Load-Balancing Optimization under Network–Compute Congestion: An Analysis

Yongjae Jang, Hanyoung Park, Taeyang Lee, Ji-Woong Choi
dept. Electrical Engineering and Computer Science
Daegu Gyeongbuk Institute of Science and Technology (DGIST)
Daegu, Republic of Korea
yj46.jang, prkhnyng, sunny626, jwchoi@dgist.ac.kr

***Abstract*—We evaluate a dynamic, cross-layer load-balancing policy for autonomous vehicles in a three-tier OBU–RSU–Cloud architecture using a city-scale, testbed-style simulation. At each slot, the policy jointly considers compute queues (OBU/RSU/Cloud) and network state (PDR, link quality) to select the processing location (local, RSU, or cloud). We compare against Local-Only and Offloading-Only baselines. The simulation instantiates 400 vehicles, 50 RSUs, and one cloud with periodic tasks and per-slot decisions. Under normal to moderate congestion, the proposed policy consistently achieves queue stability, higher average PDR, and lower energy than the baselines. However, when vehicle clustering creates an RSU hotspot, we observe a sharp PDR drop at the hotspot and ripple effects that degrade neighboring RSUs, indicating that load balancing alone cannot overcome physical-layer capacity limits. This motivates practical guardrails—such as association control (caps/biasing), interference-aware scheduling, and predictive hotspot avoidance—to secure reliability.**

***Index Terms*—Vehicular Edge Computing, Offloading, Load Balancing, Packet Delivery Ratio, Congestion, Lyapunov Optimization**

I. Introduction

Autonomous vehicles (AVs) increasingly run safety-critical perception and control together with data-hungry infotainment (AR navigation, streaming, conferencing) [1]. To meet tight latency and reliability constraints without exhausting onboard power, AV stacks rely on hierarchical computing—onboard units (OBUs), roadside edge servers (RSUs), and the cloud—plus wireless offloading [2]. In dense urban corridors, however, many AVs attempt to offload simultaneously, causing wireless contention and edge overload to co-occur. The result is a drop in packet delivery ratio (PDR), queue build-up, and end-to-end (E2E) latency spikes that directly threaten service level objectives.

The core difficulty is cross-layer coupling. Wireless quality (fading, interference, scheduling) modulates the effective offload rate; compute queues determine completion delay; both evolve under mobility. A naive "always offload" policy can saturate RSUs and air-interface resources; a conservative "always local" policy wastes available capacity and increases energy draw. Moreover, hotspots—temporary vehicle clustering at a few RSUs—induce localized collapse and ripple effects that degrade neighboring cells via interference/backoff coupling [3].

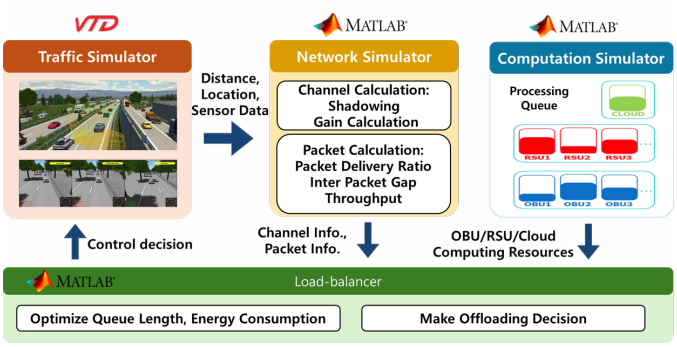


Fig. 1: Integrated simulator architecture.

Existing policies often optimize either networking (e.g., link-aware offloading) or computing (e.g., queue/energy-aware placement) in isolation, rely on small-scale or static setups, or ignore the dynamics of mobility and interference [4], [5]. These assumptions mask the non-linear interaction between PDR, effective rate, queue stability, and energy in realistic, city scale scenarios. As a result, policies that look good in siloed models can underperform when wireless congestion and compute overload coincide.

We develop and analyze a dynamic, cross-layer load-balancing policy that (i) translates instantaneous PDR into an effective offload rate, (ii) uses a Lyapunov drift-plus-penalty objective to co-optimize queue stability and energy, and (iii) operates over a three-tier OBU/RSU/Cloud architecture within a synchronized simulator. We compare Local-Only, Offloading-Only, and the proposed Dynamic policy. Results show that Dynamic sustains high PDR under typical loads by flexibly mixing local processing and offloading, while revealing a fundamental limit under extreme hotspots where localized collapse and neighbor degradation emerge.

II. System Model

A. Simulator architecture

We consider an AV network with three compute tiers (OBU, RSU, Cloud) coupled with a wireless access subsystem (e.g., PC5/Uu) [6]. Time is slotted. At each slot t, each vehicle generates a task and must choose Local, RSU offload, or Cloud offload. As shown in Fig. 1, the simulator updates mobility,

TABLE I: Network simulator parameters

Parameter	Value
Technology	LTE-V2X
Subcarrier spacing	15 kHz
Modulation order	64-QAM
Bandwidth	10 MHz
Channel coding	Turbo code
TTI duration	1 ms
Code rate	0.8333
Channel model	WINNER+
Tx power	23 dBm
Antenna gain	3 dB
Noise figure	9 dB
Thermal noise	−101 dBm

TABLE II: Computation (queue/energy) parameters

Parameter	Value
CPU frequency range	100 MHz – 1 GHz
α	2×10^{-7}
β	0.1
γ	1.1×10^{-2}
Avg. arrival	1920 Mbit
Std. arrival	500 Mbit
Avg. offload	2400 Mbit
Std. offload	480 Mbit
RSU processing capacity	360 Mbit/slot
Cloud processing capacity	2880 Mbit/slot
Network power(P_{net})	2 W
V	0.45×10^7

computes wireless KPIs (PDR, rate), updates queues/energy, and applies the controller's decision. The tables summarize the parameters used in the simulator. Table I lists the network-simulator parameters, and Table II lists the computation parameters.

B. Effective offload transmission rate

Let $\rho(t) \in [0,1]$ denote packet delivery ratio (PDR) and $R(t)$ the nominal PHY rate. The effective rate available to offloading is

$$R_{\text{eff}}(t) = \rho(t)\, R(t), \tag{1}$$

so that fading/interference reduce usable throughput linearly in ρ. As ρ falls (congestion, distance, collisions), offloading benefit diminishes sharply: arrivals to remote queues drop while retransmissions inflate delay/energy.

C. Queue model

Let $Q_i(t), Q_j(t), Q_k(t)$ be the queue backlogs at OBU i, RSU j, and Cloud k. With arrivals $a_i(t)$ and service $u(t)$, the queues evolve as

$$Q_i(t{+}1) = \max\{\, Q_i(t) + a_i(t) - u_i(t),\, 0\}, \tag{2}$$

$$Q_j(t{+}1) = \max\{\, Q_j(t) + R_{ij}^{\text{eff}}(t) - u_j(t),\, 0\}, \tag{3}$$

$$Q_k(t{+}1) = \max\{\, Q_k(t) + O_i^{\text{eff}}(t) - u_k(t),\, 0\}, \tag{4}$$

where $R_{ij}^{\text{eff}}(t) = \rho_{ij}(t)\, R_{ij}(t)$ is the effective OBU→RSU rate and $O_i^{\text{eff}}(t) = \rho_i(t)\, O_i(t)$ the effective OBU→Cloud rate.

D. Cost function

We adopt a Lyapunov drift-plus-penalty objective that trades queue stability against energy:

$$\min\ \mathbb{E}\big[\Delta L\big(Q(t)\big) + V\, E(t)\big]\,. \tag{5}$$

where $L(\cdot)$ is a quadratic Lyapunov function, $E(t)$ the slot energy, and $V > 0$ controls the delay–energy trade-off.

Local processing (with DVFS). If the OBU selects local compute at frequency f_i,

$$\text{Cost}_{\text{local}} = \min_{f_i} \Big\{\, V\big(\alpha f_i^3 + \beta\big)\ -\ \big(\tfrac{f_i}{\gamma} - a_i(t)\big) Q_i(t) \Big\}. \tag{6}$$

The f_i^3 term captures dynamic power; the queue term promotes higher f_i when Q_i is large or offloading is unappealing.

RSU offloading.

$$\text{Cost}_{\text{RSU}} = V\, P_{\text{net}} - \big(R_{ij}^{\text{eff}}(t) - a_i(t)\big) Q_i(t) + R_{ij}^{\text{eff}}(t)\, Q_j(t). \tag{7}$$

High PDR/throughput (large $R_{ij}^{\text{eff}}(t)$) helps drain , but a congested RSU (large $Q_j(t)$) raises cost.

Cloud offloading.

$$\text{Cost}_{\text{cloud}} = V\, P_{\text{net}} - \big(O_i^{\text{eff}}(t) - a_i(t)\big) Q_i(t) + O_i^{\text{eff}}(t)\, Q_k(t). \tag{8}$$

Backhaul/RTT effects are abstracted into $O_i^{\text{eff}}(t)$ and $Q_k(t)$.

Decision rule. Each slot,

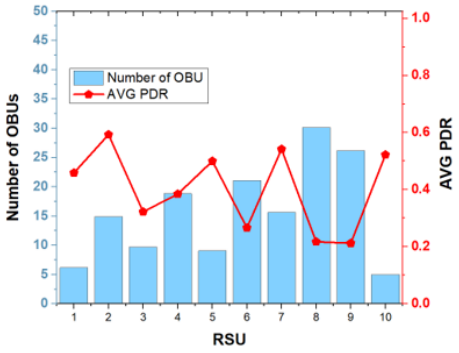
$$\text{Decision}(t) = \arg\min \Big\{ \text{Cost}_{\text{local}},\ \text{Cost}_{\text{RSU}},\ \text{Cost}_{\text{cloud}} \Big\}, \tag{9}$$

with f_i^* returned if Local is chosen.

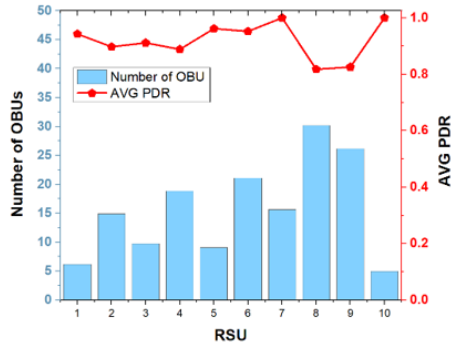
III. Simulation Results

A city-scale virtual testbed is instantiated over a real road graph. Unless stated, we use 400 vehicles, 50 RSUs, and one cloud. Vehicles generate periodic tasks per slot and associate to nearby RSUs. We compare Local-Only, Offloading-Only, and Dynamic. The primary metric is RSU-level PDR as a function of the number of simultaneously associated vehicles at that RSU.

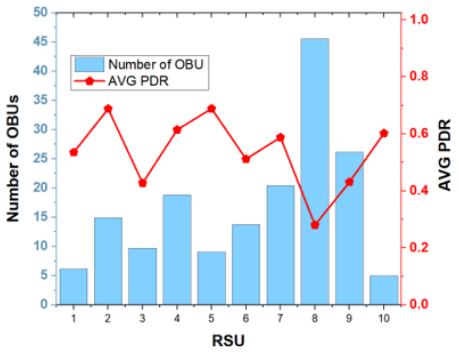
When every task is offloaded as shown in Fig. 2a, air-interface contention and RSU ingress queues grow in lock-step. Even at modest association counts, many RSUs exhibit $PDR < 0.6$. With ρ depressed, (1) yields small R_{eff}, retransmissions increase, and E2E reliability degrades. By selecting Local when links/RSUs are stressed and offloading when channels are clear, Dynamic flattens RSU load peaks and keeps most RSUs in $PDR \approx 0.8$–0.98 (Fig. 2b). The policy naturally exploits temporal diversity across tiers. For a forced cluster (AVs on one RSU), that RSU's PDR drops sharply, and neighboring RSUs also degrade due to interference/backoff coupling (Fig. 2c). Once physical-layer capacity at a hotspot is exceeded, decisions cannot locally restore reliability; nearby cells feel the ripple effect. PDR generally declines with association count, but two RSUs at the same count may differ significantly due to geometry, interference topology, and reuse patterns.



(a) Offloading-Only



(b) Normal-Dynamic



(c) Congested-Dynamic

Fig. 2: Analysis of the relationship between number of connected vehicles served by each RSU and average PDR.

IV. Conclusion

We presented a cross-layer dynamic load-balancing policy that converts PDR into effective rate, optimizes a drift-plus-penalty objective, and runs over a three-tier OBU/RSU/Cloud architecture. In city-scale simulations, Dynamic maintains high PDR under normal density by flexibly mixing local compute and offloading, yet exposes a fundamental limit under extreme hotspots where localized collapse and neighbor degradation occur. Future work will integrate interference-aware scheduling, association control, and predictive hotspot avoidance so that the controller not only reacts to congestion but also preempts it through topology-level measures.

Acknowledgment

This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT). (No. RS-2024-00442085, Development of V2X Infra Security Core Technologies for Autonomous Vehicle Services & No. RS-2024-00398157, Development of AI-Native 6G Systems Orchestrating AI-Native Services).

References

[1] Z. Ning, K. Zhang, X. Wang, L. Guo, X. Hu, J. Huang, B. Hu, and R. Y. K. Kwok, "Intelligent Edge Computing in Internet of Vehicles: A Joint Computation Offloading and Caching Solution," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, pp. 2212–2225, Apr. 2021.

[2] J. Kwak, Y. Kim, J. Lee, and S. Chong, "DREAM: Dynamic Resource and Task Allocation for Energy Minimization in Mobile Cloud Systems," *IEEE Journal on Selected Areas in Communications*, vol. 33, pp. 2510–2523, Feb. 2015.

[3] M. Gonzalez-Martín, M. Sepulcre, R. Molina-Masegosa, and J. Gozalvez, "Analytical Models of the Performance of C-V2X Mode 4 Vehicular Communications," *IEEE Transactions on Vehicular Technology*, vol. 68, pp. 1155–1166, Feb. 2019.

[4] S. Choi, P. Choi, D. Kim, J. Kwak, and J.-W. Choi, "An Integrated Process-Network Load Balancing in Edge-Assisted Autonomous Vehicles Using Multimodal Applications With Shared Workloads," *IEEE Access*, vol. 12, pp. 174654–174667, 2024.

[5] S. Choi, D. Kwon, and J.-W. Choi, "Latency Analysis for Real-Time Sensor Sharing Using 4G/5G C-V2X Uu Interfaces," *IEEE Access*, vol. 11, pp. 35197–35206, 2023.

[6] J. Kwak, H. S. Chwa, H.-S. Jo, W. Kang, J. Kim, J. Song, J. Kim, S. Lee, T. Nam, W. Seong, and J.-W. Choi, "An Integrated Network-Computing Load Balancing Simulator for VEC-Assisted Autonomous Vehicles," *IEEE Communications Magazine*, vol. 63, pp. 146–153, June 2025.