linux, and Python version 3.8.3 was used.

Table. 2: Test Environment

	Log system using CQRS	**General log system**
OS	Ubuntu 18.04.2 LTS	Ubuntu 18.04.2 LTS
CPU	Intel(R) Xeon(R) CPU E5-2640 v4 @ 2.40GHz (10C 20T) 2 vCPUs	Intel(R) Xeon(R) CPU E5-2640 v4 @ 2.40GHz (10C 20T) 2 vCPUs
Memory(GB)	80	80
DBMS	MongoDB v4.2.8	PostgreSQL v12.3

For the test data, BSMP1Summary[25] of Safety Pilot Model Deployment Data provided by the US Department of Transportation was used. The data consists of 14 attributes and 217,967 data.

4.5. Performance test results

The first performance test is the average response time of 10 times repeated reading and writing of data in each log system. The purpose of this performance test is to compare write and read performance. It measured the time to write and read data using a single thread. The log system using the CQRS pattern used MongoDB, and the log system using DBMS used PostgreSQL. Table 3 shows the result of single thread performance comparison. As a result of testing, the difference between the two log systems is around 2%. This level of difference is not a significant level of difference considering the driver characteristics of MongoDB and PostgreSQL used in Python, and it is an I/O error level that occurs when writing and reading 217,967 disks.

Table.3: Single thread performance comparison (10 averages)

	Average(m/s)	**Min(m/s)**	**Max(m/s)**	**Error(%)**
CQRS	3519	3423	3523	0
DBMS	3621	3511	3661	0

The second performance test was to read data using 40 threads at the same time while writing data to each system. This test measured the time that all threads read 217,967 data. As a result, the system applying the CQRS pattern has the advantage of being able to process 40 threads at the same time during the writing process because the writing and reading are separate. On the other hand, it can be seen that the log system using the DBMS is about 40% slower due to the lock occurring in the process of writing data and reading.

Table.4: Performance comparison of multiple threads (10 averages)

	Average(m/s)	**Min(m/s)**	**Max(m/s)**	**Error(%)**
CQRS	5521	5511	5712	0
DBMS	12822	12713	12882	0

In a general environment, the log system with CQRS or the system with DBMS does not show any difference in performance or safety. However, V2V or V2X environments are based on large systems. As a result of the second performance test, it can be expected that the slowdown occurs in the existing log system when reading and writing are performed simultaneously in a large system. The method of applying CQRS proposed in this study can process read and write operations separately in the process of storing and processing large-scale data. Therefore, this method is suitable for large-scale log systems such as V2X and V2V environments, considering the difference in performance or operation between NoSQL and DBMS.

5. Discussion

In V2V, BSM is used for communication. In V2X, various messages such as traveler information message (TIM), roadside alert (RSA), signal phase and timing (SPAT), and map intersection data (MAP) are exchanged. Messages used in V2V are used for inter-vehicle communication, but if the vehicle participates in C-ITS, the messages get stored in the log system along with each subtype of the message used in C-ITS or V2X as well as in the vehicle. Therefore, how communication messages can be stored and utilized in a particular specification or form needs to be further discussed.

In addition, the data used in BSMs should be communicated to the units that are primarily used in individual countries. As in the case of previous data, in the USA, the process of changing the existing speed to miles per hour is essential because the speed of the vehicle in the USA is measured in miles per hour. A considerable amount of the information regarding the vehicle includes not only the laws and regulations relating to the safety of the country to which the vehicle belongs, but also cultural practices. That is why much discussion is needed on how to utilize various communication messages as well as BSM.

Finally, if you want to analyze the data in the process and further apply machine learning, the computing power provided by the vehicle will also be a key issue. Thus, this paper presented a log system as a cornerstone for providing vehicle and driver safety by designing and processing various messages in the V2X environment.

6. Conclusive Remarks

This paper proposed a log system that can be applied to the V2V environment. The CQRS pattern was incorporated to compensate for the shortcomings of the existing log system. A method for stably storing and utilizing BSMs by separating writing and reading operations was proposed to enable the proposed system to reproduce data using the stored BSM.

Using the proposed log system, safety-related details can be communicated to the driver in real time. In addition, accident analysis can be performed using the log system for improving autonomous driving experience and analyzing driving-related accidents. In particular, when autonomous driving will become mainstream in the future, the log system can be used to analyze accidents and create a safer driving environment.

As future work, the log system discussed in this study will be further improved and developed into one that is suitable for application in the entire V2X environment.

7. Acknowledgements

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (2018R1D1A1B07042967)

References

Adamov, A. (2014). Data mining and analysis in depth. case study of Qafqaz University HTTP server log analysis. *2014 IEEE 8th International Conference on Application of Information and Communication Technologies* (AICT), 1-4.

Andrews, J. H. (1998). *Theory and practice of log file analysis.* Department of Computer Science, University of Western Ontario.

Cha, H.-J., Yang, H.-K., & Song, Y.-J. (2018). Cyber Physical System for Processing Sensing Data in Fog Computing. *World Journal of Wireless Devices and Engineerin*g , 7-12.

Chen, L., & Englund, C. (2016). Cooperative Intersection Management: A Survey, IEEE Trans. *Intelligent Transport System,* 571~572.

Codevilla, F., Santana, E., López, A. M., & Gaidon, A. (2019). Exploring the limitations of behavior cloning for autonomous driving. *Proceedings of the IEEE International Conference on Computer Vision*, 9329-9338.

Erb, B., Meißner, D., Ogger, F., & Kargl, F. (2018). Log Pruning in Distributed Event-sourced Systems. *Proceedings of the 12th ACM International Conference on Distributed and Event-based Systems,* 230-233.

Esposito, D. Cutting Edge (2015). Event Sourcing for the Common Application. *MSDN magazine* .

Han, S., & Choi, J.-I. (2020). Data Processing System Using CQRS Pattern and NoSQL in V2X Environment. *International Journal of Smart Device and Appliance* , 1-7.

ITF. (2015). Automated and Autonomous Driving: Regulation under Uncertainty. *International Transport Forum Policy Papers,* No. 7, OECD Publishing, Paris. https://doi.org/10.1787/5jlwvzdfk640-en.

Jayaweera, N., Rajatheva, N., & Latva-aho, M. (2019). Autonomous driving without a burden: View from outside with elevated lidar. *2019 IEEE 89th Vehicular Technology Conference (VTC2019-Spring),* 1-7.

Jeong, E., Oh, C., Kang, K., & Kang, Y. (2012). Development of traffic safety monitoring technique by detection and analysis of hazardous driving events in V2X environment. *The Journal of the Korea Institute of Intelligent Transport Systems,* 11(6), 1-14.

Johnson, D. A., & Trivedi, M. M. (2011). Driving style recognition using a smartphone as a sensor platform. *2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC)* , 1609-1615.

Kashevnik, A., Lashkov, I., Ponomarev, A., Teslya, N., & Gurtov, A. (2020). Cloud-Based Driver Monitoring System Using a Smartphone. *IEEE Sensors Journal,* 20(12), 6701-6715.

Khaliq, K. A., Chughtai, O., Shahwani, A., Qayyum, A., & Pannek, J. (2019). Road accidents detection, data collection and data analysis using V2X communication and edge/cloud computing. *Electronics,* 8(8), 896.

Kim, I., Park, J., Lee, E. Y., Lee, E. D., Shin, J., & Kim, D. (2017). Validation of Data Availability in V2V Basic Safety Message. *Journal of Auto-vehicle Safety Association*, 33-39.

Kim, J.-M., Jang, S.-B., & Baik, D.-K. (2017). An Efficient Data Management Scheme for Resolving Semantic Heterogeneity in 5G Networks. *World Journal of Wireless Devices and Engineering* , 13-18.

Kim, S. -H., & Lee, I. -Y. (2013). A Secure and Efficient Vehicle-to-Vehicle Communication based on Sensor Network. *International Journal of Security and Its Applications*, 241-248.

Kobayashi, S., Otomo, K., Fukuda, K., & Esaki, H. (2017). Mining causality of network events in log data. *IEEE Transactions on Network and Service Management*, 15(1), 53-67.

Lee, G. H., Choi, J. D., Lee, J. H., & Kim, M. Y. (2020). Object Detection Using Vision and LiDAR Sensor Fusion for Multi-channel V2X System. *2020 International Conference on Artificial Intelligence in Information and Communication (ICAIIC)* , 1-5.

Martinez, J. E. (1985). Computer Aided Automobile Accident Reconstruction-Theory, Graphlcs, Animation . *SAE Technical Paper*.

Microsoft Docs. *Command and Query Responsibility Segregation (CQRS) pattern 2019*. URL: https://docs.micro soft.com/en-us/azure/architecture/patterns/c qrs

National Highway Traffic Safety Administration. (2014). Vehicle-to-Vehicle Communications: Readiness of V2V Technology for Application, 105-107.

Ousterhout, J., & Douglis, F. (1989). Beating the I/O bottleneck: A case for log-structured file systems. *ACM SIGOPS Operating Systems Review*, 23(1), 11-28.

SAE International. (2009). Dedicated Short Range Communications (DSRC) Message Set Dictionary J2735.

US Department of Transportation. (2018). Safety Pilot Model Deployment.

Xin, C., Dan, L., & Shuo, H. (2020). Research on Deceleration Early Warning Model Based on V2X. *2020 12th International Conference on Measuring Technology and Mechatronics Automation (ICMTMA)*, 318-321.

Xu, Z., Chen, J., Peng, L., Zhao, W., & Chu, X. (2019). A General Analysis Method for Data Link Log Files. *Journal of Physics: Conference Series* (Vol. 1325, No. 1, p. 012144)..