Communications and Network, 2026, 18(3), 45-58
https://www.scirp.org/journal/cn
ISSN Online: 1947-3826
ISSN Print: 1949-2421

V2X Communication Reliability Improvement through QoS Prediction and Reservation Using Network Virtualization Technology

Tensei Kunimoto, Kenya Sato

Department of Computer and Information Science, Graduate School of Science and Engineering, Doshisha University, Kyoto, Japan
Email: tensei.kunimoto@nislab.doshisha.ac.jp

How to cite this paper: Kunimoto, T. and Sato, K. (2026) V2X Communication Reliability Improvement through QoS Prediction and Reservation Using Network Virtualization Technology. *Communications and Network*, **18**, 45-58.
https://doi.org/10.4236/cn.2026.183004

Received: May 29, 2026
Accepted: August 27, 2026
Published: August 30, 2026

Abstract

Due to the limited sensing range of vehicles, cooperative autonomous driving is being extensively researched to improve traffic safety and efficiency. This approach relies on sharing sensory data captured by surrounding vehicles and roadside units via wireless communication. By aggregating this data on a centralized server and delivering integrated environment models back to the vehicles, drivers and automated systems can perceive obstacles beyond the line-of-sight of their onboard sensors. However, if the number of vehicles within the server's management area exceeds the available communication capacity, network congestion occurs, making it difficult to guarantee the Quality of Service (QoS). Furthermore, vehicles currently lack a mechanism to determine in advance whether their required QoS will be sustained. This study proposes a novel framework that leverages network virtualization technology to centrally manage the network via software, anticipate vehicle density, and predict QoS proactively. In addition, we introduce a QoS reservation mechanism to guarantee future network performance for individual vehicles. Network simulations demonstrate that the proposed method effectively prevents communication latency from degrading for vehicles with reserved QoS, even under heavily congested network conditions.

Keywords

Cooperative Automated Driving, QoS, Network Virtualization

1. Introduction

1.1. Background

To realize a safe and efficient mobility society, research on automated driving sys-

tems—where vehicles autonomously execute driving maneuvers—has been actively conducted [1]. Recent advances in onboard sensors have enabled vehicles to perceive their immediate driving environments in high detail. Consequently, extensive efforts are underway to detect surrounding vehicles, pedestrians, and obstacles in advance to mitigate hazards by automatically decelerating or stopping.

However, the perception range remains limited when relying solely on ego-vehicle sensors; for instance, pedestrians hidden behind buildings cannot be detected. This limitation makes it challenging to handle critical scenarios, such as a pedestrian suddenly rushing out at an intersection with poor visibility. To address these issues, cooperative automated driving has emerged as a promising approach to enhance safety via Vehicle-to-Everything (V2X) communication. V2X facilitates wireless connectivity between vehicles and various entities, including other vehicles, roadside units (RSUs), and remote cloud servers [2] [3].

As illustrated in **Figure 1**, in a cooperative automated driving architecture, vehicles and RSUs transmit driving data and sensor information collected by their respective onboard sensors to a server. The server then integrates this multi-source data and transmits the comprehensive environment model back to the vehicles. Because this information incorporates data captured by other entities, vehicles can recognize peripheral environments that are completely blind to their own sensors, thereby preventing accidents in low-visibility conditions.

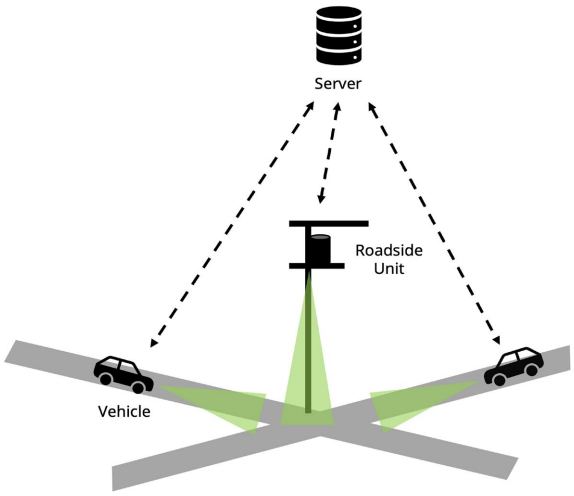

Figure 1. Cooperative automated driving with server.

The servers responsible for information integration are typically distributed geographically to balance processing loads and minimize communication latency [4]. Each server periodically aggregates data from all vehicles and RSUs within its designated control area. However, the wireless bandwidth allocated for server communication is often constrained. In scenarios where traffic density is exceptionally high, the aggregate bandwidth demand can easily exceed capacity, resulting in severe network congestion [5]. Because cooperative automated driving applications

directly impact collision avoidance, they demand ultra-low latency. Network congestion increases end-to-end latency, causing packet deliveries to exceed acceptable thresholds and degrading the overall Quality of Service (QoS). Crucially, vehicles currently attempt cooperative maneuvers without any prior knowledge of whether the network can support the required QoS, posing a severe threat to safety and traffic efficiency.

1.2. Objective

To resolve the issues of unpredictable QoS degradation and the lack of foresight in high-density vehicular environments, this study proposes a network-virtualization-based framework to predict QoS and secure future bandwidth reservations for vehicles. We construct a comprehensive communication model using a network simulator to experimentally evaluate and validate the effectiveness of the proposed method.

1.3. Structure of This Paper

The remainder of this paper is organized as follows. Chapter 2 reviews related work on QoS prediction for V2X communication in cooperative automated driving. Chapter 3 details the V2X communication control mechanism enabled by network virtualization, alongside our specific QoS prediction and reservation methods. Chapter 4 describes the experimental methodology used to evaluate the proposed framework and presents the simulation results. Chapter 5 discusses the implications of the obtained results, and Chapter 6 concludes the paper.

2. Related Work

In cooperative automated driving networks, vehicles exchange real-time data with surrounding vehicles, RSUs, and servers to build a safe and optimized transportation ecosystem. Because V2X communication is utilized for critical safety use cases such as accident prevention, it must satisfy stringent QoS requirements, particularly regarding bounded communication latency [6]. However, V2X QoS is highly dynamic and heavily influenced by factors such as high vehicle mobility and localized traffic congestion at intersections [7]. Sudden QoS degradation can paralyze cooperative driving functions, potentially jeopardizing human lives depending on the traffic context [8]. Consequently, proactive QoS prediction is paramount, as it allows vehicles and network infrastructure to deploy countermeasures prior to connection degradation.

Barmpounakis *et al.* proposed a machine-learning-based QoS prediction method that utilizes 5G network functions to collect and analyze QoS-related metrics. Their approach divides map information into spatial grids and clusters them based on network performance indicators [9]. Through network simulations, they demonstrated a fundamental trade-off between the number of clusters and prediction accuracy. Similarly, Xu *et al.* addressed the spatio-temporal dynamics of V2X communication by employing Informer [10], a long-term time-series fore-

casting model derived from the Transformer architecture [11] [12]. By gathering V2X communication data from environments replicating real-world traffic congestion, they achieved enhanced prediction accuracy for both packet loss rates and end-to-end latency.

While these existing studies focus primarily on applying and optimization various machine learning algorithms to improve computational efficiency under strict V2X constraints, they predominantly rely on historical data patterns. For instance, if historical logs indicate recurrent congestion on a specific road segment during morning commute hours, the models predict a similar pattern for the following day. However, if traffic flows deviate abruptly due to unpredictable events like road construction or traffic accidents, history-based models fail to accurately predict localized vehicle concentration.

Therefore, it is essential to predict near-future QoS by incorporating real-time operational data, such as current vehicular route plans. Furthermore, because individual vehicles only possess localized status updates, they cannot infer whether a future waypoint will suffer from network degradation caused by other vehicles. This highlights the necessity for a centralized mechanism that predicts network-wide QoS based on collective routing information and explicitly notifies vulnerable vehicles in advance.

3. Proposed Method

3.1. Overview

3.1.1. Network for Cooperative Autonomous Driving

While several architectural variations for vehicle-to-server communication have been explored, this study focuses on a cellular-network-based approach where vehicles communicate via base stations (gNBs/eNBs). In this setup, geographically distributed servers are deployed to manage specific regions. While the allocation mapping between servers and base stations varies in literature, we consider a topology where a single edge server manages multiple base stations. The server treats the collective coverage areas of its assigned base stations as its operational control area, managing all V2X data aggregation and processing for vehicles within this boundary.

3.1.2. Communication Control Using Network Virtualization Technology

To predict and guarantee network performance, this study introduces network virtualization technology to logically control communication paths across vehicles, base stations, and servers. Network virtualization enables programmatic, centralized network management via software. Within this architecture, a centralized controller monitors global network states and dynamically reconfigures communication paths to adapt to real-time traffic variations. As shown in **Figure 2**, we assume a topology where a single controller orchestrates communication across multiple edge servers and their respective vehicular fleets.

The controller establishes wired connections with the edge servers to aggregate the data required for QoS prediction. It then executes the QoS prediction algo-

rithms detailed in Section 3.2 and enforces the QoS assurance mechanisms described in Section 3.3. Since the controller-to-server links are wired and high-capacity, their internal communication latency and packet loss are assumed to be negligible.

To manage resources efficiently, the controller classifies the total available bandwidth between vehicles and servers into three distinct logical categories (Figure 3):

Bandwidth for QoS Prediction and Assurance:

This slice is exclusively reserved for the controller to collect telemetry data from vehicles and transmit QoS reservation confirmation notices back to them. If this control-plane communication is disrupted during heavy congestion, subsequent QoS predictions and notifications would fail. Thus, this bandwidth is permanently provisioned. Its capacity is dimensioned to comfortably accommodate the maximum theoretical physical limit of vehicles that could simultaneously occupy the server's control area. While the individual data overhead per vehicle remains constant, the aggregate volume scales linearly with the vehicle count.

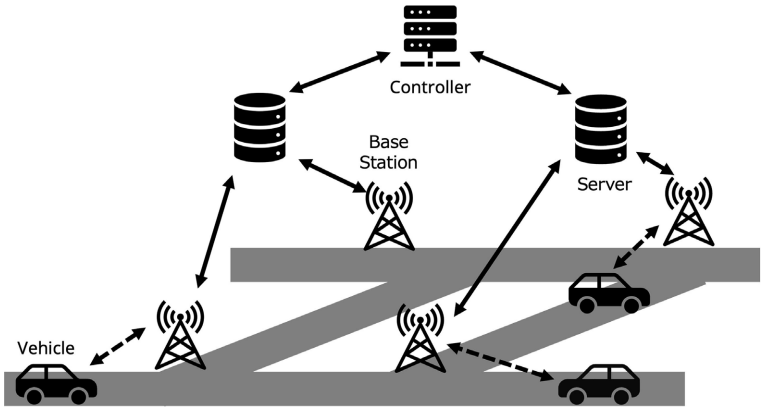

Figure 2. Control of V2X communication using network virtualization technology.

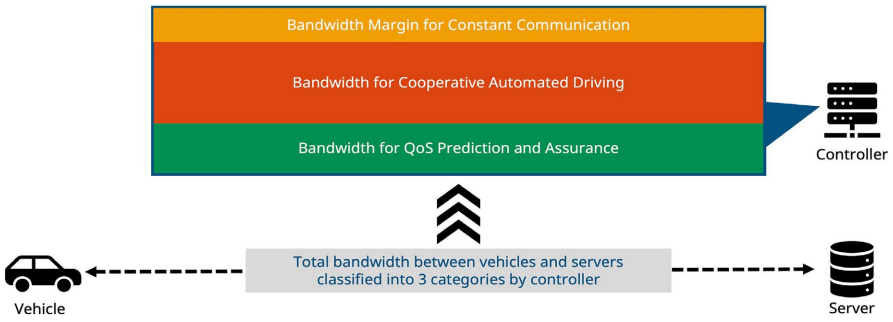

Figure 3. Classification of bandwidth by the controller.

Bandwidth for Cooperative Automated Driving:

This data-plane slice supports the core cooperative driving applications, handling the uplink transmission of ego-vehicle sensor data and the downlink distribution of integrated environment models. The traffic volume fluctuates heavily

depending on vehicle maneuvers and environmental complexity. Because this slice demands significantly higher capacity than the control plane, it is highly susceptible to congestion in high-density scenarios. Thus, our proposed prediction and reservation mechanisms target this specific bandwidth slice.

Bandwidth Margin for Constant Communication:

This static buffer is maintained to absorb instantaneous traffic bursts and minor mathematical errors in the QoS prediction model. It does not scale with vehicle density and occupies a smaller fraction of the total spectrum compared to the other two slices. This margin is strictly excluded from the "allocatable" pool during the QoS reservation process.

3.2. QoS Prediction

3.2.1. Information Managed by the Controller

To accurately forecast bandwidth congestion, the controller must predict the spatial concentration of vehicles and their projected data traffic within the control area over a specific look-ahead time horizon. To achieve this, the controller maintains a real-time database of the following static and dynamic parameters:

- Digital map topology of the managed coverage area
- Geographical coordinates of managed edge servers and base stations
- Maximum capacity B_{max} of each server
- Bandwidth allocations B_{saved} currently committed to active reservations (Section 3.3)

Additionally, every vehicle periodically uploads the following telemetry via the control plane:

- A unique vehicle identifier (Vehicle ID)
- Future travel plan trajectories (route and speed profiles)

Using this aggregated dataset, the controller evaluates network states for the next time interval.

3.2.2. QoS Prediction Method

The controller predicts network state suitability by calculating the projected aggregate data traffic and evaluating it against the maximum allocatable bandwidth. The structural workflow is illustrated in **Figure 4**. From the periodic travel plans, the controller projects the exact base station sector each vehicle will occupy after a designated time horizon. Let N represent the predicted number of vehicles within a specific server's domain at that future timestamp, and let B_{pred} denote the baseline control overhead required per vehicle. The minimum overhead bandwidth is expressed as:

$$N * B_{pred} \tag{1}$$

Since reservations are evaluated continuously over sliding windows, portions of the future bandwidth may already be locked by long-term maneuvers. Letting B_{saved} represent these pre-existing commitments, the net allocatable bandwidth available for core cooperative driving applications is formulated as:

$$B_{\max} - \left(N * B_{pred} + B_{saved} \right) \tag{2}$$

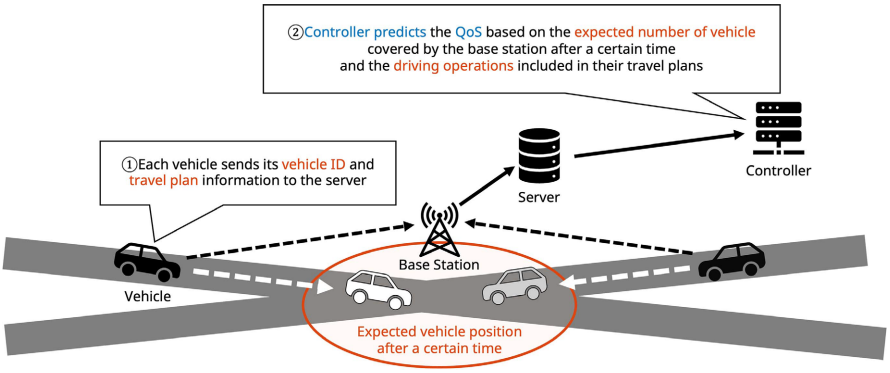

Figure 4. QoS prediction procedure.

To evaluate if the network can sustain unconstrained communication, the controller estimates the application-layer demand. Let B_{CADk} represent the predicted V2X traffic generated by vehicle k. If the aggregate demand violates the available capacity, the controller predicts an impending QoS failure for the fleet, as defined by inequality (3):

$$\sum_{k=1}^{n} B_{CAD_k} > B_{\max} - \left(N * B_{pred} + B_{saved}\right) \tag{3}$$

To account for estimation uncertainties and traffic bursts, the controller integrates the safety margin B_{extra}, or B_{margin} from Section 3.1.2. The definitive congestion criteria is thus established via inequality (4):

$$\sum_{k=1}^{n} B_{CAD_k} > B_{max} - \left(N * B_{pred} + B_{saved} + B_{extra}\right) \tag{4}$$

3.3. QoS Assurance

3.3.1. QoS Ensuring Method

If vehicles enter a congested sector without prior warning, they will attempt to transmit full-rate sensor data blindly. This behavior exacerbates packet queuing at the base station, pushing latency past acceptable cooperative driving thresholds. Consequently, vehicles are forced to abruptly fall back to local sensors, severely compromising safety.

To prevent this, the proposed framework explicitly coordinates transmission rights in advance based on the QoS prediction outputs (**Figure 5**). If the network state satisfies inequality (5), the controller concludes that capacity is sufficient for the entire local fleet:

$$\sum_{k=1}^{n} B_{CAD_k} \le B_{\max} - \left(N * B_{pred} + B_{saved} + B_{extra}\right) \tag{5}$$

In this optimal state, the controller approves reservations for all N vehicles and dispatches confirmation tokens via their Vehicle IDs. Conversely, if inequality (4) is triggered, indicating imminent congestion, the controller intercepts potential drops in performance by selectively admitting a subset of vehicles. Vehicles denied reservation tokens are instructed to suspend non-critical V2X transmissions,

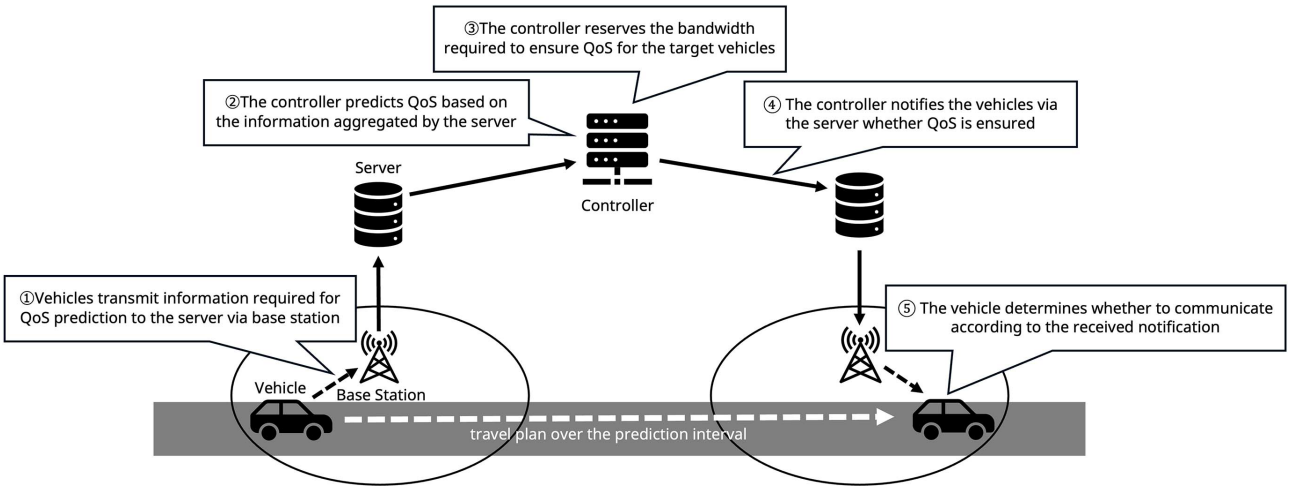

Figure 5. QoS assurance procedure.

thereby keeping the channel clean for admitted users. Admission priority is dynamically determined by the critical nature of each vehicle's immediate driving maneuver. For example, a vehicle executing a complex left/right turn at a blind intersection requires highly coordinated negotiation and is granted higher priority over a vehicle traveling straight on a clear, homogenous road segment. Tie-breaks among identical priority levels are resolved via random selection. (Note: Vehicles denied V2X bandwidth must utilize onboard fail-safes or alternative routing, which lies outside the scope of this paper).

3.3.2. QoS Assurance Time

Consider a scenario where a vehicle is prioritized to execute a right turn at an intersection (**Figure 6**). If the controller re-evaluated and re-allocated bandwidth strictly at rigid, rapid time intervals, a vehicle could theoretically be granted QoS at the start of a turn but stripped of its allocation mid-maneuver due to a sudden influx of new vehicles. Abruptly cutting off V2X data feeds mid-maneuver introduces catastrophic safety risks. To prevent this volatility, the controller parses the received travel profiles to determine the full expected duration of a critical maneuver and pre-allocates a continuous, multi-frame block of bandwidth. This ensures that once a maneuver is green-lit, its QoS remains strictly locked until completion. This locked bandwidth is added to the global B_{saved} parameter. Upon successfully completing the maneuver, the vehicle transmits a completion flag to the server, prompting the controller to immediately release the allocated slice back into the available pool.

4. Experiments

4.1. Evaluation Using Network Simulation

4.1.1. Experimental Setup

To quantify the impact of the proposed QoS prediction and reservation framework on V2X performance, we conducted discrete-event simulations using the

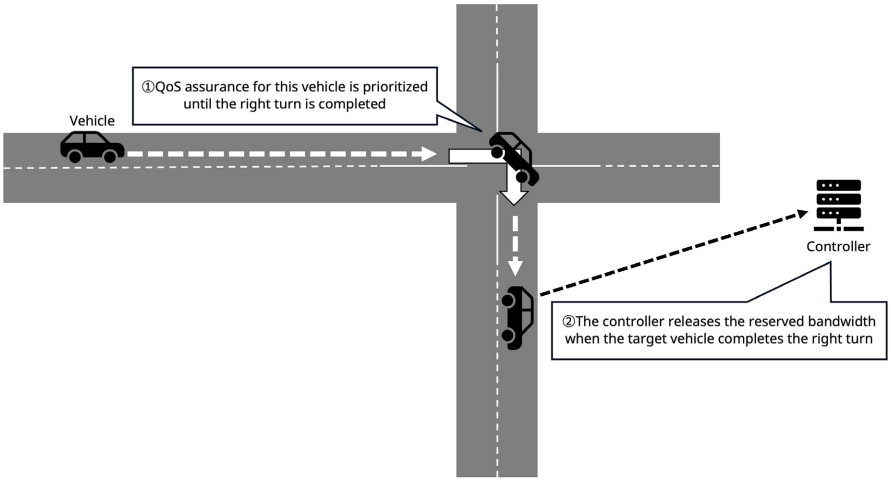

Figure 6. QoS assurance flow during a right turn.

ns-3 simulator [13] equipped with the 5G-LENA NR module. The evaluated topology consists of a single edge server connected to one base station managing a variable fleet size ranging from 20 to 70 vehicles (Figure 7). We compared two operational modes: Without QoS Prediction and Assurance (where all vehicles transmit data-plane packets greedily regardless of network load) and With QoS Prediction and Assurance (where the controller dynamically limits channel access to matching capacity). For the proposed method, the control overhead B_{pred} = 10 Kbps/vehicle was explicitly subtracted from the total available channel capacity to model overhead realistically. The explicit simulation parameters are defined in Table 1. The primary evaluation metric is the end-to-end downlink communication latency from the server to the vehicles. For the baseline mode, latency is averaged across the entire fleet; for the proposed mode, latency is measured across the admitted, QoS-guaranteed vehicles.

Table 1. Experimental parameters.

Virtual Environment	VirtualBox 7.0.8
Guest OS	Ubuntu 22.04
Simulator	ns-3 5G-LENA NR module
Transimission rate	25 Mbps
Packet size	1000 bytes
Transmission Interval	100 ms
Bandwidth for QoS Prediction per vehicle	10 Kbps
Number of Vehicle	20 - 70

4.1.2. Results

Figure 8 shows the evaluation results. In the baseline mode (Without QoS), the average latency escalated sharply as density increased: 22.0 ms (20 vehicles), 29.8 ms (30 vehicles), 53.2 ms (40 vehicles), 94.5 ms (50 vehicles), 103.9 ms (60 vehi-

cles), and 143.6 ms (70 vehicles).

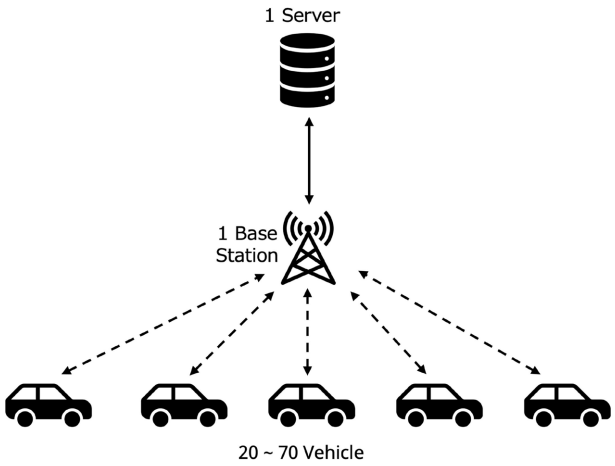

Figure 7. Simulation network model in ns-3.

In contrast, under the proposed framework (With QoS), the latency for guaranteed vehicles remained tightly bounded: 21.0 ms (20 vehicles), 31.1 ms (30 vehicles), 54.1 ms (40 vehicles), 92.1 ms (50 vehicles), 92.1 ms (60 vehicles), and 89.1 ms (70 vehicles).

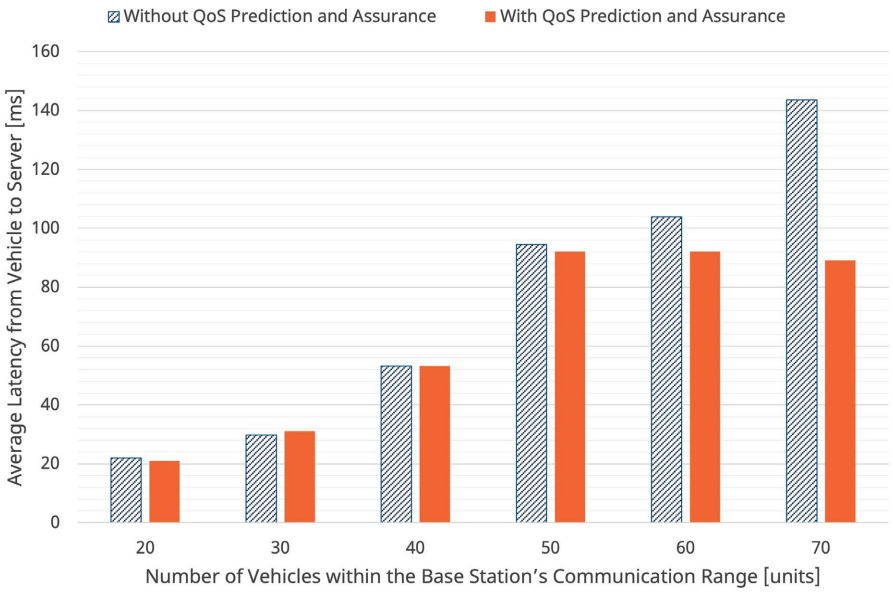

Figure 8. Comparison of latency via network simulation.

4.2. Estimation of Latency in a High-Density Traffic Scenario

To evaluate scalability in macro-scale environments, such as a massive urban intersection, we constructed a mathematical latency estimation model extrapolated from our ns-3 simulation logs, scaling the vehicle density from 200 to 1000 units. The structural application configuration (packet size, intervals, rates) remained identical to Table 1.

The comparative projections are illustrated in Figure 9. Without QoS mechanisms, latency surged to unusable levels: 275.4 ms (200 vehicles), 294.2 ms (400 vehicles), 311.1 ms (600 vehicles), 326.0 ms (800 vehicles), and 339.1 ms (1000 vehicles). Under the proposed framework, the latency experienced by admitted vehicles stabilized completely, hovering consistently around the 90 ms mark: 86.2 ms (200 vehicles), 94.5 ms (400 vehicles), 87.2 ms (600 vehicles), 94.3 ms (800 vehicles), and 96.8 ms (1000 vehicles).

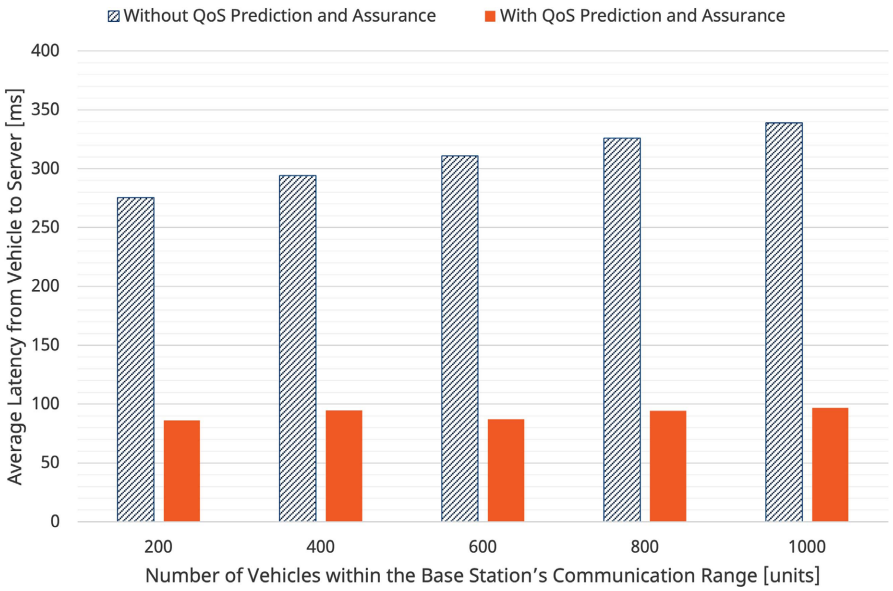

Figure 9. Comparison of estimated latency in a high-density traffic scenario.

5. Discussion

Based on the ns-3 simulation results (Section 4.1.2), the proposed QoS assurance mechanism effectively isolates critical traffic from network degradation. In lower-density scenarios (20 to 50 vehicles), both modes exhibit a typical linear latency increase matching traffic volume. However, at 60 and 70 vehicles, the baseline mode's latency surpasses the 100 ms threshold—the maximum acceptable latency budget for real-time cooperative automated driving applications.

Conversely, the proposed method caps the latency for admitted vehicles, keeping it safely below 100 ms even when the underlying channel is fully saturated. While the observed latency at high saturation (approx. 89 - 92 ms) approaches the safety limit, this can be easily mitigated in deployment by expanding the safety margin B_{extra} within the controller's allocation logic.

We must also analyze the architectural overhead introduced by our control plane. At intermediate densities (30 and 40 vehicles), the proposed method exhibits roughly 1 to 2 ms higher latency than the baseline. This minor penalty represents the capacity slice dedicated to B_{pred}, which slightly compresses the data-plane bandwidth. However, at 20 vehicles, this penalty disappears due to low overall utilization.

Interestingly, at 70 vehicles, the latency for admitted vehicles actually dropped by approximately 2 ms compared to the 50-vehicle case. This behavior occurs because as the total vehicle count scales up, the aggregate control overhead $N * B_{pred}$ expands, shrinking the remaining allocatable data-plane pool. Consequently, the controller admits fewer vehicles to the data plane, maintaining an un-congested sub-channel for the prioritized users.

The high-density analytical estimations (Section 4.2) further validate this behavior. Without control, latency scales to catastrophic levels (exceeding 300 ms), completely disabling cooperative functions. With the proposed framework, latency remains clamped near 90 ms regardless of whether 200 or 1000 vehicles occupy the area. The slight non-linear fluctuations in the controlled latency curve directly reflect the dynamic resizing of the admitted fleet size to match available capacity.

By enforcing this strict admission control and pairing it with maneuver-based prioritization (e.g., clearing vehicles actively turning inside an intersection box while deferring vehicles waiting in a trailing queue), network virtualization can successfully safeguard critical intersection negotiations, ensuring localized high-density bottlenecks do not translate into physical collisions.

6. Conclusions

Because the detection range of onboard vehicular sensors is fundamentally constrained by line-of-sight boundaries, cooperative automated driving has become a crucial paradigm to enhance traffic safety and efficiency by distributing unified environmental views via V2X networks. While edge computing infrastructure is deployed to distribute processing loads, localized vehicular spikes can easily overwhelm regional wireless links. When bandwidth saturates, packet delivery times slip past the strict latency boundaries required for cooperative safety applications, causing uncoordinated fallbacks that threaten passenger safety.

To overcome this vulnerability, this study introduced a proactive QoS prediction and reservation framework powered by network virtualization. By utilizing a software-defined controller to aggregate look-ahead routing profiles directly from the fleet, the network can model impending spatial vehicular densities and compute projected data demands. Comparing these projections against dynamic capacity matrices allows the controller to predict localized congestion prior to its physical onset. Under projected saturation, the controller enforces strict admission control, prioritizing bandwidth allocation for vehicles executing high-risk maneuvers (such as turning at blind intersections) while safely throttling non-critical nodes.

We validated the framework through both discrete-event ns-3 network simulations and macro-scale analytical scaling evaluations. The results conclusively demonstrate that the proposed framework successfully shields prioritized vehicles from the effects of network congestion, maintaining end-to-end latency within the strict 100 ms safety threshold required for cooperative maneuvers. As V2X

applications grow more data-intensive, this proactive reservation model offers a scalable solution to guarantee communication reliability when and where it is critically needed.

Funding

This work was partly supported by JSPS KAKENHI Grant Number JP 24H00698.

Author Contributions

Conceptualization, methodology, Sato, K.; software, validation, Kunimoto, T.; All authors have read and agreed to the published version of the manuscript.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] Yokoyama, T., Hatano, K., Ozawa, K., Hiyama, S. and Kodaka, K. (2020) Auto Industry Initiatives to Realize Automated Driving Cars. *Trends in the Sciences*, **25**, 5_17-5_21. https://doi.org/10.5363/tits.25.5_17

[2] Tatsumi, S., Higuchi, R. and Someya, R. (2020) Latest Technology Trends and Issues for Realizing Cooperative Autonomous Driving. *The Journal of The Institute of Electrical Engineers of Japan*, **140**, 308-311. https://doi.org/10.1541/ieejjournal.140.308

[3] Suganume, H. (2021) Trends and Future of ITS and Autonomous Driving. *IEICE Communications Society Magazine*, **15**, 102-108. https://doi.org/10.1587/bplus.15.102

[4] Zhou, F., Hu, R.Q., Li, Z. and Wang, Y. (2020) Mobile Edge Computing in Unmanned Aerial Vehicle Networks. *IEEE Wireless Communications*, **27**, 140-146. https://doi.org/10.1109/mwc.001.1800594

[5] Soto, I., Calderon, M., Amador, O. and Urueña, M. (2022) A Survey on Road Safety and Traffic Efficiency Vehicular Applications Based on C-V2X Technologies. *Vehicular Communications*, **33**, Article ID: 100428. https://doi.org/10.1016/j.vehcom.2021.100428

[6] 5G Automotive Association (5GAA) (2019) Making 5G Proactive and Predictive for the Automotive Industry. White Paper, 5G Automotive Association. https://5gaa.org/content/uploads/2020/01/5GAA_White-Paper_Proactive-and-Predictive_v04_8-Jan.-2020-003.pdf

[7] Noor-A-Rahim, M., Liu, Z., Lee, H., Ali, G.G.M.N., Pesch, D. and Xiao, P. (2022) A Survey on Resource Allocation in Vehicular Networks. *IEEE Transactions on Intelligent Transportation Systems*, **23**, 701-721. https://doi.org/10.1109/tits.2020.3019322

[8] Barros, M.T., Velez, G., Arregui, H., Loyo, E., Sharma, K., Mujika, A., *et al.* (2020) CogITS: Cognition-Enabled Network Management for 5G V2X Communication. *IET Intelligent Transport Systems*, **14**, 182-189. https://doi.org/10.1049/iet-its.2019.0111

[9] Barmpounakis, S., Maroulis, N., Koursioumpas, N., Kousaridas, A., Kalamari, A., Kontopoulos, P., *et al.* (2022) AI-Driven, QoS Prediction for V2X Communications in beyond 5G Systems. *Computer Networks*, **217**, Article ID: 109341. https://doi.org/10.1016/j.comnet.2022.109341

[10] Zhou, H., Zhang, S., Peng, J., Zhang, S., Li, J., Xiong, H., *et al.* (2021) Informer: Be-

yond Efficient Transformer for Long Sequence Time-Series Forecasting. *Proceedings of the AAAI Conference on Artificial Intelligence*, **35**, 11106-11115. https://doi.org/10.1609/aaai.v35i12.17325

[11] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł. and Polosukhin, I. (2017) Attention Is All You Need. *Proceedings of the* 31*st International Conference on Neural Information Processing Systems*, Long Beach, 4-9 December 2017, 6000-6010.

[12] Xu, Y., Shi, Y., Ge, Y., Chen, S. and Wang, L. (2023) Informer-Based QoS Prediction for V2X Communication: A Method with Verification Using Reality Field Test Data. *Computer Networks*, **235**, Article ID: 109958. https://doi.org/10.1016/j.comnet.2023.109958

[13] Ali, Z., Lagen, S., Giupponi, L. and Rouil, R. (2021) 3GPP NR V2X Mode 2: Overview, Models and System-Level Evaluation. *IEEE Access*, **9**, 89554-89579. https://doi.org/10.1109/access.2021.3090855