International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9s
DOI: https://doi.org/10.17762/ijritcc.v11i9s.7423
***Article Received**: 08 May 2023 **Revised:** 30 June 2023 **Accepted:** 24 July 2023*

Real-Time IoV Task Offloading through Dynamic Assignment of SDN Controllers: Algorithmic Approaches and Performance Evaluation

Divya Lanka[1], Selvaradjou Kandasamy[2]
[1]Research Scholar: Computer Science and Engineering
Puducherry Technological University
Puducherry, India
l.divya44@ptuniv.edu.in
[2]Professor: Computer Science and Engineering
Puducherry Technological University
Puducherry, India
selvaraj@ptuniv.edu.in

Abstract—Task offloading in Internet of Vehicles (IoV) is very crucial. The widespread use of IoT applications frequently interacts with the cloud, thereby increasing the load on centralized cloud controllers. Centralized network management in cloud infrastructure is not feasible for the latest IoT trends. Decentralized and decoupled network management in Software Defined Networks (SDN) can enhance IoV services. SDN and IoV coupling can better handle task offloading in ubiquitous and dynamic IoV environments. However, appropriate SDN controller assignment and allotment strategies play a prominent role in IoV communication. In this study, we developed algorithms for SDN controller assignment and allotment namely 1) Next Fit Allotment and Assignment of SDN Controller in IoV (NFAAC), 2) Dynamic Bin Packing Allotment and Assignment of SDN Controller in IoV (DBPAAC), and 3) Dynamic Focused and Bidding Allotment and Assignment algorithm of SDN Controller in IoV (DFBAAC). These algorithms were simulated using open-flow switch controllers. The controllers were modeled as Road Side Units (RSU) that can allocate bandwidth and resource requirements to vehicles on the road. Our results show that our proposed algorithm works efficiently for SDN controller assignment and allocation, outperforming the existing work by a significant improvement of 13.5%. The working of the proposed algorithms are verified, tested, and analytically presented in this study.

Keywords-SCPP; Task offloading; real time scheduling; SDN controller allotment; IoV.

I. INTRODUCTION

Confinement to legacy networks is moving network trends towards network virtualization and Software-Defined Networks (SDN). SDN provides a path for innovative decentralized network administration models by removing barriers to adaptation to dynamic network conditions. The abstraction of a network from the viewpoint of an SDN controller needs to maintain salient features, such as agility, reliability, flexibility, and scalability, over traditional legacy networks. IoV has evolved from vehicular ad hoc networks (VANETs) with the escalation of 5G technologies, SDN, artificial intelligence, cloud, fog, and edge computing technologies. The IoV offers flexible services for vehicular communications across heterogeneous networks. Proper network management and real-time service demands in IoV require efficient handling of the entire network [1]. Network Function Virtualization (NFV) in SDN decouples the data layer from the control layer, meeting the requirements of IoV ubiquitous environments. However, the assignment and allocation of controllers in SDN plays a crucial role in achieving high performance and Quality of Service (QoS) [2].

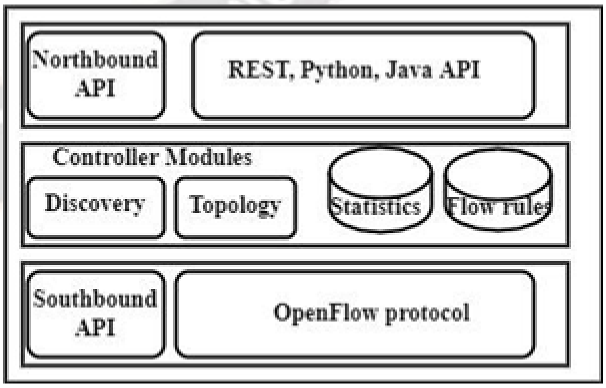

Figure 1. SDN controller functionalities.

VANETs have evolved into IoV with advanced communication facilities in real-time. Acceleration of IoV has resulted in vehicle-to-everything communication (V2X). 5G, New Radio V2X (NR V2X) telecommunication technologies with millimeter wave (mmWave) technology, and Dedicated Short-Range Communications (DSRC) are implemented in IoV to achieve connectivity for V2X communication. DSRC enables limited distance communication between vehicles and Road Side units (RSU). Mobile telecommunication technologies such as 5G provide long-distance communication between vehicles and cloud servers. Cellular V2X can handle VANET challenges by integrating SDN, fog and edge computing [3] [4]. The imperative demands of the IoV include low latency, data dissemination, reliable communication, scalability, and limited congestion. IoV provides quality services to users using the latest technologies, with minimum expenditure on Intelligent Transportation Systems (ITS). The traffic in IoV networks varies geographically; it is delay sensitive and requires more reliability. IoV acquires data from on-board sensors, global positioning systems (GPS), neighboring vehicles, mounted cameras, and surroundings for optimal routing, traffic management, navigation, and control.

IoV can be integrated with SDN to satisfy the advancements in vehicular communications. There are many advantages to integrating SDN into an IoV environment, but this cannot be achieved directly. The integration of SDN with IoV adheres to the proper assignment and allotment of SDN controllers [5]. SDN for the IoV (SDN-IoV) requires real-time controller positioning and allotment techniques to yield better results in real-time traffic handling between various planes of SDN.

The various open research issues in SDN integrated with IoV include integrity, scalability, resilience, reliability, device security, controller placement, routing data from the data to the control plane, congestion, load balancing, standardization, communication latency, vulnerability of the controller, and energy consumption [6]. Among the various issues in the SDN environment, the SDN Controller Placement Problem (SCPP) is a major issue that must be addressed because SDN controllers are crucial in SDN. Improper placement of the SDN controller results in increased delay among the control and data planes. The SCPP is very prominent when migrating from legacy networks to complete SDNs, withholding all quality parameters. Positioning the SDN controller directly affects the network efficiency and resource utilization by optimizing the cost. The controller frames the flow rules and controls the whole SDN and SDN devices. The flow rules govern the controller to execute routing, load balancing, and forward data for applications that use the controller to communicate in a data plane.

The Figure 1. shows the core functionalities of the SDN controller for both southbound and northbound Application Programming Interfaces (API). The southbound API is standardized and well-defined compared to the northbound API. OpenFlow is an example of standardization for a southbound API. The lack of a controller for northbound API standardization is the main shortfall of SDN [7]. A list of definitions in this study is provided in Table 1.

TABLE. 1 List of definitions

Term	Definition
Critical job	A job is said to be critical in IoV scenarios as the one that has the highest priority
Normal job	A job is said to be normal in IoV scenarios as the one that has normal priority in terms of execution.
Periodic job	A job is defined as periodical, if it occurs for every specific period of time.
Non Periodic job	A job is defined as non-periodic, if it occurs spontaneously without restriction on the time period.
Job precedence	A job will have higher precedence, if it is important or it should meet its deadline hardly.
RSU Switches	Handles assignment of jobs to controllers and allotment of controllers to the data plane nodes.
SDN Controller	The controller that handles the jobs arriving from data plane, and manages all the RSU switches

The key contributions of this study are the design and implementation of algorithms for SDN controller positioning and allotment in an IoV environment. The rest of this paper is structured as follows. In Section 3, various studies related to the SCPP problem in IoV are discussed. Section 4 deals with the problem statement and system model along with the mathematical models used in this study. In Section 5, the implementation of algorithms for SDN controller switches is discussed. Finally, Section 6 concludes the paper with a summary of the key contributions and discusses avenues for future enhancement.

II. LITERATURE SURVEY

The rise in IoV applications has increased communication with cloud infrastructure. Frequent communication and computation with clouds in vehicular networks require critical network management. Hence, implementing SDN in IoV can better manage vehicular data traffic with centralized network control. Scalability and controller placement problems arise within SDN with centralized control [8]. The solution for the SCPP should be dynamic and optimal for dense and pervasive IoV traffic [9], [10], [11]. Several solutions have been proposed for handling SCPP in vehicular networks. Various techniques, such as brute force, heuristic, greedy-based, clustering, linear programming, quadratic programming, bio-inspired, genetic algorithms, and simulated annealing, have been implemented to solve the SCPP problem. Most of the studies were conducted on SDN wide-area networks and VANETs, and few studies have been conducted on wireless sensor networks. Energy price and energy consumption modelling are also important when designing an SDN environment [12] [13].

A. Static controller placement strategies for SCPP problem

Placing controllers farther from the data plane introduces communication delay. Positioning the controllers in a hierarchical manner in a vehicular network limits the delay when compared with the centralized placement of controllers [14]. Ramya et al. in their work considered the SCPP in wireless sensor networks to be an NP-hard problem and provided a solution for NS 3.25. The authors used graph theory to determine ideal clusters and Pareto search to identify the location of the controllers. The authors of their other work suggested a migration for connection failures and load balancing in the network. The controller location is also essential for deploying SDN controllers in their work. An array is maintained to observe the free controllers, and the controllers are then allocated to the requirement. The works have a drawback that allocation of controllers was not dynamic and there is no caching of critical flow rules at the controllers [15] [16].

The centralized controller should carefully handle all traffic flows coming from the southbound. A multi-attribute decision-making algorithm to order identical data flows was used to allot flow paths to controllers [17]. One drawback of this study was that controller failure may affect the entire network and work in fixed-flow traffic. Multiple controllers are necessary to handle large traffic flows and to attain fault tolerance. It is also important to safeguard data traffic flow from the RSU controller.

B. Dynamic controller placement strategies for SCPP problem

A dynamic controller placement strategy was proposed in [9] using a multi-controller solution for the SCPP problem to control the delay in communication. In this study, local controllers were executed on the selected RSU’s and the main controller was executed on the cloud using MATLAB. The authors used linear programming to adjust the controllers dynamically by predicting and monitoring the real-time mobility traffic using the Barabasi-Albert topology. The limitation of this study is that there is no critical flow-rule storage at RSU’s. Wang et al. provided a solution by treating the problem as a facility location issue in vehicular networks using decentralized controller placement at RSU’s. In this the authors considered the trade-off between data transmission and placement of controller costs. The authors considered a wired connection between controllers to retain reliable and secure services, as simulated in MATLAB [10]. A limitation of this study is that the wired connectivity is unsuitable for controller failure situations. Toufga et al. solved the SCPP problem using linear programming by dynamically allocating a controller to the vehicular scenarios. The authors observed a reduction in replacement cost and latency in periodic and event-based intervals compared to static controller placement architectures [11]. The limitation of this study is that it requires more communication overhead between controllers.

In [8], the authors focused on the scalability of SDN by initially identifying the number of clusters in which the network could be divided into partitions using silhouette and static gap analysis. After identification, they used a density-based clustering algorithm and allotted controllers to each cluster in MATLAB. They optimized the latency and capital expenditure. The disadvantage of this scheme is that it runs at a slower speed and has static controller placement in nature.

C. Bio-inspired techniques for SCPP problem

Many controller placement strategies have been proposed for static and simple networks, using bio-inspired techniques. The authors of [18] proposed an SCPP solution for dynamic networks called the Improved Multi-objective Artificial Bee Colony (IMABC) technique. The authors implemented the technique in MATLAB and Python using Simulation of Urban Mobility (SUMO) for selecting and switching controllers based on real-time vehicular traffic. The IMABC limits energy utilization by selecting the required number of controllers that cover the specified region in the IoV network. This implementation results in better energy usage and minimal delay and jitter; however, network congestion may still occur. The authors of [19] developed an optimized bee Density Peak Clustering (DPC) algorithm in 5G networks to place controllers at central locations by dividing the area into subareas. The authors considered the proximity of the nodes minimize the delay. The limitation of this work is that the network density and load at the SDN controllers are not considered. The authors in [20] developed a solution for SCPP using feedback control theory, which is a variation of varna optimization. The authors considered packet loss, switches managed by each controller, and delays in propagating data from the data plane. The work was not experimented with to prove the scalability in terms of switches and obtainable loads.

D. Edge and fog computing solutions for SCPP problem

Some solutions given by the authors in [11], [18], [21], [22] used edge and fog computing in IoV to reduce computational and communication delays, supporting high mobility, tolerance to controller failure, and enhanced QoS. The position of controller placement affects the overall SDN performance. Syed et al. integrated cloud computing and fog computing in VANETs to limit the latency in message transmission. The authors prioritized the messages on the safety factor to reduce latency and time to give responses [21]. In this study, the authors considered the static controller placement and did not focus on message security. Deng et al. focused on latency in SDN and proposed designs for the transition from software-defined vehicular networking to software-defined edge-up designs in vehicular networks. Channel access is scheduled when the network is stable, and randomly accessed when it is unstable [4]. Li et al. provided a solution to the SCPP using edge computing with the louvian algorithm. To spot communities in the network, the authors used a delay metric between vehicles and controllers at the edge of the network. After forming the communities, the delay between the edge and SDN controllers is considered. Chunhui et al. [23] built a two-layer fog-based architecture for IoV to manage the critical tasks. Job completion is maximized by offloading critical jobs dynamically to the static and mobile fog nodes in an IoV network. This work is suitable for less-time-critical jobs, but not for more-time-rigid jobs. Shi et al. [24] presented job offloading for IoV jobs using an auction scheme on edge-computing nodes and dynamic programming to match the job time with available resources. The authors reduced this delay by effectively using resources. The limitation of the algorithm is that the authors did not consider the mobility of nodes between the cell towers. The authors of [25] introduced a secure protocol for SDN VANETs based on elliptic curve cryptography and hashing to provide security between the RSU and SDN controller using the Scyther tool. However, it works only for single V2V communications and not for multiple vehicles.

mmWave offers a large bandwidth for fast and high-speed data communication. mmWave technology has been adapted in new-generation vehicular networks to guarantee higher data rates and reliability in vehicular networks [26]. Millicar, a new open-source NS3 module, allows the simulation of NR V2X communication in the mmWave bands. The Millicar module was designed according to the NR V2X physical and MAC layers [27].

It is evident from the literature review that many researchers have proposed different solutions to the SCPP problem. However, we assume that in IoV, roads have different properties, such as curves, heavy traffic, and peak hours for traffic. Therefore, a specific fixed positioning of the controller is not feasible for this type of heterogeneous road condition. In this study, we propose placing controllers in a random manner. Controller assignment with effective task offloading is vital. In IoV, how efficiently traffic information is exchanged within a short transmission time to provide more accuracy and less delay for data sharing is important [29], [30]. Hence, this study has two objectives: (i) dynamic job assignment and (ii) dynamic allotment of SDN controller resources to satisfy the real-time constraints. Here, the term ‘assignment’ of the controller refers to offloading an incoming job to the controller, and ‘allotment’ refers to spending the controller resources to execute the job by the controller. In this study, we differentiated vehicular traffic into normal and critical jobs by relying on deadlines, priorities, computation time, and the nature of the jobs. These jobs are executed by real-time scheduling algorithms and then the logic of the proposed algorithms is emulated at SDN controllers on NS-3 openflow switches.

III. SYSTEM MODEL

In the IoV, vehicles join and leave the network at any time. Because a vast number of vehicles move at faster speeds and higher rates, mobility management, communication latency, and energy prices have become major issues that deteriorate bandwidth usage and QoS in networks [28]. Therefore, SDN coupled with IoV is best suited to vehicular environments for optimizing these factors. Simultaneously, proper SDN controller assignment and allotment algorithms are mandatory to handle incoming vehicular traffic better.

The open-flow controllers in the control plane handle the data traffic generated by vehicles. Because a single controller cannot handle all data units, multiple controllers are installed without compromising the bandwidth for routing constraints. All devices were connected to the respective controllers at the controller plane. Recently, SDN controller assignment and allotment have been considered crucial problems in IoV for vehicular datacenters. Vehicular datacenters are powered by many edge switches, virtual switches, and SDN controllers. In this study, we propose a multi-controller assignment and allotment strategy to handle controller allotment with efficient data handling. The proposed method should reduce congestion, latency, jitter, hop count, and energy consumption, thereby increasing the resilience and reliability of the SDN controllers.

A defined strategy for SDN controller assignment and allotment algorithms was proposed in this work. The strategy selection for the assignment and allotment of controllers depends on the volume of traffic, peak times of traffic. IoV applications benefit from dynamic controller allotment strategies, in which data rates are not known beforehand.

In the control plane, open-flow controllers handle the data traffic generated by the vehicles. A single controller cannot handle all incoming data traffic; hence, as shown in Figure 2. multiple controllers were installed without compromising the bandwidth for routing constraints. All vehicles were connected to the respective controllers at the controller plane.

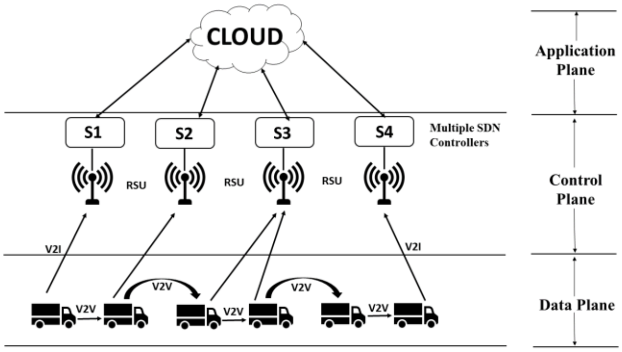

Figure 2. System model with multiple controllers at control plane

A. Assumptions

In this study, two types of jobs were used: critical and normal. Critical jobs for a vehicular network need immediate and time-critical attention; hence, they are called critical jobs, and normal jobs do not need any critical attention. Accident information or collision information, routing information, and video crowdsourcing are categorized under critical jobs, as this information requires faster connectivity, intra- and inter-vehicle communication, and RSU. Likewise, all assumptions in the proposed solution for critical jobs are listed in Table 2.

TABLE 2. Assumptions for critical jobs

Critical Jobs (CT)	**Other Vehicles**	**RSU**	**Own vehicle**	**Connectivity**
Accident information	YES	YES	YES	YES
Route information	NO	YES	YES	NO
In vehicle alert	NO	NO	YES	YES
Assistance to lane change	YES	YES	NO	NO
Video crowd-sourcing	NO	NO	YES	YES

Normal jobs such as business services, infotainment, alarms, and fuel shortages should consider factors such as connectivity, communication with itself, and other vehicles and RSU's. For example, business services need not communicate with other vehicles; they do not need to communicate over the RSU, but this job needs to communicate with their own vehicles and needs internet connectivity. The assumptions made in this study for the various types of jobs are listed in Table 3.

TABLE 3. Assumptions for normal jobs

Normal jobs (NT)	**Other Vehicles**	**RSU**	**Own vehicle**	**Connectivity**
Business services	NO	NO	YES	YES
Infotainment	NO	NO	YES	YES
Alarm	NO	YES	YES	NO
Fuel shortage	YES	YES	YES	YES
Air conditioning issue	NO	NO	YES	YES
Over heating	YES	NO	YES	YES

The values for the jobs assumed in Table 2 and Table 3 are assigned with the Worst Case Execution Time (WCET) in milliseconds, deadline to complete the execution of job, type of job (periodic or non-periodic), what type of resources need to execute the job, and priority assigned for each job (high or low). The values for critical and normal jobs were specified according to the following criteria, and are listed in Table 4.

TABLE 4. Values for critical and normal jobs

Job (Ti)	**WCET**	**Deadline**	**Type**	**Resources**	**Precedence**
Business services	4	Delayed	Non periodic	Passive	Moderate
Infotainment	3	Delayed	Periodic	Active	Moderate
Alarm	2	Critical	Non Periodic	Active	High
Fuel shortage	3	Delayed	Non periodic	Passive	High
Air conditioning issue	4	Delayed	Non Periodic	Active	Low
Over heating	2	Critical	Non periodic	Active	High
Collision of vehicles	1	Critical	Non Periodic	Passive	High
Accident information	1	Critical	Non Periodic	Active	High
Route information	1	Delayed	Both	Periodic	Moderate
In vehicle alert	2	Critical	Active	Periodic	Moderate
Assistance to lane change	2	Delayed	Periodic	Passive	Moderate
Breaking the car	2	Critical	Non Periodic	Active	High

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9s
DOI: https://doi.org/10.17762/ijritcc.v11i9s.7423
***Article Received**: 08 May 2023 **Revised:** 30 June 2023 **Accepted:** 24 July 2023*

In this study, three algorithms are proposed to handle classified jobs in an SD-IoV environment. The algorithms proposed in this work are the Next Fit Allotment and Assignment algorithm of SDN Controller in IoV (NFAAC), Dynamic Bin Packing Allotment and Assignment algorithm of SDN Controller in IoV (DBPAAC), and Dynamic Focused and Bidding Allotment and Assignment algorithm of SDN Controller in IoV (DFBAAC). All jobs were executed on the controller. The NFAAC algorithm is scheduled for normal jobs because normal car operations are known and no other issues arrive dynamically. Critical jobs were scheduled using the DBPAAC. DBPAAC is a dynamic scheduling algorithm that prioritizes each job and assigns priorities to each critical job. The DFBAAC algorithm is used to handle both critical and normal jobs by considering the controller utilization. DFBAAC works upon bidding and is best suited to dynamic environments.

The following notations were made for all the three dynamic placement algorithms:

a_i- Arrival time of the tasks
T_i – Jobs/tasks in the system
P_i- Period of the task
p_i- Controllers
dl_i- Deadline of the tasks
e_i- Execution time of the tasks
R_i- Response time
CT- Critical Tasks given in Table 3.
NT- Normal Tasks specified in Table 4.
RB - Request for Bidding
B_i- Time after the requesting process to examine the bids
W_i- Waiting time of the task (after arrival, the task waits to get the processor).
U- Utilisation of the processor
n- number of jobs
pb_i- Probability of scheduling the tasks is high

The Utilisation factor (U) is calculated based on the equations

$$U = \sum_{i=0}^{n} ei/pi < 1 \quad (1)$$

$$U = n(2^{\frac{1}{n}} - 1) < 1 \quad (2)$$

Algorithm 1 Next Fit Allotment and Assignment of SDN Controller in IoV (NFAAC)

Require: Let T_1, T_2, T_3 ... T_n be the Jobs from the data plane, N_1, N_2,... N_i be nodes in the data plane, C_1, C_2, C_3...C_i be the controllers

Ensure: The jobs (T_i) should be executed successfully by decentralized static allotment of controllers using class bounds

```
NFAAC initialization:
while $T_i \leq U(C_i)$
    do
        $T_i$ schedules with $C \in C_i$
        Check for job execution and allocate job to class
        if $T_i \in C_i$ then
        Assign $T_i$ to $C_i$
        Execute the job
        else
        $C_i = C_{i+1}$, $T_i$ scheduled with $C_i$ +1
        Adding new class of controller
        end if
end while
```

$$Wi(t) \leq \tau i \quad (3)$$

$$Wi(t) <= \sum_{j=1}^{i} e_i \lceil \frac{t}{P_j} \rceil \quad (4)$$

where W is the weight assigned and t is the period of the task computed while running the RM algorithm given in Equation 4. The above equations are solved only for a uni-controller system, but all the controllers of the IoV system should exhibit multi-controller scheduling.

In the DBPAAC algorithm provided in Algorithm 2, the tasks listed in Table 4 were scheduled according to the execution time and task period. The tasks were pre-emptive and independent. This period was the same as the relative deadline of the tasks. The tasks require only controller time for completion. The tasks will be scheduled and executed on the processor provided that the processor usage does not exceed 100%; in such a case, the tasks will be scheduled to another processor.

Algorithm 2 Dynamic Bin Packing Allotment and Assignment of SDN Controller in IoV (DBPAAC)

Require: Let T_1, T_2, T_3 ... T_n be the Jobs from the data plane, N_1, N_2,... N_i be nodes in the data plane, C_1, C_2, C_3...C_i be the controllers

Ensure: The jobs (T_i) should be executed successfully by the decentralized dynamic allotment of controllers

```
    while $T_i$ arrives at $L_c$ do Check for $T_i$ execution and identify
$N_i$ with enough resources
    if $\frac{e_i}{p_i} = u_i \leq 1$
    then
      $C_i \Leftarrow min(D_i)$
      $T_i \in L_c$, schedule
     else
      $T_i$ schedule to $C_{i+1}$
    end if
    end while
```

The controller utilization is found using Equation 5.

$$p_i = \sum_{0}^{P_i} u_i < U < 1 \quad (5)$$

Where i= {0 ,1,2,3....T }

Not all processors are loaded with 100% CPU, but if a task cannot be accommodated by one processor, it is scheduled to another processor. This system works well for newly arrived tasks. When new vehicles enter the RSU zone, they can obtain information from the RSU (which is modelled as an SDN controller) regarding safety messages and bandwidth provisioning. If a particular RSU cannot service the job beyond its full load, the request is processed by the next RSU, and the process continues.

In the DFBAAC algorithm mentioned in Algorithm 3, the critical tasks listed in Table 2 are scheduled by processor pi based on resource availability and processor time. However, when a normal task arrives, as mentioned in Table 3, which is not critical, the processor that encounters Ni may attempt to run based on the calculation of resources and Ti. Otherwise, pi transfers the task to other processors in the system. If any processor has sufficient resources, it can handle Ni comfortably. For Ni, the bidding process runs under the schedule. The best bidder performs the task comfortably.

Algorithm 3 Dynamic Focused and Bidding Allotment and Assignment Algorithm of SDN Controller in IoV (DFBAAC)

```
Require: Let $T_1, T_2, T_3 ... T_n$ be the Jobs from the data plane, $N_1, N_2, ... N_i$ be nodes in the data plane, $C_1, C_2, C_3...C_i$ be the controllers
Ensure: The jobs ($T_i$) should be executed successfully by decentralized controllers
Focused Algorithm initialization:
while $T_i \subset C_i$ do
    if (($T_i$ =true) && $C_i$ =MAX(Resources))
        $T_i \subset L_c$
        $T_i$=TRUE
    else
        $T_i$=FALSE
    end if
    end while
    Bidding Algorithm initialization;
    while $T_i \subset C_i$ do
        Check for k controllers with a surplus amount of resources, The number of controllers were taken for successful completion of all jobs. Then send RB
    if (likelihood (Ci)==true) then
        $T_i \in C_i$ with best bid;
        $T_i$=TRUE
    else
        $T_i$=False, No best bid available
    end if
end while
```

$$B_i = D - (pb_i + \text{time taken to reach the bidder} + W_i + e_i), \quad (6)$$

B_i is high, if $pb(pi) \geq 50\ \%$

If B_i=LESS and if the probability of pi is < 50%, Ti=fails to execute.

Therefore, in the DFBAAC algorithm, the RSU is programmed using these algorithms, which in turn can run the critical tasks required by the vehicles, and the normal tasks can run based on Equation 6. Most of the time, bidding occurs if the time between arrival and deadline are very high. If time period is within nominal range, the vehicles will receive a service from the RSU for any service or safety message. The DFBAAC algorithm is shown in Algorithm 3 for the assignment and allotment strategies of the SDN controllers.

SDN controllers perform their bid calculation for the DFBAAC algorithm on the reception of requests to bid messages and then implement focused addressing strategies. Job Guarantee Execution (TGE) is given by Equation 7.

GE(T,c) = number of jobs T that controller C executes (7)

Upon receiving a RB to send a message, controller finds the maximum resources and time it can allocate for a job as given in Equation 8. Using the maximum bid, the controllers checks whether the job can be successfully executed. If so, then job is assigned to the controller with more bid.

Maximum Bid = Minimum (required resources for a job)/Time required for completion (8)

Many classical algorithms have been proposed for optimal placement of SDN controllers. In this study, NFAAC and DBPAAC were used, based on the timelines associated with these controllers. Each controller is supposed to handle the bandwidth required by data planes, switches, or hosts. In these cases, the controller uses the NFAAC and DBPAAC algorithms, in which the response time and period by which the nature of jobs are scheduled on these controllers. This study mainly deals with IoV traffic that requires networking requirements through the RSU.

Therefore, in this work, the RSU’s are modelled as switches, and the switches are controlled by SDN controllers, namely, the learning controller (LC) or the drop controller (DC), where the hosts are the number of vehicles attached to the switches. If a switch connected to the controller receives a job from an unknown host, then the LC dynamically creates a flow rule that determines the port. However, upon receiving a job from an unknown host, a DC drops the packet.

$$\text{Success ratio}=\frac{\textit{number of successful job execution by the controller}}{\textit{Total jobs requested}} \quad (9)$$

If job T is finished earlier, it’s early completion time is given by (EC_i) is given by $EC_i = d - F_i$, F_i is the finishing time of job i and d is the deadline. If job i is finished after the expected time, its tardiness is found by Equation 10. These are taken from [31].

$$\text{tardiness}= F_i - d \quad (10)$$

Figures 3 and 4 show the success ratio when transactions are encountered at controllers (controllers are finite in numbers). In Figure 4, the DBPAAC algorithm is shown for a number of controllers in 4, 8, and 12, and the success ratio either increases or remains almost the same as the number of controller’s increases. The same phenomenon occurred in both the NFAAC and DBPAAC algorithms. Because NFAAC and DBPAAC are easier to implement in an SDN controller, they solve the problem most of the time. However, the objective is to achieve the optimal success ratio of the DFBAAC algorithm. In DFBAAC, the hosts and switches bid for the controller, which increases the success ratio by increasing the controller count. However, the complexity of the DFBAAC algorithm is high, and it has some limitations.

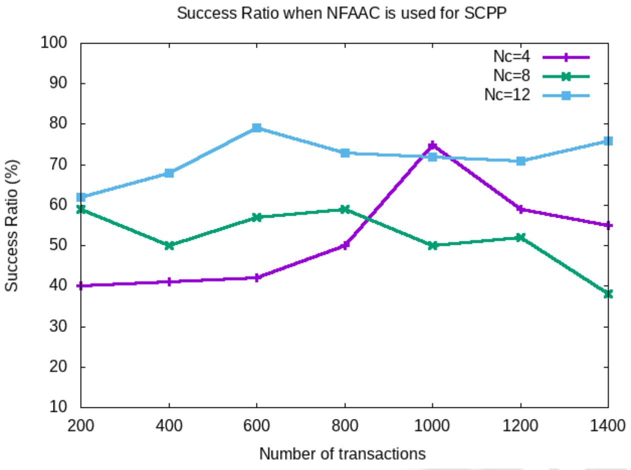

Figure 3. SCPP Success ratio for NFAAC algorithm

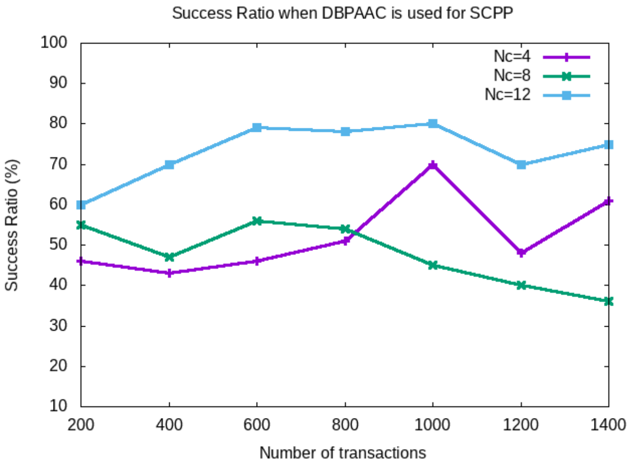

Figure 4. SCPP Success ratio for DBPAAC algorithm

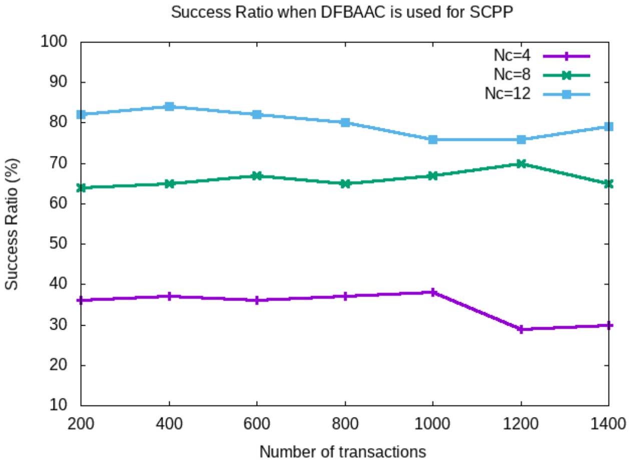

Figure 5. SCPP Success ratio for DFBAAC algorithm

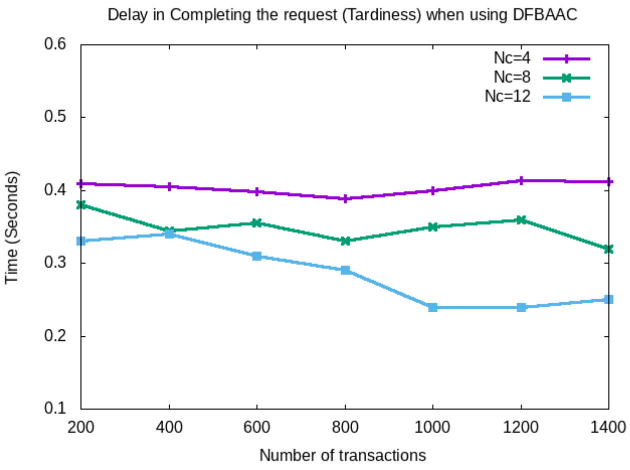

Figure 6. Tardiness for DFBAAC algorithm

Figure 5 shows the success ratio when using the DFBAAC algorithm on the same set of SDN controllers. The success ratio is common for the number of transactions involved; however, when the number of controllers increases, the success ratio also increases. Additionally, Figure 6 shows the average delay in completing a given task, namely tardiness. When the number of controllers increases, tardiness also decreases, implying that the tasks are completed on time with a small delay. This suggests that these algorithms are classical in scheduling in a real-time system, but suitable for IoV as it is also a real-time system whenever the nodes or the hosts need the service from the controller, it should be serviced and safety messages should be sent whenever there is any issue in the traffic or in transportation failure or accidents. In the next section, experiments are conducted using a real open-flow switch with various controllers, namely, external controllers and internal controllers such as learning and drop controllers, switches, hosts, and vehicles (nodes).

IV. EXPERIMENTAL EVALUATION AND RESULT ANALYSIS

SDN controllers can be optimally placed when the demand for bandwidth increases due to the broadcast of safety messages in vehicular networks. This was proven theoretically in the previous sections on various aspects of SDN controllers based on the success ratio and tardiness in completing the transactions as requested by vehicles with RSU's. An experimental evaluation was performed, and the outcomes were analyzed analytically and plotted against the constants. The two simulation methods used in this study were as follows:

1. Use of an (OF) switch with multiple controllers, switches, and suitability of wide area networks and controller placement in multiple locations, namely 10, 20, and 30 locations. The experimental parameters are presented in Table 5.

2. Use OFSoftSwitch 1.3 version with multiple domain controllers and multiple controllers with a set of hosts. The hosts were configured as RSU for vehicular networks. Each host was connected to multiple vehicles in the network. The experimental parameters are presented in Table 6.

This simulation was performed using the software NS3 (Network Simulator 3) which is an event-driven simulator. This study also uses an open-source SDN controller, namely the open-flow switch module, to model the experiment. The simulation was performed as follows:

1. Payload like demand setting by the vehicles for the safety messages.
2. A topology control algorithm to set the controller placement position.
3. Capacity of the channel for the nodes in the topology.
4. A control link to simulate the Wide Area Network Precedence or priority of the jobs.

TABLE. 5 Experimental parameters taken for Experiment1

Name of the parameter	**Value of the parameter**
Latency	8ms (Approximate)
Addressing	Southbound APIs and Control Plane addressing
SDN Controller parameters	Energy, capacity, energy price, fixed energy, Tx consumption, topology control messages, number of responses, locations, demand, number of messages.
Router or Switch parameters	Number of messages, number of requests, transmission consumption, traffic, energy price, size of the request
Simulation time	50, 100, 150 seconds
Topology	Tree
Message size	1 KB
Maximum number of packets	100 MB
Number of locations	10, 20, 30
Switch	OFSoftSwitch 1.3 version

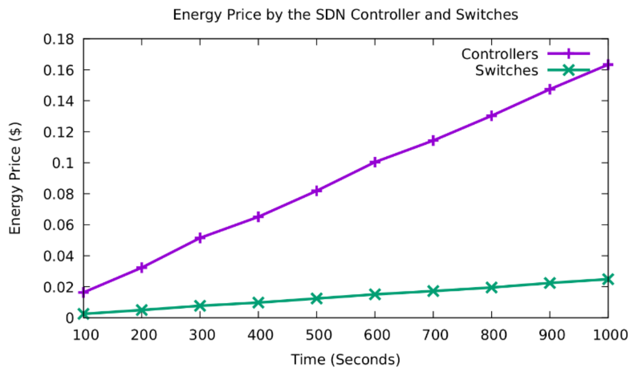

Figure 7. Energy Price estimation while using the SDN controllers and switches.

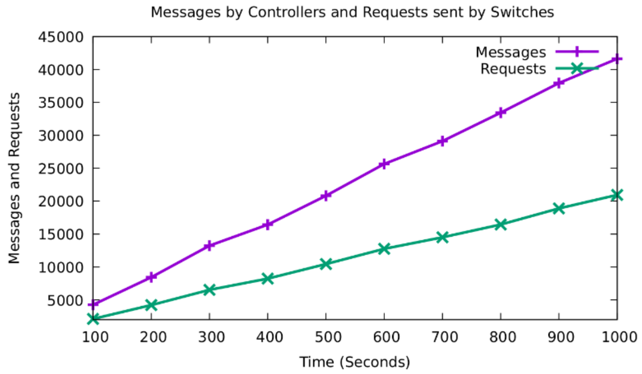

Figure 8. Messages and requests by the controller and switches respectively

Figure 8 depicts the messages and the corresponding requests. It can be seen from the graph that there are more than 40000 messages sent by the controllers that were used for 1000 s, and linearly, it uses fewer messages when it is used for less time. Simulations were performed by varying the durations and their corresponding messages and requests, which were calculated for energy pricing. So it is prevalent that the energy pricing is one of the important metrics for SDN controller placement algorithms which is achieved in this work. In this experiment, two controllers were used for SDN switches: the DC and the LC, which creates a flow for every packet when it is ignored, and the learning controller is a complicated SDN switch for simulating the controller placement problem.

TABLE. 6 Evaluation parameters in NS3

Experimental parameters	**Values of the parameter**
Number of SDN Switches	1, 2, 3, 4
Transport Layer protocols	Transmission Control and User Datagram Protocols (TCP, UDP)
Number of Hosts or RSU's	20
Number of Vehicles	Finite number of vehicles that can interact with RSU's
Controller Assignment Algorithm	DFBAAC, DBPAAC, NFAAC
Parameters evaluated	Hop Count, Throughput, Delay, Packet Delivery Ratio (PDR),
Existing protocol	IMABC against DFBAAC
Vehicular Generation	SUMO
Speed of the vehicle	Average 8 m/s
Real world Map	Open Street Maps (OSM)
Actual Map	Pondicherry Map was used

In this experiment, the switches were programmed using the DFBAAC, DBPAAC, and NFAAC algorithms, and tested based on the task sets. The learning controller acts as a highly powerful and complicated net device that can analyze PDR and packets/s. Because the allotment and assignment algorithms are based on multiple controllers, we tested the controller with 1 or 2 or 3 or 4 switches, where one switch acts as a single controller, and two, three, and four switches act as multiple controllers. More controllers might also be placed as and when bandwidth is needed; however, this experimentation is decentralized, so that the controllers can talk to the RSU's. For this reason, the RSU was set to 20. In the first simulation, the RSU talks to a single controller switch, and later, the RSU talks to either two or three controller switches so that the load and bandwidth are equally shared among the controllers; hence, the controllers are placed nominally for the network. The following figures show the various performance metrics of the controller placement and allotment problems.

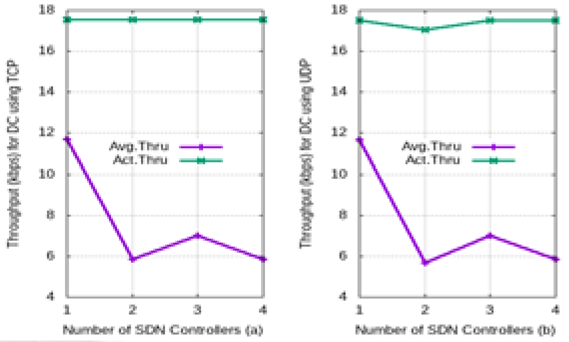

Figure 9. (a) Throughput of Drop Controller with TCP (b) Throughput of Drop Controller with UDP

Figures 9 and 10 show the average throughput and actual throughput when the drop and learning controllers were used along with TCP and UDP as traffic generators. Clearly, the actual throughput with UDP is higher than that of the TCP protocol. The controller switches were predominantly tested for UDP as a traffic generator for controllers to vehicles for packet transfer, but the TCP protocol is preferred only for transmitting Basic Safety Messages (BSM) between vehicle-to-vehicle and controller-to-vehicles, and vice versa. Hence, the throughput is lighter for the TCP protocol because the BSM must be reliable. In addition, our work is compared with the existing literature for the case of IMABC used in [18]. In our SCPP algorithm DFBAAC, we used two protocols, namely TCP (variable bit rate or file transfer application) and UDP (constant bit rate) to handle traffic between the controllers and vehicles. The evaluation metrics specified in Table 6 were computed and compared using The IMABC algorithm, The IMABC algorithm did not handle anything in the transport layer, and this work was mainly on the networking layer. However, the proposed method concentrates on all the layers of the Open Systems Interconnection (OSI) model.

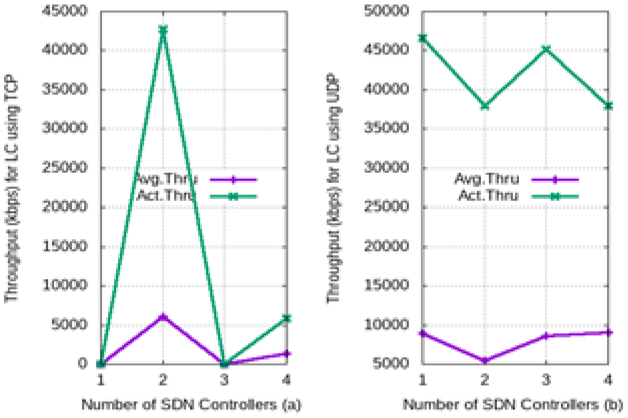

Figure 10. (a) Throughput of Learning Controller with TCP (b) Throughput of Learning Controller with UDP

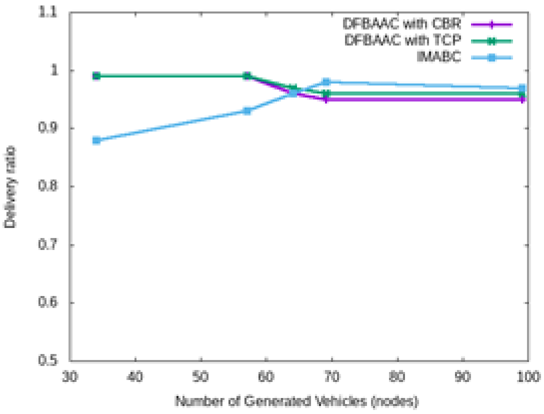

Figure 11. Delivery Ratio

Figures 11 and 12 depicts various metrics compared with the existing SCPP problem, IMABC. Figure. 11 shows that the PDR is higher in the DFBAAC algorithm with CBR and TCP than that in the existing work. When the number of vehicles increased, the PDR was almost high and remains at an average of 97%. The existing study also performed well for PDR but lacked the property of using a transport layer protocol.

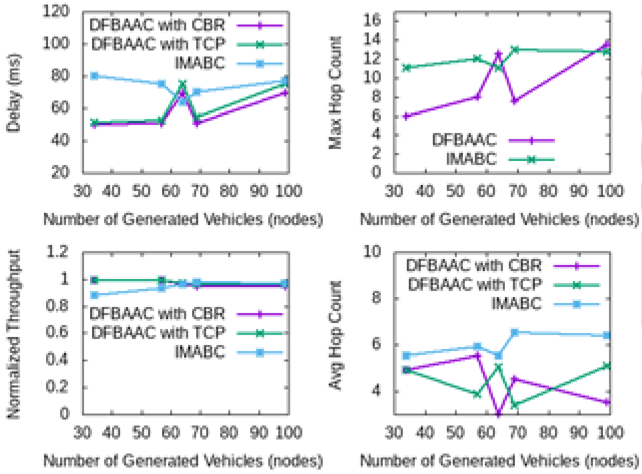

Figure 12. Various metrics compared with IMABC and DFBAAC

Figure 12 shows four properties: end-to-end delay, maximum hop count, average hop count, and throughput. The delay factor is better handled in the DFBAAC protocol, where the delay is between 50 and 70 ms. The existing literature handles the delay at approximately 80 ms. In addition, the maximum hop count is maintained between 6 and 12, and the energy cost is reduced when the intermediate nodes are low. Compared with existing work, our algorithm is improved by a 20% reduction in the number of hops. The average hop count is also better when compared to the existing work, where DFBAAC is maintained between three and five, whereas the existing work handles anywhere from five to seven average hop counts, which directly affects the energy cost.

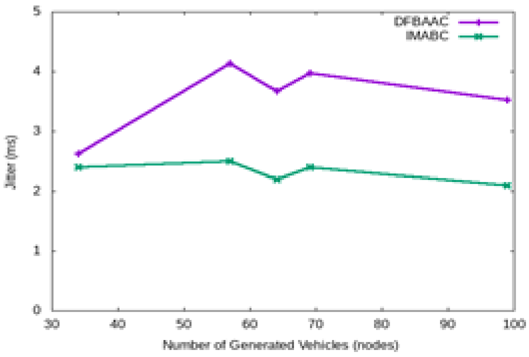

Figure 13. Comparison of Jitter

Figure 13 shows the jitter computation, which was slightly higher than that of the IMABC protocol. The jitter was calculated based on the network performance in existing work, where the actual vehicular waiting time was not considered. However, Jitter is calculated by considering the average waiting time of the vehicles. Hence, the jitter was higher in this study than in the existing work.

V. CONCLUSION

In this study, three algorithms were proposed, namely DFBAAC, DBPAAC, and NFAAC, for controller switches modelled as RSU's, which are the nodes handled by these switches for bandwidth and basic safety message communication. Various simulations were performed to calculate metrics such as delay, throughput, jitter, the number of controllers to be deployed, and hop counts to measure the performance of the SDN controller placement problem. This work also compares the existing work, namely IMABC, which was proposed as a controller-placement problem. Analytically, the results compared with this protocol and the DFBAAC protocol outperformed the IMABC protocol in most aspects with an improvement score of 13.5% in throughput, delay, and PDR.

ACKNOWLEDGMENT

The authors would like to thank the support of Puducherry Technological University, Puducherry, India and SRKR Engineering College, Bhimavaram, India. The authors would like to thank the editor and reviewers to get quality research work.

REFERENCES

[1] Zhuang, Weihua, Qiang Ye, Feng Lyu, Nan Cheng, and Ju Ren. "SDN/NFV-empowered future IoV with enhanced communication, computing, and caching." Proceedings of the IEEE 108, no. 2 (2019): 274-291. https://doi.org/10.1109/JPROC.2019.2951169

[2] Lanka, Divya, and Selvaradjou Kandasamy. "A Hybrid Trust Probability Based C-ACO Model for Large Software Defined Networks." In Innovations in Bio-Inspired Computing and

---

Applications: Proceedings of the 11th International Conference on Innovations in Bio-Inspired Computing and Applications (IBICA 2020) held during December 16-18, 2020 11, pp. 55-65. Springer International Publishing, 2021. https://doi.org/10.1007/978-3-030-73603-3_5

[3] Nkenyereye, Lionel, Lewis Nkenyereye, SM Riazul Islam, Yoon-Ho Choi, Muhammad Bilal, and Jong-Wook Jang. "Software-defined network-based vehicular networks: A position paper on their modeling and implementation." Sensors 19, no. 17 (2019): 3788. https://doi.org/10.3390/s19173788

[4] Deng, Der-Jiunn, Shao-Yu Lien, Chun-Cheng Lin, Shao-Chou Hung, and Wei-Bo Chen. "Latency control in software-defined mobile-edge vehicular networking." IEEE Communications Magazine 55, no. 8 (2017): 87-93. DOI: 10.1109/MCOM.2017.1601165

[5] K. Kaur, S. Garg, G. Kaddoum, N. Kumar, F. Gagnon, Sdn-based internet of autonomous vehicles: An energy-efficient approach for controller placement, IEEE Wireless Communications, 26 (6), 2019, 72-79.

[6] Narendhiran, R., K. Pavithra, P. Rakshana, and P. Sangeetha. "Software Defined Based Pure VPN Protocol for Preventing IP Spoofing Attacks in IOT." International Journal on Recent and Innovation Trends in Computing and Communication 7, no. 3: 14-20. https://doi.org/10.17762/ijritcc.v7i3.5251

[7] Silva, Fabricio A., Azzedine Boukerche, Thais RMB Silva, Eduardo Cerqueira, Linnyer B. Ruiz, and Antonio AF Loureiro. "Information-driven software-defined vehicular networks: Adapting flexible architecture to various scenarios." ieee vehicular technology magazine 14, no. 1 (2019): 98-107. http://dx.doi.org/10.1109/MVT.2018.2867356

[8] Whangkuanklang, V. ., Siangdang, Y. ., Kampagdee, M. ., Akkarakultron, P. ., Supamattra, S. ., Phasinam, K. ., & Watcharinrat, D. . (2023). An Scientific Approach of Design and Development of a Garlic Peeling Machine. International Journal of Intelligent Systems and Applications in Engineering, 11(3s), 115–118. Retrieved from https://ijisae.org/index.php/IJISAE/article/view/2548

[9] Honarpazhooh, Sanaz. "Controller Placement in Software-defined Networking Using Silhouette Analysis and Gap Statistic." Turkish Journal of Computer and Mathematics Education (TURCOMAT) 12, no. 13 (2021): 4848-4863.

[10] Maity, Ilora, Ravi Dhiman, and Sudip Misra. "Mobiplace: Mobility-aware controller placement in software-defined vehicular networks." IEEE Transactions on Vehicular Technology 70, no. 1 (2021): 957-966. http://dx.doi.org/10.1109/TVT.2021.3049678

[11] Wang, Junpeng, Zhumu Fu, Pengju Si, and Fazhan Tao. "A Distributed Algorithm for Controller Placement in Software Defined Vehicular Networks." In Proceedings of the 2019 8th International Conference on Networks, Communication and Computing, pp. 142-148. 2019. https://doi.org/10.1145/3375998.3376012

[12] Toufga, Soufian, Slim Abdellatif, Hamza Tarik Assouane, Philippe Owezarski, and Thierry Villemur. "Towards dynamic controller placement in software defined vehicular networks." Sensors 20, no. 6 (2020): 1701. https://doi.org/10.1016/j.comnet.2021.107981

[13] Kumar, TS Pradeep, and P. Venkata Krishna. "A survey of energy modeling and efficiency techniques of sensors for IoT systems." International Journal of Sensors Wireless Communications and Control 11, no. 3 (2021): 271-283. http://dx.doi.org/10.2174/2210327910999200614001521

[14] Kumar, TS Pradeep, and P. Venkata Krishna. "Power modelling of sensors for IoT using reinforcement learning." International Journal of Advanced Intelligence Paradigms 10, no. 1-2 (2018): 3-22. https://doi.org/10.1504/IJAIP.2018.089491

[15] Liyanage, Kushan Sudheera Kalupahana, Maode Ma, and Peter Han Joo Chong. "Controller placement optimization in hierarchical distributed software defined vehicular networks." Computer Networks 135 (2018): 226-239. https://doi.org/10.3390/s20061701

[16] Ramya, G. and Rajendiran Manoharan. “Enhanced Multi-Controller Placements in SDN.” 2018 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET) (2018): 1-5. https://doi.org/10.1109/WISPNET.2018.8538519

[17] Ramya, G., and R. Manoharan. "Enhanced optimal placements of multi-controllers in SDN." Journal of Ambient Intelligence and Humanized Computing 12 (2021): 8187-8204. https://doi.org/10.1007/S12652-020-02554-2

[18] Rossi, G., Nowak, K., Nielsen, M., García, A., & Silva, J. Machine Learning-Based Risk Analysis in Engineering Project Management. Kuwait Journal of Machine Learning, 1(2). Retrieved from http://kuwaitjournals.com/index.php/kjml/article/view/114

[19] Shirmarz, Alireza, and Ali Ghaffari. "Automatic software defined network (SDN) performance management using TOPSIS decision-making algorithm." Journal of Grid Computing 19 (2021): 1-21. https://doi.org/10.1007/s10723-021-09557-z

[20] Lin, Na, Qi Zhao, Liang Zhao, Ammar Hawbani, Lu Liu, and Geyong Min. "A novel cost-effective controller placement scheme for software-defined vehicular networks." IEEE Internet of Things Journal 8, no. 18 (2021): 14080-14093. https://doi.org/10.1109/JIOT.2021.3069878

[21] Mr. Rahul Sharma. (2015). Recognition of Anthracnose Injuries on Apple Surfaces using YOLOV 3-Dense. International Journal of New Practices in Management and Engineering, 4(02), 08 - 14. Retrieved from http://ijnpme.org/index.php/IJNPME/article/view/36

[22] Cui, Xin, Xiaohui Gao, Yan Ma, and Weihan Wang. "A controller deployment scheme in 5G-IoT network based on SDN." EURASIP Journal on Wireless Communications and Networking 2020 (2020): 1-19. https://doi.org/10.1186/s13638-020-01853-8

[23] Aly, Wael Hosny Fouad, Hassan Kanj, Samer Alabed, Nour Mostafa, and Khaled Safi. "Dynamic Feedback versus Varna-Based Techniques for SDN Controller Placement Problems." Electronics 11, no. 14 (2022): 2273. https://doi.org/10.3390/electronics11142273

[24] Syed, Sidra Abid, Munaf Rashid, Samreen Hussain, Fahad Azim, Hira Zahid, Asif Umer, Abdul Waheed, Mahdi Zareei, and Cesar Vargas-Rosales. "QoS aware and fault tolerance

---

based software-defined vehicular networks using cloud-fog computing." Sensors 22, no. 1 (2022): 401. https://doi.org/10.3390/s22010401

[25] Barış, K. A. R. A., and Bahri Atay ÖZGÖVDE. "Effect of RSU placement on autonomous vehicle V2I scenarios." Balkan Journal of Electrical and Computer Engineering 8, no. 3 (2020): 272-284. https://doi.org/10.17694/bajece.668615

[26] Liu, Chunhui, Kai Liu, Songtao Guo, Ruitao Xie, Victor CS Lee, and Sang H. Son. "Adaptive offloading for time-critical tasks in heterogeneous internet of vehicles." IEEE Internet of Things Journal 7, no. 9 (2020): 7999-8011. https://doi.org/10.1109/JIOT.2020.2997720

[27] Yang, Shi. "A task offloading solution for internet of vehicles using combination auction matching model based on mobile edge computing." IEEE Access 8 (2020): 53261-53273. https://doi.org/10.1109/ACCESS.2020.2980567

[28] Amin, Ruhul, Isha Pali, and Venkatasamy Sureshkumar. "Software-defined network enabled vehicle to vehicle secured data transmission protocol in VANETs." Journal of Information Security and Applications 58 (2021): 102729. https://doi.org/10.1016/j.jisa.2020.102729

[29] Mezzavilla, Marco, Menglei Zhang, Michele Polese, Russell Ford, Sourjya Dutta, Sundeep Rangan, and Michele Zorzi. "End-to-end simulation of 5G mmWave networks." IEEE Communications Surveys & Tutorials 20, no. 3 (2018): 2237-2263. https://doi.org/10.1109/COMST.2018.2828880

[30] Drago, Matteo, Tommaso Zugno, Michele Polese, Marco Giordani, and Michele Zorzi. "MilliCar: An ns-3 module for mmWave NR V2X networks." In Proceedings of the 2020 Workshop on ns-3, pp. 9-16. 2020. https://doi.org/10.1145/3389400.3389402

[31] Ali, Muhammad, Saargunawathy Manogaran, Kamaludin Mohamad Yusof, and Muhammad Ramdhan Muhammad Suhaili. "Analysing vehicular congestion scenario in Kuala Lumpur using open traffic." Indonesian Journal of Electrical Engineering and Computer Science 10, no. 3 (2018): 875-882. DOI: 10.11591/ijeecs.v10.i3.pp875-882

[32] Fadhil, Jawaher Abdulwahab, and Qusay Idrees Sarhan. "Internet of vehicles (IoV): A survey of challenges and solutions." In 2020 21st International Arab Conference on Information Technology (ACIT), pp. 1-10. IEEE, 2020. https://doi.org/10.1109/ACIT50332.2020.9300095

[33] Talpur, Anum, and Mohan Gurusamy. "On attack-resilient service placement and availability in edge-enabled iov networks." IEEE Transactions on Intelligent Transportation Systems (2023). https://doi.org/10.1109/TITS.2023.3249830

[34] Alharkan, Ibrahim M., T. Aziz, and Mohammed Hatem Mohammed Alhaag. "Minimization of earliness & Tardiness penalties with common due dates problem using Tabu search." In 2015 International Conference on Industrial Engineering and Operations Management (IEOM), pp. 1-12. IEEE, 2015. https://doi.org/10.1109/IEOM.2015.7093848