Project Report of Master of Engineering

Design of V2X-based Vehicular Contents Delivery Network for Autonomous Driving

자율주행을 위한 V2X 기반 차량 CDN 설계

February 2021

Graduate School of Engineering Practice
Seoul National University
Department of Engineering Practice

Sanghoon Lee

Design of V2X based Vehicular Contents Delivery Network for Autonomous Driving

자율주행을 위한 V2X 기반 차량 CDN 설계

Professor Seong-Woo Kim

Submitting a Master's Project Report

February 2021

Graduate School of Engineering Practice
Seoul National University
Department of Engineering Practice

Sanghoon Lee

Confirming the master's Project Report written by
Sanghoon Lee
February 2021

Chair	Hoon Han	(Seal)
Examiner	Seong-Woo Kim	(Seal)
Examiner	Young-Hoon Park	(Seal)

Abstract

Recent technical innovation has driven the evolution of autonomous vehicles. To improve safety as well as on-road vehicular experience, vehicles should be connected with each other or to vehicular networks. Some specification groups, e.g., IEEE and 3GPP, have studied and released vehicular communication requirements and architecture. IEEE's Wireless Access in Vehicular Environment focuses on dedicated and short-range communication, while 3GPP's New radio V2X supports not only sidelink but also uplink communication. The 3GPP Release 16, which supports 5G New Radio, offers evolved functionalities such as network slice, Network Function Virtualization, and Software-Defined Networking. In this study, we define and design a vehicular network architecture compliant with 5G core networks. For localization of autonomous driving vehicles, a high-definition map needs to contain the context of trajectory . We also propose new methods by which autonomous vehicles can push and pull map content efficiently, without causing bottlenecks on the network core. We evaluate the performance of V2X and of the proposed caching policy via network simulations. Experimental results indicate that the proposed method improves the performance of vehicular content delivery in real-world road environments.

Keywords : Autonomous driving, C-V2X, HD Map, Vehicular Content Centric Networks.

Student Number : 2019-25271

Contents

I. Introduction **1**

II. Related Works **6**

2.1 V2X Standardization 6

2.1.1 IEEE WAVE 6

2.1.2 3GPP C-V2X 9

2.2 Geographic Contents 14

2.3 Vehicular Content Centric Network 17

III. System Modeling **20**

3.1 NR-V2X Architecture Analysis 20

3.2 Caching Strategy for HD Map Acquisition 23

IV. Evaluation **30**

4.1 Contents Replacement Strategy 30

4.2 V2X Characteristics 36

4.3 Edge Performance in Driving on the Road 38

4.4 Edge Performance on 3D Point Clouds Caching for Localization 44

V. Conclusion **47**

Bibliography **49**

Abstract **54**

List of Figures

Figure 1. Architecture of a Connected Autonomous Vehicle Platform. 3
Figure 2. IEEE WAVE Architecture. 7
Figure 3. WAVE Channel Spectrum. 7
Figure 4. Reference points of WAVE and C-V2X. 10
Figure 5. Channel allocation in LTE C-V2X and NR V2X. . . . 11
Figure 6. Architecture of NR V2X. 13
Figure 7. HD Map Layers. 15
Figure 8. Nodes and Packets in Content-Centric Network (CCN). 17
Figure 9. Local traffic steering at edge side. 21
Figure 10. Requesting chunks Procedure. 23
Figure 11. On receipt of interest from requester, cache server checks if PIT has a pending interest on desired nonce. 24
Figure 12. On the receipt of data from face, PIT checks the pending records and forwards data 25
Figure 13. Overview of vehicular content delivery. 27
Figure 14. Weight and request vector at the chunk n. 28
Figure 15. Overview of grid road edge simulator. 31
Figure 16. Cache replacement. 32
Figure 17. Geospatial cache initialization. 33
Figure 18. Random cache initialization. 34
Figure 19. Hit ratio to size of vehicle buffer. 35

Figure 20. Packet Error Ratio to vehicle popularity by ns-3. 37
Figure 21. HD map of Yeouido (Latitude : 37.5289, Longitude : 126.9169). 39
Figure 22. Trajectory of Downtown scenario. 40
Figure 23. Trajectory of Motorway scenario 40
Figure 24. Velocities of trajectories. 41
Figure 25. Elapsed Time of Core Server. 42
Figure 26. Elapsed Time of Edge Server. 43
Figure 27. Elapsed Time of Edge and Core Server for Point Cloud. 45
Figure 28. Feature Map through Edge Server. 46
Figure 29. Feature Map through Core Server. 46

List of Tables

Table 1. SAE Automation Levels. 1

Table 2. List of 3GPP NR V2X specification. 12

Table 3. Comparison of Geospatial Database. 14

Table 4. PQI list of NR Sidelink. 36

Table 5. Parameter of ns-3 simulation. 37

Table 6. Specification of experimental devices. 38

Chapter 1

Introduction

The evolution of technologies such as AI, telecommunication, and big data affects not only the IT industry but also traditional machinery and process industries such as automobiles, which can apply these innovations in autonomous driving. The Society of Automotive Engineers (SAE) International defined six levels of automated driving [1]. In Table 1, Level 5 Autonomous Vehicle (AV) is defined that it perceives a road environments and controls a maneuver by itself.

Level	Status	Driver Status	Monitor	Control
0	No Automation	Hands On	Human	Human
1	Driver Assistance	Hands On	Human	Both
2	Partial Automation	Hands Off	Human	Machine
3	Conditional Automation	Eyes Off	Both	Machine
4	High Automation	Mind Off	Machine	Machine
5	Full Automation	Driver Off	Machine	Machine

Table 1: SAE Automation Levels.

As per SAE automation levels, a Level 5 Autonomous Vehicle (AV) should be able to perceive a road environment and maneuver itself independently. A crucial prerequisite for a higher level of automation is the ability to guarantee the safety of the vehicle and its passengers by precisely predicting risks and devising a driving plan that eliminates the possibility of accidents.

Contemporary self-driving platforms focus on realizing perception abilities matching those of human drivers. However, given the inherent uncertainty involved in driving on roads , they cannot mitigate all risks entirely. Thus, research on risk awareness to ensure the safety of autonomous driving is an ongoing process [2, 3].

A functional architecture for autonomous driving consists of modules focused on perception, decision, control, and manipulation [4]. For high-level automation, a sensor-equipped vehicle must detect the surroundings in detail and perform instantaneous driving decision planning and control. AI can extend an AV's sensing range. However, AVs are unable to perceive beyond the Non-Line of Sight (NLOS) of the sensors embedded in it.

In order to overcome this limitation, AVs could be connected to each other or to a vehicular network in order to share sensing data. The concept of Connected Autonomous Vehicle (CAV) over Vehicle-to-Everything (V2X) communication for complementing the perception and awareness of AVs has been discussed by the academic community, standardization groups, and telecommunication operators, all of whom have contributed to vehicular communications [5]. The connectivity of vehicles, infrastructure, and pedestrians is expected to improve safety as well as the driving experience.

The characteristics of the vehicular contents differ from existing client-server communication models. Figure 1 illustrates that the CAV merges its own collected sensor data with information received from adjacent vehicles, infrastructure, and pedestrians. Cooperative perception has been a major focus of V2X applications [6]. Since the first V2X standard IEEE 802.11p was released in July 2010, subsequent research has helped the architec-

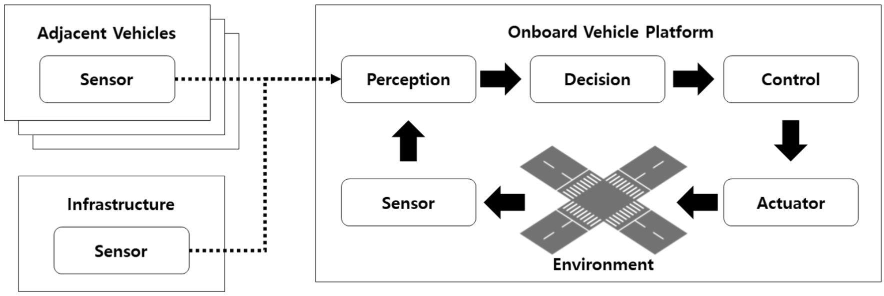

Figure 1: Architecture of a Connected Autonomous Vehicle Platform.

ture, specifications, protocols, and applications for vehicular communication to evolve considerably. IEEE Wireless Access Vehicular Environments (WAVE) and ETSI ITS-G5 are IEEE 802.11p-based Dedicated Short-Range Communications (DSRC), which disseminate content within the range as a wireless LAN. WAVE adopted SAE J2735 message protocols for cooperative awareness, e.g., Basic Safety Message (BSM). WAVE utilizes broadcast transmission via a dedicated control channel, which improves the delay performance as compared to unicast. However, as the number of participants and the content increases, the probability of transmission success decreases owing to congestion [7]. Consequently, it is important to identify the priority of vehicular content and distribute it over multiple channels.

Perception information and content in networks is used to determine precise localization and mapping, which are crucial for autonomous driving [8]. Localization is a necessary procedure for AV perception. An AV estimates the precise position orientation by comparing the sensor data and the built-in geographical content, e.g., High-Definition (HD) Map [9]. As on-board storage is a limited computing resource, it is cumbersome to carry complete geographical and updated data within a single vehicle. Offloading HD maps to the remote cloud server could be an alternative, but this causes a bottleneck effect when multiple AVs request data simultaneously, as demonstrated in our experiments. Caching and distribution on the edge side must be considered in order to mitigate those effects.

In this paper, we propose a new architecture and the corresponding algorithms for HD map acquisition for autonomous driving with the analysis of a cooperative perception procedure. Although previous studies on V2X attempted to implement cooperative perception [6] and connectivity between vehicles via Cell-V2X [10], to the best of our knowledge, this study is the first to propose a vehicular cache model from the perspective of content delivery at the edge of V2X. The contributions of this paper can be summarized as follows:

- We propose an architecture and the corresponding procedures of NR 5GC-based vehicular content delivery networks for real-time HD map acquisition.
- In this architecture, we propose a cache replacement strategy to maximize the efficiency of edge utilization in vehicular content delivery scenarios.
- We verify the feasibility and performance of our proposal through actual experiments on an urban road in order to suggest the guiding strategy for real deployment.

The remainder of the paper is organized as follows. Chapter 2 describes the literature on cooperative perception algorithms over V2X, Vehicular Content-Centric Networks (VCCN), and caching on the vehicular edge node. Chapter 3 describes the 3GPP-enhanced C-V2X architecture and defines the roles and functions of each network entity. Based on C-V2X architecture, a methodology involving cooperative perception via multiple channels and offloading geometric information on the edge node are proposed. In Chapter 4, we evaluate the performance of the proposed algorithm using simulation tools and experiments. In Chapter 5, we conclude our proposals and suggest ways to overcome further challenges.

Chapter 2

Related Works

In this chapter, we will review V2X, geometric content, and content delivery networks, which are the key elements of VCCNs for autonomous driving.

2.1 V2X Standardization

2.1.1 IEEE WAVE

IEEE WAVE was introduced to support V2V and V2I for traffic information broadcast. IEEE released IEEE 802.11p originating from 802.11a in 2010, which reduced the channel bandwidth to 10 MHz and utilized the frequency spectrum around 5.9 GHz [11]. WAVE adopts IEEE 802.11p, which defines the specification of PHY and MAC layers, and the 1609 series for LLC sublayer, WAVE short message protocol (WSMP), and security shown in Figure 2.

In the PHY layer, the Orthogonal Frequency-Division Multiplexing (OFDM) transmission scheme enables multiplexing and maintains orthogonality between subcarriers, resulting in high frequency efficiency. The MAC layer adopts Distributed Control Function (DCF) CSMA/CA. If there are pending requests, messages will be re-transmitted after random back-off time for collision avoidance. As shown in Figure 3, seven channels, i.e.,

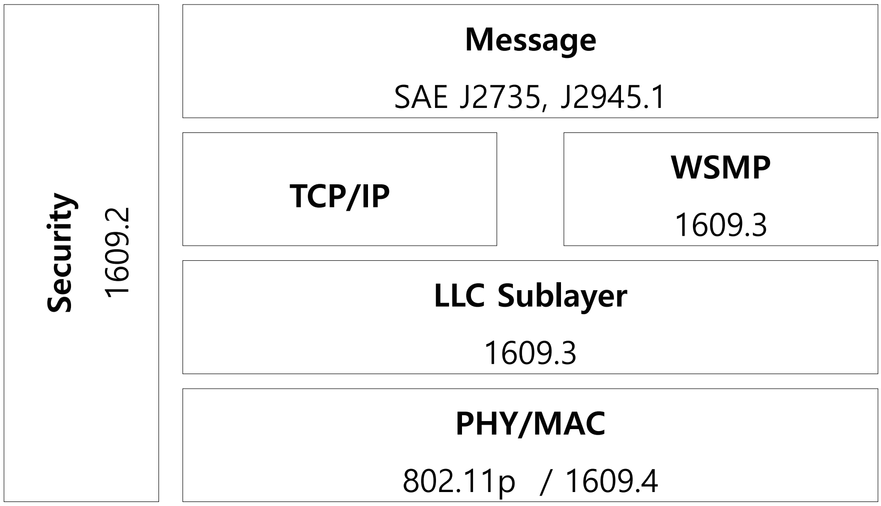

Figure 2: IEEE WAVE Architecture.

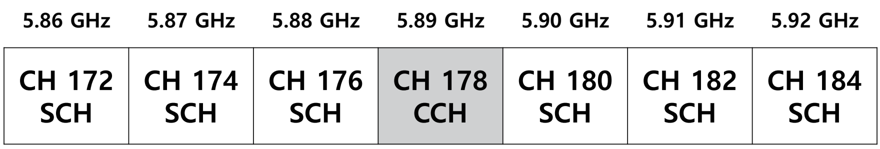

Figure 3: WAVE Channel Spectrum.

one Control Channel (CCH) and six Service Channel (SCH) are allocated for vehicular communication. Vehicles continuously alternate the CCH for safety message exchanges and the SCH for complementary applications.

For V2X services, SAE J2735 describes 17 protocols for messages, data frame format, and application architecture, and SAE J2945.1 defines the minimal requirements. Seventeen messages include the Basic Safety Message (BSM) that broadcasts vehicular positions and events and Single Phase and Timing (SPaT) that notifies the cycle information of traffic light and map data.

WAVE can deploy a distributed and decentralized network regardless of core network controls. However, IEEE 802.11p has limitations, such as a hidden node problem [11]. WAVE includes contention-window techniques and adaptive message frequency techniques to decrease the congestion, but these cannot be applied in high-density networks. IEEE 802.11p is not suitable for high-data ultra-low latency applications as it has a large communication overhead that consumes bandwidth and increases latency in error retransmission [11]. The NLOS of radio propagation and dynamic conditions of V2X always interrupt the connection [12].

2.1.2 3GPP C-V2X

The 3rd Generation Partnership Project (3GPP) is an alliance of telecommunications standard development organizations, which covers cellular telecommunications technologies including radio access, core network, and service capabilities. In addition to DSRC, a cellular vehicle network can be an alternative, thereby solving a communication coverage problem and transmitting more data [13].

3GPP initiated the V2X working group and released the first LTE-based V2X specifications, Release 14, in 2017. Cellular V2X (C-V2X) can utilize the uplink communication offered by Bases Station (BSs) and Device-to-Device (D2D), like DSRC. A feasible network scheme supports various vehicular scenarios including V2V, V2I, and V2P, as shown in Figure 4. 3GPP defines V2V interface as sidelink telecommunication.

Owing to the emergence of 5G NR (New Radio), 3GPP has also evolved V2X requirements and schemes based on new keywords, such as millimeter wave, beam forming, adaptive numerology, and network slice. Release 16, approved in 2018, had enhanced V2X use cases, e.g., platooning, extended sensor, remote driving, and advanced driving. The main items include sidelink design, Uu link enhancement to support enhanced V2X use cases, Uu link-based sidelink resource allocation/configuration, QoS management, and coexistence.

In the sidelink communication in LTE V2X, only the broadcast method was supported. However, sidelink in NR V2X, which should support new use cases, supports not only broadcast but also unicast and groupcast. A

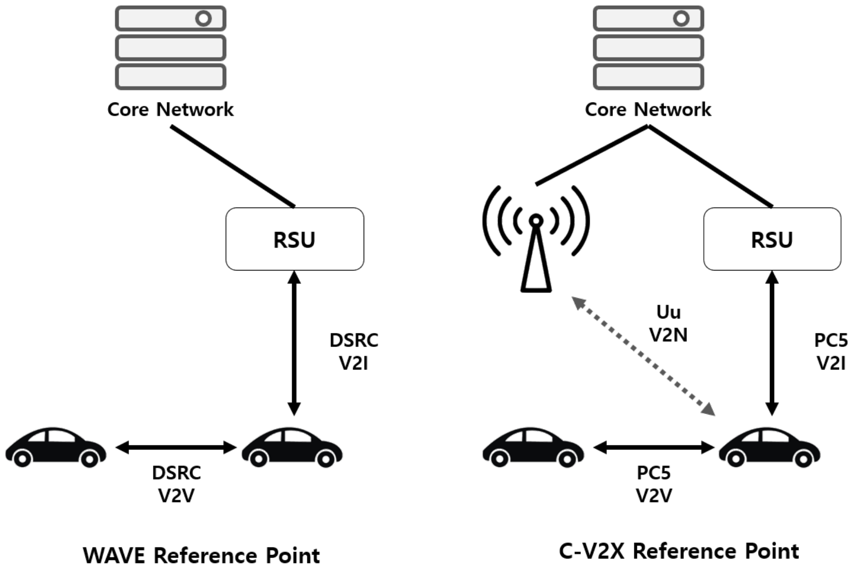

Figure 4: Reference points of WAVE and C-V2X.

new channel type for feedback sharing, Physical Sidelink Feedback Channel (PSFCH), was defined because of issues on link adaption, Hybrid Automatic Request (HARQ) feedback, and Channel State Information (CSI) feedback in LTE V2X [14]. Physical Sidelink Control Channel (PSCCH), control channel for sidelink resource allocation, and Physical Sidelink Shared Channel (PSSCH) for data channels decided to follow LTE V2X, but details on allocation and multiplexing were changed; channels were allocated by time domain and frequency domain, as shown in the right side of Figure 5.

Initial synchronization can happen by receiving a signal from a base station (gNB or eNB). However, when the vehicle terminal is outside the coverage area, it is possible to obtain synchronization by receiving a global navigation satellite system (GNSS) signal or a Sidelink Synchronization

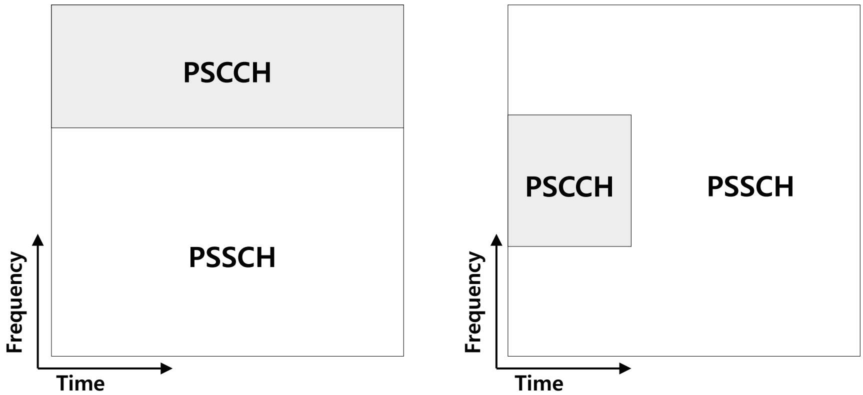

Figure 5: Channel allocation in LTE C-V2X and NR V2X.

Signal from an adjacent terminal.

In case of allocation, terminals and cell can choose the Sidelink mode. In Mode 1, a vehicle (User Equipment) allocates resources of adjacent vehicles. In mode 2, the vehicle can sense through adjacent vehicles. Modes 3 and 4 support direct V2V communications but differ on how radio resources are allocated. Resources are allocated by the cellular network under Mode 3. Mode 4 does not require cellular coverage; vehicles autonomously select their radio resources using a distributed scheduling scheme supported by congestion control mechanisms [15].

Compared with LTE core network, the 5G Core showed major differences in structural and procedural terms [16], where new terminologies of network entities were introduced, Control Plane and User Plane were separated, and network functions were modularized. Table 2 lists the the 3GPP V2X specifications released with 5G network.

As the 5G core architecture was renewed, V2X specifications were also

Number	Title
TS 22.185	Service requirements for V2X services
TS 22.186	Service requirements for enhanced V2X scenarios
TR 22.886	Study on enhancement of 3GPP support for 5G V2X services
TS 23.285	Architecture enhancements for V2X services
TS 23.286	Application layer support for Vehicle-to-Everything (V2X) services; Functional architecture and information flows
TS 23.287	Architecture enhancements for 5G System (5GS) to support Vehicle-to-Everything (V2X) services
TR 23.764	Study on enhancements to application layer support for V2X services
TR 23.795	Study on application layer support for V2X services
TR 38.885	Study on NR Vehicle-to-Everything (V2X)
TR 38.886	V2X Services based on NR; User Equipment (UE) radio transmission and reception

Table 2: List of 3GPP NR V2X specification.

changed. TS 23.287 defined NR V2X reference points and architecture, as shown in Figure 6. A vehicle as User Equipment (UE) supports up/downlink via the Uu reference point and sidelink via the PC5 reference point. The V2X Application Server provides vehicular services, such as accident and congestion warning, platoon driving, and HD Map acquisition, where V2X AS controls UE requests and prioritizes transactions. Among the potential scenarios, event detection via cooperative perception in C-V2X has been proposed [10].

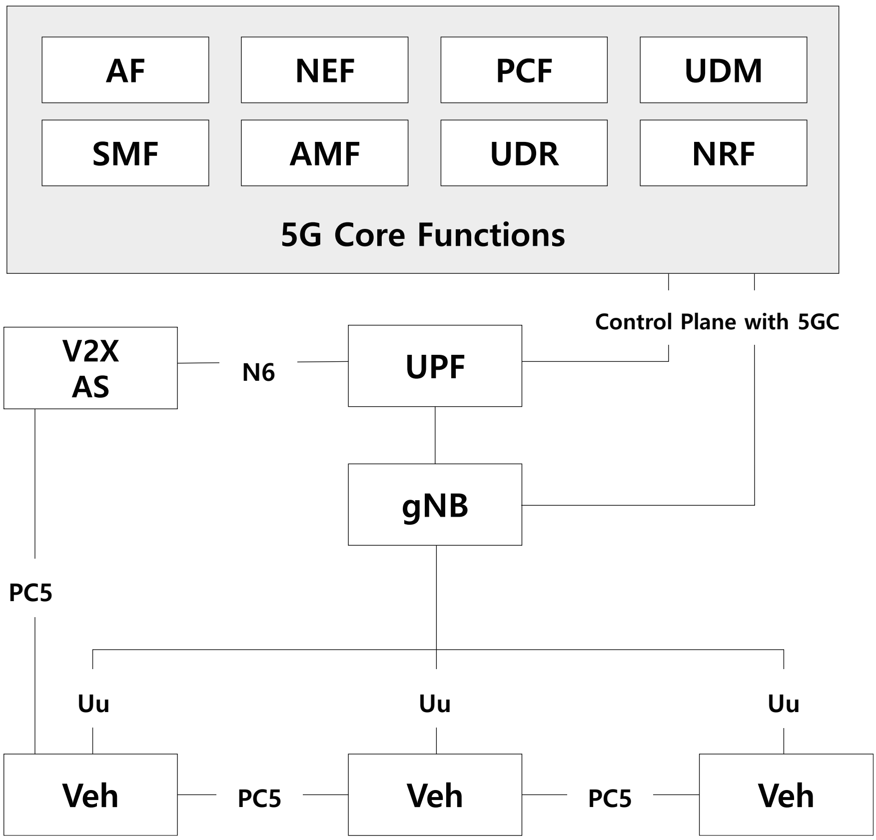

Figure 6: Architecture of NR V2X.

2.2 Geographic Contents

Geometries in a map for navigation can be described by the group of points, line strings, and polygons. GeoJSON is an open standard format designed to systematically express terrain based on location information by IETF RFC 7946 [17]. GeoJSON is compatible with GPX, which makes it possible to compress the polyline and load the application program with light-weight data. As geographic content becomes more complex and larger in size, efficient database management is required. A DBMS specialized for geographic content and rapid querying is proposed. Table 3 summarizes the features of a NoSQL database supporting geospatial functions [18].

Database	**Supported Geometry Objects**	**Supported Spatial Indexes**	**Data Format**
MongoDB	Point, LineString, Polygon, MultiPoint, MultiLineString, MultiPolygon, GeometryCollection	2dsphere index, 2d index based on geohash	GeoJSON
Couchbase	Point, LineString, Polygon, MultiPoint, MultiLineString, MultiPolygon, GeometryCollection	R-Tree, designed by users	GeoJSON
Cassandra	Point, LineString, Polygon	Lucene index	WKT
Redis	Point	geohash	GeoJSON

Table 3: Comparison of Geospatial Database.

ETSI defined the interfaces and architecture of Local Dynamic Maps [19]. As shown in Figure 7, the HD map has a layered structure. Layer 1 represents a base map such as geometric information. Layer 2 shows the layout

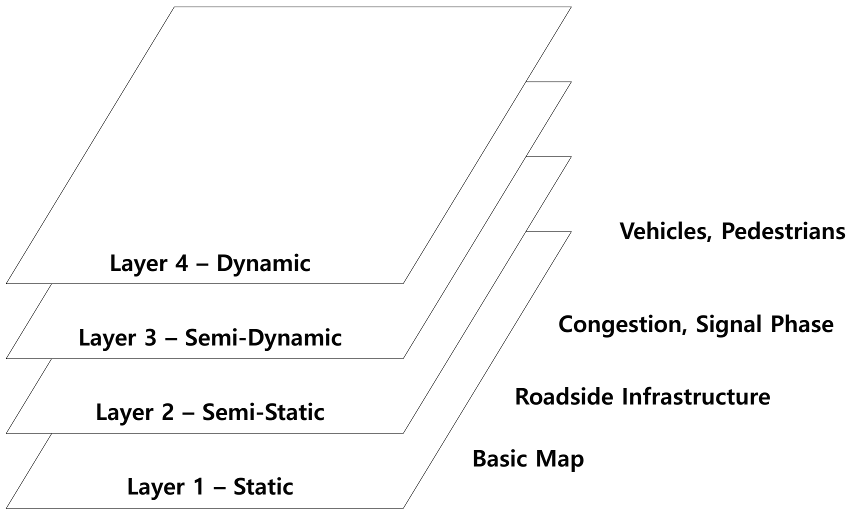

Figure 7: HD Map Layers.

of roadside infrastructure, e.g., traffic lights, road signs, and lane markings. Layer 3 represents time-variant information such as traffic congestion and signal phase of traffic light. Layer 4 shows dynamic information where vehicles and pedestrians move, which can be shared through V2V direct communications.

With GNSS, such as GPS, AVs can estimate their positions. However, it needs to localize more accurately itself in the real-world environment. An HD Map, which has high precision at centimeter-level, is essential for autonomous driving [20]. A navigation map contains only basic information, link, and node information of roads. Figure 21 shows another snapshot of an HD Map that contains information on traffic lights, signs, curbs, road marks, and various structures as well as information on each lane level such as the road center line and boundary line in a 3D digital format. This will be explained in Section 4.3. By feature mapping between map elements and detected objects, autonomous vehicle can calculate centimeter-accurate positions and dynamics.

During the localization process, an autonomous vehicle integrates GNSS, Lidar, Radar, and Camera vision detects [21]. Integrating more and higher sensor equipment reduces the error caused by device specification and environmental noise but increases costs and resource usage. Utilizing an HD map can reduce the diversity of on-board equipment and ensure better performance. Sven *et al.* [9] introduced the particle filter-based localization algorithm utilizing HD maps, GNSS, and odometry. Hao *et al.* [19] estimated vehicular position using only HD maps, monocular vision, and GNSS. The road signs perception information by the LIDAR scanner is matched with features on HD map so that the vehicular system can infer the position accurately [22], [23].

2.3 Vehicular Content Centric Network

Content-Centric Network (CCN) and Information-Centric network (ICN) are emerging as next-generation network architecture, in contrast to Host-Centric Network. A Host-Centric Network consists of clients and servers, where all network entities route and forward data by resolving the host address. By contrast , CCN decides based on the contents of data instead of the host address, which designs elements of the network that utilize the identifiers or names of contents instead of the address of the host [24].

In CCN networks, a content consuming node asks for specific content by sending an Interest Packet via the network. A receiver that possesses the relevant content responds by sending a Data Packet, as shown in Figure 8. The Interest Packet includes a detailed description of content name, preference in case multiple types of content match, and items for security. The Data Packet contains signed information of public key digest, time stamps, and encryption description.

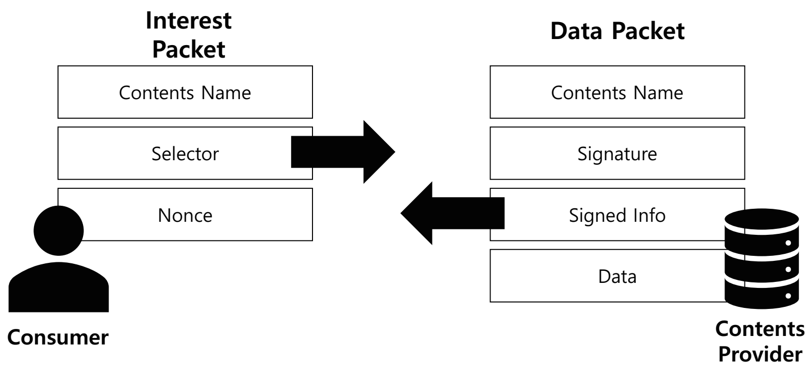

Figure 8: Nodes and Packets in Content-Centric Network (CCN).

The CCN forwarding mechanism has three main components: Content Store (CS), Pending Interest Table (PIT), and Forwarding Information Base (FIB). CCN supports multiple faces connected with other interfaces. Similarly, with the buffer memory of a router, CS stores data content and lists and exchanges caches based on the Least Recently Used (LRU) or Least Frequently Used (LFU) policy. PIT records and manages the face at which the interest packet is transmitted or received after transmitting the interest packets. FIB is used to forward the interest packet, where the policy is determined in consideration of security and bandwidth efficiency. FIB list is created and removed according to the determined policy.

Recently, studies on the feasibility of applying CCN concept to vehicular network have been conducted. CCN is advantageous in dynamic and mobile environments such as for vehicular communications [25]. The CCN architecture applied to V2X is named Vehicular Content-Centric Network (VCCN), which has certain specific characteristics e.g., content naming, name resolution, and assigning roles of CCN nodes like CS, FIB, and PIT [26].

For minimizing delivery delay, VCCN leverages a caching algorithm that considers popularity, vehicle mobility, and cache capacity in edge nodes [27]. Road Side Unit (RSU) can play a role not only as a base station but also as caching storage for vehicles. In vehicular content delivery, interesting content depends on the given positions of consumers, so content returned by the nearest provider can reduce the number of resources needed [28]. Vehicles can be consumers as well as providers of information. With sidelink communication, vehicles can share content with adjacent consumers and manage their own on-board caches. Mobile vehicular caches can improve the availability of network for pedestrians and other types of mobile users [29]. To this end, we will build a new cache policy for the essential functions of autonomous driving on the V2X and VCCN.

Chapter 3

System Modeling

In this chapter, we propose a system model that enables autonomous vehicles to extend their perception ranges and precise localization through CCN.

3.1 NR-V2X Architecture Analysis

In this research, we will define the roles and procedures of each network entity for HD map cache and acquisition services according to the NR V2X specification introduced in Section 2.1. 5GC in Release 16 supports vertical service networks and edge computing. Profiles of vehicles should be provisioned and registered in Unified Data Management (UDM) before attaching to networks. UDM manages a V2X service provision profile. Configurations, such as radio parameters, QoS, and Public Land Mobile Network (PLMN) properties stored in User Data Repository (UDR) are forwarded to Application Servers, gNodeB (gNB), and vehicles over a control plane via Policy Control Function (PCF).

On the edge side, contents stream from UEs branches into Local Data Network and Core Data Network according to the ETSI Mobile Edge Computing (MEC) specification [30]. As shown in Figure 9, local User Plane Function (UPF) supporting MEC steers the data stream as per HD map in-

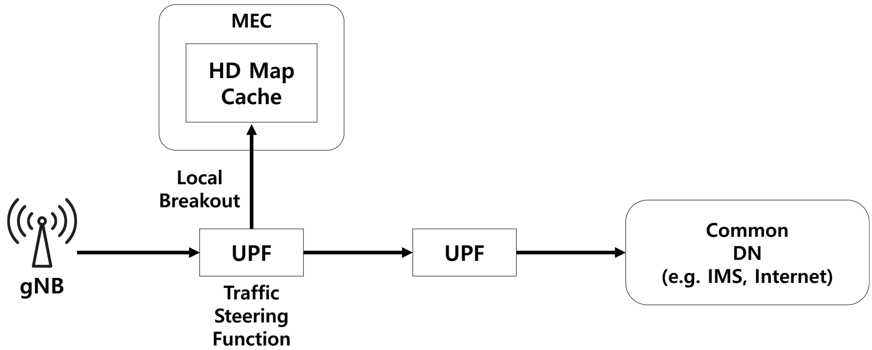

Figure 9: Local traffic steering at edge side.

terest and interest packets are processed at the edge server on MEC. Local steering reduces the number of nodes between map consumer and provider, which can prevent bottlenecks at the core side.

Cache servers supporting NR V2X Uu and PC5 sidelink are responsible for the specific geographic region allocated by V2X AS. Each cache server consists of CS, FIB, and PIT, and maintains content based on the caching policy.

The performance of wireless communication can be determined by Packet Error Rate (PER) and End-to-End delay time. Let δ be the PER interval that is divided into **w** sub-window intervals, normally set to one second. $\delta = n \cdot w$, where n is normally set to five seconds PER value. At the end of each sub-window interval w the PER is calculated at the end of each δ for the last n sub-windows, which can be formulated as

$$PER = \frac{\text{missed number of packets}}{\text{total number of packets}}. \tag{3.1}$$

According to [31], PER can be approximated by an exponential function as Equation (3.2), where γ denotes SNRs and m denotes MCS index, which can be formulated as follows:

$$PER^{(m)}(\gamma) = \begin{cases} 1 & \text{for } \gamma < \gamma^{(m)} \\ \exp(-\alpha^{(m)}(\gamma - \gamma^{(m)})) & \text{for } \gamma \geq \gamma^{(m)}, \end{cases} \tag{3.2}$$

where $\alpha^{(m)}$ and $\gamma^{(m)}$ are an exponential decay ratio and SNR computed by simulations at each MCS index m, respectively.

3.2 Caching Strategy for HD Map Acquisition

A global map is divided into several chunks for edge caching. Cache servers are located on the edge side of the vehicular network. Initialized cache servers store only the nearest chunks from a responsible location. Assuming that a vehicle has set up path plans, it is necessary to fetch maps of the trajectory. If a vehicle does not possess maps on the desired trajectory in the vehicular buffer, it will try to request maps from the edge network. The cache server stores chunks of the map for the relevant region as per the cache initialization policy, as shown in Figure 10. If the cache server stores the desired chunks, it will respond to the interest packet and transmit content to the vehicle. If not, a local cache will forward the interest packet to the upper-level server on the core of network or an adjacent cache server and record on FIB and PIT.

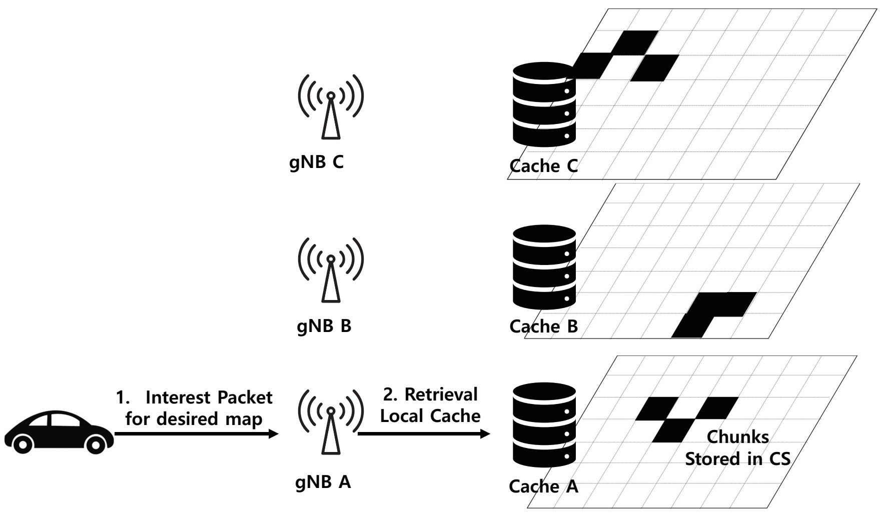

Figure 10: Requesting chunks Procedure.

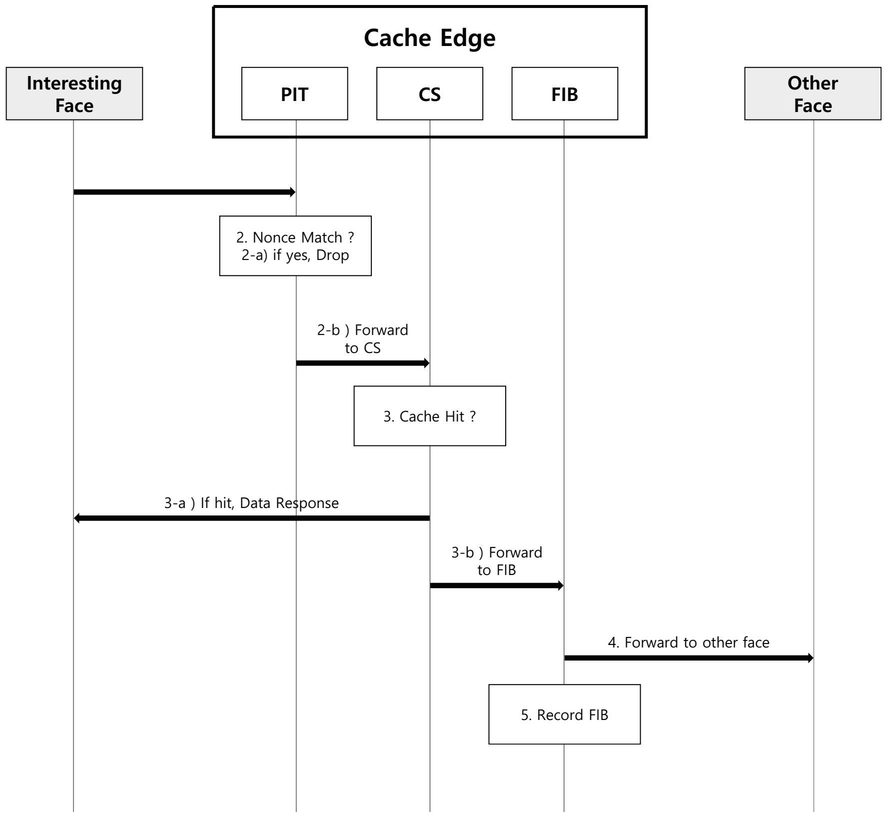

Figure 11: On receipt of interest from requester, cache server checks if PIT has a pending interest on desired nonce.

First, the cache server checks if PIT has a pending interest on desired nonce, as shown in Figure 11. In the case of PIT match, PIT discards the interest; otherwise, PIT forwards the interest to CS. In this case, it is possible to reduce delay in responding to the consumer. CS searches matched chunks in the storage. If cache hits, CS will answer to the interesting face. If cache misses, CS will relay the interest to the other face via FIB. FIB records the transaction with the other face.

When data chunks arrive at the face of cache server, PIT checks the

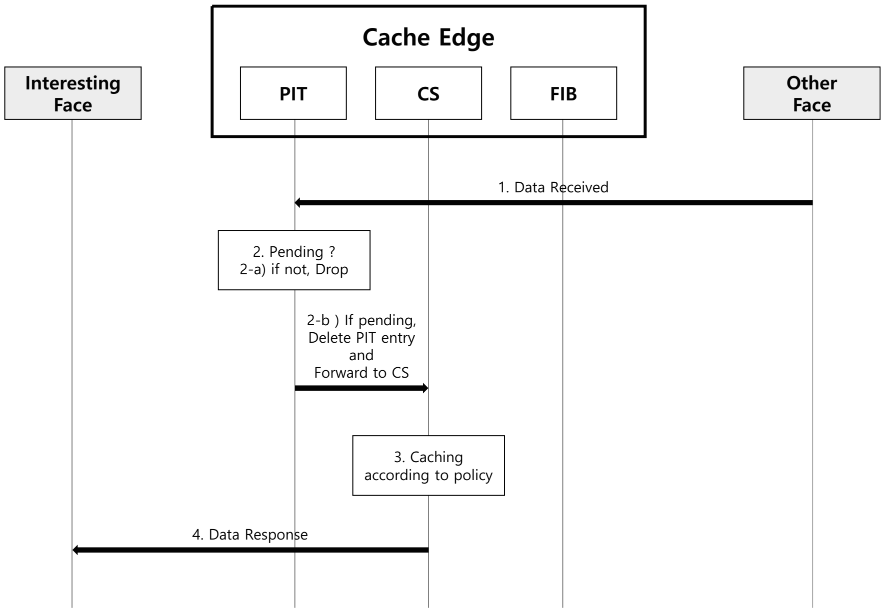

Figure 12: On the receipt of data from face, PIT checks the pending records and forwards data

pending records, as shown in Figure 12. PIT searches the pending records with respect to the received data. If a record matches, PIT will forward to CS, and CS determines whether or not to store the data on the server. If not , PIT discards the data.

As described in the procedures above, a caching policy affects an immediate delivery of interest and response messages, which can predict the chunks to be requested because it has limited space to store data. Therefore, the cache server should manage requests from the consumer and orchestrate chunks in storage effectively. The main objective of the cache strategy is to increase the probability of hitting its caches without fetching the chunks from the origin content server.

A cache algorithm is a conventional concept that has been studied for a long time in computer science, where they have researched to provide immediate accessibility in processing logic between processor and memory. A fundamental operation is cache replacement, which erases redundancy and writes a candidate to be read by the processor. Classical methods, e.g., First in, First out (FIFO), Last in, Last out (LIFO), Least Recently Used (LRU), and Least Frequently Used (LFU) are utilized to design a processor and web server architecture. FIFO evicts the old cache first and then adds a new one, whereas LIFO clears recent used data. FIFO and LIFO leverage stack architecture of memory, so that they operate immediately. However, LRU and LFU operate according to statistics, where LRU deletes the cache that has been unused for the longest time. By contrast, LFU places only frequent data and removes rarely used data.

Unlike cache replacement in a conventional domain, vehicular content, such as static HD map and dynamic road environment perception, should be handled using a specific approach. In contrast to general content forwarding consumers and producers of vehicular content are not spatially fixed but continuously moving. The purpose of conventional caches is to predict which content consumer will place a request, whereas in vehicular content cache the mission is to determine where the consumer will request the specific chunks.

To solve the new problem, we assume a road environment with wireless stations that are connected to vehicles and store vehicular content at the edge. A vehicle has its own buffer to load the HD map chunks, and according to buffer policy, will request other chunks on its trajectory in the future.

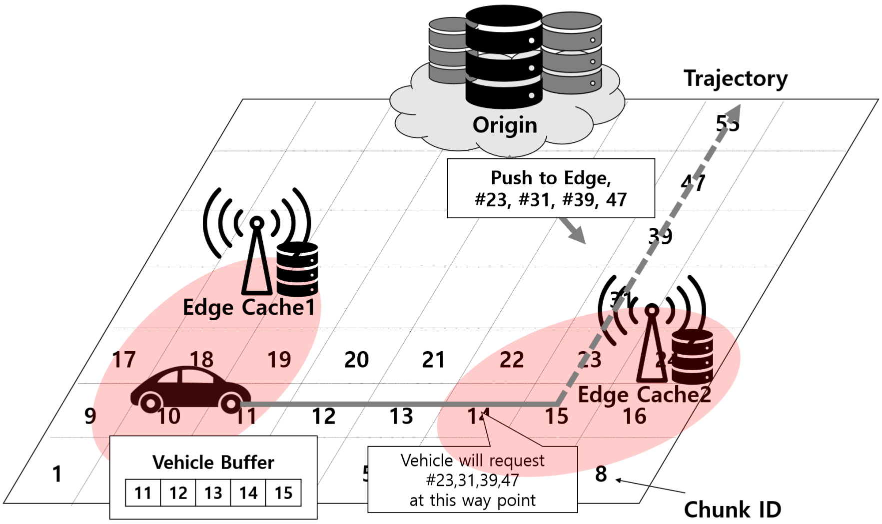

Figure 13: Overview of vehicular content delivery.

In Figure 13, at this time, Edge 1 is responsible for the vehicle content, although this will be disconnected soon. The vehicle will move to an area regard of chunk ID 14 and will fetch the chunks 23, 31, 39, 47, and 55. If Edge 2 stocks up those chunks, the vehicle will hit the caches and load its buffer in a shorter time. Most vehicles have a trajectory plan to the usual destination and can estimate the approximate direction and position in the near future. Therefore, we propose the method to reserve the future chunks with its trajectory information.

Trajectory consists of n chunks corresponding to way points. t_n represents the time to reach at the way point of n-th chunk. The probability to go off the scheduled path plan will increase over time. Therefore, reservation weight value decreases with time as described in Equation (3.3). We assume that k vehicles subscribe to the HD map service.

$$\vec{w_{n,i}} = \alpha \cdot \exp(-\beta \cdot t_n) \cdot \vec{v}, \quad |\vec{v}| = 1, \tag{3.3}$$

where $\vec{v}$ is the i-th vehicle's direction vector at the center point of chunk, which can be calculated based on trajectory direction. This assumption will be validated through experiments. The vehicle reports reservation weight value corresponding to each chunk except for chunks already loaded in the buffer to V2X Application server. The V2X application server merges weight values from vehicles and calculates total weight value of the chunk. Therefore, we can derive the total weight vector of each chunk.

$$\vec{w_n} = \frac{1}{k}\sum_{i}^{k} \vec{w_{n,i}} = |\vec{w_n}| \cdot (-\vec{r}). \tag{3.4}$$

In Equation (3.4), $\vec{r}$ represents the request possibility vector, which indicates the direction of candidate to request, as shown in Figure 14. The probability of an interest occurring in the direction indicated by $\vec{r_n}$ is high.

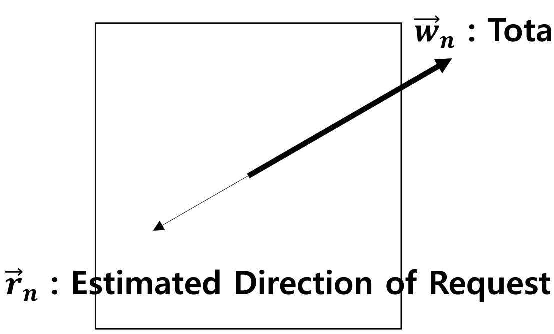

Figure 14: Weight and request vector at the chunk n.

When cache replacement is needed, the cache server retrieves chunk weight values and vectors collected from the vehicles in V2X AS. The cache server compares the values and chooses the chunk that has a high probability to be requested at the cache server. $\vec{ec}$ indicates a position vector between cache server and chunk n. The cache server selects the proper chunk n^* that maximizes the dot product of weight vector and position vector. The optimal chunk can be formulated as follows:

$$n^* = \arg\max_{n} \vec{ec} \cdot \vec{w}_n. \tag{3.5}$$

For an efficient response to a subsequent request, the cache replaces chunks according to the weighted eviction policy described above. Each chunk has the reservation weight. If chunks need to be fetched from the upper layer, it searches through whole chunks and evicts the least weighted chunk from the cache.

Chapter 4

Evaluation

In this chapter, we evaluate the proposed system of cache policy through extensive simulations and real experiments with vehicles on the road.

4.1 Contents Replacement Strategy

To verify the proposed architecture and cache replacement algorithm, we model and implement our Manhattan Grid Simulator. The Simulator can generate road grids, vehicles, and a cache server based on input variables. As shown in Figure 15, the simulator consists of three objects: environment, a vehicle, and a cache server. The environment object can generate multiple players such as vehicles and caches and measure vehicular mobility and content delivery. The vehicle object generates a path plan randomly according to traffic condition and communicates with the nearest cache server to obtain geographical content. The cache server object transmits the chunks requested when a cache hits; otherwise, it replaces and reorganizes caches. In our proposed method, each vehicle disseminates reservation messages for future chunks and the origin server merges and calculates the weight value and direction of each chunk.

We compared the proposed algorithm to classic cache replacement methods, e.g., LRU and LFU. Researchers have tried to apply classical

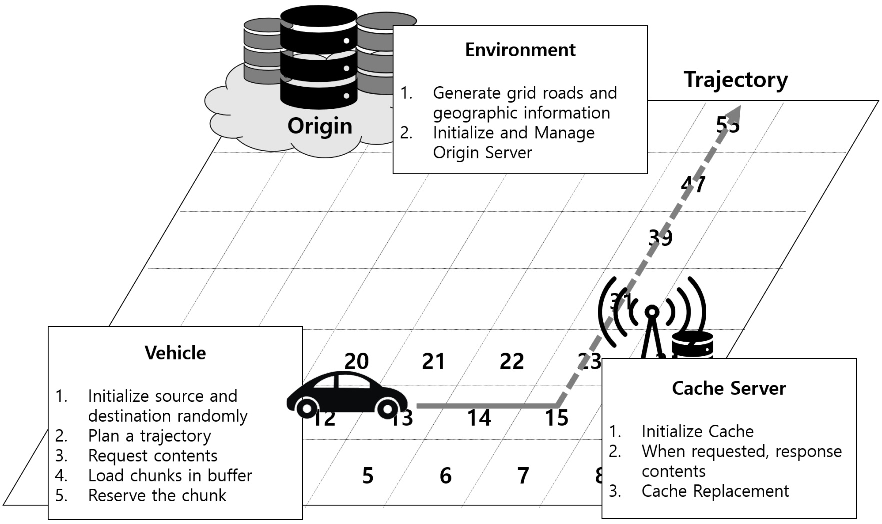

Figure 15: Overview of grid road edge simulator.

methods to vehicular content-centric networks [32]. Although the proposed method judges by reservation weight value, LRF and LFU decide admission and eviction based on requested time and number.

First, we measured the effect of the edge storing capacity. The cache server without replacement stores chunks based on geographic information, i.e., the nearest chunks from the server, and maintains chunks without any change. Capacity of Edge refers to the number of maps stored in each edge among all the maps. If the cache server is assumed to possess 10% of maps, hit ratios of classical methods score 12.21% and 12.08%, as shown in Figure 16. The proposed algorithm outperforms the 16.83% hit ratio when the cache server possesses 10% of the entire contents. The hit ratios of other methods are similar with the cache to total content ratio.

When the edge server starts to serve, storage is empty. It will be filled

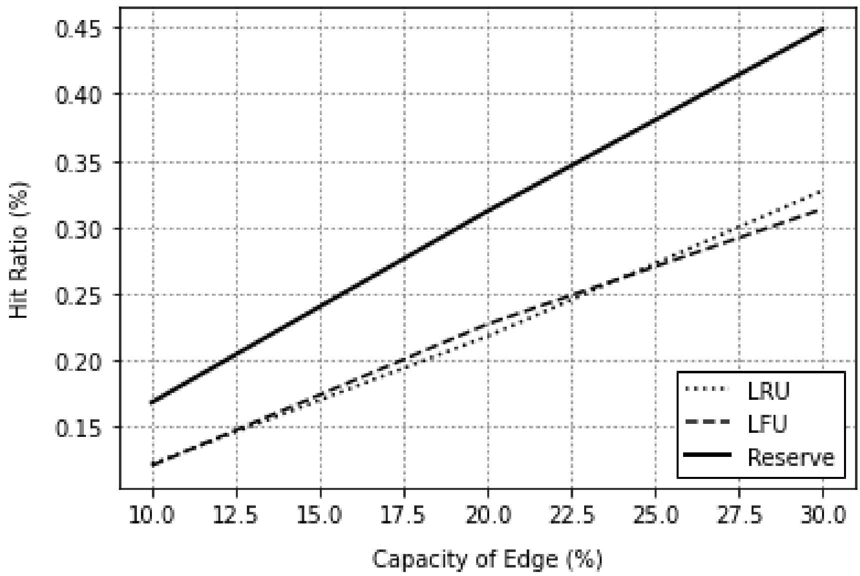

Figure 16: Cache replacement.

with content by the initialization policy. The cache server initiates the contents according to initialization policy such as random and the nearest selection. Geospatial initialization guides the edge server to store the nearest content from the position of the server. Figure 17 shows the result of random initialization and Figure 18 represents the effect of spatial cache initialization to store the nearest chunks. In the case of the proposed method, random initialization is better than spatial initialization . LRU and LFU also achieved higher hit ratio by random initialization. However, the initialization policy did not have a big effect on improving the hit ratio performance of the proposed method.

The vehicle storage is determined based on a trade-off between energy consumption, weights, and cost. According to Figure 19, LRU and LFU are

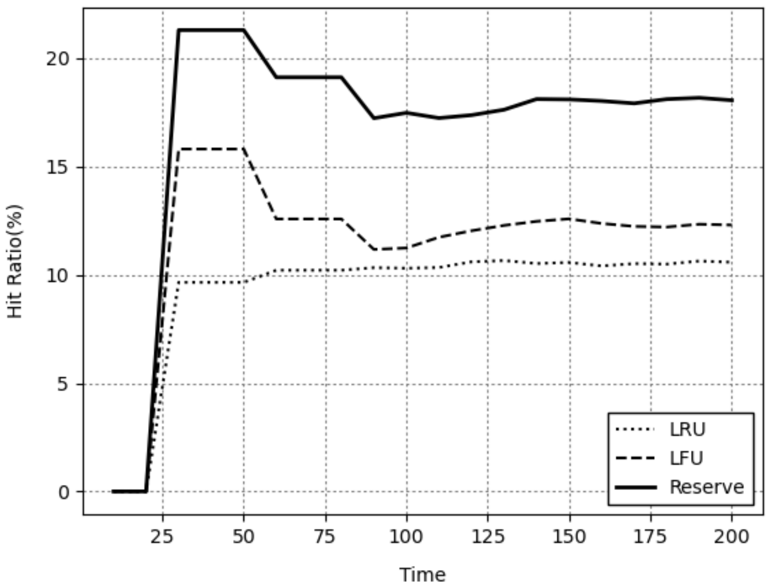

Figure 17: Geospatial cache initialization.

not affected by vehicle buffer size. The greater the vehicle buffer increase, hit ratio of caching by reservation are measured 25.66% in the case of 10 chunks buffer size.

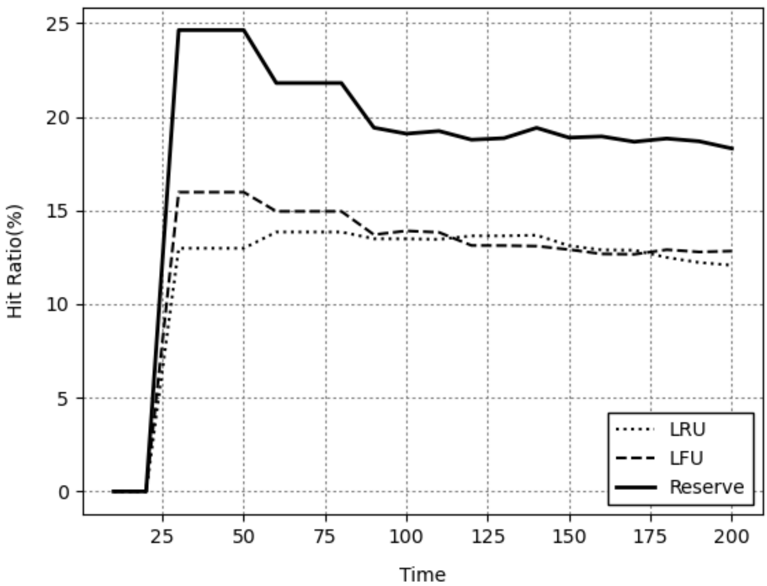

Figure 18: Random cache initialization.

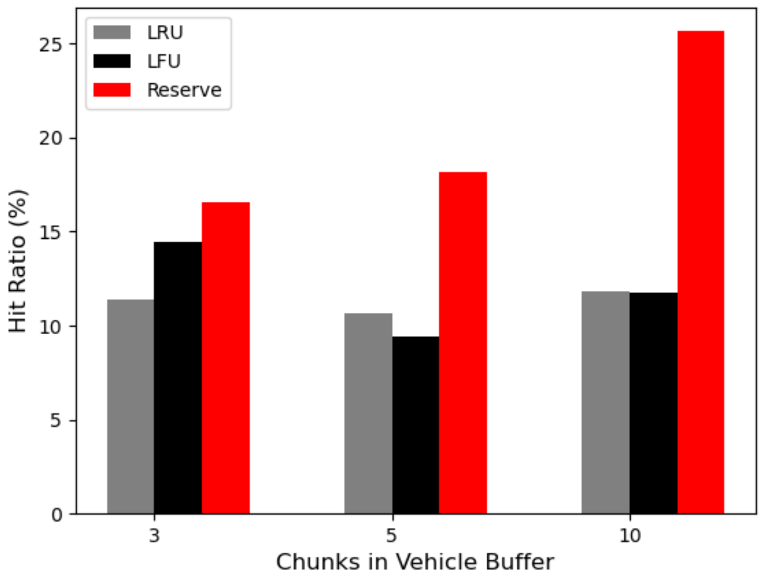

Figure 19: Hit ratio to size of vehicle buffer.

4.2 V2X Characteristics

The dissemination of V2X sidelink communication has the characteristic that it is unnecessary to resolve the host address and route. Consequently, content can be delivered simultaneously to multiple destinations quickly. However, sidelink channel resources are affected by the number of allocated wireless channels and the vehicle density. According to 3GPP TS 23.287, QoS is required to satisfy the specification listed in Table 4.

Resource Type	PQI Value	Packet Delay Budget	Packet Error Rate
GBR	21	20 ms	10^{-4}
	22	50 ms	10^{-2}
	23	100 ms	10^{-4}
Non-GBR	55	10 ms	10^{-4}
	56	20 ms	10^{-1}
	57	25 ms	10^{-1}
	58	100 ms	10^{-2}
	59	500 ms	10^{-1}
Delay Critical GBR	90	10 ms	10^{-4}
	91	3 ms	10^{-5}

Table 4: PQI list of NR Sidelink.

To verify the relationship between PER and Radio environment, we model vehicular communication environments using a network simulator, ns-3 C-V2X [33]. Table 5 lists the parameters of simulation. We evaluate the packet error ratio depending on number of subchannels and vehicles on the road.

Figure 20 shows the results of simulation. The higher the number of vehicles disseminating packets, the greater is the Packet Error Rate increase

parameter	value
Index of MCS	20
size of subchannel	10 MHz
length of Packets	300 Bytes
UE Tx Power	23 dBm

Table 5: Parameter of ns-3 simulation.

over requirement of specification. If the vehicular network allocates more sub-channels for PC5 interface, packet success stays stable in case of 5 and 10 subchannels. In a crowded environment, it is impossible to support delay-critical GBR.

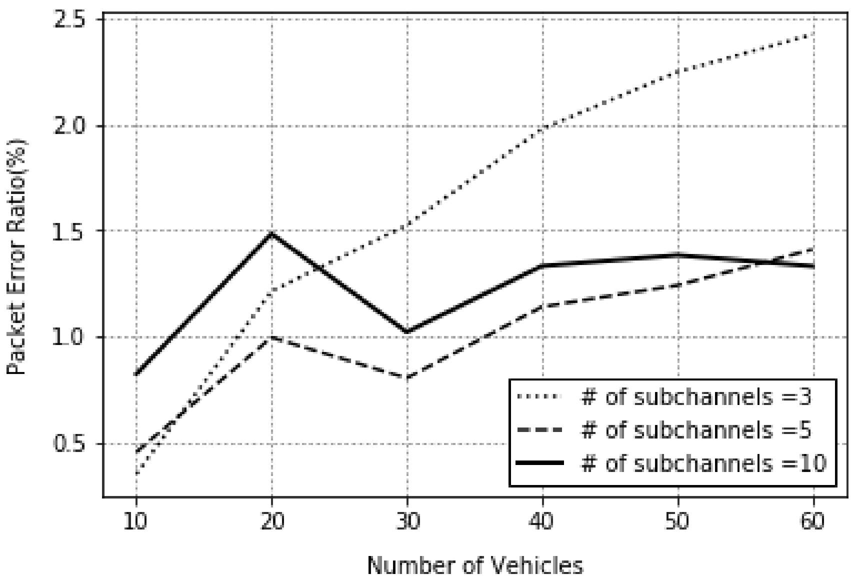

Figure 20: Packet Error Ratio to vehicle popularity by ns-3.

4.3 Edge Performance in Driving on the Road

To prove the efficiency of the edge network in a real road environment, we implemented vehicular client and HD map servers on the core and edge. The vehicular client estimates vehicular dynamics and records trajectory. Table 6 lists the specifications of each device. The vehicular client communicates with core server via LTE network and with edge server via a IEEE 802.11a wireless network. Core Server is located over Internet Service Provider and equipped with logical 72 computing processors and an embedded NoSQL Database, MongoDB, which supports geometric queries.

Device	**Client**	**Edge Server**	**Core Server**
Type	Laptop	SBC	Server
Memory	8GB	4GB	256GB

Table 6: Specification of experimental devices.

Core Server contains entire HD maps and operate a REST API web server process by Flask, which pushes the chunks of maps to the edge server or vehicular client whenever content is requested. Maps are divided into $2\,\text{km} \times 2\,\text{km}$ areas and layered by context including Lane, Link, Nodes, and Surf Signs. The size of each context layer of chunk is approximately 20 MB. Figure 21 shows the example of requested HD map [34].

The experiment is performed in two varied scenarios, which are at downtown and motorway for the purpose of comparison with different road conditions. As shown in Figure 22, the downtown scenario runs through Yeouido business district and Han River and encounters 18 LTE cell tow-

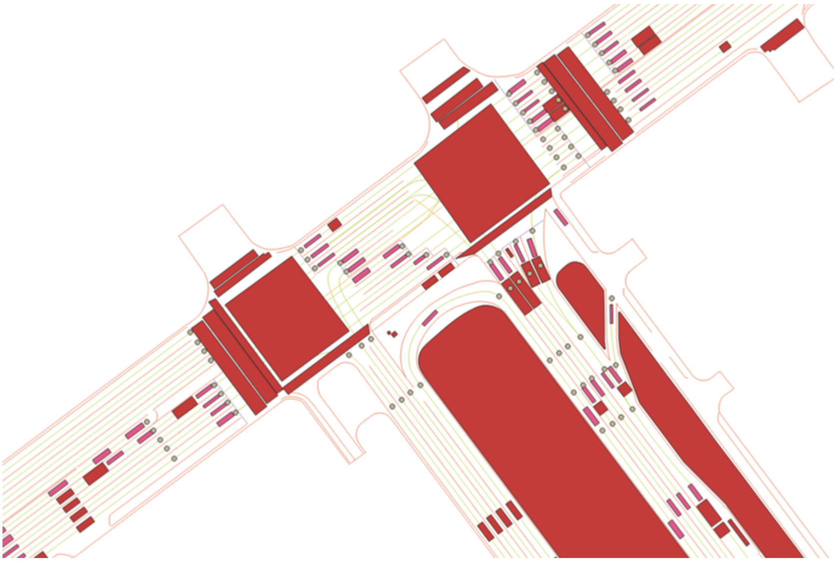

Figure 21: HD map of Yeouido (Latitude : 37.5289, Longitude : 126.9169).

ers, with the total distance being 5.7 km. As shown in Figure 23, motorway scenario runs through Gangbyeonbuk-ro motorway along the northern Han River side. The vehicle encounters 19 LTE cell towers. The total distance is 12.6 km. The Vehicle client equipped with UBLOX 6M GPS sensor samples geometric information every second. The client measures positions and velocities at each waypoint. In downtown scenario, each cell tower serves the vehicle for approximately 35 seconds. In motorway scenario, it takes 22 seconds, which shows that the handover occurs more frequently in the high velocity environment.

Figure 24 represents the amount of change in vehicle movement during a driving scenario. In downtown scenario, the vehicle tends to stop and go alternately, but keeps moving over 40 $\mathrm{km\,h^{-1}}$ velocity in motorway.

Vehicle clients are connected to Core and Edge HD Map server and re-

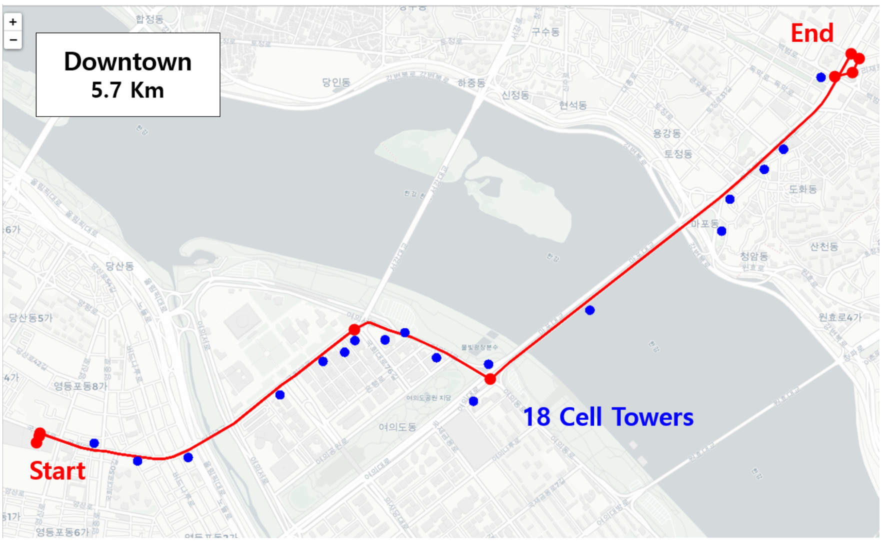

Figure 22: Trajectory of Downtown scenario.

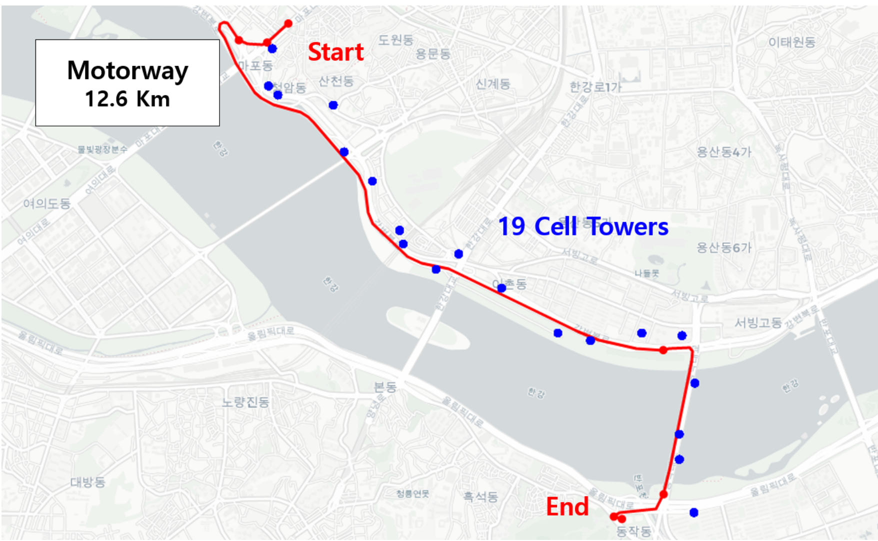

Figure 23: Trajectory of Motorway scenario

quest map content every 10 seconds to measure the performance of content delivery.

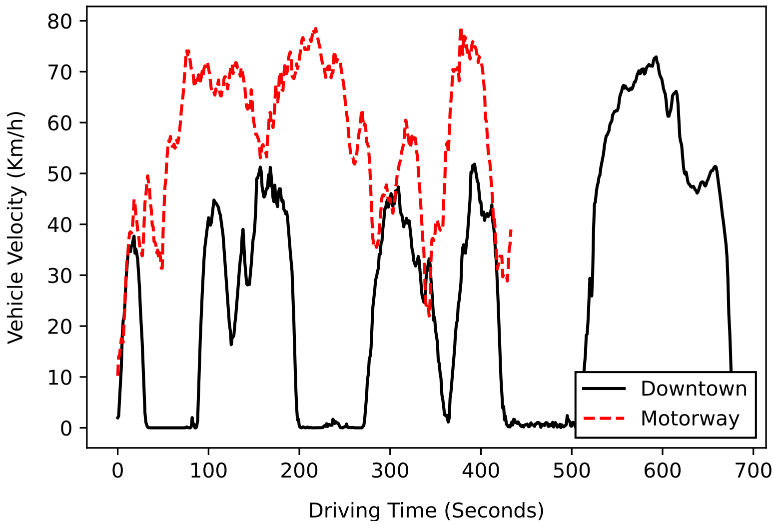

Figure 24: Velocities of trajectories.

$$T_{Elapsed} = T_{Request} + T_{Process} + T_{Response}, \tag{4.1}$$

where the elapsed time to query REST API server consists of request delivery time $T_{Request}$, content processing time on server $T_{Process}$, and response delivery time $T_{Response}$ [35].

When the server receives an HTTP request message, it parses a message and searches requested records in Mongo Database system. The processing time depends on computing resources of the device. The core server with 72 computing processors takes 740 ms, whereas the edge server with 4 computing processors takes 1500 ms to query the database.

As shown in Figure 25, elapsed time to query to core network server through internet network was measured to be an average of 2.48 s in the

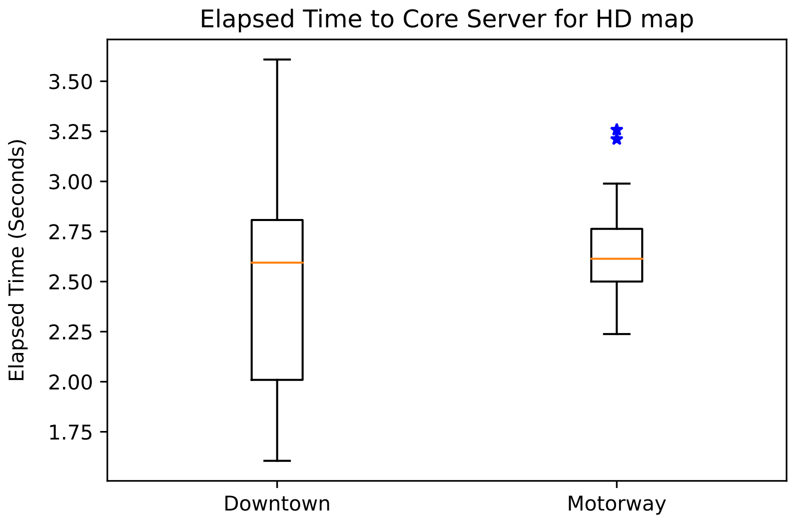

Figure 25: Elapsed Time of Core Server.

downtown scenario and 2.66 s in the motorway scenario. The downtown scenario showed higher variance of time than the motorway scenario. It can be assumed that various variables in downtown, such as massive pedestrians and other vehicles, cause an instability of network performance.

As shown in Figure 26, the round-trip time to edge server is 1.9 s in downtown and 1.89 s in motorway, when there is no significant difference between the two scenarios. Storing content on the edge can significantly shorten the time to request and receive content, while guaranteeing the stable network quality even in variable environments.

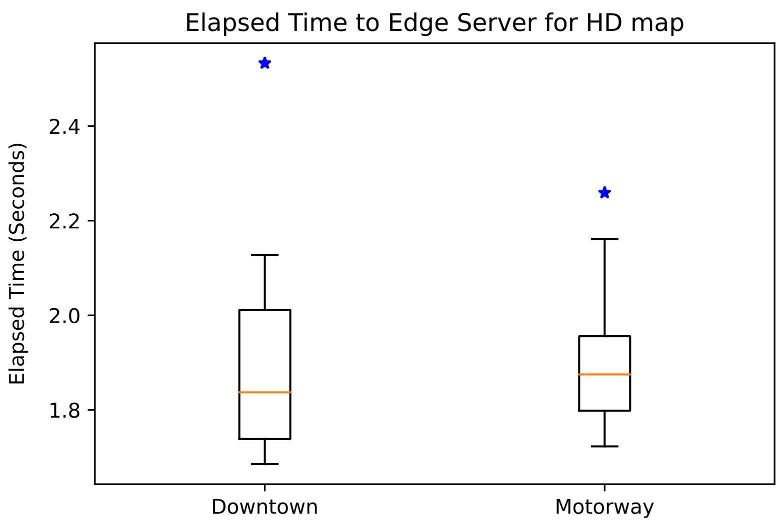

Figure 26: Elapsed Time of Edge Server.

4.4 Edge Performance on 3D Point Clouds Caching for Localization

Localization and detection by 3D point clouds acquired by lidar sensors are widely used in autonomous driving [36, 37]. However, they require a larger storage capacity than other content, such as semantic maps and 2D images. For a simultaneous localization, a vehicle should load entire point clouds on its trajectory and it is difficult to load new content immediately when the trajectory changes unexpectedly. We assume that a vehicle receives point cloud chunks from the edge server and processes the localization task through them.

A unit chunk contains points over a $100\,\mathrm{m} \times 100\,\mathrm{m}$ area and are resolved by $10\,\mathrm{cm} \times 10\,\mathrm{cm}$. The size of the chunk is 40 MB. The edge server contains chunks and responds to the requests of vehicles. Vehicles build an adjacent feature map through received point clouds and estimate the precise position of the vehicle.

As with Section 4.3, the experiment is performed on the urban road with an average speed of approximately $60\,\mathrm{km\,h^{-1}}$. The vehicle requests content from the point cloud to the core and edge server, respectively. Each server delivers content through TCP bytes stream. The vehicle client measures the elapsed time through the network and delivers them to the localization module.

Figure 27 shows that the average elapsed times of the edge and core server are 420 ms and 565 ms, respectively. The localization process returns the result approximately within 100 ms. The delivery through the edge

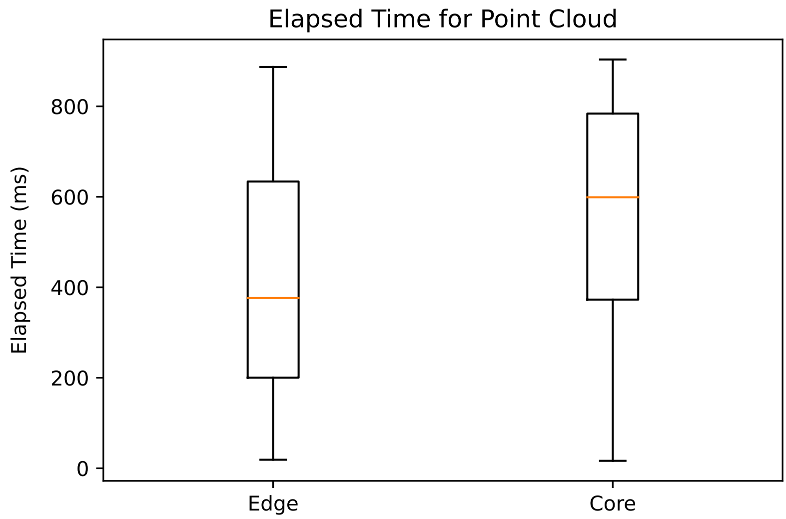

Figure 27: Elapsed Time of Edge and Core Server for Point Cloud.

server is 25.7% faster than through the core server. Therefore, delivery through the edge server can generate a denser feature map and estimate the precise position more accurately, as shown in Figure 28 and Figure 29. As described above, each chunk contains information of 100 m, where it takes 6 s to consume a chunk if a vehicle drives at $60\,\mathrm{km\,h^{-1}}$. This experiment shows that our proposal supports the HD maps and localization requirements for autonomous driving.

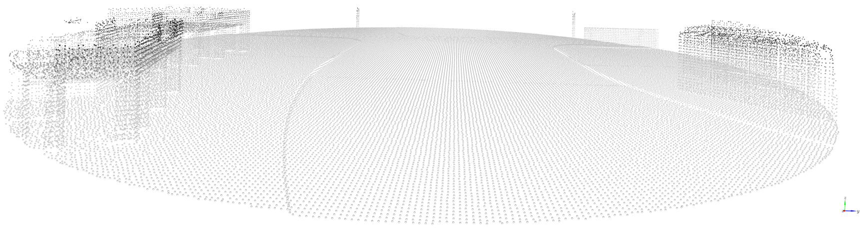

Figure 28: Feature Map through Edge Server.

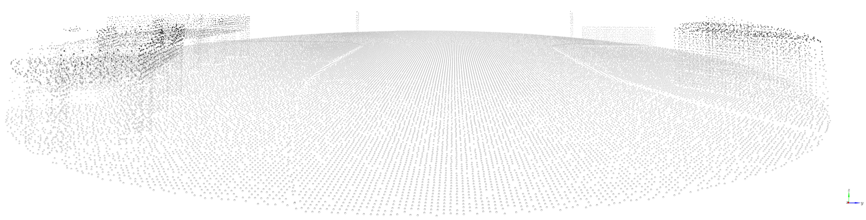

Figure 29: Feature Map through Core Server.

Chapter 5

Conclusion

As autonomous vehicle technology becomes a reality, further evolution of vehicle communication is required to satisfy the requirement of reliability and performance. Compared to conventional communication methods, vehicle communication has different characteristics, which are based on mobility and geospatial features. HD map is the representative example of vehicular content. For HD map delivery of autonomous vehicles, we researched vehicular communications and new emerging communication trends, such as 5G new radio and Content-Centric Network (CCN).

In this paper, we proposed an architecture and a cache replacement algorithm for V2X mobile edge. NR V2X supports sidelink and uplink communication, where the vehicle chooses the appropriate method based on environmental conditions.

We analyzed the newly released NR V2X specification and defined the role and procedure of V2X entities. Unlike conventional caching algorithms, consumers generate a reservation message for hit cache in the future. As the V2X application server orchestrates the reservation information, the edge cache server can pull appropriate candidate chunks with high possibility to hit cache. Edge storage can reduce the time to deliver content and load on the core. In two driving scenarios, content delivery through the edge server guarantees a reliable quality of service.

The proposed method improves the hit ratio performance more than other classic caching algorithms, where 16.33% of requests hit the adjacent cache server in the condition of 20 chunks stored in the cache. The larger the size of cache storage and vehicle buffer increase, the higher the improvement in performance, where the content delivery time can be reduced. We tested the architecture and method in autonomous driving scenarios using real data and experiments and demonstrated that it can contribute to the efficiency of vehicular content delivery via C-V2X.

Bibliography

[1] SAE, "Taxonomy and definitions for terms related to driving automation systems for on-road motor vehicles," *SAE International,(J3016)*, 2018.

[2] S.-W. Kim, W. Liu, M. H. Ang, E. Frazzoli, and D. Rus, "The impact of cooperative perception on decision making and planning of autonomous vehicles," *IEEE Intelligent Transportation Systems Magazine*, vol. 7, no. 3, pp. 39–50, 2015.

[3] S.-W. Kim and W. Liu, "Cooperative autonomous driving: A mirror neuron inspired intention awareness and cooperative perception approach," *IEEE Intelligent Transportation Systems Magazine*, vol. 8, no. 3, pp. 23–32, 2016.

[4] S. Behere and M. Torngren, "A functional architecture for autonomous driving," in *2015 First International Workshop on Automotive Software Architecture (WASA)*, pp. 3–10, IEEE, 2015.

[5] H. Zhou, W. Xu, J. Chen, and W. Wang, "Evolutionary v2x technologies toward the internet of vehicles: Challenges and opportunities," *Proceedings of the IEEE*, vol. 108, no. 2, pp. 308–323, 2020.

[6] S.-W. Kim, B. Qin, Z. J. Chong, X. Shen, W. Liu, M. H. Ang, E. Frazzoli, and D. Rus, "Multivehicle cooperative driving using cooperative perception: Design and experimental validation," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 2, pp. 663–680, 2014.

[7] A. J. Ghandour, M. Di Felice, H. Artail, and L. Bononi, "Dissemination of safety messages in ieee 802.11 p/wave vehicular network: Analytical study and protocol enhancements," *Pervasive and mobile computing*, vol. 11, pp. 3–18, 2014.

[8] F. Bellotti, S.-W. Kim, and F.-L. Lian, "Introduction to the special issue on applications and systems for collaborative driving," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 12, pp. 3457–3460, 2017.

[9] S. Bauer, Y. Alkhorshid, and G. Wanielik, "Using high-definition maps for precise urban vehicle localization," in *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, pp. 492–497, IEEE.

[10] M. Boehme, M. Stang, F. Muetsch, and E. Sax, "Talkycars: A distributed software platform for cooperative perception," in *2020 IEEE Intelligent Vehicles Symposium (IV)*, pp. 701–707, IEEE.

[11] J. B. Kenney, "Dedicated short-range communications (dsrc) standards in the united states," *Proceedings of the IEEE*, vol. 99, no. 7, pp. 1162–1182, 2011.

[12] F. Lyu, H. Zhu, H. Zhou, L. Qian, W. Xu, M. Li, and X. Shen, "Momac: Mobility-aware and collision-avoidance mac for safety applications in vanets," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 11, pp. 10590–10602, 2018.

[13] 3GPP, "3rd generation partnership, http://www.3gpp.org/main.html."

[14] "Study on nr vehicle-to-everything (v2x)," Technical Report (TR) 38.885, 3rd Generation Partnership Project (3GPP), 03 2019. Version 16.0.0.

[15] R. Molina-Masegosa and J. Gozalvez, "Lte-v for sidelink 5g v2x vehicular communications: A new 5g technology for short-range vehicle-to-everything communications," *IEEE Vehicular Technology Magazine*, vol. 12, no. 4, pp. 30–39, 2017.

[16] "System architecture for the 5g system (5gs)," Technical Specification (TS) 23.501, 3rd Generation Partnership Project (3GPP), 04 2019. Version 16.0.0.

[17] H. Butler, M. Daly, A. Doyle, S. Gillies, T. Schaub, and T. Schaub, “The GeoJSON Format.” RFC 7946, Aug. 2016.

[18] D. Guo and E. Onstein, “State-of-the-art geospatial information processing in nosql databases,” *ISPRS International Journal of Geo-Information*, vol. 9, no. 5, p. 331, 2020.

[19] ETSI, “302 895 (v1. 1.0): Intelligent transport systems-extension of map database specifications for local dynamic map for applications of cooperative its,” report, Technical report, ETSI, 2014.

[20] W.-S. Hur, S.-T. Choi, S.-W. Kim, and S.-W. Seo, “Precise free space detection and its application to background extraction,” in *2015 IEEE 7th International Conference on Cybernetics and Intelligent Systems (CIS) and IEEE Conference on Robotics, Automation and Mechatronics (RAM)*, 2015.

[21] S.-W. Kim, G.-P. Gwon, W.-S. Hur, D. Hyeon, D.-Y. Kim, S.-H. Kim, D.-K. Kye, S.-H. Lee, S. Lee, M.-O. Shin, *et al.*, “Autonomous campus mobility services using driverless taxi,” *IEEE Transactions on intelligent transportation systems*, vol. 18, no. 12, pp. 3513–3526, 2017.

[22] F. Ghallabi, F. Nashashibi, G. El-Haj-Shhade, and M.-A. Mittet, “Lidar-based lane marking detection for vehicle positioning in an hd map,” in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pp. 2209–2214, IEEE.

[23] F. Ghallabi, G. El-Haj-Shhade, M.-A. Mittet, and F. Nashashibi, “Lidar-based road signs detection for vehicle localization in an hd map,” in *2019 IEEE Intelligent Vehicles Symposium (IV)*, pp. 1484–1490, IEEE.

[24] W. Chun, T.-H. Lee, and T. Choi, “Yanail: yet another definition on names, addresses, identifiers, and locators,” in *Proceedings of the 6th International Conference on Future Internet Technologies*, pp. 8–12.

[25] S. H. Bouk, S. H. Ahmed, and D. Kim, "Vehicular content centric network (vccn) a survey and research challenges," in *Proceedings of the 30th annual ACM symposium on applied computing*, pp. 695–700.

[26] M. A. Yaquba, S. H. Ahmedb, S. H. Boukc, and D. Kimd, "Interest forwarding in vehicular information centric networks: A survey," 2016.

[27] H. Wu, W. Xu, J. Chen, L. Wang, and X. Shen, "Matching-based content caching in heterogeneous vehicular networks," in *2018 IEEE Global Communications Conference (GLOBECOM)*, pp. 1–6, IEEE.

[28] X. Wang and X. Wang, "Vehicular content-centric networking framework," *IEEE Systems Journal*, vol. 13, no. 1, pp. 519–529, 2018.

[29] Y. Zhang, C. Li, T. H. Luan, Y. Fu, W. Shi, and L. Zhu, "A mobility-aware vehicular caching scheme in content centric networks: Model and optimization," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 4, pp. 3100–3112, 2019.

[30] "Mobile edge computing (mec);framework and reference architecture," Group Specification (GS) GS MEC 003, European Telecommunications Standards Institute (ETSI), 03 2016. Version 1.1.1.

[31] A. I. Pérez-Neira and M. R. Campalans, "Chapter 2 - different views of spectral efficiency**this chapter is co-authored by dr. stephen pfletschinger.," in *Cross-Layer Resource Allocation in Wireless Communications* (A. I. Pérez-Neira and M. R. Campalans, eds.), pp. 13 – 33, Oxford: Academic Press, 2009.

[32] C. Chen, C. Wang, T. Qiu, M. Atiquzzaman, and D. O. Wu, "Caching in vehicular named data networking: architecture, schemes and future directions," *IEEE Communications Surveys Tutorials*, 2020.

[33] F. Eckermann, M. Kahlert, and C. Wietfeld, "Performance analysis of C-V2X mode 4 communication introducing an open-source C-V2X simulator," in *2019 IEEE 90th Vehicular Technology Conference (VTC-Fall)*, (Honolulu, Hawaii, USA), September 2019.

[34] "Naver labs dataset, https://hdmap.naverlabs.com."

[35] J. Touch, J. Heidemann, and K. Obraczka, "Analysis of http performance," *URL: http://www. isi. edu/touch/pubs/http-perf96/¿, ISI Research Report ISI/RR-98-463,(original report dated Aug. 1996), USC/Information Sciences Institute*, 1998.

[36] A. Y. Hata and D. F. Wolf, "Feature detection for vehicle localization in urban environments using a multilayer lidar," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 2, pp. 420–429, 2015.

[37] Y. Jung, S.-W. Seo, and S.-W. Kim, "Curb detection and tracking in low-resolution 3d point clouds based on optimization framework," *IEEE Transactions on Intelligent Transportation Systems*, 2019.

Abstract

Design of V2X-based Vehicular Contents Delivery Network for Autonomous Driving

Sanghoon Lee
Graduate School of Practical Engineering
Seoul National University

최근들어 기술의 혁신은 자율주행 자동차의 발전을 가속화 하고 있다. 보다 높은 수준의 자율 주행을 구현하기 위해서, 차량은 네트워크를 통해 서로 연결되어 있어야 하고 차량의 안전과 편의성을 향상 시킬 수 있도록 정보를 공유 할 수 있어야 한다. 표준화 단체인 IEEE와 3GPP는 차량 통신 요구사항, 아키텍처를 연구하고 개정해왔다. IEEE가 전용 채널을 통한 근접 지역 통신에 초점을 맞추는 반면에, 3GPP의 New Radio V2X는 Sidelink 뿐만 아니라 Uplink 통신을 동시에 지원한다. 5G 통신을 지원하는 3GPP Release 16은 Network Slice, NFV, SDN과 같은 새로운 통신 기능들을 제공한다. 이 연구에서는 새롭게 정의된 5G Core Network Architecture를 바탕으로 차량 네트워크를 정의하고 설계하였다. 자율주행 자동차의 측위를 위해서, 고해상도 지도는 각 구성요소들의 의미와 속성을 자세하게 포함하고 있어야 한다. 우리는 이 연구에서 V2X 네트워크 상에 HD map을 중계할 수 있는 Edge Server를 제안 함으로써, 중앙에서 발생할 수 있는 병목현상을 줄이고 전송 Delay를 최소화한다. 또한 Edge의 컨텐츠를

등록하고 삭제하는 정책으로 기존의 LRU, LFU가 아닌 새로운 컨텐츠 교체 알고리즘을 제안하였다. 실제 주행 시험과 시뮬레이션을 통한 실험을 통해 전송 품질을 향상시켰으며, Edge 컨텐츠의 활용도를 높였다.

Keywords : Autonomous driving, C-V2X, HD Map, Vehicular Content Centric Networks.

Student Number : 2019-25271