Vehicular Traffic Event Transmission Method Using IPv4 Option Field in V2I Local Networks

HyeonMuk Park
Information & Communication Engineering
Chungbuk National University
Chungju 28644, Republic of Korea
qga0809@chungbuk.ac.kr

WonSeok Choi
Information & Communication Engineering
Chungbuk National University
Chungju 28644, Republic of Korea
choiws@cbnu.ac.kr

SeongGon Choi
Information & Communication Engineering
Chungbuk National University
Chungju 28644, Republic of Korea
choisg@cbnu.ac.kr

***Abstract*— We propose a vehicular traffic event transmission method for Vehicle-to-Infrastructure (V2I) local networks, using a fixed 20 bytes IPv4 option field to carry vehicular traffic event information such as car accidents, vehicular traffic congestion, road construction, and emergency situations. The motivation is to address critical limitations in current V2I communication systems, which suffer from high processing latency due to complex payload parsing, dependency on centralized servers that create bottlenecks, and delays in safety critical message transmission that can compromise road safety. In the proposed protocol, vehicles embed vehicular traffic event information directly into IPv4 option fields and transmit packets through roadside infrastructure. Intermediate nodes forward these packets based solely on header inspection without payload processing or server communication, enabling direct event transmission across the network. The protocol was implemented and tested in a laboratory environment, and its end-to-end operation was successfully verified.**

Keywords—LAN, Event Transmission, IPv4 Option Field, V2I

I. INTRODUCTION

Rapid urbanization and the exponential growth in the number of vehicles have significantly intensified vehicular traffic congestion and increased the frequency of accidents in modern cities [1]. Addressing these challenges, Vehicle-to-Infrastructure (V2I) communication–driven Intelligent Transportation Systems (ITS) have gained attention as an effective approach, facilitating the exchange of real-time data between vehicles and roadside units [2].

In conventional V2I deployments, the forwarding of real-time vehicular traffic data is typically achieved through payload-oriented message formats, including the Basic Safety Message (BSM), Cooperative Awareness Message (CAM), Decentralized Environmental Notification Message (DENM), and Signal Request Message (SRM) [5][6][7]. In these standardized message formats, the entire meaning is contained within the payload. Consequently, intermediate devices such as RSUs and gateways must perform a full payload inspection to interpret the message. Additionally, the processed information is typically forwarded to cloud-hosted back-end servers for further analysis before any action can be taken [3]. During emergencies, this server side processing path can experience link congestion and additional hop delays [4], which reduces the timeliness of warnings and may degrade overall vehicular traffic-management reliability. Previous event transmission approaches like DV-CAST and REMBP have attempted to address these issues but suffer from substantial processing overhead, deployment complexity, and delays in dense vehicular traffic conditions [2][8], highlighting the need for more efficient V2I communication methods.

To overcome these issues, this paper proposes a V2I traffic event transmission method that supports efficient local transmission of vehicular traffic events such as car accidents, vehicular traffic congestion, road construction, and emergency situations. The system inserts event information directly into the IPv4 option field of the packet header, keeping the payload intact. A fixed 20-byte option structure holds fields such as Type, Severity, Timestamp, Location Data, Node ID, and others, while remaining within the IPv4 option-space limit.

This paper is organized as follows. Section II reviews related work, beginning with an overview of current V2I communication and message formats, and then summarizing prior research on event transmission in V2I networks. Section III introduces the proposed IPv4-based vehicular traffic event transmission method, describing the system components, overall architecture, and the IPv4 option-field structure that carries the event data. Section IV presents the implementation details and laboratory test environment. Section V concludes the paper and outlines directions for further study.

II. RELATED WORK

To clearly position the proposed method within the academic and technical landscape, this section outlines existing message formats used in V2I communication and analyzes the limitations of prior studies and protocols designed for event transmission.

A. V2I Communication and Message Formats

In current Vehicle-to-Infrastructure (V2I) communications, four standardized message formats are commonly used, each embedding its meanings in the payload. The Basic Safety Message (BSM) specified in SAE J2735 broadcasts a vehicle's position, speed, and heading ten times per second and typically carries 300–400 bytes payload. In U.S. DSRC deployments, every received BSM is validated through the ITS-AID security framework to confirm authenticity and sender legitimacy [5]. Europe uses the Cooperative Awareness Message (CAM) defined by ETSI ITS standards. CAM augments status data such as position, speed and acceleration with descriptors of vehicle size, type, and role, and the declared role governs a station's transmission rights [6]. Event reporting relies on the Decentralized Environmental Notification Message (DENM), which advertises accidents, congestion, or roadworks for the duration of the event and must be fully parsed at the payload level for correct interpretation [7]. Finally, the Signal Request Message (SRM) enables emergency or public-transport vehicles to request signal-priority service at intersections, following the detailed authorization and validation procedures defined in ETSI TS 103 301 [6].

Each message format serves a different purpose but shares two structural characteristics that introduce latency. First, because the entire meaning of the message is contained in the payload, devices such as RSUs, gateways, and cloud servers must inspect the full payload rather than relying on a few header bytes. Second, after this inspection, the information is typically forwarded to a cloud server for further processing, and the result must return before any action can be taken. Previous studies indicate that these two steps, namely payload inspection at intermediate nodes and round-trip processing through the server path, increase delay under dense vehicular traffic conditions and can weaken the real time responsiveness required by safety-critical systems [3][4].

B. Prior Research on Event Transmission in V2I

Several studies have explored broadcasting incidents and congestion in vehicular networks, yet each approach exposes practical limitations. Distributed Vehicular Broadcast (DV-CAST), designed for vehicle-to-vehicle (V2V) communication, relies on one-hop neighbor topology information and uses a store-carry-forward strategy to bridge network partitions. Periodic recalculation of the local topology and continuous link-connectivity checks introduce substantial overhead, so message delivery can be delayed in dense vehicular traffic conditions [8]. Reliable Emergency Message Broadcast Protocol (REMBP) operates in a vehicle-to-infrastructure (V2I) setting with an RSU-centric design in which the RSU receives, processes, and rebroadcasts emergency messages. Because it depends on a hybrid RF/VLC communication stack and requires additional processing functions at the RSU, deploying REMBP on existing infrastructure is challenging. Hong et al. proposed a conceptual V2I framework that forwards event data from vehicles to a gateway and then to neighboring RSUs [2], but the study does not specify the concrete event-message format or where it is placed in the packet.

Therefore, to address the payload inspection and server round-trip delays inherent in existing V2I message formats, and to overcome the deployment complexity and processing overhead limitations of prior event transmission protocols, we propose a V2I traffic event transmission method that places event information in a fixed 20-byte IPv4 option field.

III. PROPOSED METHOD

This section presents a vehicular traffic event transmission method based on the IPv4 option field, designed to overcome the limitations of existing V2I communication protocols. The proposed approach fundamentally shifts from payload-centric message processing to header-based event identification, enabling intermediate network devices to make forwarding decisions without deep packet inspection. Unlike conventional V2I systems that require complex application-layer parsing and centralized server coordination, our method leverages the underutilized IPv4 option space to embed critical vehicular traffic event information directly within the packet header structure. The method maintains full compatibility with existing IP infrastructure while providing the low-latency, distributed communication capabilities essential for safety-critical vehicular applications.

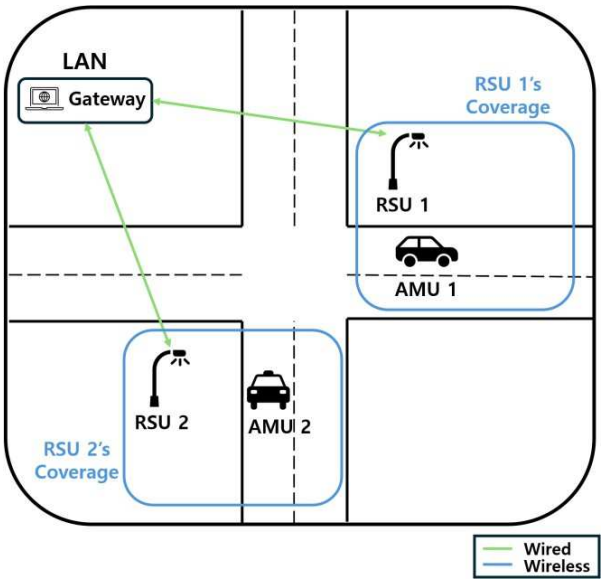

Fig. 1. System environment

The proposed system comprises three main components. The first, Autonomous Moving Units (AMUs), refers to vehicles equipped with communication capabilities that can detect vehicular traffic incidents such as accidents, congestion, and roadworks and transmit event information to the infrastructure. Roadside Units (RSUs) serve as access points that provide connectivity between vehicles and the network infrastructure within their coverage areas. The Gateway acts as a central hub that connects multiple RSUs and manages vehicular traffic event transmission across the entire network.

AMUs communicate with RSUs via wireless links, while RSUs connect to the gateway through a wired network infrastructure composed of standard routers and switches. This hierarchical structure enables efficient message transmission from vehicles to the broader network infrastructure.

To enable AMUs to transmit vehicular traffic events to the gateway, each RSU periodically broadcasts beacon messages containing the gateway's IP address. When an AMU enters an RSU's coverage area, it receives these beacon transmissions and learns the gateway address, allowing it to establish connectivity with the infrastructure network. This beacon-based discovery mechanism ensures that AMUs can immediately participate in event message distribution upon entering any RSU's service area without requiring manual configuration or complex discovery protocols.

In the proposed method, when an AMU detects incidents, it writes the event data into 20 bytes IPv4 option field and sends the packet to its serving RSU. Any AMU that later receives such a packet can react immediately, for example by recalculating its route, slowing down, or activating hazard lights.

The RSU forwards packets from AMUs to the gateway unchanged at Layer 2 and, in the opposite direction, rebroadcasts the gateway's packets over the air to nearby AMUs. The gateway checks only the predefined option code in the IPv4 header to verify that an incoming packet carries an event message. When a packet is identified, the gateway broadcasts it to all attached RSUs without examining the

payload. Standard routers and switches form the wired network and are configured to pass IPv4 option fields without modification. Figure 1 illustrates this environment, showing the roles of AMUs, RSUs, and the gateway, together with the green wired links and blue wireless connections that interconnect them.

A. Event Message Structure

Figure 2 depicts the layout of the proposed IPv4 option, showing how the 20 bytes event field is positioned immediately after the Layer-3 header and before the payload. This layout allows intermediate nodes to parse and classify events solely by inspecting the IP header, thereby minimizing processing latency and enabling header-only identification without payload parsing.

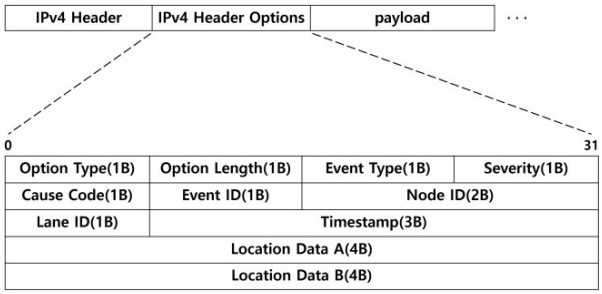

Fig. 2. Packet format

TABLE I. EVENT MESSAGE FIELDS

Field	*Size(Byte)*	*Description*
Option Type	1	Newly defined V2I event option type
Option Length	1	Total option length
Event Type	1	Event category identifier
Severity	1	Event severity level
Cause Code	1	Cause of the event
Event Code	1	Deduplication / identification code
Node ID	2	Sender identifier
Lane ID	1	Lane number where the event occurred
Timestamp	3	Event generation time
Location A	4	Source position A
Location B	4	Source position B

As detailed in Table I, the proposed event message is confined to a 20-byte option field, remaining within the IPv4 standard range. Figure 2 illustrates how each field is mapped into the option block at the byte level. Table I lists the byte-level definition of the 20-byte IPv4 option assigned to V2I event messages. The first two bytes follow IPv4 rules, indicating the option type selected for V2I events and the total option length. They are followed by four single-byte fields that include Event Type, Severity, Cause Code, and Event Code. These fields collectively describe what occurred, how serious it is, why it happened, and provide a small identifier for fast de-duplication. Two bytes Node ID then specifies the sender, after which a one byte Lane ID points to the exact lane in multi-lane road segments. The remaining bytes supply temporal and spatial context. Three bytes Timestamp marks the generation time with millisecond precision, and two four-bytes coordinates for Location A and Location A capture the vehicle's position in a fixed-point format. For example, Location A and Location B can represent the x-axis and y-axis coordinates in two-dimensional coordinate system, respectively. The total is exactly 20 bytes, allowing the option to fit within a single IPv4 header extension so that intermediate devices can classify events by inspecting the header alone, without touching the payload.

IV. IMPLEMENTATION AND TEST ENVIRONMENT

To evaluate the feasibility of the proposed IPv4 option-based event transmission scheme, we conducted practical network level experiments using commercial off-the-shelf devices. This section describes the experimental setup and validation results.

A. Testbed Configuration

The laboratory testbed was organized to mirror a compact urban V2I path of AMU → RSU → Gateway → RSU → AMU.

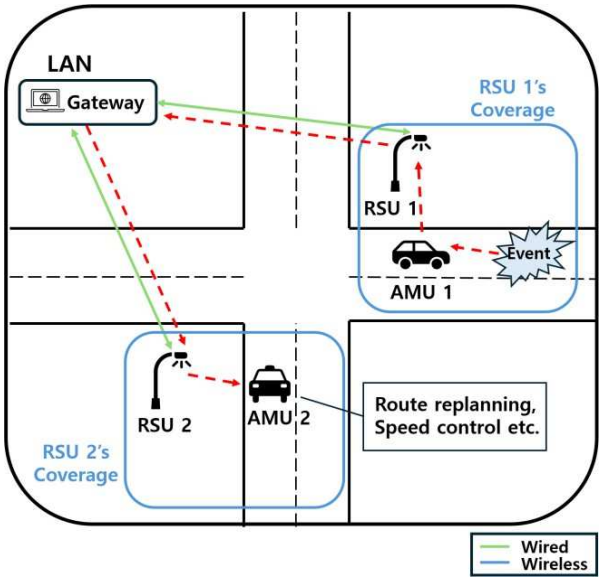

Fig. 3. System operation upon event occurrence

Fig. 3 illustrates an operating scenario that demonstrates the complete end-to-end vehicular traffic event transmission process in the proposed system. AMU 1 detects an incident inside RSU 1's coverage zone and immediately generates an IPv4 packet with the event information embedded in the 20-byte option field. The packet is transmitted wirelessly to RSU 1, which forwards it through the wired infrastructure to the gateway. Upon receiving the packet, the gateway performs header-only inspection to identify the vehicular traffic event and initiates a broadcast transmission. The gateway forwards the packet over the green wired network to all connected RSUs, including RSU 2, which then retransmits the event information over blue wireless links to all devices within its coverage area. AMU 2, located inside RSU 2's coverage zone, receives the forwarded alert and can immediately respond to the vehicular traffic event, completing the distributed

communication cycle without requiring centralized server processing.

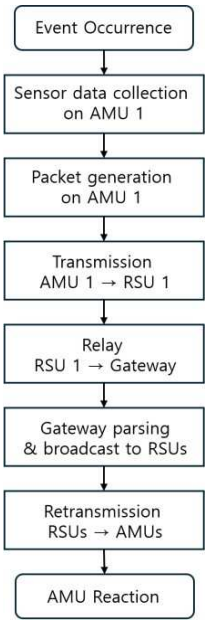

Fig. 4. End-to-end workflow.

Fig. 4 illustrates the operational flow of the proposed transmission system in seven logical steps. The process begins when an event occurs, followed by sensor data collection and packet generation by AMU 1. The packet is then transmitted to RSU 1, which relays it to the gateway. Upon receiving the message, the gateway processes the message and broadcasts it to all connected RSUs. Each RSU subsequently retransmits the message over the air to nearby AMUs. Finally, AMUs receiving the message take appropriate actions based on the event information. This flow demonstrates that event alerts can be transmitted across the network without modifying the payload at intermediate nodes.

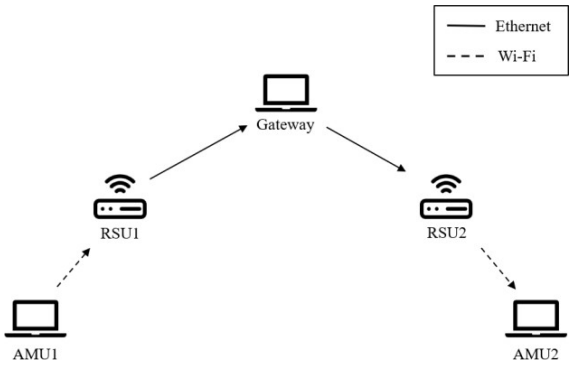

Fig. 5. Implementation of laboratory testbed.

Figure 5 depicts the laboratory testbed that implements the logical architecture of Figure 3. A commercial laptop serving as AMU 1 uses the Scapy packet-crafting library, which is a Python framework for creating and sending custom packets, to generate IPv4 packets containing a 20-byte custom option block. These packets are sent over Wi-Fi to RSU 1. Both RSU 1 and RSU 2 are off-the-shelf consumer Wi-Fi routers configured in Access Point mode, so they forward Ethernet frames that include the IPv4 packets with custom options without inspection or modification.

A commercial laptop serving as AMU 1 generates IPv4 packets with manually configured test data in the 20-byte option field and transmits them to RSU 1. A desktop PC acting as the gateway then receives the unmodified frame, parses only the IPv4 option code, and broadcasts the packet by setting the Ethernet destination MAC to the Layer 2 broadcast address FF:FF:FF:FF:FF:FF. The switch receives this broadcast frame and floods it to all connected RSUs including RSU 2. RSU 2 then floods the frame over Wi-Fi to all devices in its coverage area, including nearby AMUs and neighboring RSUs. Finally, AMU 2 using another commercial laptop receives the incoming frame and, using Wireshark, verifies byte-by-byte that the 20-byte option block remains intact end-to-end.

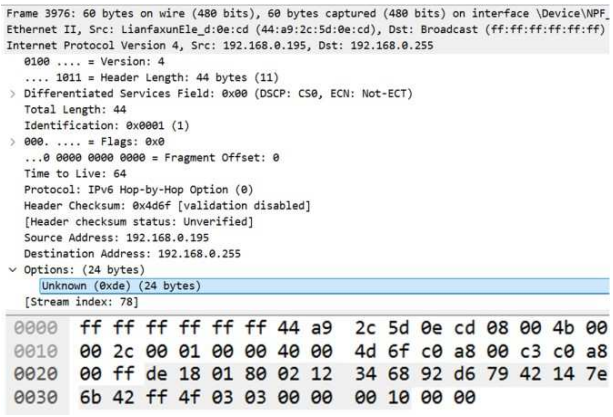

Fig. 6. Wireshark view confirming event message delivery

Figure 6 shows the packet captured by AMU 2 after the broadcast transmission. The Internet Header Length remains at 11, and Wireshark reports Options (24 bytes) because the 20-byte custom option block is preceded by a one-byte type and one-byte length field and padded with two bytes to meet the 4-byte alignment requirement. Despite traversing two RSUs and rebroadcasting by the gateway, no IP option fields were altered and the full custom option message was preserved end-to-end.

B. Testbed Components and Specifications

TABLE II. HARDWARE SUMMARY OF THE TESTBED

Role	*Device*	*Key Specs*	*OS/Firmware*
AMU 1	LG Gram 14Z90Q	Intel i5, Wi-Fi 6	Ubuntu 20.04 LTS
AMU 2	LG Gram 14Z90Q	Intel i5, Wi-Fi 6	Windows 11
RSU 1	ipTIME A1004NS	802.11ac Wi-Fi 5, 4×1 GbE	Firmware v11.14
RSU 2	ipTIME A1004NS	802.11ac Wi-Fi 5, 4×1 GbE	Firmware v11.14
Gateway	Desktop PC	Intel Core i7-12700, 16 GB RAM	Ubuntu 20.04 LTS
Switch	ipTIME 8005	Unmanaged Layer-2, 5 × 1 Gbps ports	

Table II lists the off-the-shelf hardware used in the testbed. Two LG Gram laptops act as AMUs, while a desktop PC with an Intel i7-12700 serves as the gateway. Commercial Wi-Fi 5

routers (ipTIME A1004NS) function as RSUs and connect to the gateway through an unmanaged 1 GbE switch, demonstrating that the proposed scheme operates on standard consumer equipment without modification.

V. Conclusion

This paper proposed a vehicular traffic event transmission scheme based on the IPv4 option field for V2I communication. Event data is embedded in fixed 20 bytes option field, allowing message transmission without relying on the payload. To examine feasibility, a test environment was constructed using laptops, consumer routers, and a desktop gateway. Customized IPv4 packets containing user-defined option fields were transmitted across the network. Packet analysis at the receiver confirmed that the option field remained unchanged through intermediate nodes. These results indicate that the proposed method is structurally compatible with standard IP-based network equipment.

Further studies will evaluate the performance of the scheme under varying network conditions, including mobility and congestion. Additional investigation will address potential extensions such as support for IPv6 extension headers and integrity protection of the option field.

Acknowledgment

This work was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (No. RS-2020-NR049604, 100%).

*Corresponding author is S.G. Choi(choisg@cbnu.ac.kr).

References

[1] W. Albattah, S. Habib, and M. F. Alsharekh, "An Overview of the Current Challenges, Trends, and Protocols in the Field of Vehicular Communication," Electronics, vol. 11, pp. 3509

[2] M. H. Hong, W. S. Choi, and S. G. Choi, "Event Information Delivery Method Using Location Information for Route Replanning in V2I Local Environment," in Proc. 27th Int. Conf. on Advanced Communications Technology (ICACT), 2025.

[3] V. Milanes, J. Villagra, and J. Godoy, "An Intelligent V2I-Based Traffic Management System," IEEE Trans. on Intelligent Transportation Systems, vol. 13, no. 1, pp. 49-58, Mar. 2012.

[4] M. Gupta, J. Benson, and F. Patwa, "Secure V2V and V2I Communication in Intelligent Transportation Using Cloudlets," IEEE Trans. on Services Computing, vol. 15, no. 4, pp. 2312-2325, Aug. 2022.

[5] J. Li, T. Han, W. Guan, and X. Lian, “A preemptive-resume priority MAC protocol for efficient BSM transmission in UAV-assisted VANETs,” *Appl. Sci.*, vol. 14, no. 5, Art. 2151, Mar. 2024. doi: 10.3390/app14052151.

[6] ETSI TS 103 301 V2.2.1, "Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Facilities layer protocols and communication requirements for infrastructure services," Aug. 2024.

[7] Z. Lokaj et al., “Automated evaluation of C-ITS message content for enhanced compliance and reliability,” *Appl. Sci.*, vol. 14, no. 20, Art.

[8] J. K. Ng, S. Song, C. Li, and W. Zhang, "Optimal Path Routing Protocol for Warning Messages Dissemination in VANET," Sensors, vol. 22, no. 18, pp. 6839, 2022.

[9] F. Gont and R. Atkinson, "Recommendations on Filtering of IPv4 Packets Containing IPv4 Options," RFC 7126, Feb.