Optimal Allocation and Deployment of Roadside Units in Cloud-Based Internet of Vehicles Framework

Nay Myo Sandar[1*] Surekha Lanka[2] Thinzar Aung Win[3] Shuvra Tripura[4]

[1*,2,3,4]*Faculty of Business and Technology, Stamford International University, Bangkok, Thailand*

*Corresponding Author. E-mail address: naymyo.sandar@stamford.edu

Received: 13 February 2025; Revised: 4 June 2025; Accepted: 8 June 2025
Published online: 26 December 2025

Abstract

The research focuses on internet of vehicles (IoV) where vehicles are equipped with cameras and sensors to monitor traffic jams, accidents, and locations to ensure safety and comfort for drivers. To process sensor data effectively, cloud computing is used because of vast storage and processing capabilities. However, transferring data from sensors to cloud can be challenging due to bandwidth and memory constraints. Therefore, a cloud-based internet of vehicles framework is proposed incorporating Roadside Units (RSUs). RSUs can buffer video streams from vehicles and send them to cloud services. With RSUs in the framework, total latency for transferring video streams to cloud services can be significantly enhanced. In this research, dynamic programming approaches are applied to determine how many RSUs are needed at the lowest cost and greedy algorithm is implemented to prove the optimal solution from dynamic programming. Furthermore, K-means clustering algorithm is applied to find the best locations for RSUs. According to numerical results, the proposed methods can determine the optimal number of 6 RSUs with the minimum cost and allocation of RSUs to serve video streams across regions.

Keywords: Cloud-based internet of vehicles, Dynamic programming approach, Greedy algorithm, K-means clustering algorithm, Roadside units (RSUs)

I. INTRODUCTION

Today, the number of vehicles used in many countries has exceeded 1 billion [1], largely driven by economic and population growth, and is expected to reach 2 billion by 2035 [2]. Due to the increasing number of vehicles, it can raise traffic congestion and accidents on the roads. Until recently, the internet of vehicles (IoV) is emerged to solve the challenging issues in current transportation systems [3]. In IoV system, the intelligent sensor devices are installed in vehicles and traffic lights which can be able to absorb data from the environment and exchange information between vehicles to vehicles and vehicles to roads [4]. The use of IoV system can provide vehicle transportation safety and improve users' convenience [5]. However, data readings from sensors installed in vehicles and traffic lights need to be processed to become useful and meaning information. As a result, cloud computing has been emerged to provide scalable storage capacity and processing power for achieving value-added services such as data analytics, backup, database management system, respectively [6]. Therefore, data streamed from sensors can be transferred to the cloud for storing and processing in order to utilize value-added services. Then, data processed or stored by the cloud services can be remotely monitored and managed by the users through their Internet connected devices such as surveillance monitors, personal computers, smart phones, tablets, and set-top boxes.

However, although cloud computing provides a large pool of resources to use cloud services, serious network latency could be encountered when sensors with limited bandwidth transfer data to the cloud services over long distance [7]. As a result, the small amount of buffer memory allocated in sensors will be overflow and data can be lost. To address such problem, Cloud-based Internet of Vehicles framework in which stream aggregation (SA) approach is applied. The SA approach provides Roadside Units (RSUs) which are buffering servers with high speed network connectivity between sensors and cloud services [8]. An RSU can be allocated close to vehicles on the roads and collect the data from sensors and move to the cloud services with better bandwidth. In such case, RSUs are manufactured by some companies Savari, Fluidmesh Networks, Beijing Juli Science & Technology Co., Ltd., etc. [9].

Since the framework is done by RSUs, it is challenging to determine how many RSUs and where should be located. For the proposed issues, we applied dynamic programming optimization approach to minimize the total cost for allocating the optimal number of RSUs. Dynamic programming is problem solving technique which breaks down problem into simpler subproblems, solves each by one, stores the results, and finally combines the results [10]. After that, greedy algorithm is applied to prove the optimal number of RSUs found by dynamic programming. A greedy algorithm is problem solving approach which makes the best choice in every stage and finally lead to overall optimal solution [11]. Moreover, K-means clustering algorithm is utilized to find the optimal location for installing RSUs in areas of Internet of vehicles in the proposed framework. This algorithm groups data into clusters by assigning to the nearest centroid [12]. Then, we evaluate the proposed optimization approach and algorithms by performing numerical studies.

This research paper presents literature review in Section 2, research methods in Section 3, results in Section 4, and conclusion and discussion in Section 5.

II. LITERATURE REVIEW

Recently, earlier researchers have studied the combination of cloud computing and internet of vehicles. For example, a novel multilayered vehicular data cloud platform was proposed in [13] by using cloud computing and internet of vehicles to store and process transportation related information such as traffic control and management, car location tracking and monitoring, road condition, car warranty, and maintenance information for drivers. In [14], edge-cloud computing is deployed for autonomous vehicles to provide a large amount of computation resources. Cloud-assisted Internet of Things Intelligent Transportation System (CIoT-ITS) was proposed in [15] to handle traffic management's challenges.

Based on the papers, cloud computing has integrated with the internet of vehicles since it can provide the benefits of virtual servers that allow remote storage and computational capacities. Data stored in clouds can be retrieved from anywhere and can be shared among the vehicles. However, one critical issue of network scalability problem needs to be considered when many vehicles transfer data to cloud services over long range communication.

Dealing with these problems, the concept of roadside units (RSUs) is introduced. RSUs can perform as an aggregator which collects data from vehicles and transfers to the cloud for further storage and processing. In [16], aggregators are applied in a smart (electrical) grid to achieve scalable connectivity which can transfer data about metered power usage from many smart meters to the same target. In [17], a smart QoE-Aware Adaptive Video Bitrate Aggregation scheme is proposed for HTTP live streaming on smart edge computing to improve efficiency of live video streaming. In [18], the researchers presented roadside units (RSUs) which collect traffic data from vehicles and transmit them to the traffic control center. The article in [19] proposed intelligent transportation system with roadside unit (RSU) using the concept of internet of things. Based on the concepts established in previous research, this paper applies cloud computing technology and roadside units to enhance connectivity for Internet-connected vehicles.

Although previous studies have proposed cloud computing technology and the use of roadside units for the Internet of Vehicles, they have not addressed the unified challenge of simultaneously optimizing the number of roadside units and determining optimal deployment locations for roadside units. Different from the previous works, this paper utilizes dynamic programming approach for roadside unit (RSU) allocation, and the K-means clustering algorithm for RSU deployment to minimize the overall cost while meeting the demand of video streams within the proposed framework.

III. RESEARCH METHODS

In this section, we initially proposed the cloud-based internet of vehicles framework for traffic monitoring system as shown in Figure 1 where the bottom layer is internet of vehicles (IoV) that is a self-organized network by connecting vehicles which are equipped with wireless sensors in order to collect road data such as traffic jam, accident detection, post-accident investigation, and intersection collision, respectively. The collection of large amounts of road data needs to be processed to provide useful information for a wide variety of services (e.g., transport safety, traffic management, convenient driving, etc.). However, vehicle sensors are normally constrained by resources of storage and computation so that it becomes difficult to achieve those services. As a result, cloud computing is a very promising technology which can offer an excessive amount of storage resources and computation resources. By integrating cloud computing to the internet of vehicles, road data could be collected by vehicles and transmitted to the cloud for processing and in return, information could be broadcasted to vehicles for better traffic control and safety on the road.

In the framework, vehicles transmit road data to the traffic lights beside the road. However, traffic lights have limited wireless bandwidth so that undesirable network latency can be experienced for uploading data to the cloud. For this issue, Roadside Units (RSUs) are further installed between traffic lights and cloud infrastructure which can perform as stream aggregation approach which collects the data reported by traffic lights and send to the cloud for further analyzing and processing. After the cloud processes data, it can provide traffic information which can help drivers to choose the optimal routes by avoiding potential accidents, dangerous situations, and traffic congestion along the road.

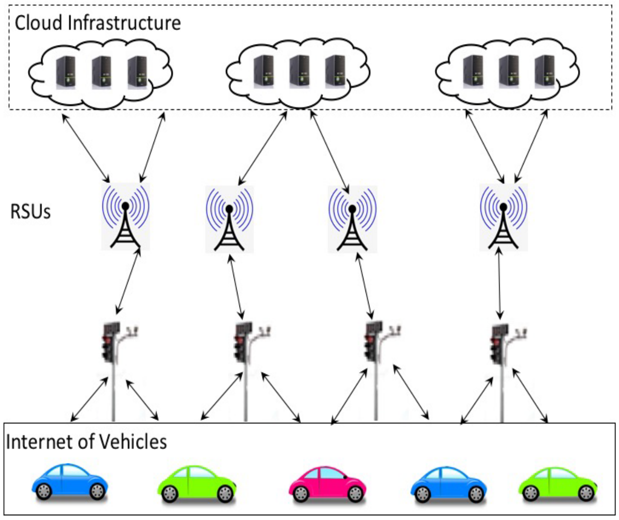

Figure 1: Cloud-based Internet of vehicles framework

Since RSUs are applied in the framework, the installation of RSUs is very expensive and it is difficult to determine the number of required RSUs to cover several vehicles on the road. Additionally, deploying RSUs along the road to ensure coverage of vehicles is also challenging. To solve these issues, a dynamic programming optimization approach is applied to obtain the optimal number of RSUs in the framework. Then, we implemented greedy algorithms to verify the optimal number of RSUs obtained by dynamic programming. Furthermore, K-means clustering algorithm is also applied to determine in which locations for deploying RSUs on the roadside. Then, we perform numerical studies to evaluate the optimal solutions.

A. Dynamic Programming Approach to Roadside Units Allocation

First, a dynamic programming approach is applied for determining the optimal number of Roadside Units (RSUs) with the minimum cost needed to handle video streams from traffic lights on the road. Each RSU has a specific amount of deployment cost and has maximum capacity to handle up to a certain number of video streams. Let streams $[i]$ represent the number of video streams generated by traffic lights i. Let $dp[i]$ is dynamic programming array which represents the minimum number of RSUs needed to handle video streams from traffic lights i.

B. Greedy Algorithm Approach to Prove Roadside Units Allocation

Greedy algorithm is also applied to prove the optimal number of Roadside Units (RSUs) obtained by dynamic programming approach.

Table 1: Greedy algorithm for optimal allocation of RSU

```
Input:
Video Streams: List of stream sizes
Max Capacity: Maximum capacity per RSU

Output:
Number of RSUs: Total RSU used

Step 1: Initialize variables
RSUs = [ ]
currentRSUCapacity = 0

Step 2: Sort video streams in descending order
sortedStreams = sort (videoStreams, descending)

Step 3: Iterate through video streams
  if currentRSUCapacity + stream <= maxCapacity:
    currentRSUCapacity += stream
    else:     RSUs.append(currentRSUCapacity)
currentRSUCapacity = stream

Step 4: Add the last RSU capacity if not empty
    if currentRSUCapacity > 0:
      RSUs.append(currentRSUCapacity)

Return the total number of RSUs allocated
Return length (RSUs)
```

As shown in Table 1, The greedy algorithm involves four main steps. First, the algorithm checks for each stream if it can accommodate into the existing RSU without exceeding the maximum capacity. Second, if the stream can fit into existing RSU without exceeding its capacity, it is added to that RSU. This is greedy because it utilizes the available resources of existing RSU before creating new ones. Third, if the stream doesn't fit into any existing RSU, the algorithm creates a new RSU for that stream. This algorithm iterates through each vehicle's streams from all traffic lights. After iterating through all video streams from all traffic lights, it calculates the total number of RSUs needed and the total cost.

C. K-means Clustering Algorithm for Roadside Units Deployment

Next, we implement K-means clustering algorithm as shown in Table 2 to find the optimal locations for installing RSUs along the roadway to meet video stream demands. Each RSU needs to cover specific number of video streams from traffic lights with minimal distance to improve connectivity.

Table 2: K-means clustering algorithm for optimal placement of RSU

```
Input: Traffic Lights = [{index: 0, demand: 0}, {index: 1, demand: 3}, …}

Step 1: Initialize RSUs at random traffic light positions
Initialize K RSU centers at random traffic light indices

Repeat until convergence:
Step 2: Assign traffic lights to closest RSU
For each traffic light:
        Find nearest RSU with remaining capacity
        Assign traffic light if RSU capacity allows
        Update RSU's remaining capacity

Step 3: Update RSU centers to centroid of assigned traffic lights
For each RSU with assigned traffic lights:
        Set RSU center to average position of its traffic lights

End Repeat
Output: Final RSU positions and assigned traffic lights
```

The input Traffic Lights array contains each traffic light's index and demand of video streams. We startplacing K RSUs at random traffic light positions along the road. These initial placements serve as RSU locations which will be adjusted in each iteration to cover traffic light demands. For each traffic light, the algorithm calculates the nearest RSU based on Euclidean distance. If an RSU has enough remaining capacity to serve the traffic light's demand, the traffic light is assigned to it. After assignment, the RSU's remaining capacity is updated for the new demand. If an RSU cannot accommodate to a traffic light due to capacity constraints, the algorithm tries to assign to the next closest RSU with available capacity. Once all traffic lights are assigned, each RSU is updated to the centroid of all traffic lights. The centroid means average position to traffic lights associated with that RSU, ensuring that each RSU center is optimally located to handle the data demands from traffic lights while minimizing travel distance between each RSU and traffic lights.

IV. RESULTS

First, we calculate the minimum number of roadside units (RSUs) required to handle video streams from 10 traffic lights using the dynamic programming approach. It is assumed that there are 10 video streams from 10 traffic lights streams[i] = [3,2,4,1,5,2,3,4,1,2]. Each RSU costs \$100 and it can handle up to 5 video streams.

For each traffic light i, the minimum number of RSUs required to process streams from traffic lights i can be calculated using the formula:

$$dp\_i = \min(dp[j] + 1), where \sum_{k=j}^{i} S_k \leq C \tag{1}$$

- $dp[i]$ represents the minimum number of RSUs needed to handle the video streams from traffic light i.
- $dp[j] + 1$ accounts for the new RSU to cover video streams from previous traffic light j to traffic light i.

- $\sum_{k=j}^{i} S_k \leq C$ ensures that the number of streams from traffic light j to i does not exceed the capacity of a single RSU. If this condition is met, then a single RSU can handle the streams from traffic light j to traffic light i.

Then, cumulative streams are calculated to get the number of streams between any two traffic lights.

streams[i] = [3,2,4,1,5,2,3,4,1,2] (2)

Cumulative_streams = [3,5,9,10,15,17,20,24,25,27]

Based on the numerical results in Table 3, 6 RSUs are needed to process the video streams from 10 traffic lights. After we obtain the required number of RSUs, we calculate the minimum total cost using the formula:

Total Cost = RSUs x Cost per RSU (3)

Total Cost = 6 x \$100 = \$600

The minimum total cost for installing the optimal number of RSU required to process the video streams from traffic lights is \$600. Then, we prove this optimal solution using greedy algorithm. Greedy algorithm is implemented in Java program to allocate videos streams from traffic lights to RSUs efficiently. The goal is to minimize the number of RSUs needed while ensuring each RSU does not exceed its capacity. Figure 2 shows the generated output of greedy algorithm.

Traffic Light Index	Video Streams	RSU Allocation Status
1	3	New RSU 1 (Total: 3)
2	2	Added to RSU 1 (Total: 5)
3	4	New RSU 2 (Total: 4)
4	1	Added to RSU 2 (Total: 5)
5	5	New RSU 3 (Total: 5)
6	2	New RSU 4 (Total: 2)
7	3	Added to RSU 4 (Total: 5)
8	4	New RSU 5 (Total: 4)
9	1	Added to RSU 5 (Total: 5)
10	2	New RSU 6 (Total: 2)

RSU Management Summary
Total RSUs needed: 6 Total cost: \$600

Figure 2: Numerical results of greedy algorithm

As shown in Figure 2, traffic light indexes 1, 3, 5, 6, 8, and 10 require new RSUs because the streams cannot be accommodated into the existing RSUs. Traffic light indexes 2, 4, 7, and 9 can add streams to existing RSUs to utilize the available resources. Finally, the program displays the same numerical results that 6 RSUs are required with the overall total cost of deployment of \$600.

To determine the areas where 6 RSUs should be installed based on the vehicle stream data, K-means clustering approach is applied. Based on the K-means clustering approach, we distribute the traffic light indices into clusters and each cluster represents an RSU location along the road. The clustering result for 6 RSUs is presented in Table 4 (in the next page).

Table 3: Numerical results for optimal number of RSUs

Traffic Light Index i	Video Streams	Cumulative Streams	Formula	Calculation	RSUs Needed
1	3	3	dp[1]=dp[0]+1	dp[1]=0+1=1	1 (Fits in RSU 1)
2	2	5	dp[2]=dp[1]+0	dp[2]=1+0=1	1 (Fits in RSU 1)
3	4	9	dp[3]=dp[2]+1	dp[3]=1+1=2	2 (New RSU needed)
4	1	10	dp[4]=dp[3]+0	dp[4]=2+0=2	2 (Fits in RSU 2)
5	5	15	dp[5]=dp[4]+1	dp[5]=2+1=3	3 (New RSU needed)
6	2	17	dp[6]=dp[5]+1	dp[6]=3+1=4	4 (New RSU needed)
7	3	20	dp[7]=dp[6]+0	dp[7]=4+0=4	4 (Fits in RSU 4)
8	4	24	dp[8]=dp[7]+1	dp[8]=4+1=5	5 (New RSU needed)
9	1	25	dp[9]=dp[8]+0	dp[9]=5+0=5	5 (Fits in RSU 5)
10	2	27	dp[10]=dp[9]+1	dp[10]=5+1=6	6 (New RSU needed)

Table 4: Clustering results for optimal location of RSUs

RSUs	Traffic Light Indices	Total Streams	Cluster Center Position (Centroid)
1	1, 2	5	(1+2) ÷2 = 1.5
2	3, 4	5	(3+4) ÷2 = 3.5
3	5	5	5
4	6, 7	5	(6+7) ÷2 = 6.5
5	8, 9	5	(8+9) ÷2 = 8.5
6	10	2	10

In Table 4, it describes 6 RSUs, traffic light indices covered each RSU, total video streams managed by each RSU while meeting the RSU capacity constraints, and approximate road position or centroid of each RSU by averaging traffic light indices within each cluster. The results indicate to deploy RSU 1 between traffic light index 1 and 2, RSU 2 between traffic light index 3 and 4, RSU 3 at traffic light index 5, RSU 4 between traffic light index 6 and 7, RSU 5 between traffic light index 8 and 9, and RSU 6 at traffic light index 10.

Each RSU location serves a cluster of traffic light indices with a demand sum not exceeding 5 streams, achieving optimal placement along the road. The centroids indicate where each RSU should be installed to minimize distances and ensure coverage. With 6 RSUs, this clustering balances the demand in RSUs while meeting capacity requirements.

In this work, we applied centroid based K-means clustering algorithm to determine optimal RSU locations. This algorithm is suitable for stable stream demand and equally distributed clusters. However, it is more reasonable if the video streams are uncertain and non-uniform in real world vehicular environment. In this scenario, other algorithm such as Gaussian Mixture Model (GMM) is more suitable and flexible in variably sized demand regions. Compared to K-means and GMM, K-means can allow each stream to go to one RSU while GMM can share the streams between RSUs based on the distance between vehicles and RSUs.

V. CONCLUSION AND DISCUSSION

This research proposes the framework of Cloud-based Internet of Vehicles (IoV) which integrates Roadside Units (RSUs) to enhance traffic monitoring and ensure driver safety. The incorporation of RSUs facilitates the buffering of video streams from traffic lights, leading to a significant reduction in latency when transferring data from traffic lights to cloud services. In addition, we applied dynamic programming approach for cost-effective RSU allocation and greedy algorithm to prove the optimal solution for RSU allocation obtained by dynamic programming. K-means clustering algorithm is also applied for optimal placement of RSUs. According to the results and findings, the framework not only optimizes the allocation of RSU resources but also enhances the placement of RSUs in proximity to traffic lights in order to minimize distances and ensure coverage, hence improving the efficiency of traffic data management.

For the future work, we will focus on exploring resource allocation for RSUs under the uncertainty demand of video streams from traffic lights. This investigation aims to further enhance the framework's adaptability to dynamic traffic conditions and demand fluctuation. This research is a foundational step toward advancing IoV applications and developing intelligent transportation systems that prioritize safety and efficiency.

REFERENCES

[1] United States International Trade Commission, "Passenger vehicles: Summary of trends summary," 2013. [Online]. Available: https://www.usitc.gov/publications/332/pub_ITS_09_PassengerVehiclesSummary5211.pdf

[2] Global Growth Insights. "Automobile Market." GLOBALGROWTHINSIGHTS.com. https://www.globalgrowthinsights.com/market-reports/automobile-market-109995 (accessed Jun. 4, 2025).

[3] R. Dhanare, K. K. Nagwanshi, S. Varma, and S. Pathak, "The future of internet of vehicle : challenges and applications," in *Proc. Int. Conf. Comput. Perform. Eval. (ComPE)*, Shillong, India, Dec. 2021, pp. 023–026.

[4] F. A. Butt, J. N. Chattha, J. Ahmad, M. U. Zia, M. Rizwan, and I. H. Naqvi, "On the integration of enabling wireless technologies and sensor fusion for next-generation connected and autonomous vehicles," *IEEE Access*, vol. 10, pp. 14643–14668, 2022.

[5] M. B. Mollah *et al.,* "Blockchain for the Internet of Vehicles towards intelligent transportation systems: A survey," *IEEE Internet Things J.*, vol. 8, no. 6, pp. 4157–4185, 2021.

[6] R. Cherekar, "Cloud-based big data analytics: Frameworks, challenges, and future trends," *Int. J. AI, Bigdata, Comput. Manage. Stud*., vol. 4, no. 1, pp. 35–46, 2023.

[7] J. Beuchert, F. Solowjow, S. Trimpe, and T. Seel, "Overcoming bandwidth limitations in wireless sensor networks by exploitation of cyclic signal patterns: An event-triggered learning approach," *Sensors*, vol. 20, no. 1, 2020, Art. no. 260.

[8] S. Akhavan Bitaghsir and A. Khonsari, "Modeling and improving the throughput of vehicular networks using cache enabled RSUs," *Telecommun. Syst.*, vol. 70, pp. 391–404, 2019.

[9] Valuates Reports, "Global Market Share and Ranking, Overall Sales and Demand Forecast 2024-2030," 2024. [Online]. Available: https://reports.valuates.com/market-reports/QYRE-Auto-7R15191/global-roadside-unit-rsu

[10] D. Liu, S. Xue, B. Zhao, B. Luo, and Q. Wei, "Adaptive dynamic programming for control: A survey and recent advances," *IEEE Trans. Syst., Man, Cybern.: Syst.*, vol. 51, no. 1, pp. 142–160, 2021.

[11] D. Jungnickel, "The greedy algorithm," in *Graphs*, *Networks and Algorithms*, 2nd ed. Berlin, Germany: Springer, 1999, ch. 5, pp. 129–153.

[12] R. V. Singh and M. S. Bhatia, "Data clustering with modified K-means algorithm," in *Proc. 2011 Int. Conf. Recent Trends Inf. Technol. (ICRTIT)*, Chennai, India, Jun. 2011, pp. 717–721.

[13] N. S. Rajput *et al*., "Amalgamating vehicular networks with vehicular clouds, AI, and big data for next-generation ITS services," *IEEE Trans. Intell. Transp. Syst.*, vol. 25, no. 1, pp. 869–883, 2024.

[14] Y. Sasaki *et al*., "An edge-cloud computing model for autonomous vehicles," in *Proc. 11th Int. Workshop Planning, Perception, Navigation Intell. Vehicle*, Macau, China, Nov. 2019, pp. 99–104.

[15] C. Liu and L. Ke, "Cloud assisted Internet of Things intelligent transportation system and the traffic control system in the smart city," *J. Control Decis.*, vol. 10, no. 2, pp. 174–187, 2023.

[16] S. Tonayali, K. Akkaya, N. Saputro, A. S. Uluagac, and M. Nojoumian, "Privacy-preserving protocols for secure and reliable data aggregation in IoT-enabled smart metering systems," *Future Gener. Comput. Syst.*, vol 78, pp. 547–557, 2018.

[17] X. Ma *et al.*, "QAVA: QoE-aware adaptive video bitrate aggregation for HTTP live streaming based on smart edge computing," *IEEE Trans. Broadcast.*, vol. 68, no. 3, pp. 661–676, 2022.

[18] P. K. N. Kouonchie, V. Oduol, and G. N. Nyakoe, "Roadside units for vehicle-to-infrastructure communication: An overview," in *Proc. Sustain. Res. and Innov. Conf.*, Nairobi, Kenya, Oct. 2021, pp. 69–72.

[19] P. Pyykonen, J. Laitinen, J. Viitanen, P. Eloranta, and T. Korhonen, "IoT for intelligent traffic system," in *IEEE 9th Int. Conf. Intell. Comput. Commun. and Process. (ICCP)*, Cluj-Napoca, Romania, Sep. 2013, pp. 175–179.