ROLEX: Reliable Offloading and Latency Efficiency in HAPS-based V2X System

Daniar Estu Widiyanti, *Soo Young Shin
Department of IT Convergence Engineering,
Kumoh National Institute of Technology (KIT), Gumi, South Korea
Email: {daniarestuw, *wdragon}@kumoh.ac.kr

Abstract

Vehicular Edge Computing (VEC) enables low-latency, computation-intensive services for intelligent transportation by allowing vehicles to offload tasks to roadside units (RSUs). However, due to high vehicular mobility, the contact time between a vehicle and an RSU is often insufficient to complete offloaded tasks, leading to high task failure rates. To address this challenge, this paper propose a ROLEX, a High-Altitude Platform Station (HAPS)-assisted offloading framework that introduces HAPS as a persistent computing tier to complement traditional RSU-based Multi-access Edge Computing (MEC). Our system explicitly incorporates RSU contact time into task scheduling decisions to ensure reliable task completion. Tasks exceeding the estimated RSU contact duration are offloaded to HAPS, improving task success under mobility constraints. The proposed approach enhances the reliability of task execution without increasing vehicle-side complexity or relying heavily on cloud connectivity.

I. Introduction

The continuous evolution of intelligent transportation systems has been supported by the development of Vehicular Edge Computing (VEC), in which computational tasks are offloaded from vehicles to nearby Roadside Units (RSUs) to reduce latency and processing load on vehicular devices [1]. However, due to the high mobility of vehicles and the limited coverage area of RSUs, reliable task execution has often been hindered by short and intermittent contact durations [2].

Consequently, tasks may be interrupted or dropped before completion, adversely affecting service reliability and quality [3]. To mitigate these challenges, the use of High-Altitude Platform Stations (HAPS) as a complementary computing tier has been proposed in recent research [4], [5]. Through persistent coverage and higher endurance, HAPS systems have been integrated to support offloading continuity when RSU contact becomes insufficient.

In this work, a HAPS-assisted offloading framework is introduced, where the contact duration between vehicles and RSUs is explicitly considered in task scheduling decisions to enhance reliability in dynamic vehicular environments..

II. Proposed System and Problem Formulation

In the proposed framework, a three-tier architecture is considered, consisting of vehicular clients, Roadside Units (RSUs), and High-Altitude Platform Stations (HAPS). Vehicles generate computation-intensive tasks that are first offloaded to the nearest RSU within communication range. However, due to high vehicular speed, the contact duration with RSUs is often insufficient for completing tasks, especially when execution latency exceeds the available connection time.

To ensure task reliability, HAPS is introduced as a persistent and wide-coverage computing layer, capable of receiving, caching, and executing tasks that are interrupted or rejected at the RSU level. This layered design enables seamless task migration from RSUs to HAPS when reliability thresholds cannot be satisfied locally. A simplified illustration of this architecture is shown in Fig. 1.

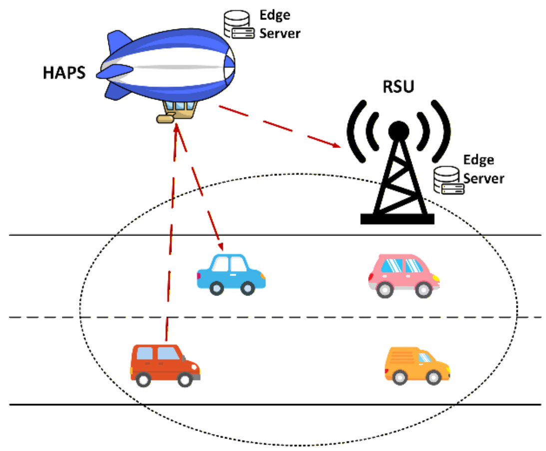

Figure 1 Proposed System

To enhance task reliability in vehicular edge computing, the offloading decision is formulated as a binary optimization problem. Each task is assigned either to a RSU or HAPS, depending on the estimated execution time and available contact duration. The objective is to maximize the number of tasks completed within their deadlines by selecting the most suitable execution tier based on mobility conditions and resource availability. A binary variable x_i is defined, where $x_i = 1$ indicates execution at the RSU, and $x_i = 0$ assigns the task to HAPS. The execution time of task i at tier $j \in \{\text{RSU}, \text{HAPS}\}$ is represented by T_i^j, and the task deadline is denoted d_i.

$$\begin{aligned} \max_{\{x_i\}} \quad & \sum_i \mathbb{I}\left(T_i^{(j_i)} \le d_i\right) \\ \text{s.t.} \quad & \text{(C1)} \quad x_i \cdot T_i^{\text{RSU}} \le t_i^{\text{RSU}}, \quad \forall i \\ & \text{(C2)} \quad (1 - x_i) \cdot T_i^{\text{HAPS}} \le d_i, \quad \forall i \\ & \text{(C3)} \quad \sum_{i \in \mathcal{T}_j} \rho_i \le C_j, \quad \forall j \in \{\text{RSU}, \text{HAPS}\} \end{aligned} \tag{1}$$

This optimization problem aims to maximize the number of vehicular tasks that can be completed before their deadlines by intelligently deciding whether each task should be offloaded to a nearby RSU ($x_i = 1$) or to HAPS ($x_i = 0$). Three constraints are enforced: (C1) tasks assigned to RSUs must finish within the vehicle's contact time with the RSU; (C2) tasks assigned to HAPS must complete before their individual deadlines; and (C3) the total computing load assigned to either RSU or HAPS must not exceed their respective resource capacities.

III. Result

To evaluate the effectiveness of the proposed HAPS-assisted offloading framework, a simulation was conducted using MATLAB with 100 randomly generated vehicular tasks. Each task was assigned a random size between 100 KB and 1000 KB and a deadline uniformly distributed between 0.5 s and 1.5 s. The required CPU cycles were set to 5000 cycles per KB, and RSUs and HAPS were assumed to operate at 10 GHz and 5 GHz, respectively. The RSU contact duration, limited by vehicular mobility, was randomly assigned between 0.3 s and 1.2 s.

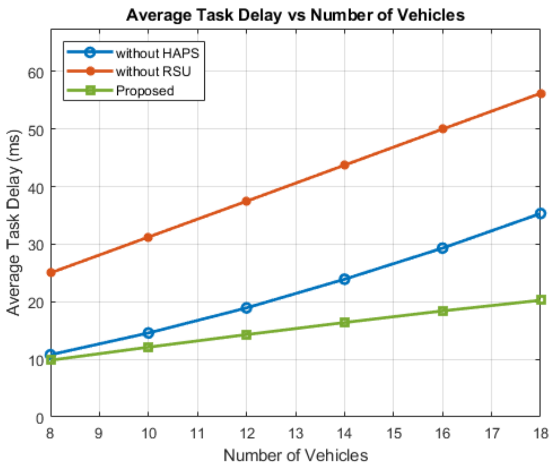

Figure 2. Average task delay versus number of vehicles

As shown in Fig. 2, the proposed approach consistently achieved the lowest average task delay across all traffic levels. Compared to schemes without RSU or HAPS support, significant delay reductions were observed, especially as the number of vehicles increased. This demonstrates that integrating HAPS as a complementary tier improves offloading efficiency and scalability under mobility constraints.

IV. Conclusion

A HAPS-assisted offloading framework for vehicular edge computing was proposed to address task failures caused by limited RSU contact time. By leveraging HAPS as a persistent computing tier, task reliability and delay performance were improved under varying traffic conditions. Simulation results showed that the proposed approach outperformed RSU-only and HAPS-only schemes, offering lower task delays and better scalability.

ACKNOWLEDGEMENT

This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (RS-2025-00553810, 50%) and This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education (2018R1A6A1A03024003, 50%).

REFERENCE

[1] Liu, L., Chen, C., Pei, Q., Maharjan, S., & Zhang, Y. (2021). Vehicular edge computing and networking: A survey. Mobile networks and applications, 26(3), 1145-1168.

[2] R. Men, X. Fan, K. -L. A. Yau, A. Shan and G. Yuan, "Hierarchical Aerial Computing for Task Offloading and Resource Allocation in 6G-Enabled Vehicular Networks," in IEEE Transactions on Network Science and Engineering, vol. 11, no. 4, pp. 3891-3904, July-Aug. 2024.

[3] Q. Ren, O. Abbasi, G. K. Kurt, H. Yanikomeroglu and J. Chen, "Handoff-Aware Distributed Computing in High Altitude Platform Station (HAPS)–Assisted Vehicular Networks," in IEEE Transactions on Wireless Communications, vol. 22, no. 12, pp. 8814-8827, Dec. 2023.

[4] Nguyen, T. H., & Park, L. (2023). HAP-assisted RSMA-enabled vehicular edge computing: A DRL-based optimization framework. Mathematics, 11(10), 2376.

[5] I. Rzig, W. Jaafar, M. Jebalia and S. Tabbane, "Dependency-Aware Task Offloading in Cooperative UAV-HAPS-Assisted Vehicular Networks," 2024 International Wireless Communications and Mobile Computing (IWCMC), Ayia Napa, Cyprus, 2024, pp. 1541-1546.