Delay Tolerant Precaching Scheme based on Proximal Policy Optimization in Content-Centric Vehicular Networks

Jeongtak Na, Jongpil Youn, Irina Em, and Euisin Lee
School of Information and Communication Engineering
Chungbuk National University
Cheongju, Republic of Korea
{jeongtak_na, whdvlf1919, ira.em, eslee}@chungbuk.ac.kr

Youngju Nam
School of Software
Kunsan National University
Gunsan, Republic of Korea
imnyj@kunsan.ac.kr

***Abstract*—The surge in multimedia demand from autonomous driving and infotainment stresses RSUs; mobility and limited wireless capacity cause delays and QoS degradation. While CCVN precaching has been studied, delay-tolerant services remain underexplored. We present a PPO-based delay-tolerant precaching scheme featuring path-aware RSU discovery, proportional precaching decisions, and opportunistic transmission within deadlines. A tailored reward balances delivery success, backhaul reduction, and traffic efficiency. Simulation results in Large-scale SUMO experiments show that our scheme outperforms a heuristic scheme in content delivery success ratio and traffic overhead, demonstrating scalable and adaptive learning for delay-tolerant vehicular content delivery.**

***Index Terms*—content-centric vehicular networks, delay-tolerant precaching, reinforcement learning, PPO.**

I. Introduction

The rapid advancement of autonomous driving and the increasing sophistication of in-vehicle displays and entertainment systems have led to an explosive growth in passenger demand for rich multimedia content [1]. The size and diversity of such content continue to expand, imposing substantial requirements on vehicular communication infrastructures. However, due to vehicle mobility and the inherent limitations of wireless communications between vehicles and roadside units (RSUs), passengers frequently experience content download delays and degraded quality of service (QoS) [2]. Ensuring seamless content delivery under these dynamic conditions remains a critical challenge in vehicular networks.

To mitigate these issues, research has focused on content precaching in content-centric vehicular networks (CCVNs) [3]. Precaching allows RSUs to proactively store content in advance based on predicted vehicle mobility patterns and user requests, thereby reducing backhaul traffic and improving delivery latency [4]. By leveraging mobility-aware predictions, CCVNs can enhance the reliability of content delivery, particularly for high-demand multimedia services, and support a better quality of experience for passengers.

Despite these efforts, most existing studies have concentrated primarily on delay-sensitive content, where immediate delivery is critical [5], [6]. In contrast, research on delay-tolerant content remains limited. Recently, heuristic-based schemes have been proposed to address delay-tolerant precaching [7]. While such methods can provide partial improvements, their reliance on static heuristics restricts their adaptability to the highly dynamic and stochastic nature of vehicular networks. Consequently, these heuristic approaches often fail to achieve consistently superior performance in realistic deployment scenarios.

To overcome these limitations, we propose a novel delay-tolerant precaching scheme based on the Proximal Policy Optimization (PPO) algorithm [8], a state-of-the-art reinforcement learning method. Our approach dynamically determines both the participating RSUs and the proportion of content to be precached, enabling adaptive and efficient utilization of RSU resources. Through extensive large-scale simulations conducted in a SUMO-based vehicular environment, we demonstrate that the proposed PPO-based scheme outperforms a heuristic approach in terms of content delivery ratio and traffic overhead, thereby validating its effectiveness in real-world vehicular networking scenarios.

II. Network Model

We consider a CCVN with content servers, RSUs $\mathbb{R} = \{R_1, \dots, R_J\}$ deployed at signalized intersections (with caching storage and CPT), and vehicles $\mathbb{V} = \{V_1, \dots, V_I\}$. RSUs connect to servers via fiber backhaul and disseminate content over WAVE; vehicles move along routes with speeds updated every $0.1\,\mathrm{s}$ by a skewed-Gaussian acceleration model. The content set $\mathbb{C} = \{C_1, \dots, C_K\}$ comprises delay-tolerant items, each with a tolerant delay Tol_k and fixed-size chunks.

We integrate PPO for delay-tolerant precaching: at decision epoch t, the state s_t aggregates vehicle context (speed, location, sojourn), content traits (size, category, Tol_k), and RSU info (cache status, distance). The action $a_t = [p_1, \dots, p_N]$ allocates precaching proportions across candidate RSUs on the path, and the reward r_t promotes success within Tol_k while penalizing backhaul use, waste, and deadline violations. This model couples the physical network with learning-based control so that RSUs act as cache nodes and PPO-guided decision points, enabling adaptive, delay-aware, and resource-efficient delivery.

III. PPO-based Delay Tolerant Precaching Scheme

A. Path-aware RSU Discovery

A vehicle requesting content with tolerant delay Tol_k shares its path $\mathcal{P} = \{x_1, \ldots, x_K\}$ via a lightweight *Path_Announce*. The RSU builds the candidate set $\mathcal{C}_t = \{R_i \in \mathbb{R} \mid R_i \in \mathcal{P} \wedge \text{reachability}(R_i, Tol_k) = 1\}$. For each $R_i \in \mathcal{C}_t$, arrival/departure $(\hat{t}_i^{\text{arr}}, \hat{t}_i^{\text{dep}})$ and sojourn $\hat{\tau}_i$ are inferred from the speed/acceleration model with a confidence interval $\text{CI}_\alpha(\hat{\tau}_i)$.

Each RSU advertises a compact *Cache_Report*: cache sketch $\mathbf{c}_i$, free space C_i^{free}, link stats $\bar{r}_i$, backhaul load b_i, and timestamp τ_i^{ts}. The deliverable budget is $\widehat{L}_i = \bar{r}_i \cdot \mathbb{E}[\hat{\tau}_i] \cdot (1-\rho_i)$, where $\rho_i = f(b_i) \in [0, 1)$. We prune RSUs failing feasibility ($\widehat{L}_i \leq \epsilon$), freshness (stale $> \delta$), or reachability ($\Pr(\text{visit } R_i) < \kappa$). Remaining RSUs are sorted by distance d_i, and the top-$N_{\max}$ are passed to the agent.

B. PPO Formulation for Delay-Tolerant Precaching

1) State and Action: The state s_t (56-D) includes: **1) Vehicle & Content (6-D):** normalized (x, y), avg. network speed, content size, agent-selected tolerant delay, and max speed. **2) Path-Aware RSUs (50-D):** for the first 25 on-path RSUs, normalized cumulative distance and a binary selection flag.

The action a_t (50-D) consists of: **1) RSU Scores (25-D):** raw scores for top-k selection. **2) Chunk Proportions (25-D):** normalized content shares for selected RSUs.

2) Reward Function: The total reward balances delivery success and resource efficiency:

$$\begin{aligned} r_{total} = R_{\text{succ}} W_{succ} - \frac{C_{\text{cons}}}{C_{\text{total}}} W_{cons} - \frac{C_{\text{waste}}}{C_{\text{total}}} W_{waste} \\ - \frac{P_{\text{wrong}}}{N_{\text{RSU}}} W_{wrong} + \frac{T_{tol} - T_{\text{actual}}}{T_{tol}} W_{time} \\ + \frac{C_{cached}}{C_{total}} W_{saved}, \end{aligned} \tag{1}$$

where each component encourages a specific behavior:

- R_{succ}: large bonus for completing the full download within T_{tol} (primary objective), weighted by W_{succ}.
- $\frac{C_{\text{cons}}}{C_{\text{total}}}$: penalty for consumed data volume (transmission cost), weighted by W_{cons}.
- $\frac{C_{\text{waste}}}{C_{\text{total}}}$: heavier penalty for precached but unused data, weighted by W_{waste}.
- $\frac{P_{\text{wrong}}}{N_{\text{RSU}}}$: penalty for selecting RSUs not encountered on the route, weighted by W_{wrong}.
- $\frac{T_{tol} - T_{\text{actual}}}{T_{tol}}$: reward/penalty based on slack or overrun relative to the deadline, weighted by W_{time}.
- $\frac{C_{cached}}{C_{total}}$: bonus for leveraging cached delivery (backhaul savings), weighted by W_{saved}.

where R_{succ} rewards timely completion; C_{cons} and C_{waste} penalize transmission cost and unused data; P_{wrong} penalizes invalid RSU selection; the time term rewards slack relative to the deadline; and C_{cached} incentivizes backhaul savings. The weights (W) are optimized via Optuna.

3) PPO Objective: The clipped surrogate objective is $L^{\text{PPO}}(\theta) = \mathbb{E}_t[\min(\rho_t(\theta)\hat{A}_t, \text{clip}(\rho_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t)]$, where $\rho_t(\theta)$ is the probability ratio and $\hat{A}_t$ is the advantage function.

C. Content Transmission Process

Once the PPO policy determines a_t, content portions are precached. As the vehicle traverses the path, RSUs transmit assigned portions upon contact, minimizing backhaul dependency while meeting Tol_{req}. The overall procedure of our proposed scheme, from state observation to content download, is summarized in Algorithm 1.

Algorithm 1: PPO-based Delay Tolerant Precaching

```
Input: Trained PPO Policy π, Vehicle V with path P,
  Content Request (ID, Size), Tolerable Delay Tol_req;
Output: Final Download Status;

Phase 1: Decision and Precaching;
Assemble state s_t from (V, P, ID, Size, Tol_req);
a_t ← π(s_t) Determine allocation vector;
foreach RSU R_i ∈ P do
    C_chunk ← getAllocatedChunk(a_t, R_i);
    if C_chunk > 0 then
        R_i caches C_chunk of ID from backhaul;
    end
end

Phase 2: Vehicle Traversal and Download;
Size_down ← 0, T_start ← CurrentTime;
while Size_down < Size and V is on path P do
    Update V position; R_contact ← getRSU(V);
    if R_contact ≠ null then
        C_assign ← getAllocatedChunk(a_t, R_contact);
        if C_assign > 0 and is cached then
            data ← R_contact.transmit(ID, C_assign);
        end
        else
            data ← Backhaul.transmit(ID);
        end
        Size_down ← Size_down + data;
    end
end
T_actual ← CurrentTime − T_start;
if Size_down ≥ Size and T_actual ≤ Tol_req then
    return Success;
else
    return Failure;
end
```

IV. Performance Evaluation

A. Environment and Training Setup

We use **SUMO** via TraCI API for simulations. Table I shows the environmental parameters in our simulations.

Optimal reward weights were determined using **Optuna** (TPE sampler, 100 trials) to maximize the objective:

TABLE I
SIMULATION ENVIRONMENT AND TRAINING PARAMETERS

Parameter	Value
Simulation Map Size	15×15 km (Manhattan Grid)
Vehicle Speed Range	$10 \sim 100$ km/h
Number of Vehicles	1000 per episode
RSU Coverage Radius	700 m
Content Size	$100 \sim 700$ MB
Tolerable Delay (T_{tol})	$2000 \sim 4800$ s
Transmission Rate	6 Mbps
PPO Learning Rate	3×10^{-4}
Discount Factor (γ)	0.99
Clip Range (ϵ)	0.2

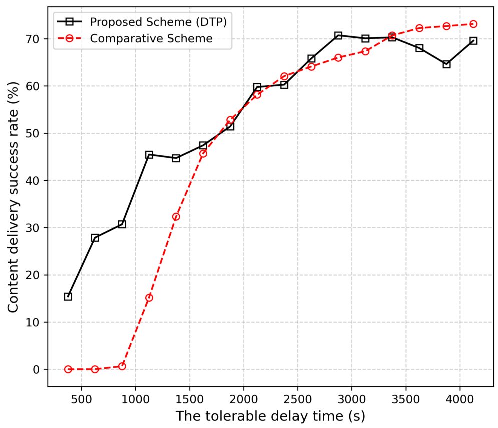

Fig. 1. Content delivery success rate vs. tolerable delay time.

(Hit Ratio$\times$100)$-$(Delay$\times$0.01). The optimized weights are: $W_{succ}{\approx}148.6$, $W_{cached}{\approx}30.3$, $W_{cons}{\approx}83.2$, $W_{waste}{\approx}22.4$, $W_{wrong}{\approx}15.4$, and $W_{time}{\approx}53.2$.

B. Performance Metrics

We focus on two metrics: **1) Content Delivery Success Rate:** The percentage of completed requests within the deadline. **2) Overhead Ratio:** The ratio of total transmission cost (including unused cache) to the effective content size.

C. Simulation Results and Analysis

We compare our PPO-based DTP scheme against the **Average Speed-based Allocation** (Comparative Scheme), which estimates dwell time based on collective traffic flow.

Fig. 1 and Fig. 2 illustrate the performance comparison. As shown in Fig. 1, our **Proposed Scheme (DTP)** demonstrates significantly superior performance in scenarios with tight deadlines ($400s \sim 1500s$). While the Comparative Scheme fails to deliver content effectively when the time budget is limited, the PPO agent intelligently identifies the most critical RSUs for immediate caching.

Although the Comparative Scheme shows a slightly higher success rate in loose delay conditions ($> 2500s$), Fig. 2 reveals the cost. The Comparative Scheme maintains a consistently high overhead ratio ($\approx 62\%$), indicating excessive redundant caching. In contrast, the Proposed Scheme maintains a significantly lower overhead ratio ($\approx 35\%$). This confirms that PPO

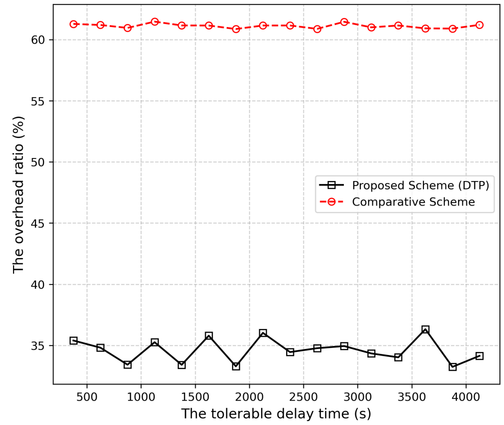

Fig. 2. Overhead ratio vs. tolerable delay time.

balances delivery success and resource efficiency, whereas the heuristic relies on inefficient over-provisioning.

V. CONCLUSION

This paper presents a PPO-based delay-tolerant precaching (DTP) framework for CCVNs that respects delivery deadlines. The scheme combines path-aware RSU discovery, a compact state, continuous precaching allocation, and a deadline-aware reward. Large-scale SUMO experiments show consistent gains over a heuristic baseline in delivery success and traffic overhead, while forward-pass inference enables millisecond-level decisions for real-time deployment. The reward balances success, backhaul savings, waste reduction, wrong-RSU penalties, and time slack, aligning learning with operational key performance indicators.

REFERENCES

[1] Q. Yuan *et al.*, “Toward Efficient Content Delivery for Automated Driving Services: An Edge Computing Solution,” *IEEE Netw.*, vol. 32, no. 1, pp. 80–86, 2018.
[2] S. Ullah *et al.*, “RSU assisted reliable relay selection for emergency message routing in intermittently connected VANETs,” *Wireless Netw.*, vol. 29, pp. 1311–1332, 2023.
[3] X. Wang and X. Wang, “Vehicular Content–Centric Networking Framework,” *IEEE Syst. J.*, vol. 13, no. 1, pp. 519–529, 2019.
[4] M. T. R. Khan *et al.*, “Proactive Content Retrieval Based on Value of Popularity in Content-Centric Internet of Vehicles,” *IEEE Trans. Intell. Transp. Syst.*, vol. 25, no. 8, pp. 8514—8526, 2024.
[5] L. Yao *et al.*, “A cooperative caching scheme based on mobility prediction in vehicular content centric networks,” *IEEE Trans. Veh. Technol.*, vol. 67, no. 6, pp. 5435–5444, 2018.
[6] B. Feng *et al.*, “Proactive content caching scheme in urban vehicular networks,” *IEEE Trans. Commun.*, vol. 71, no. 7, pp. 4165–4180, 2023.
[7] Y. Nam, H. Choi, and E. Lee, “Content Storage Management and Precaching Scheme in Content-Centric-Network-Based Internet of Vehicles,” *IEEE Internet Things J.*, vol. 12, no. 9, pp. 12927-12947, 2025.
[8] P. Saikia *et al.*, “Proximal Policy Optimization for RIS-Assisted Full Duplex 6G-V2X Communications,” *IEEE Trans. Intell. Veh.*, vol. 9, no. 7, pp. 5134-5149, 2024.