HERMES: Hybrid Execution and Resource Management for Edge Systems in Vehicular Environment for Future Communication

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

Abstract

The Internet of Vehicles (IoV) demands low-latency and reliable computation to support real-time, safety-critical applications. However, traditional edge computing architectures struggle to meet these requirements due to the growing number of vehicles and the dynamic nature of vehicular networks. Vehicular Edge Computing (VEC) addresses this challenge by enabling task offloading to road-side units (RSUs) and cooperative vehicles. Yet, many existing approaches ignore inter-task dependencies, resulting in inefficient scheduling and increased delays. To overcome these limitations, this paper proposes HERMES, a V2X-based hybrid offloading strategy that explicitly accounts for task dependencies and adapts to wireless variability. By jointly optimizing task distribution and resource allocation, HERMES effectively reduces latency and enhances overall system performance in dynamic vehicular environments.

I. Introduction

The rapid development of the Internet of Vehicles (IoV) has enabled a wide range of applications in intelligent transportation systems, including autonomous driving, real-time navigation, and cooperative safety. These applications demand ultra-low latency, high reliability, and efficient computation [1]. Traditional cloud computing architectures, however, fall short in meeting these requirements due to network congestion and the physical distance from end users. To bridge this gap, Vehicular Edge Computing (VEC) has emerged as a promising solution, leveraging computational resources available at Road Side Units (RSUs) and within vehicles to process tasks closer to the source [2].

Despite its advantages, VEC faces critical challenges when applied to complex vehicular environments. One of the major issues lies in task offloading strategies, which must account for the dynamic and unpredictable nature of Vehicle-to-Everything (V2X) communication. Furthermore, many existing offloading schemes treat tasks as isolated units, often overlooking the dependencies that exist among them, such as data processing pipelines, sensor fusion, or multi-stage decision-making. Ignoring such interdependencies can lead to inefficient scheduling, increased delays, and degraded system performance, especially in latency-sensitive scenarios [3].

To address these limitations, this paper proposes HERMES — a Hybrid Execution and Resource Management framework for edge systems in vehicular environments. HERMES introduces a V2X-based offloading strategy that explicitly models inter-task dependencies and dynamically orchestrates execution across vehicles and RSUs. By recognizing task dependencies and adjusting execution accordingly, the framework is able to reduce delay and improve overall processing efficiency.

Through dependency-aware scheduling and V2X-based hybrid offloading, HERMES adapts to both computational load and real-time communication conditions. Simulation results demonstrate that HERMES achieves lower latency and higher task success rates than baseline methods, particularly in dense or rapidly changing vehicular scenarios.

II. Proposed System and Problem Formulation

In the proposed HERMES framework, computation tasks generated by vehicles (TVs) are partially offloaded to nearby RSUs or to cooperative vehicles (CVs/R-CVs) using V2X communication. To minimize total task latency, decisions are made regarding how tasks are divided, where they are executed, and how system resources are allocated across computing and communication components.

Each task may consist of multiple dependent parts, with some executed locally and others offloaded. Offloading introduces two types of delays: **transmission delay** (due to wireless data transfer) and **computation delay** (due to processing at the RSU or another vehicle). To optimize performance under such conditions, HERMES dynamically adjusts several key system parameters:

- **Transmission power** on the cellular link (Uu interface),
- **Packet transmission frequency** over the direct V2V link (PC5 interface),
- **CPU speeds** of both the RSU and the receiving vehicle,
- **The task offloading decision**, which determines how tasks are split and where they're executed.

The objective is to minimize the total latency for executing all tasks, especially when tasks are interdependent. The optimization problem is formulated as:

$$\min_{\boldsymbol{\Xi}, \boldsymbol{f}, \boldsymbol{p}, \beta} \sum_{n=1}^{N} \max\left\{T_n^{\text{loc}},\, T_n^{\text{off}}\right\} \tag{1}$$

In this formulation, T_n^{loc} denotes the total time required to execute task n locally. If the task is offloaded, the total time is given by $T_n^{off} = T_n^{td} + T_n^{comp}$, which includes both the transmission delay and the computation delay. The offloading decisions are represented by the matrix Ξ, which determines the execution location of each task. The variables f, p, and β represent the CPU frequency, transmission power, and V2V packet transmission frequency, respectively. This optimization is subject to several constraints:

- Power constraint: total transmission and V2V power must be within the vehicle's power budget.
- Bandwidth constraint: V2V link usage must not exceed the available channel capacity.
- Computation constraint: CPU allocation at RSUs and vehicles must stay within their hardware limits.
- Dependency constraint: task execution must respect the DAG structure, i.e., a task can only start once all its predecessors are completed.

Due to the complexity of solving this optimization, the problem is decomposed into two interrelated subproblems:

1. Resource Allocation: Optimizing transmission power, frequency, and CPU speeds under system constraints.
2. Offloading Strategy: Deciding how much of each task to offload and to whom, while respecting task dependencies.

III. Result

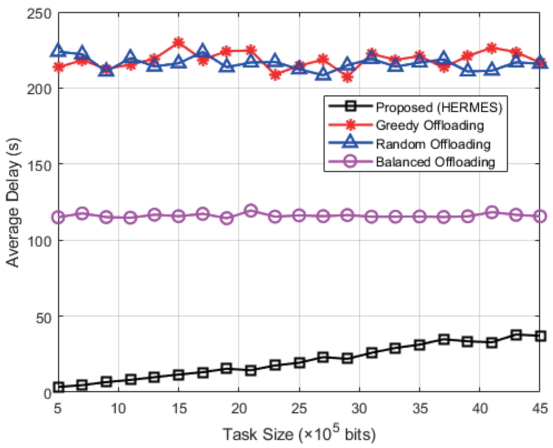

Figure 1. Result (Average Delay per-Task)

Figure 2 presents the average delay performance of the proposed HERMES framework compared to three baseline offloading strategies: greedy offloading, random offloading, and balanced (uniform) offloading. The simulation is configured with the following parameters: vehicle arrival rate $\Lambda = 1$, channel capacity $C_m \sim U[500, 1000]$, VEC server CPU frequency $f^{ser,max} \sim U[1, 2]$ GHz, 10 cooperative vehicles (M = 10), maximum transmission power P_{max} = 23 dBm, and four vehicular lanes.

To assess system performance under varying workloads, multiple task dependency graphs (DAGs) are generated with task data sizes ranging from 5 to 45 x 10^5 bits. In the greedy offloading strategy, tasks are assigned to the fastest available CV, R-CV, or RSU. Random offloading distributes tasks arbitrarily between vehicles and the RSU, while balanced offloading splits tasks evenly regardless of context. As shown in the figure, HERMES consistently achieves lower average delay across all task sizes, highlighting its effectiveness in optimizing offloading and resource allocation while respecting task dependencies.

IV. Conclusion

This paper proposed HERMES, a V2X-based offloading framework that accounts for inter-task dependencies in vehicular edge computing. By optimizing task distribution and resource allocation, HERMES effectively reduces latency and improves performance under dynamic network conditions. The results demonstrate its potential for enabling efficient and scalable computation in intelligent transportation systems.

ACKNOWLEDGMENT

This research was supported by the MSIT(Ministry of Science and ICT), Korea, under the ITRC(Information Technology Research Center) support program(IITP-2025-RS-2024-00437190) supervised by the IITP(Institute for Information & Communications Technology Planning & Evaluation, 50%) and This work was supported by Innovative Human Resource Development for Local Intellectualization program through the Institute of Information & Communications Technology Planning & Evaluation(IITP) grant funded by the Korea government(MSIT) (IITP-2025-RS-2020-II201612, 50%).

REFERENCE

[1] S. Yang, J. Tan, T. Lei and B. Linares-Barranco, "Smart Traffic Navigation System for Fault-Tolerant Edge Computing of Internet of Vehicle in Intelligent Transportation Gateway," in IEEE Transactions on Intelligent Transportation Systems, vol. 24, no. 11, pp. 13011-13022, Nov. 2023.

[2] L. Yin, J. Luo, C. Qiu, C. Wang and Y. Qiao, "Joint Task Offloading and Resources Allocation for Hybrid Vehicle Edge Computing Systems," in IEEE Transactions on Intelligent Transportation Systems, vol. 25, no. 8, pp. 10355-10368, Aug. 2024.

[3] Y. -J. Ku, S. Baidya and S. Dey, "Uncertainty-Aware Task Offloading for Multi-Vehicle Perception Fusion Over Vehicular Edge Computing," in IEEE Transactions on Vehicular Technology, vol. 72, no. 11, pp. 14906-14923, Nov. 2023.