This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSE.2020.3047454, IEEE Transactions on Network Science and Engineering

ARTNet: **AI**-based **R**esource Allocation and **T**ask Offloading in a Reconfigurable Internet of Vehicular **Net**works

Muhammad Ibrar, Aamir Akbar, Syed Rooh Ullah Jan, Mian Ahmad Jan[§], Lei Wang[†], Houbing Song, Nadir Shah

***Abstract*—The convergence of Software-Defined Networking (SDN) and Internet of Vehicular (IoV) integrated with Fog Computing (FC), known as Software Defined Vehicular based FC (SDV-F), has recently been established to take advantage of both paradigms and efficiently control the wireless networks. SDV-F tackles numerous problems, such as scalability, load-balancing, energy consumption, and security. It lags, however, in providing a promising approach to enable ultra-reliable and delay-sensitive applications with high vehicle mobility over SDV-F. We propose ARTNet, an AI-based Vehicle-to-Everything (V2X) framework for resource distribution and optimized communication using the SDV-F architecture. ARTNet offers ultra-reliable and low-latency communications, particularly in highly dynamic environments, which is still a challenge in IoV. ARTNet is composed of intelligent agents/controllers, to make decisions intelligently about (i) maximizing resource utilization at the fog layer, and (ii) minimizing the average end-to-end delay of time-critical IoV applications. Moreover, ARTNet is designed to assign a task to fog nodes based on their states. Our experimental results show that considering a dynamic IoV environment, ARTNet can efficiently distribute the fog layer tasks while minimizing the delay.**

***Index Terms*—Internet of Vehicles, Software Defined Network, Machine Learning, Fog Computing, Task Offloading.**

I. Introduction

In the current era, Internet-of-Things (IoT) [1] are becoming increasingly accessible irrespective of their known limitations (i.e., restricted storage, computing power, and processing speed) of being resource constraints. Similarly, Artificial Intelligence (AI) or Machine Learning (ML) [2] is another popular area that has seen significant attention recently. The integration of IoT and AI technologies dramatically improves the effectiveness of IoT devices in the production of social and economic applications that are useful [3]–[5]. For example, the Intelligent Transportation System (ITS) for the Internet of Vehicles (IoV) [6] is a typical application of IoT technology and is essentially a transport system for the coming decades.

IoV is a distributed, large-scale wireless communication and information exchange system. It includes the convergence of three networks: (a) inter-vehicle (V2V), (b) intra-vehicle (V2I), and mobile internet vehicle. Based on these technologies, the architecture is also called Vehicle to Everything (V2X) [7], [8]. The IoV/V2X provides several services, e.g., enhanced traffic safety, improved traffic efficiency, and controlled supervision. In the applications developed for IoV/V2X, services such as data aggregation, data mining, data storage, and computation power can also be supported using communication technologies, e.g., cellular data networks (4G/5G), WAVE (Wireless Connectivity in Vehicular Environment), and IEEE 802.11p to access virtual resources in the vehicular cloud layer. While the vehicular cloud layer may provide significant storage and processing power if numerous vehicles access data simultaneously, it can be vulnerable to a considerable delay in storing or retrieving the data. This can lead to core/backbone network congestion, which in turn causes a significant reduction in Quality-of-Service (QoS) and longer end-to-end delays.

A new paradigm called Vehicular Fog Computing (VFC) has recently been introduced [9], which addresses the problem of end-to-end delays in IoV. As FC provides a layer between the cloud and the IoV, it reduces the delays and improves QoS. However, there are several challenges in designing an efficient and reliable VFC system, such as congestion avoidance, fault tolerance, load balancing, a balanced allocation of resources, guaranteed delay, and security [10]. To overcome these challenges, a feasible approach can be Software-Defined Networking (SDN) combined with VFC, referred to as Software-Defined Vehicle Network-based Fog Computing (SDV-F).

FC provides IoV cloud-like services, i.e., virtualization, task offloading, and fog-to-fog resource pooling at the edge of the network and consists of interconnected and small virtual data centers called fog nodes. These nodes can be hosted in RoadSide Units (RSUs) or Base Stations (BSs) to create a three-tier architecture, i.e., Vehicles-Fog-Cloud. The research community has proposed several approaches for SDV-F that address various issues such as energy usage, scalability of control planes, and security [10]–[18]. However, it lags in providing a promising solution with high vehicle mobility for supporting ultra-reliable and delay-sensitive applications over SDV-F.

As wireless networks and applications become more complex and intelligent in SDV-F, there is an urgent need for

M. Ibrar and L. Wang are with the School of Software, Dalian University of Technology. (email: mibrar@mail.dlut.edu.cn; lei.wang@dlut.edu.cn)

A. Akbar, MA. Jan and SRU. Jan are with the Department of Computer Science, Abdul Wali Khan University Mardan. (email: amirakbar@awkum.edu.pk; mianjan@awkum.edu.pk; roohullah-syed@awkum.edu.pk)

N. Shah is with the department of computer science, COMSATS University, Wah Campus. (email: nadirshah82@gmail.com)

H. Song is with the department of electrical, computer, software, and systems engineering, Embry-Riddle Aeronautical University. (email: h.song@ieee.org)

† Corresponding author in Dalian University of Technology.

§ Corresponding author in Abdul Wali Khan University.

effective management of the performance of increasingly complex tasks, based on the requirements of the IoV application. Moreover, the problem is compounded by the highly volatile service requirements of end-users and the uncertainties associated with the availability of services at fog nodes. Therefore, in fog networks, where different applications are concurrently running over the same network, the selection of appropriate nodes and proper resource allocation and load balancing is important. Load balancing in FC refers to the efficient allocation of incoming workload through a network of processing fog nodes to increase the required tasks' efficiency. Load balancing can be categorized into two types. Prior knowledge of task requests is used in static load balancing to distribute the workload, which is measured at the beginning of the execution. Its key drawback is that during the process execution, the task allocation can not be changed to reflect traffic load changes. In dynamic load balancing, under-loaded nodes are dynamically allocated to the task. In other words, it will adjust the allocation of tasks constantly, based on the latest understanding of traffic loads. Accurate real-time load prediction is, therefore, critical for efficient load balancing.

The authors in [19]–[22] have considered minimizing the overall cost of energy usage, computation offloading, end-to-end delay, and to allocate computation and communication resources in IoV networks. However, dynamic IoV topologies, fog-to-fog resource pooling, and dynamic computational costs for load balancing and minimizing IoV networks' delay have not been considered.

In this work, we propose *ARTNet*: **A**I-based **R**esource allocation and **T**ask offloading in a reconfigurable internet of vehicular **Net**works. ARTNet uses the benefits of both AI and SDV-F to guarantee ultra-reliable and low latency communication in a highly dynamic environment, which can not be assured by the existing underlying networks. To distribute the load efficiently and minimize the end-to-end delay, the SDN controller in ARTNet selects the fog nodes as secondary agents/controllers at the fog layer. In addition, ARTNet efficiently (based on AI) distribute the available resources within complex and dynamic networks like SDV-F.

We summarize the main contributions of this work as follows.

- In this paper, we propose an AI-based hierarchical framework for SDV-F, in which the SDN controller selects the secondary agents, e.g., fog nodes, at the fog layer dynamically with maximum computation power. If the SDN controller fails, the secondary agents provide the services to the underlying architecture.
- The proposed ARTNet framework supports horizontal (fog-to-fog) and vertical (vehicles-to-fog or fog-to-cloud) traffic offloading to attain adaptive resource scaling and minimize the end-to-end delay.
- The secondary agents at the fog layer decide to offload, depending on the reward feature. This is a desirable attribute since the secondary agents have the freedom to determine the compensation role based on the expected performance.

The rest of this paper is organized as follows. Section II discusses the related work. Section III-A discusses the system architecture and modelling based on the ARTNet framework. Section IV explains the proposed solution. Section V discusses the experimental results. Finally, Section VI concludes the paper with future directions.

II. STATE-OF-THE-ART

Over the last decade, SDN and FC based paradigms for IoV service architectures have been made possible by a range of studies. Some of the studies are discussed as follows.

In [23], an SDN-based architecture, was proposed to enable rapid network innovation in heterogeneous vehicular communication environments. Furthermore, a 5G-enabled Software-Defined Vehicular Network (5G-SDVNs) for service provision in IoV was proposed in [24]. The authors in [25] proposed a solution to temporal information services with time constraints in heterogeneous vehicular networks using SDN-based scheduling. In [26], a solution to improve communication performance in dynamic vehicular networking environments using an SDN-based Medium Access Control (MAC) protocol is proposed. Moreover, another approach [27] that addressed the problem of enhancing the available bandwidth usage in an SDN-based service architecture for IoV by combining vehicular caching and network coding has been proposed.

Furthermore, the authors in [28] proposed a general-purpose Mobile Cloud Hybrid (MCH) framework. The proposed framework was used to optimize power consumption, and network usage of MCH applications developed for battery-powered Android-based devices [29] and robotics [30]. A similar type of framework for mobile cloud computing systems with a Computing Access Point (CAP) was proposed in [19]. The CAP serves mobile users as both a network connectivity portal and a data service provider. From the mobile devices' perspective, independent tasks of each mobile user can be processed both locally at the CAP or a remote cloud server. The proposed method aims at optimizing the offloading decision process, allocating computing and communication resources, minimizing the overall energy costs, processing, and delays for all users. This problem is NP-hard in general. Therefore, the author proposed an efficient three-step algorithm comprising of Semi-Definite Relaxation (SDR), Alternating Optimization (AO), and Sequential Tuning (ST).

Emerging vehicle applications, such as real-time situation awareness and cooperative lane change, require ample computational resources at the edge to execute time-critical and data-intensive activities. A novel approach called Folo [9] is proposed to address the latency issues and task allocation in vehicular fog computing. Also, Folo is designed to support cars' mobility, including those producing tasks and those acting as fog nodes. The authors articulate the task allocation mechanism across stand-alone and mobile fog nodes into a mutual optimization problem, with service latency, quality loss, and fog power consumption constraints. It is an NP-hard problem; therefore, the authors used the Mixed Integer Linear Programming (MILP) technique to linearize the optimization equation and then solve it.

Fog nodes are deployed closer to the vehicles, where the data is generated. Besides, SDN might also be deployed to

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSE.2020.3047454, IEEE Transactions on Network Science and Engineering

support the use of large-scale fog-enabled Vehicle Networks (VN) services [17], [18]. However, the current management of each wireless network that composes the VN has restricted the exploration of fog infrastructures for scalable VN services. Therefore, VN architecture's design principles are still an open issue, mainly because it is necessary to address the diversity of VN fog applications. The authors investigate the design principles for fog-enabled Vehicular SDN (VSDN), focusing on the systems, networking, and services perspectives [11]. The authors evaluated these design principles of fast traffic accident rescue for emergency vehicle use cases, using real traffic accident-related data.

The distributed FC architecture can reduce the latency at the end devices and improve the QoS. For example in [31], the authors proposed a heuristic algorithm called Minimum Response (MinRE) to solve the service placement problem based on service demand, e.g. delay, power consumption, and computational resources. MinRE is based on distributed FC architecture, which categorizes the services at the end devices into two categories: 1) normal services, 2) critical services. For normal services, the aim is to reduce energy consumption while the critical service target is to minimize the response time for critical service. The authors in [32] proposed an offloading scheme to minimize the delay and energy consumption in the FC architecture. The authors consider heterogeneous CPU frequencies in the proposed scheme to measure the mission. Also, the proposed model in [33] simultaneously optimizes communication and computing resources in FC, which are used by end-users to offload tasks. To measure the task in the tolerable latency, the primary node obtains the resources from neighbour fog nodes or clouds.

The recent work [10]–[22] no doubt offers excellent solutions to minimize the energy usage, average end-to-end delay, and to manage the task offloading issues in the SDV-F architecture. However, they are limited in using the multi-agent system and the horizontal fog layer resource pooling. We believe that the multi-agent architecture and horizontal fog layer resource pooling substantially decrease the end-to-end delay.

III. System Model

In this section, we first explain the system architecture of the proposed ARTNet model, which aims to enhance wireless IoV architecture's scalability. The architecture also aims to efficiently distribute the tasks at the fog layer (i.e., horizontal distribution). The proposed ARTNet model provides the services at very low latency to IoV (i.e., vertical distribution) using Machine Learning (ML).

A. System Architecture

This section provides the high-level system architecture for the ARTNet model, as shown in Fig. 1. Fig. 1 includes the following layers:

1) A significant number of IoV devices are in the bottom layer, which is also called the IoV layer. On-board devices (OBUs) containing a processing unit, sensors, a navigation system (e.g., GPS), and a radio transceiver chip are provided for the vehicles. As a networking technology to communicate with a Base Station (BS) or RSUs, these IoV devices use WiMAX/3G/4G/WAVE. The devices in IoV architecture reflect the link between the vehicles and provide connectivity between infrastructure, coordination between pedestrians, and roads. Moreover, according to a recent forecast [34], in the coming years, more than 300 million vehicles are projected to appear in the IoV market.
2) The fog layer consists of static modules like RSUs and BSs, with storage and computing capabilities. The fog nodes provide networking, storage, and computing facilities to IoV with closer proximity. BS provides communication channels that link the RSUs, BSs, and vehicles using the underlying communication networks such as WAVE, WiMAX, LTE, and 5G. The fog node frequently uploads the real-time status of vehicles to the controller. However, since a significant number of connected vehicles generate a large number of processing resources, it is challenging to satisfy low latency requirements for a single BS or RSU processing operation. To control the load and reduce latency, it is also vital to perform distributed computing and implement a load balancing technique.
3) The AI-based SDN controller separates the data plane from the control plane, improves the system's evolution, and facilitates network management. The AI unit includes an intelligent agent module, Big data analytics module, and a deep learning module. An intelligent agent is a rational entity that perceives the environment (input) and acts accordingly (output). It takes data from the environment as input and provides instructions as output. Using deep learning networks and the Intelligent Agent Module, the Big Data Analytics module allows computers to make intelligent decisions without human input. The deep learning module offers the fog nodes the optimal model performed on each fog node, taking into account their available computing resources.

In the ARTNet, the SDN controller is responsible for data collection and decision-making. On the other hand, the fog nodes serve the vehicles directly and transmit reports to the controller based on the gathered traffic data and queue status. Additionally, the proposed ARTNet architecture supports both vertical, i.e., Fog node-to-IoV and Fog node-to-cloud, and horizontal, i.e., Fog node-to-Fog node (inter-Fog nodes), traffic offloading, to attain adaptive resource scaling. Inter-Fog nodes resource scaling also called resource pooling.

B. Problem Formulation

To provide distributed and efficient resource management in SDV-F architecture, fog nodes are connected to the SDN controller, where the controller can make decisions. At the same time, the data plane serves forwarding and computational tasks. Additionally, a fog node consists of multiple virtual machines assigned to different tasks on demand. Therefore, achieving synchronization and load-balancing among multiple agents (e.g., central SDN controller and fog nodes that the

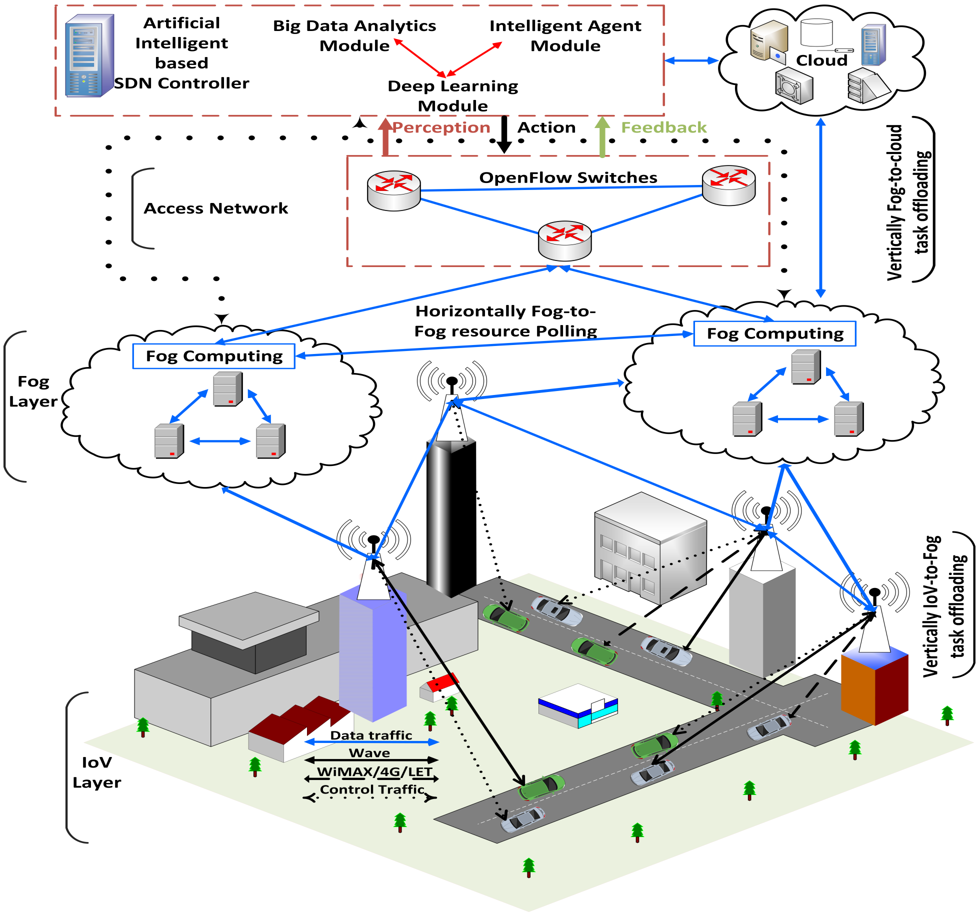


Fig. 1: System Architecture based on ARTNet Model

SDN controller has chosen) in the SDV-F environment is still a big challenge [35].

1) **Synchronization Among Agents***:* FC architecture consists of distributed fog nodes. Therefore, synchronization among the fog nodes is important to increase the Quality-of-Service (QoS). To demonstrate the advantages of coordination of agents for optimal load balancing and to minimize the delay, we assume a fog node f_n^i can provide better services in terms of computation power and delay than an f_n^j. However, due to the highly dynamic nature of a wireless network, agents are not always synchronized. As a result, the task may be assigned to f_n^j based on outdated information, which leads to performance degradation in an SDV-F network. Besides, it is possible to obtain the real-time status of a fog node using a synchronization process. Therefore, using reinforcement learning (RL) techniques (which has been very successful in solving complex problems recently), we address the load balancing problem among multiple agents by formulating it as a Markov Decision Process (MDP) function.

MDP Functions: States, Actions, and Rewards: In the SDV-F network, we consider the SDN controller as the main agent. Apart from the main agent and the high dynamic nature of the SDV-F setting, the following two other key components are needed in the design of the RL algorithm. *(a)* Action Selection Policy and *(b)* State-Action Quality Function

The SDN controller (i.e., the primary agent) receives actions based on the environment's perceived states. The quality function shows the quality difference between the current and previous state-action pairs. Consequently, the long-term reward signifies the total rewards an SDN controller can expect to accumulate over time concerning each environment state. In contrast, the reward is given after the agent's current action. This revenue provides the long-term value of states concerning the future states, followed by their corresponding rewards. Finally, the system model imitates a real-life system's management and foreseen a good reward and quality of the next state from the current ones. Besides, with the help of using MDP and RL, our approach will enable the agent to make effective decisions autonomously.

Furthermore, the MDP offers a mathematical basis for system modelling and using the RL. It is an effective way to formulate a sequential decision problem in a stochastic and fully measurable Markov Transition Model (MTM) settings. Generally, MDP is represented with a 4-tuple (S, A, P, R), where $s \in S$ represents a set of states, $a \in A$ means a set of actions, and $r \in R$ shows a set of rewards. Additionally, P offers the transition probability, i.e., P: S × A × S → [0, 1]. The P($s^{'}|s$, a) shows a new state of the environment provided that the environment is in state s and the chosen action is a. $R\left(s, a, s'\right)$ shows the reward function i.e., $R : S \times A \mapsto \mathbb{R}$ after the action. The reward function's quality represents with discount factor γ, where the value of the discount factor is

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSE.2020.3047454, IEEE Transactions on Network Science and Engineering

$0 < \gamma < 1$. The main goal of the ARTNet is to optimally synchronizes the agents.

Action Selection Policy: The computable quantities, as an MDP solution, select an action for each state referred to as a π policy and a sum of rewards called a utility. In the current setting, the projected utility demonstrates the consistency of the policy. Consequently, an agent seeks to maximize the required utility that generates the optimal policy of π. One of the agent's successes metrics is the number of rewards for the traversed states, which can be written as a utility function based on environmental history, i.e., $UT_h = [(s_0, s_1, s_2..., s_n)]$.

State-Action Quality Function: In MDP, before the execution of the controlled system, *P* and *R* are calculated and can be solved with dynamic programming. The primary principle of dynamic programming is to use the value function to search for the best choices. The optimal quantities in MDP are the value of state *(*s), expect utility $U^*(s)$, and Q-state (s, a). Accordingly, the optimal policy $\pi^*(s)$ which is an optimal action from state *s*. The details of the optimal policy selection are presented in Algorithm 1. In MDP, the long-term rewards (i.e., the discounted sum of the expected immediate rewards) determined each state's optimal action. Therefore, the optimum action satisfies the Bellman Equations, as follows.

$$U^*(s) = max_a Q^*(s_t, a_t) \tag{1}$$

$$Q^*(s_t, a_t) = \sum_{t=0}^{T} P(s_t, a_t, s')\left[R(s_t, a_t, s') + \gamma^t U^*(s')\right] \tag{2}$$

$$U^*(s_t, a_t) = \max_a \sum_{t=0}^{T} P(s_t, a_t, s')\left[R(s_t, a_t, s') + \gamma^t U^*(s')\right] \tag{3}$$

where $s_{t},\ a_t$ shows the state-action quality pair at time *t*. Additionally, T represents the time limit of synchronizing the agents' optimization problem in the proposed model. Precisely, $Q(s_t, a_t)$ shows the quality of taking action *a* at the current state s_t. Consequently, when the agent needs to choose an action for the current state s_t, it compute $Q(s_t, a_t)$ for each possible action and decides the next action according to these quality values.

2) **Task-offloading and load-balancing at the fog layer***:* The load balancing algorithms intended to provide a node workload measurement compared to a global average usually based on a load index. It functions to locate a load imbalance condition used to verify if a load index is much lower or higher than the other nodes' load index. For a successful load index, the processor queue length is one of the essential parameters. To find performance, this type of load index can be used on time-shared workstations—furthermore, the agents obtain information and decide. Simultaneously, the nodes represent end-users directly and provide information based on the traffic information gathered and the state of the queue to the controllers.

The proposed model's main goal is to optimize offloading action at each system to maximize the utility while minimizing the processing delay and distributing the IoV tasks optimally. Therefore, the *R(s, a)* given an *a* at *s* as follows.

$$R(s, a) = U(s, a) - (L(s, a) + D(s, a)), \tag{4}$$

where, *L(s, a)* signifies the traffic load probability function of a fog node f_n^j and *D(s, a)* indicate the end-to-end delay function. The utility function *U(s, a)* can be computed as,

$$U(s, a) = r_u log(1 + k^o), \tag{5}$$

where, r_u and k^o indicate the utility reward and number of task to be offload to a fog node f_n^j.

In the proposed model, the controller/agents compute the the traffic load probability function *L(s, a)* of a fog node f_n as follows:

$$L(s, a) = \kappa_o \frac{k^{pro} P_{trafficload,pro} + k^o P_{trafficload,o}}{k^{pro} + k^o}, \tag{6}$$

$$P_{trafficload,j} = \frac{\max(0, k_{arrival-rate} - (Q_{j,max} - Q_j^{'}))}{k_{arrival-rate}}, \tag{7}$$

$$Q_j^{'} = min(max(0, Q_j - c_p wr^j) + k^j, Q_{j,max}), \tag{8}$$

where, κ_o indicates the traffic load weight, k^{pro} shows the currently processing tasks. Consequently, $k_{arrival-rate}$ shows the arrival rate of tasks at fog node f_n^j, which can be modeled by a *Poisson* process. Additionally, $Q_j^{'}$ shows the next estimated queue state of a fog node f_n^j in state *s* and action *a* is taken.

In the proposed task offloading algorithm, the end-to-end delay for a task is computed as follows:

$$D(s, a) = \kappa_d \frac{d^e + d^q + d^t}{k^{pro} + k^o}, \tag{9}$$

where, κ_d, d^e, d^q, and d^t indicate the delay weight, execution delay, queue delay, and transmission delay, respectively. In the proposed model, queue delay shows the waiting time at the queue of the f_n^j and execution delay depends upon the CPU speed of the f_n^j. In the wireless network, we compute the transmission delay as follows:

$$d^t = \frac{2D\kappa^o}{TR_{s,f_n^j}}, \tag{10}$$

$$TR_{s,f_n^j} = Bwd.log(1 + \frac{CG_{s,f_n^j} P_{tx,s}}{B.N_p}), \tag{11}$$

where, D indicate the data size of a task and TR_{s,f_n^j} is the transmission rate from source *s* node to f_n^j node. The channel gain $CG \cong \omega_i d_{s,f_n^j}^{\omega_j}$ between the nodes with distance d_{s,f_n^j} and path loss constant ω_i and exponent ω_j. The variables *Bwd*, P_{tx}, and N_p show the bandwidth per node, transmission power, and noise power. The normalized thermal noise power density is *174 dBm/Hz.*

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSE.2020.3047454, IEEE Transactions on Network Science and Engineering

IV. ARTNET: THE PROPOSED METHOD FOR SYNCHRONIZATION AND LOAD BALANCING

The proposed ARTNet consists of two stages: 1) synchronization of agents and 2) load balancing at the fog layer in a dynamic SDV-F environment.

A. An optimized synchronization algorithm

In the proposed ARTNet, we use the Q-function of the deep neural network (DNN) to foresee the quality of a state-action pair value.

$$S \times A \rightarrow R. \tag{12}$$

$$Q^{*}(s,a) = E[R(s,a) + \gamma \max_{a^{'} \in A_{s^{'}}} Q^{*}(s^{'}, a^{'})], \tag{13}$$

where Q-function for a state-action pair is given in Eq. (13). Additionally, $A_{s^{'}}$ indicates the set of actions at the next state $s^{'}$ and ($s^{'}$, $a^{'}$) is the state-action pair at the next time slot. Since in the proposed model, we use DNN as the function approximation of the agent's Q-function *Q(s, a)*. It is parametric by the set of adjustable parameters λ, representing the weights of the DNN. The update, λ, is adjusted to reduce the gap between the estimated and the optimal values. In particular, the following loss function using the mean-squared error measurement is defined for adjusting λ:

$$loss(\lambda) = E[(x - Q_{\lambda}(s,a))^2], \tag{14}$$

s.t.,

$$x = R(s,a) + \gamma \max_{a^{'} \in A_{s^{'}}} Q_{\lambda}(s^{'}, a^{'})], \tag{15}$$

is the estimation of the maximum accumulated future reward for the agents synchronization process. After the synchronization process, this paper also considers a dynamic load balancing technique using distributed fog computing mechanism in which nodes can offload their computation tasks to a neighboring node with available queue spaces in terms of computing capabilities and task demands distributions.

B. An optimized task-offloading algorithm

Based on the dynamic nature of IoV, the AI controller or secondary controller cannot predict the P and R correctly. Due to which the reward R or probability distribution is not changed. To handle this issue, in the proposed ARTNet, we have used the RL technique. In RL, the agent makes the decision based on experiences and improve system performance. Therefore, in the proposed ARTNet model, we use the Q-Learning to predict the optimal (s, a) policy in the system. For the current s, a, Q-Learning learn continuously and observes the new state $s^{'}$ and the r. Therefore, continuously interact with the environment and based on observations, the Q-Leaning algorithm update the Q-function to make the optimal decision for the new task, as shown in Eq. 16,

$$Q^{*}(s,a) = \leftarrow (1-\beta)Q^{*}(s,a) + \beta[R(s,a) + \alpha \overset{'}{\max_{a}} \in A_{s^{'}} Q^{*}(s^{'}, a^{'})], \tag{16}$$

where, β shows the learning rate, i.e., $(0 < \beta < 1)$, of a Q-learning, that modify the reward function R based on the new learning rate and learned estimations from the SDV-F architecture.

Algorithm 1: EVALUATE THE POLICY

```
Input: ∀s ∈ $S$, $\pi(s) = a \in A$, $U(s) = 0$
Output: Evaluate the policy $\pi(s)$
1  $policy - stable = True$
2  for $s \in S$ do
3      tmp = $\pi(s)$
4      /*compute new policy*/
5      compute Eq. 3
6      /*policy is stable or not?*/
7      if tmp $\neq \pi(s)$ then
8          policy-stable = False
9  if policy-stable then
10     exit
11 /*if policy is change then recompute s */
12 else
13     goto step-2
```

In ML, the Greedy algorithm straightforward selects the action with the height foreseen value, i.e., $a_t = arg \max_{a} Q_t(a)$. Consequently, the greedy action selection eternally utilizes the network's current information to maximize the next reward function. Hence, the ε-greedy algorithm is a crucial factor in Q-Learning and independently selects greedily random actions with equal likelihoods. The RL calls this greedy selection and the ε likelihood of stochastic election as exploration and exploitation policies. In Q-Learning, the exploitation maximizes the foreseen reward function on the one-step, and exploration yields the most eminent total reward in the long term.

The proposed model used Eq. 4 to compute the appropriate reward value about the selection of fog node for task computation and an approximate next state $s^{'}$ is obtained after defining its components, as mentioned in Eq. 4. The next fog node with the task to be allocated in the proposed model and the tasks' size following each node task arrival are stochastic. Therefore, the proposed scheme demonstrates them using Poisson random variables. Algorithm 2 explain the optimized task-offloading algorithm.

V. PERFORMANCE EVALUATION AND RESULTS

A. Experiment Setup

For the experimental setup, we consider an IoV environment, which is based on the SDV-F framework. The total coverage area we considered is $250 \times 250m^2$. In addition, data from each vehicle (i.e., a content) in the IoV setting are being transmitted at the rate of $t^i \in [0, 5]$ using Poison process [36], [37]. Consequently, we distributed 7 RSUs as Fog nodes in the network at the roadside. For consistency in the experimental environment, we set the parameters as follows. We have kept the content size constant ($cs = 20$ Mb), the required CPU size for each content is $Rcpu = 1$ cycle/bit,

Algorithm 2: EFFICIENT TASK OFFLOADING IN SDV-F BASED ON Q-LEARNING

```
Input: task arrival rate ($k_{arrival-rate}$), $\beta$, $\lambda$, $\gamma$, $d$
       (distance), and $\varepsilon$.
Output: Optimized task offloading process: the best
        available node
Set Q(s, a) = 0, s $\in$ S, a $\in$ A, i = 0,
for $i \leq max_i$ do
    select $a \in A$
    task offloading based on $a$
    observe $s^{'}$ and $r^{'}$
    update Q-function (Eq. 16)
    $s \leftarrow s^{'}$
    $i \leftarrow i+1$
end
```

the transmission capacity, $Txpw$ is 100 mW, the transmission delay is $Txd = 0.2$ second(s) and the bandwidth is 100 MB/s.

In the proposed ARTNet system, the $Pathloss[dB] = 128.2 + 37.5log10(d[m])$ indicates the channel gain when being offloaded to the fog nodes, and the noise power is set to $174dBm/hz$. During the computation process, the fog nodes consume energy, $8 \times 10^{-5}W$, and the energy constraint is $Ecr = 1W.h/GHz$ for the use of computing resources. Furthermore, the total content types are 60. The size of each content is 25Mb. In the simulation, we vary the number of IoVs from 100 to 500 and also vary the size of a fog node's computational power from 5 to 40 GHz.

We considered two baseline models for the performance assessment of the proposed ARTNet framework: No RSU Peer Offloading (NRPO) [38] and Average Energy Constraint (AEC) [39]. To reduce the energy usage of end devices [38], the authors suggested the energy-based offloading scheme. The proposed model combines resource planning and dynamic offloading to mitigate full-time collaborative application minimization, including energy consumption. The proposed model performs peer offloading and their related RSUs process the obtained tasks from vehicles. Without analyzing long-term energy constraints, this technique will accomplish tasks. The author implemented a scheme in [39] that studies the mobility function along with background traffic. By modelling an average restriction on RSU in each time slot, this scheme aims to obey the long-term energy constraint rigidly.

B. Results and Discussion

In the simulation of the proposed ARTNet model and baselines, we consider the following parameters for evaluation.

1) Network Performance under different time-slots: We compare the results of the proposed ARTNet and baselines in-terms of latency (s), energy consumption (J), and energy shortfall, as shown in Fig 2. From the results, we can analyze that the performance of ARTNet is much better than NRPO and AEC. The average latency in ARTNet much less than AEC and NRPO. However, the time-slot increases from 20, the average latency of ARTNet, and AEC becomes almost the same. Additionally, the average latency of NRPO is more than ACE and ARTNet, as shown in Fig 2(a).

Energy consumption in ARTNet is significantly less than the baselines. The NRPO performs poorly in terms of average latency as well as more energy consumption, as shown in Fig 2(b).

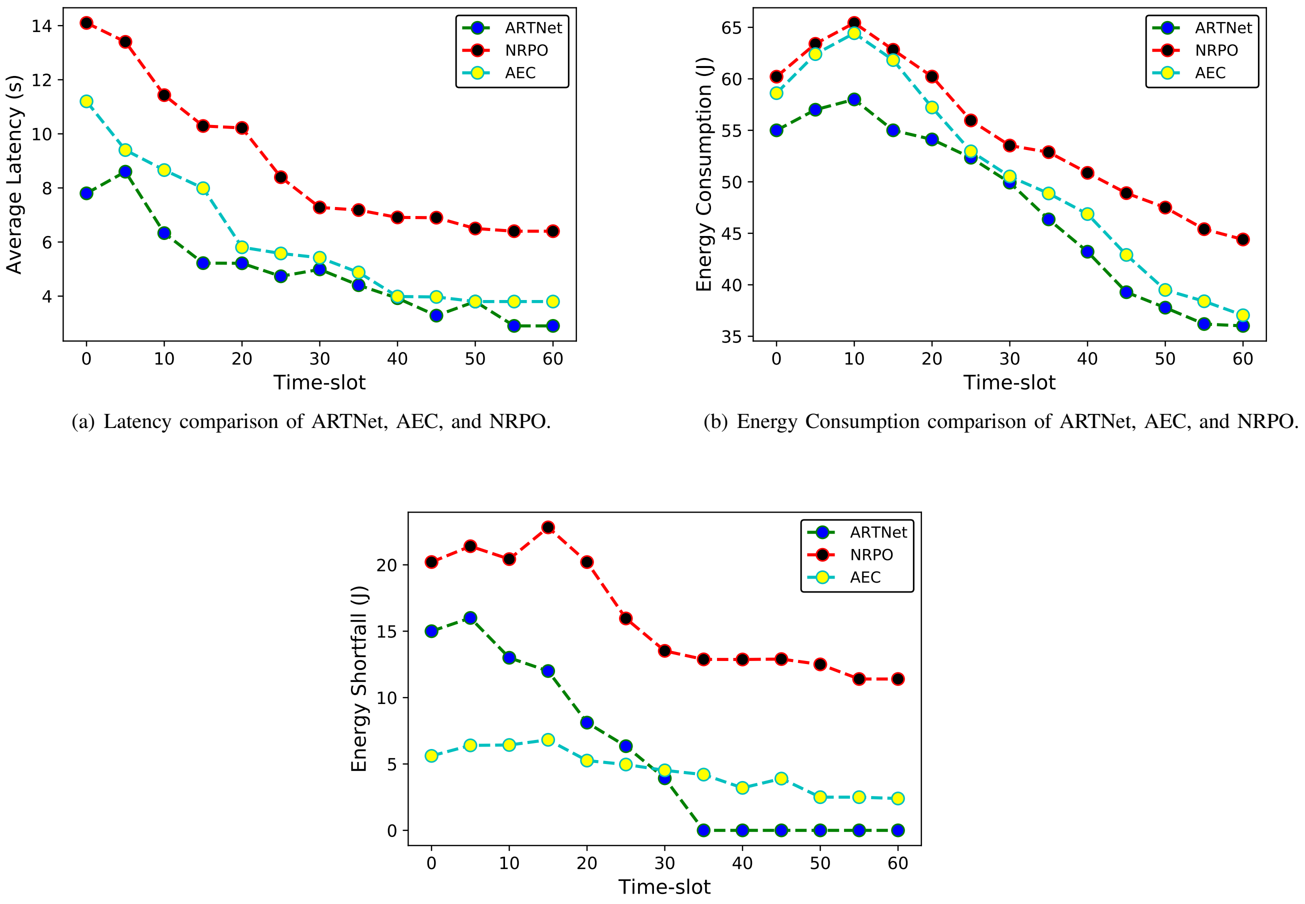
In Fig. 2(c), the energy shortfall for all models is shown. The results showed that the energy shortfall in the AEC model is very less at the start. However, when the time-slot increases from 30, then the energy shortfall in our proposed ARTNet approach becomes less than the AEC model. Energy shortfall in ARTNet becomes zero when the time-slot increases from 35. However, the NRPO model performs poorly in all parameters. The main reason is that NRPO does not use peer offloading; that's why it has consumed more energy and have large latency. The ARTNet model's success in terms of lower latency, less energy consumption, and lower energy shortfall, is it intelligently distributed the tasks at the fog layer using resource pooling. Secondly, the ARTNet assigns the tasks to those fog nodes, which have fewer tasks for execution. This is done through the agent's synchronization process. The average latency of ARTNet is almost 5s, energy consumption is almost 15%, and energy shortfall is 15% less than NRPO.

2) Impact of No. of IoVs: In Fig.4, the performance evaluation of our ARTNet model with baselines in the contexts of various numbers of IoVs in the SFV-F architecture is shown. Our proposed ARTNet performs better in terms of average energy consumption, average latency, average energy shortfall, and average overload probability when the IoVs varies from 100 to 500 in the simulation. From the results, we can notice that average energy consumption increases linearly when the number of IoVs increase in all models. However, the average energy consumption in NRPO is more than ACE and ARTNet. Additionally, when the number of IoV increases, ARTNet and AEC consume almost the same energy. However, ARTNet has lower latency and less energy shortfall when the IoV reaches 500, as shown in Fig. 3(a), Fig. 3(b) and Fig. 3(c). ARTNet consumes almost 10% less energy than NRPO.

In Fig. 3(d), the average overload probability is shown. In NRPO, the overload probability is more than 60% when the IoVs reach 500, and in AEC, the overloading probability is more than 50%. However, the proposed model distributed the tasks efficiently. Therefore the overloading probability is much less than the baselines.

The above simulation results show that when the agent assigns the task to fog nodes based on task load, task arrival rate, and computing power, it decreases the average latency and energy consumption. The synchronization process also increases the performance of the SDV-F network. This can be achieved with the help of Q-learning based offloading decision.

3) Failure Recovery of SDN Controller: In this study, we confirm how the secondary/fog agents' fallback mechanism in the ARTNet is used. The proposed ARTNet gives low average latency if the connection to the SDN controller is lost. The failure scenario where the SDN controller fails for 10, 20 and 30 seconds are shown in Fig. 4(a). The results show that when the centralized SDN controller is down, the average latency increases instantly. This means that the SDN controller



(a) Latency comparison of ARTNet, AEC, and NRPO.

(b) Energy Consumption comparison of ARTNet, AEC, and NRPO.

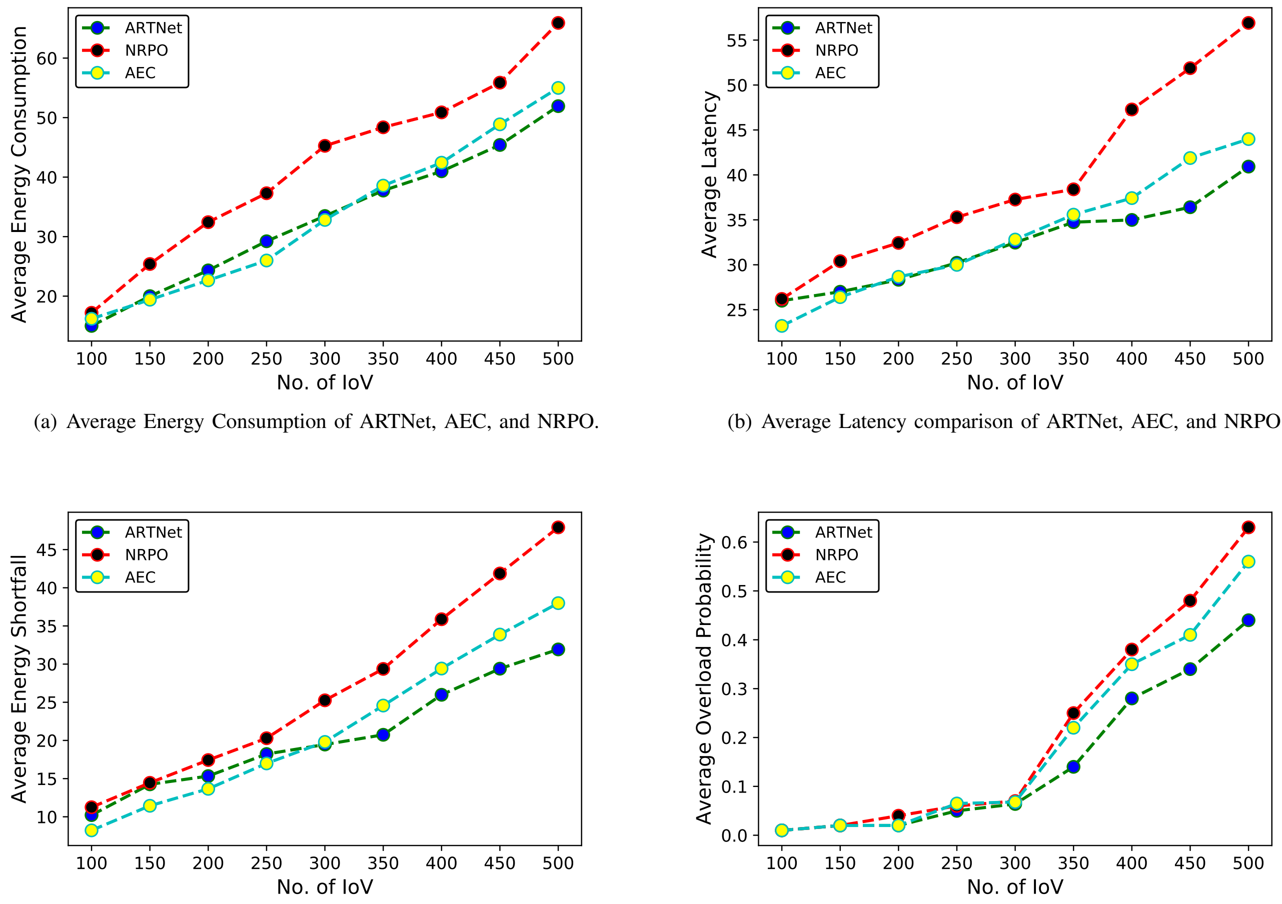
(c) Energy Shortfall comparison of ARTNet, AEC, and NRPO.

Fig. 2: Performance comparison of ARTNet, AEC, and NRPO w.r.t time-slots.

no longer implements the ARTNet task estimation rules. As seen in Fig. 4(a), the long downtime of the SDN controller raises the average latency, and also affects the latency when the SDN controller becomes working. After a brief downtime, e.g., 10 and 20 seconds, devices are equipped with almost normal services by the controller. However, due to network congestion, the controller delivers service with a delay for a longer downtime, e.g., 30 seconds. The results show that average latency reaches almost 100ms when the controller downtime reaches 30 seconds. The proposed ARTNet considers secondary agents to minimize the average latency. In ARTNet, the secondary fog control serves as a secondary controller when the primary controller fails. As shown in Fig 4(b), the results show that the secondary controller offers services to devices with low latency. We can see that the secondary controller in the ARTNet effectively reduces the latency. Also, the findings show that when the controller downtime exceeds 30 seconds, the average latency is less than 50ms, which is almost 50 ms less than Fig. 4(a).

VI. Conclusion

In this paper, we introduced a novel framework (ARTNet), which ensures ultra-reliable low-latency communications in highly dynamic environments in V2X using SDV-F architecture. The primary purpose of ARTNet is to address the challenges of maximizing resource utilization and optimizing the distribution of traffic. There are many requests simultaneously in the IoV setting, while the queuing and processing of requests results in the end-to-end latency of time-critical applications. Therefore, the proposed ARTNet distributes the fog layer's traffic load according to the computational power and load on each fog node. We have modeled the transmission process of time-critical applications data in SDV-F and computing nodes' selection as a Markov decision process. ARTNet has the potential to become a flagship model for the design of the future cooperative intelligent transport system. ARTNet can also integrate, deploy, and maintain an intelligent, cooperative, distributed environment for connected cars. This will lead to a transformation in society that will result in shorter travel time, less congestion, and reduced road accidents. The load balancing approach used in ARTNet is based on a centralized strategy. To improve fog networks'



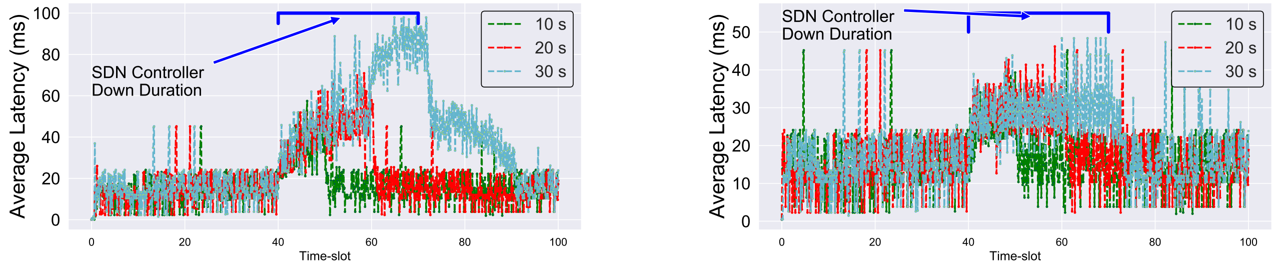
(a) Average Energy Consumption of ARTNet, AEC, and NRPO.

(b) Average Latency comparison of ARTNet, AEC, and NRPO.

(c) Average Energy Shortfall comparison of ARTNet, AEC, and NRPO.

(d) Average Overload Probability comparison of ARTNet, AEC, and NRPO.

Fig. 3: Performance comparison of ARTNet, AEC, and NRPO w.r.t No. of IoV.



(a) SDN controller failure without Secondary Agents

(b) SDN controller failure with Secondary Agents

Fig. 4: Impact of Failure of SDN Controller.

survivability and reliability, we will consider a distributed load balancing strategy suitable in fog networks in future work.

References

[1] J. Lin, W. Yu, N. Zhang, X. Yang, H. Zhang, and W. Zhao, "A survey on internet of things: Architecture, enabling technologies, security and privacy, and applications," *IEEE Internet of Things Journal*, vol. 4, no. 5, pp. 1125–1142, Oct 2017.

[2] S. Russell and P. Norvig, "Artificial intelligence: a modern approach," 2002.

[3] M. Ibrar, L. Wang, G.-M. Muntean, A. Akbar, N. Shah, and K. R. Kaleem, "Prepass-Flow: A Machine Learning based Technique to Minimize ACL Policy Violation due to Links Failure in Hybrid SDN," *Computer Networks*, vol. DOI: https://doi.org/10.1016/j.comnet.2020.107706, 2020.

[4] A. Akbar, M. Ibrar, M. A. Jan, A. K. Bashir, and L. Wang, "SDN-enabled Adaptive and Reliable Communication in IoT-Fog Environment using Machine Learning and Multi-Objective Optimization," *IEEE In-*

ternet of Things Journal, pp. 1–1, 2020.

[5] M. Ibrar, L. Wang, G. M. Muntean, J. Chen, N. Shah, and A. Akbar, "IHSF: An intelligent solution for improved performance of reliable and time-sensitive flows in hybrid SDN-based FC IoT systems," *IEEE Internet of Things Journal*, pp. 1–1, 2020.

[6] X. Wang, Z. Ning, X. Hu, L. Wang, L. Guo, B. Hu, and X. Wu, "Future communications and energy management in the internet of vehicles: Toward intelligent energy-harvesting," *IEEE Wireless Communications*, vol. 26, no. 6, pp. 87–93, 2019.

[7] A. Bhati and P. Singh, "Analysis of various routing protocol for vanet," in *Cybernetics, Cognition and Machine Learning Applications*. Springer, 2020, pp. 315–325.

[8] M. Dixit, R. Kumar, and A. K. Sagar, "Vanet: Architectures, research issues, routing protocols, and its applications," in *2016 International Conference on Computing, Communication and Automation (ICCCA)*. IEEE, 2016, pp. 555–561.

[9] C. Zhu, G. Pastor, Y. Xiao, Y. Li, and A. Ylae-Jaeaeski, "Fog following me: Latency and quality balanced task allocation in vehicular fog computing," in *2018 15th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*. IEEE, 2018, pp. 1–9.

[10] A. J. Kadhim and S. A. H. Seno, "Maximizing the utilization of fog computing in internet of vehicle using sdn," *IEEE Communications Letters*, vol. 23, no. 1, pp. 140–143, 2018.

[11] J. C. Nobre, A. M. de Souza, D. Rosário, C. Both, L. A. Villas, E. Cerqueira, T. Braun, and M. Gerla, "Vehicular software-defined networking and fog computing: Integration and design principles," *Ad Hoc Networks*, vol. 82, pp. 172–181, 2019.

[12] N. B. Truong, G. M. Lee, and Y. Ghamri-Doudane, "Software defined networking-based vehicular adhoc network with fog computing," in *2015 IFIP/IEEE International Symposium on Integrated Network Management (IM)*. IEEE, 2015, pp. 1202–1207.

[13] J. Khoury, H. Sami, H. Safa, and W. El-Hajj, "On the use of software defined wireless network in vehicular fog computing environments," in *2019 15th International Wireless Communications & Mobile Computing Conference (IWCMC)*. IEEE, 2019, pp. 1198–1203.

[14] Y. Zhang, H. Zhang, K. Long, Q. Zheng, and X. Xie, "Software-defined and fog-computing-based next generation vehicular networks," *IEEE Communications Magazine*, vol. 56, no. 9, pp. 34–41, 2018.

[15] E. Borcoci, T. Ambarus, and M. Vochin, "Distributed control plane optimization in sdn-fog vanet," *ICN 2017*, p. 135, 2017.

[16] A. A. Khan, M. Abolhasan, and W. Ni, "5g next generation vanets using sdn and fog computing framework," in *2018 15th IEEE Annual Consumer Communications & Networking Conference (CCNC)*. IEEE, 2018, pp. 1–6.

[17] A. J. Kadhim and S. A. H. Seno, "Energy-efficient multicast routing protocol based on sdn and fog computing for vehicular networks," *Ad Hoc Networks*, vol. 84, pp. 68–81, 2019.

[18] M. Arif, G. Wang, T. Wang, and T. Peng, "Sdn-based secure vanets communication with fog computing," in *International Conference on Security, Privacy and Anonymity in Computation, Communication and Storage*. Springer, 2018, pp. 46–59.

[19] M.-H. Chen, B. Liang, and M. Dong, "Joint offloading and resource allocation for computation and communication in mobile cloud with computing access point," in *IEEE INFOCOM 2017-IEEE Conference on Computer Communications*. IEEE, 2017, pp. 1–9.

[20] M. Whaiduzzaman, A. Naveed, and A. Gani, "Mobicore: Mobile device based cloudlet resource enhancement for optimal task response," *IEEE transactions on services computing*, vol. 11, no. 1, pp. 144–154, 2016.

[21] J. Shuja, A. Gani, K. Ko, K. So, S. Mustafa, S. A. Madani, and M. K. Khan, "Simdom: A framework for simd instruction translation and offloading in heterogeneous mobile architectures," *Transactions on Emerging Telecommunications Technologies*, vol. 29, no. 4, p. e3174, 2018.

[22] M.-H. Chen, M. Dong, and B. Liang, "Joint offloading decision and resource allocation for mobile cloud with computing access point," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 3516–3520.

[23] Z. He, J. Cao, and X. Liu, "Sdvn: Enabling rapid network innovation for heterogeneous vehicular communication," *IEEE network*, vol. 30, no. 4, pp. 10–15, 2016.

[24] X. Huang, R. Yu, J. Kang, Y. He, and Y. Zhang, "Exploring mobile edge computing for 5g-enabled software defined vehicular networks," *IEEE Wireless Communications*, vol. 24, no. 6, pp. 55–63, 2017.

[25] P. Dai, K. Liu, X. Wu, Z. Yu, H. Xing, and V. C. S. Lee, "Cooperative temporal data dissemination in sdn-based heterogeneous vehicular networks," *IEEE Internet of Things Journal*, vol. 6, no. 1, pp. 72–83, 2018.

[26] G. Luo, J. Li, L. Zhang, Q. Yuan, Z. Liu, and F. Yang, "sdnmac: A software-defined network inspired mac protocol for cooperative safety in vanets," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 6, pp. 2011–2024, 2018.

[27] K. Liu, L. Feng, P. Dai, V. C. Lee, S. H. Son, and J. Cao, "Coding-assisted broadcast scheduling via memetic computing in sdn-based vehicular networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 8, pp. 2420–2431, 2017.

[28] A. Akbar and P. R. Lewis, "Towards the optimization of power and bandwidth consumption in mobile-cloud hybrid applications," in *2017 Second International Conference on Fog and Mobile Edge Computing (FMEC)*, 2017, pp. 213–218.

[29] A. Akbar and P. R. Lewis, "The importance of granularity in multiobjective optimization of mobile cloud hybrid applications," *Transactions on Emerging Telecommunications Technologies*, vol. 30, no. 8, p. e3526, 2019.

[30] A. Akbar, P. R. Lewis, and E. Wanner, "A self-aware and scalable solution for efficient mobile-cloud hybrid robotics," *Frontiers in Robotics and AI*, vol. 7, p. 102, 2020.

[31] H. O. Hassan, S. Azizi, and M. Shojafar, "Priority, network and energy-aware placement of iot-based application services in fog-cloud environments," *IET Communications*, vol. 14, no. 13, pp. 2117–2129, 2020.

[32] M. Mukherjee, V. Kumar, S. Kumar, R. Matamy, C. X. Mavromoustakis, Q. Zhang, M. Shojafar, and G. Mastorakis, "Computation offloading strategy in heterogeneous fog computing with energy and delay constraints," in *ICC 2020-2020 IEEE International Conference on Communications (ICC)*, 2020, pp. 1–5.

[33] M. Mukherjee, S. Kumar, M. Shojafar, Q. Zhang, and C. X. Mavromoustakis, "Joint task offloading and resource allocation for delay-sensitive fog networks," in *ICC 2019-2019 IEEE International Conference on Communications (ICC)*. IEEE, 2019, pp. 1–7.

[34] K. Liu, X. Xu, M. Chen, B. Liu, L. Wu, and V. C. Lee, "A hierarchical architecture for the future internet of vehicles," *IEEE Communications Magazine*, vol. 57, no. 7, pp. 41–47, 2019.

[35] H. Li, M. Dong, and K. Ota, "Radio access network virtualization for the social internet of things," *IEEE Cloud Computing*, vol. 2, no. 6, pp. 42–50, 2015.

[36] M. Khabazian and M. K. M. Ali, "A performance modeling of connectivity in vehicular ad hoc networks," *IEEE Transactions on Vehicular Technology*, vol. 57, no. 4, pp. 2440–2450, 2008.

[37] M. T. Abbas, A. Muhammad, and W.-C. Song, "Sd-iov: Sdn enabled routing for internet of vehicles in road-aware approach," *Journal of Ambient Intelligence and Humanized Computing*, vol. 11, no. 3, pp. 1265–1280, 2020.

[38] S. Guo, B. Xiao, Y. Yang, and Y. Yang, "Energy-efficient dynamic offloading and resource scheduling in mobile cloud computing," in *IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on Computer Communications*. IEEE, 2016, pp. 1–9.

[39] Q. Liao and D. Aziz, "Modeling of mobility-aware rrc state transition for energy-constrained signaling reduction," in *2016 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2016, pp. 1–7.