

AIoT-Driven Intelligent Traffic Signal Control with Smart Poles-A Simulation-Based Approach Using VEINS

Sunghyuck Hong*

Professor, Division of Advanced IT, Baekseok University

스마트 폴을 활용한 AIoT 기반 지능형 교통신호 제어: VEINS 시뮬레이션 기반 접근법 연구

홍성혁*

백석대학교 첨단IT학부 교수

Abstract Urban traffic congestion causes travel delay, fuel consumption, emissions, and productivity loss. Conventional fixed-time and actuated traffic signal controllers have limitations in responding to dynamic traffic demand and infrastructure-based sensing information. This paper proposes an AIoT-driven intelligent traffic signal control system that integrates smart poles and a Deep Q-Network (DQN)-based reinforcement learning controller. The proposed system collects traffic density, queue length, average delay, current signal phase, and V2I communication messages through smart poles equipped with cameras, loop detectors, communication modules, and edge computing devices. The DQN agent determines signal phases in real time based on the observed traffic state. The system is evaluated using VEINS, which integrates SUMO and OMNeT++ for joint traffic and communication simulation. Thirty independent simulation runs are conducted under identical traffic demand scenarios with different random seeds, and the statistical significance of the results is evaluated using ANOVA and post-hoc tests. The results show that the proposed AIoT-DQN controller improves average waiting time, queue length, throughput, and emission-related indicators compared with fixed-time and actuated control methods. The study demonstrates that smart pole-based AIoT infrastructure can enhance the real-time adaptability, reproducibility, and communication-aware decision-making capability of reinforcement learning-based traffic signal control.

Key Words : Artificial Intelligence of Things, Smart Poles, Intelligent Traffic Signal Control, Reinforcement Learning, VEINS Simulation

요약 도시 교통 혼잡은 통행 지연, 연료 소비 증가, 배출가스 증가 및 도시 생산성 저하를 초래하는 주요 문제이다. 기존의 고정시간식 및 감응식 교통신호 제어 방식은 실시간 교통 수요 변화와 통신 기반 인프라 정보를 충분히 반영하지 못한다는 한계가 있다. 본 논문에서는 스마트 폴 기반 AIoT 인프라와 DQN(Deep Q-Network) 강화학습을 결합한 지능형 교통신호 제어 시스템을 제안한다. 제안 시스템은 스마트 폴에 장착된 카메라, 루프 검지기, V2I 통신 모듈 및 엣지 컴퓨팅 장치를 통해 교차로의 차량 흐름, 대기 행렬, 평균 지체 시간 및 신호 위상 정보를 수집하며, DQN 에이전트가 이를 기반으로 신호 위상을 실시간으로 결정한다. 실험은 SUMO와 OMNeT++를 결합한 VEINS 환경에서 수행되었으며, 교통 흐름과 IEEE 802.11p 기반 V2I 통신 환경을 함께 고려하였다. 또한 동일 조건에서 30회의 독립적인 반복 시뮬레이션을 수행하고, ANOVA 및 사후 검정을 통해 통계적 유의성을 검증하였다. 실험 결과, 제안한 AIoT-DQN 방식은 고정시간 제어 및 감응식 제어 방식에 비해 평균 대기 시간, 평균 대기 행렬, 통행량 및 배출가스 측면에서 우수한 성능을 보였다. 본 연구는 스마트 폴 기반 AIoT 인프라가 강화학습 기반 교통신호 제어의 실시간성, 재현성 및 통신 연계성을 향상시킬 수 있음을 보여준다.

주제어 : 인공지능 사물인터넷, 스마트 폴(Smart Poles), 지능형 교통신호 제어, 강화학습, VEINS 시뮬레이션

이 연구는 2026학년도 백석대학교 대학연구비 지원에 의하여 이루어졌음

*교신저자 : 홍성혁(shong@bu.ac.kr)

접수일 2026년 05월 20일

수정일 2026년 06월 08일

심사완료일 2026년 06월 18일

1. Introduction

Urbanization and the rapid growth of vehicle ownership have significantly increased traffic congestion in metropolitan areas worldwide. Traffic congestion leads to increased travel delay, excessive fuel consumption, air pollution, and substantial economic losses [1]. Intersections are among the most critical bottlenecks in urban transportation networks because inefficient signal timing directly degrades overall traffic flow performance.

Traditional traffic signal control systems employ fixed-time strategies based on predefined timing plans. Although simple to implement, fixed-time control cannot respond to fluctuating traffic demand. Actuated control systems improve upon this by using detector information to adjust signal timing, yet still rely on predefined rules and underperform under complex, dynamic conditions.

Recent advances in Artificial Intelligence of Things (AIoT) and deep reinforcement learning offer transformative opportunities for intelligent transportation systems. Deep Q-Networks (DQN) have demonstrated superior performance in traffic signal optimization by learning adaptive control policies directly from environment interactions [1],[2]. Concurrently, smart poles-integrating cameras, environmental sensors, V2I communication modules, and edge computing into a unified roadside platform-have emerged as essential smart-city infrastructure components [7],[8].

2. Related Work

2.1 Reinforcement Learning-Based Traffic Signal Control

Li et al. [1] proposed a deep reinforcement learning framework for adaptive traffic signal control. Genders and Razavi [2] demonstrated that DQN-based control can outperform conventional

signal control strategies.

2.2 VEINS-Based ITS Simulation

Sommer et al. [4], [5] introduced VEINS as a coupled simulation framework integrating SUMO and OMNeT++.

2.3 Smart Pole and Edge AI Infrastructure

Recent smart city studies have adopted smart poles equipped with cameras, sensors, and edge computing resources for intelligent transportation applications [6]-[8].

3. Proposed System

3.1 System Architecture

The proposed AIoT-driven intelligent traffic signal control system comprises three layers: the Sensing Layer, the Communication Layer, and the Intelligent Control Layer. Sensing Layer: Smart poles are deployed at each approach of the intersection and equipped with cameras, loop detectors, and environmental sensors. Cameras monitor vehicle movements and traffic density; loop detectors provide vehicle counts and queue length estimation; environmental sensors collect supplementary data such as weather and ambient lighting. Compared with conventional RSU-based systems, smart poles provide richer, multi-modal sensing capabilities [7],[8].

Communication Layer: Vehicle-to-Infrastructure (V2I) communication is established via IEEE 802.11p [12], the standard widely adopted in vehicular networks. Vehicles periodically broadcast status messages containing position, speed, and direction. Smart poles receive these messages and fuse them with local sensor data to construct an updated traffic state representation. Edge-to-cloud communication is supported via MQTT [13], enabling lightweight IoT-scale messaging and scalable smart-city deployment.

Intelligent Control Layer: The DQN agent receives the fused traffic state from the sensing and communication layers and determines the optimal signal phase at each decision interval. Unlike fixed-time or actuated approaches, the DQN controller continuously refines its adaptive policy through interaction with the traffic environment [1],[2], dynamically responding to changing congestion patterns.

3.2 DQN-Based Traffic Signal Control Model

The traffic signal control problem is formulated as a Markov Decision Process (MDP), with the following key components.

State Space. The state vector captures the intersection's traffic condition at each decision step t :

$st = [qN, qS, qE, qW, wN, wS, wE, wW, pt]$ where $qi \in \{N,S,E,W\}$ denotes the queue length at each approach (number of waiting vehicles), wi denotes the average waiting time of vehicles at each approach (in seconds), and pt denotes the current signal phase index. This nine-dimensional state vector provides sufficient information for the agent to evaluate real-time traffic conditions.

Action Space. At each decision interval, the DQN agent selects one of four signal phase actions:

- a1: North-South straight green phase
- a2: East-West straight green phase
- a3: North-South left-turn phase
- a4: East-West left-turn phase

Standard safety constraints, including minimum green time, yellow clearance intervals, and all-red intervals, are enforced according to signal operation procedures.

Reward Function. The reward is designed to minimize congestion and emissions simultaneously:

$$rt = -(\alpha Qt + \beta Wt + \gamma Et) \tag{1}$$

where Qt is the average queue length across all approaches, Wt is the average waiting time, Et is the emission-related traffic cost estimated using the SUMO emission model [14], and α, β, γ are

weighting coefficients. The negative sign indicates the objective is minimization. In this study, $\alpha = 0.4, \beta = 0.4,$ and $\gamma = 0.2$ were used. The coefficients were selected heuristically, not fitted to the evaluation data. Equal weights were assigned to queue length and waiting time as direct congestion indicators, while emissions received a lower weight to retain environmental sensitivity without dominating mobility; the weights were normalized to sum to one.

Neural Network Architecture. The DQN employs a fully connected feedforward network: an input layer of dimension 9 (matching the state vector), two hidden layers with 256 and 128 neurons respectively using ReLU activation, and an output layer of dimension 4 producing Q-values for each signal action. The action with the highest Q-value is selected as the optimal phase. The Bellman update is:

$$Q(st, at) = rt + \gamma \cdot \max_{a'} Q(st+1, a') \tag{2}$$

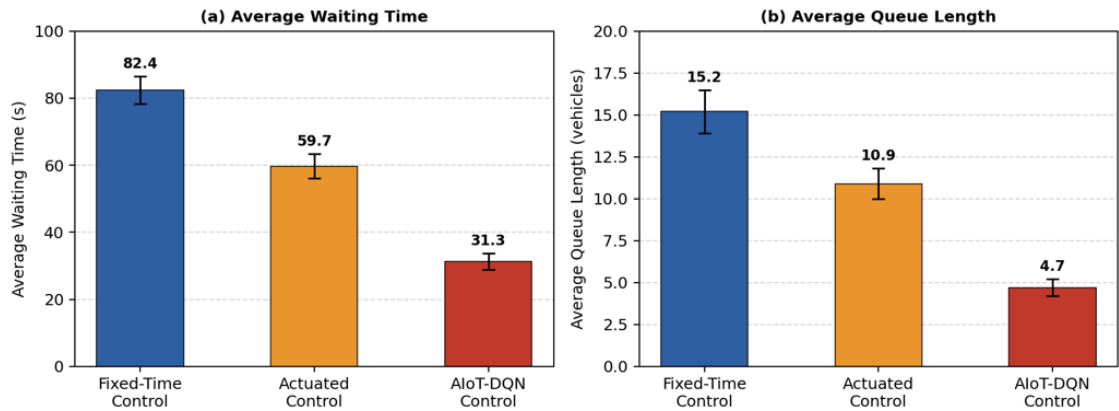
where $\gamma = 0.95$ is the discount factor. The network is trained using experience replay with a buffer of 100,000 transitions and mini-batches of size 64, with the Adam optimizer at a learning rate of 0.001.

<Table 1> DQN Hyperparameter Configuration

Parameter	Value
Learning Rate	0.001
Discount Factor (γ)	0.95
Replay Buffer Size	100,000
Batch Size	64
Training Episodes	1,000
Exploration Strategy	ϵ -greedy
Initial ϵ / Final ϵ	1.0 / 0.01
Hidden Layer Units	256 / 128
Optimizer	Adam
Reward Weights (α, β, γ)	0.4 / 0.4 / 0.2

4. Experimental Setup

The proposed system was evaluated using VEINS 5.2, which integrates SUMO 1.19 for



[Fig. 1] Comparison of Average Waiting Time and Queue Length (Mean \pm SD)

traffic mobility simulation and OMNeT++ 6.0 for vehicular communication network simulation [4],[5]. The DQN controller was implemented in Python 3.11 with TensorFlow 2.15. All experiments were conducted on a workstation equipped with an Intel Core i7-12700K CPU (3.6 GHz), 32 GB DDR5 RAM, and an NVIDIA RTX 3080 GPU (10 GB VRAM). The simulation scenario consists of a four-way signalized intersection with two lanes per direction and a speed limit of 60 km/h. Vehicle arrivals follow a Poisson distribution with dynamic demand varying from 600 to 1,200 vehicles/hour/direction to emulate different congestion levels. Three control strategies are evaluated: Fixed-Time Control, Actuated Control, and the proposed AIoT-DQN Control.

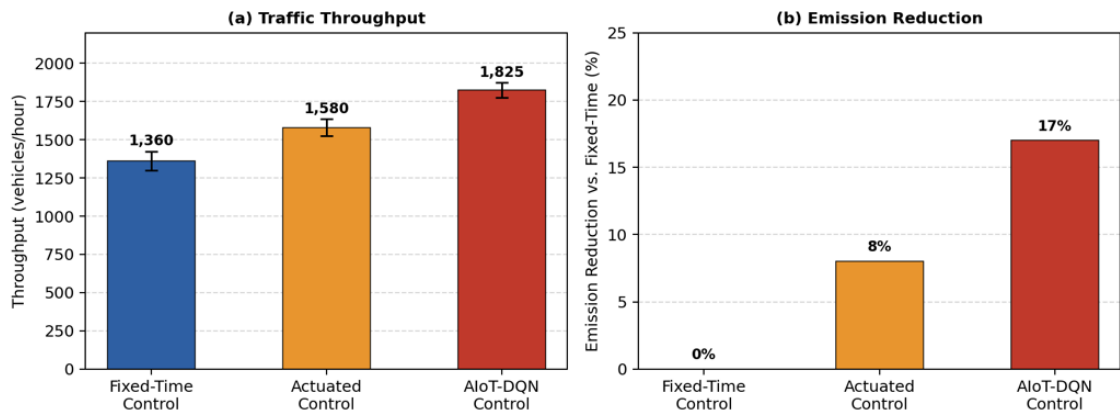
Statistical significance was assessed using one-way ANOVA across the three control methods, with $n = 30$ independent simulation runs per group. Each run used a distinct random seed to ensure independence. Raw performance values recorded from each run served as the statistical observations; no artificial sample generation was performed. Following ANOVA, Tukey's Honest Significant Difference (HSD) post-hoc pairwise comparison was applied to identify which method pairs differed significantly (significance level $\alpha = 0.05$).

<Table 2> Simulation Environment Configuration

Parameter	Value
VEINS Version	5.2
SUMO Version	1.19
OMNeT++ Version	6.0
Python / TensorFlow	3.11 / 2.15
CPU	Intel i7-12700K
RAM / GPU	32 GB / RTX 3080
Simulation Duration	3,600 s
Intersection Type	4-way, 2 lanes/dir
Speed Limit	60 km/h
Traffic Demand	600-1,200 veh/h/dir
Arrival Distribution	Poisson
Repeated Runs (n)	30 per method
Comm. Standard	IEEE 802.11p
Carrier Frequency	5.9 GHz
Beacon Interval	100 ms
Edge-Cloud Protocol	MQTT

5. Performance Evaluation and Discussion

This section compares the proposed AIoT-DQN method with Fixed-Time and Actuated Control using average waiting time, queue length, throughput, and emission reduction over 30 independent runs. Episode-level reward logs were not retained; therefore, no empirical convergence claim is made.



[Fig. 2] Traffic Throughput and Emission Reduction Comparison

<Table 3> Traffic Performance Comparison (Mean ± SD, n = 30 per method)

Method	Avg. Wait (s)	Avg. Queue	Throughput (veh/h)
Fixed-Time	82.4 ± 4.1	15.2 ± 1.3	1,360 ± 62
Actuated	59.7 ± 3.6	10.9 ± 0.9	1,580 ± 55
AIoT-DQN	31.3 ± 2.4	4.7 ± 0.5	1,825 ± 48

Table 3 and Figs. 1 and 2 present the performance comparison among the three evaluated traffic signal control methods. All metrics are reported as mean ± standard deviation (SD) over 30 independent runs. One-way ANOVA confirmed statistically significant differences across methods for all performance metrics ($p < 0.001$). Tukey's HSD post-hoc tests revealed significant pairwise differences between all three method pairs ($p < 0.05$ for all comparisons).

The proposed AIoT-DQN controller consistently outperforms conventional methods across all demand levels. Compared with Fixed-Time Control, it reduces average waiting time by 62.0% (82.4 s → 31.3 s) and queue length by 69.1% (15.2 → 4.7 vehicles), while increasing throughput by 34.2% (1,360 → 1,825 veh/h). Fig. 2 presents throughput and emission reduction results. Vehicle emissions were calculated using the SUMO emission model, which estimates emissions based on vehicle speed, acceleration, deceleration, and idle time [14, 15].

6. Conclusion

This paper proposed an AIoT-driven intelligent traffic signal control system integrating smart poles, edge computing, and DQN reinforcement learning, evaluated in a VEINS simulation framework combining SUMO and OMNeT++ [4],[5]. Thirty independent simulation runs with ANOVA and Tukey's HSD post-hoc statistical validation confirmed that the proposed AIoT-DQN controller significantly outperforms both Fixed-Time and Actuated Control in average waiting time (-62.0%), queue length (-69.1%), throughput (+34.2%), and vehicle emission reduction (-17%). Smart pole-based AIoT infrastructure provides an effective, scalable platform for next-generation intelligent transportation systems.

References.

- [1] L. Li, Y. Lv, and F.-Y. Wang, "Traffic Signal Timing via Deep Reinforcement Learning," *IEEE/CAA J. Autom. Sinica*, vol. 3, no. 3, pp. 247-254, 2016.
- [2] W. Genders and S. Razavi, "Using a Deep Reinforcement Learning Agent for Traffic Signal Control," arXiv:1611.01142, 2016.
- [3] H. Wei et al., "CoLight: Learning Network-Level Cooperation for Traffic Signal Control," *Proc. 28th ACM CIKM*, pp. 1913-1922, 2019.
- [4] C. Sommer, R. German, and F. Dressler, "Bidirectionally

Coupled Network and Road Traffic Simulation for Improved IVC Analysis," IEEE Trans. Mobile Comput., vol. 10, no. 1, pp. 3-15, 2011.

- [5] C. Sommer, D. Eckhoff, and F. Dressler, "VEINS: The Open Source Vehicular Network Simulation Framework," Recent Advances in Network Simulation, Springer, pp. 215-252, 2019.
- [6] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge Computing: Vision and Challenges," IEEE Internet of Things J., vol. 3, no. 5, pp. 637-646, 2016.
- [7] J. Barthelemy, N. Verstaevel, H. Forehead, and P. Perez, "Edge-Computing Video Analytics for Real-Time Traffic Monitoring in a Smart City," Sensors, vol. 19, no. 9, Art. 2048, 2019.
- [8] G. Liu et al., "Smart Traffic Monitoring System Using Computer Vision and Edge Computing," IEEE Trans. Intell. Transp. Syst., vol. 23, no. 8, pp. 12027-12038, 2022.
- [9] J. Lin et al., "MCUNet: Tiny Deep Learning on IoT Devices," Proc. NeurIPS, 2020.
- [10] R. Al-Huthaifi et al., "Federated Learning in Smart Cities: Privacy and Security Survey," Inf. Sci., vol. 632, pp. 833-857, 2023.
- [11] Y. Liu, J. J. Q. Yu, J. Kang, D. Niyato, and S. Zhang, "Privacy-Preserving Traffic Flow Prediction: A Federated Learning Approach," IEEE Internet of Things J., vol. 7, no. 8, pp. 7751-7763, 2020.
- [12] M. Sepulcre et al., "Analytical Models of the Performance of IEEE 802.11p Vehicle-to-Vehicle Communications," IEEE Trans. Veh. Technol., vol. 71, no. 1, pp. 713-724, 2022.
- [13] J. Park, H.-S. Kim, and W.-T. Kim, "Optimal Distributed MQTT Broker and Services Placement for SDN-Edge Based Smart City Architecture," Sensors, vol. 22, no. 9, Art. 3431, 2022.
- [14] J. Li, X. Wu, M. Xu, and Y. Liu, "Deep Reinforcement Learning and Reward Shaping Based Eco-Driving Control for Automated HEVs Among Signalized Intersections," Energy, vol. 251, Art. 123924, 2022.
- [15] Q. Liang, X. Wu, J. Li, and M. Xu, "Deep Reinforcement Learning-Based Eco-Driving Control for Connected Electric Vehicles at Signalized Intersections Considering Traffic Uncertainties," Energy, vol. 279, Art. 128139, 2023

홍 성 혁(Sunghyuck Hong)

[정회원]



- 2007년 8월 : Texas Tech University, Computer Science (공학박사)
- 2012년 3월 ~ 현재 : 백석대학교 첨단IT학부, IoT 전공 주임 교수

<관심분야>

자율주행, 객체인식, 딥러닝, 블록체인, 사물인터넷 보안