# On System Stability in Multitier Roadside Computing Toward an Intelligent Transportation

Nhu-Ngoc Dao, *Senior Member, IEEE*, Duc-Nghia Vu, Anh-Tien Tran, Trung V. Phan, *Member, IEEE*, Schahram Dustdar, *Fellow, IEEE*, and Sungrae Cho

***Abstract*—Owing to the heterogeneity and massiveness of data generated by connected vehicles, multitier roadside computing (MRC) plays a key role in an intelligent transportation system (ITS). MRC provides a localized cloudization capability in close proximity to the connected vehicles. Because the massive data correspondingly necessitate a high computing energy consumption, stable workload processing with respect to energy efficiency is a crucial problem of MRC. To address this problem, we propose an energy-efficient workload (E2W) scheduling algorithm for flexibly handling the random offloading traffic from connected vehicles. In our E2W algorithm, an MRC is transformed into a multitier queuing system, where the workload arrived from the vehicles and the computing capability of the roadside units are considered to be arrival and departure processes, respectively. The departure rate that closely impinges on the computing energy consumption is supervised using the Lyapunov drift-plus-penalty policy to achieve efficient energy reduction while maintaining service satisfaction. In addition, the deterministic upper bound of the Lyapunov optimization provides the MRC system with stable operation. Simulation results demonstrate that the E2W algorithm outperforms existing optimization strategies in terms of energy efficiency and system stability.**

***Index Terms*—multitier roadside computing, system stability, vehicular communication, mobile edge computing.**

## I. Introduction

IOTIZATION has dramatically promoted the evolution of next-generation intelligent transportation systems (ITSes), where a large number of vehicles are connected in a unified networking infrastructure, realizing an Internet of vehicles (IoV) paradigm. In an IoV paradigm, the vehicles, regardless of their hardware performance, are connected to innovate heterogeneous services, which have diverse requirements in terms of latency, bandwidth, and reliability [1], [2]. As vehicles are considered to be lightweight devices from a computing perspective, they are seen to increasingly transfer their workload to the network for an offloaded processing [3]. For instance, sensor and tracking-camera information in autonomous driving cars is offloaded to the network in order to acquire optimal guidance for drivers. An in-vehicle infotainment system requires comprehensive image processing to offer passengers a virtual-reality game for relaxation on the move [4], [5]. Under these circumstances, owing to the heterogeneity and massiveness of the offloaded data, flexible and powerful networking and computing infrastructures are required to achieve satisfactory performance.

Emerging multi-access edge computing (MEC) technologies [6], [7] are considered to be a solution to the aforementioned challenges. As defined by the European telecommunications standards institute (ETSI), MEC provides *cloud-computing capabilities and an IT service environment at the edge of the network* [8]. Exploited the advantages introduced by MEC technologies, a fog-enabled access network [9]–[11] has been developed, which is constituted by high power nodes (a.k.a. macro remote radio heads) and multiple fog access points located at small base station/femtocell/remote radio heads. In this model, fog access points have low computational power but very low response times while high power nodes have high computational power but higher response times. Matching into this model, we proposed a 2-tier computing model for connected vehicle networks, referred to as multitier roadside computing (MRC) platforms. Here, an integration of MEC into ITSes can offer the advantage of a flexible hierarchical computing infrastructure. In particular, MRC orchestrates computing capabilities among heterogeneous roadside units (RSUs) such as macro base stations (MBSs) with high computing power in upper tier and small base stations (SBSs) and road traffic control (RTC) devices with small computing power in lower tier. Fig. 1 depicts a typical two-tier MRC system. Almost all vehicles (including personal devices of passengers) connect to RSUs in the lower tier, i.e., tier-$\alpha$ MRC. A part of the workload that is offloaded from the vehicles is delivered to the upper tier (tier-$\beta$ MRC) for further processing. In this model, tier-$\beta$ is assumed to possess higher cloudization capability and latency compared to tier-$\alpha$. For instance, a federated learning based navigation system may deploy pre-configured learning models at every tier-$\alpha$ RSUs for local traffic training while the central scheme which fuses these local models is located at tier-$\beta$ RSUs owing to high processing requirement.

As aforementioned, an effective scheduler, which manages workload distribution among RSUs, plays an important role in harmonizing the computing power of diverse RSUs in the MRC platform. In particular, massive IoV data generated by

This work was supported in part by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. 2021R1G1A1008105) and in part by the National Research Foundation of Korea (NRF) under Grant NRF-2019R1A2C1090447 funded by the Korea Government (Ministry of Science and ICT). *(Corresponding author: Sungrae Cho).*

N.-N. Dao is with Sejong University, Department of Computer Science and Engineering, Seoul, Republic of Korea (email: nndao@sejong.ac.kr).

D.-N. Vu, A.-T. Tran, and S. Cho are with Chung-Ang University, School of Computer Science and Engineering, Seoul, Republic of Korea (email: dnvu@uclab.re.kr, attran@uclab.re.kr, srcho@cau.ac.kr).

T. V. Phan is with Technische Universität Chemnitz, Chair of Communication Networks, Germany (email: trung.phan-van@etit.tu-chemnitz.de).

S. Dustdar is with the Distributed Systems Group, TU Wien, Austria. (email: dustdar@dsg.tuwien.ac.at).

the connected vehicles correspondingly results in a high computing energy consumption. Consequently, energy efficiency is of importance in the MRC platform to reduce the overall cost of network operations for such in-network computation services. In contrast, the heterogeneity of IoV data requires a flexible schedule in order to assign each workload to an appropriate RSU for user service satisfaction. Motivated from this status quo, stable and dynamic workload processing with respect to energy efficiency has been considered in our study as one of the main purposes of the scheduler to reduce the overall cost of system operations.

Literature reviews [12], [13] have shown that cutting-edge workload scheduling techniques can be classified into two categories: *user satisfaction-aware* and *system resource-aware* approaches. User satisfaction-aware approaches aim at maximizing user satisfaction such as response latency and service availability. These targets may require the approaches with resource exhaustion in MRC systems in order to achieve the one-handed optimization for user devices. In contrast, system resource-aware approaches mainly focus on minimizing the amount of MRC resources consumed for offloaded workload execution within baseline requirements instead of high quality of service. Therefore, to overcome the imbalanced performance of these approaches, several hybrid solutions have been proposed [14]–[16]. However, none of them have paid particular attention to internal orchestration among heterogeneous computing entities (i.e., RSUs) inside the MRC system.

In this study, we propose an energy-efficient workload (E2W) scheduling algorithm to resolve the aforementioned problems in heterogeneous MRC platforms. The contributions of this paper are summarized as follows.

- The MRC platform is modeled as a multitier queuing system from a computing perspective. In this model, traffic offloaded from the connected vehicles and the computing capability of the RSUs play the roles of arrival and departure processes, respectively. To ensure the generality, the arrival process is considered to be stochastic while the departure process is controllable.
- A dynamic trade-off between energy consumption and service buffer, referred as energy-efficient workload (E2W) scheduling algorithm, is developed using the *Lyapunov drift-plus-penalty* (DPP) policy [17]. In this context, the energy consumption is minimized by controlling the RSUs' computing capabilities. On the contrary, a deterministic upper bound provided by the Lyapunov optimization ensures the MRC system buffer stable.
- Comprehensive simulation analysis was conducted to prove the outperformance of the proposed E2W algorithm compared to existing solutions. The evaluation consists of three folds: trade-off factor selection, workload distribution, and performance metrics.

The remainder of this paper is organized as follows. Related work is presented in section II, and the system model is illustrated in section III. The proposed E2W algorithm is described and analyzed in section IV. Section V discusses performance evaluation. Finally, the paper is concluded in section VI.

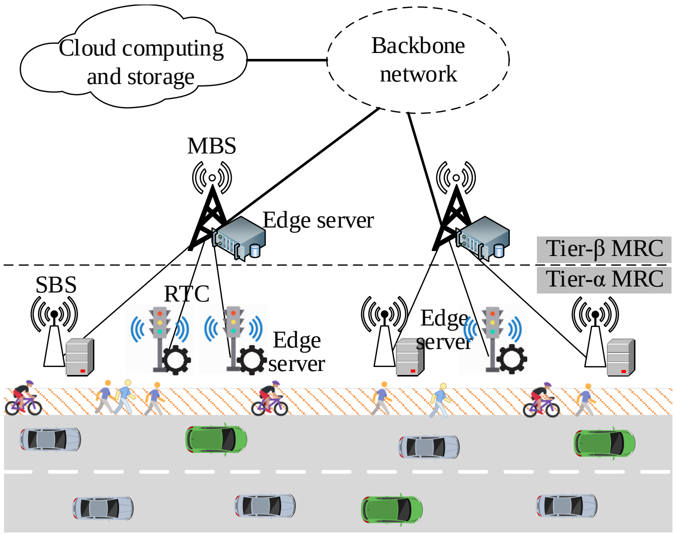


Fig. 1: Multitier roadside computing platforms.

## II. Literature Review

Following the taxonomy described in Section I, existing workload scheduling techniques can be classified as (*i*) user satisfaction-aware [18]–[23] and (*ii*) system resource-aware approaches [24]–[28], as well as (*iii*) their hybrid solutions [14]–[16], [29]

In user satisfaction-aware approaches, service availability and latency minimization, two fundamental factors of quality of service (QoS), are highly prioritized. In [18], Munoz *et al.* described a method to adjust the uplink data rate that minimizes the latency experienced by users with respect to the targeted energy cost. The decision is made by carefully evaluating the impacts of the transmission rate and the load of the system on the QoS. To utilize potential dynamical mobile users' connectivity, Pu *et al.* [19] proposed a device-to-device (D2D) fogging framework to achieve energy-efficient task executions for network-wide users. This framework devises efficient task scheduling policies and proactively adapt to various features of the task type, user amount, and task generation frequency. In the field of mobile applications, Dolezal *et al.* [20] implemented a computation offloading framework to cope with low-level communication between applications and a small cell cloud, which consists of cloud-enabled small cells (CeSCs) serving as radio end-points for mobile users. The offloading framework has a user stack, in addition to the application in compile-time to facilitate low-level offloading operations, to statically decide whether offloading should be performed instead of local execution on user devices according to the user's choice. Following this, the alleviated latency and reduced energy consumption of the UEs are proved by using an augmented reality (AR) application as a testbed basement. In another line of research, Liu *et al.* [21] set the minimum delay and the average power consumption at a mobile device as the goal of proposed efficient one-dimensional search algorithm to find the optimal task scheduling policy. The algorithm adopted a Markov decision process approach to address the problem of power-constrained delay minimization to schedule the computation tasks based on the queuing state of the task buffer, the execution state of the local processing unit, and the state of the transmission unit. By investigating the problem of minimizing the average energy consumed by all users under average delay constraints [22], Labidi *et al.* jointly optimized radio resource scheduling and computation offloading (CO) via offline and online dynamic programming approaches. The proposed solutions, derived from application rates, select only one user for scheduling and offloading and decide whether other users undergo local processing or stay idle. In [23], Jovsilo and Dan developed a theoretical game model of peer-aware and edge computing offload to improve user task execution performance in terms of latency minimization.

There have been several studies conducted on system resource-aware approaches, where approaches mainly focus on minimizing the amount of MRC resources consumed for offloaded workload execution within baseline requirements instead of high quality of service. For example, Yang *et al.* [24] derived an energy-efficient offloading optimization problem from mutual computational tasks and transmission requirements. The considered problem is addressed by an artificial fish swarm algorithm-based scheme to reach the global optimum in terms of energy efficiency. In [25], Wang *et al.* proposed an alternating direction method of a multipliers-based decentralized algorithm to find the global optimal solution for resource-aware perspectives such as computation offloading decision, resource allocation, and content caching strategy. In [26], Chen *et al.* designed a distributed offloading algorithm to achieve superior offloading performance and scale with an incremental user size. This algorithm transformed the offloading decision-making problem among multiple users into a potential game, proved its Nash equilibrium state, and used the advantages of game theory to solve it. In another study, Zhang *et al.* [27] took the energy cost of both task computing and file transmission into consideration, designed a three-stage energy-efficient CO scheme to jointly optimize offloading decisions and radio resource allocation strategies while preserving latency constraints. In this scheme, through type classification and priority assignment for users, the optimization problem is definitively processed in polynomial complexity. By taking into account the completion time and energy, Yu *et al.* [28] formulated a system cost minimization problem for MEC and proposed a distributed algorithm consisting of offloading strategy selection, clock frequency configuration, transmission power allocation, and channel rate scheduling. All optimal results exhibited a higher energy-efficient offloading performance compared to other existing algorithms.

To overcome the imbalanced performance of the above approaches, several hybrid solutions have been proposed. They mainly concentrate on finding an optimal solution that compromises between QoS and energy efficiency. In [14], a lightweight heuristic stabilized green cross-haul orchestration scheme, which utilizes Lyapunov-theory-based drift-plus-penalty policies, was proposed to jointly consider stabilization, energy efficiency, and latency for dense IoT offloading services. The scheme aimed at time-average minimization of energy consumption by providing an adjustable computing latency threshold. In another perspective, upstreaming IoT offloading services in fog radio access networks spurred Vu *et al.* [15] to formulate a joint energy and latency optimization scheme to strictly manage energy consumption, load balancing, and critical IoT service level of satisfaction. In [16], Cui *et al.* speculated on the problem of computation offload in a centralized MEC network with multi-cells to obtain a trade-off between average energy consumption of the system and users' latency. This issue was formulated into a constrained multi-objective optimization problem and was solved by a modified fast elitist non-dominated sorting genetic algorithm. Deng *et al.* [29] formulated a workload allocation problem in fog-cloud computing toward power consumption with service delay and used an approximate approach to decompose it into three subproblems. The optimal workload allocation, determined by the generalized Benders decomposition algorithm and Hungarian algorithm, showed that communication bandwidth and transmission latency can be saved by sacrificing modest computation resources.

These aforementioned studies have significantly improved edge computing performances from multiple perspectives. However, neither their one-handed optimization focuses in the user satisfaction- and system resource-aware categories nor the environmental adaptations of the hybrid category have not sufficiently taken into account an internal cross-tier orchestration among computing entities in the MRC system. This lack has inspired our study in this paper.

## III. Multitier Roadside Computing Platforms

### A. Computational Model

As described in Section I, an MRC system typically consists of two tiers, i.e., tier-$\alpha$ and tier-$\beta$. Tier-$\alpha$ has low computing power; however, it issues low response latency owing to its positioning in proximity to vehicles. On the contrary, tier-$\beta$ equips high computing capability and a high response latency for handling aggregated complex traffic. According to queuing theory [30], the MRC system can be modeled as a hierarchical queuing system. In this model, the offloaded workloads arrive at tier-$\alpha$ randomly; consequently, a part of them are delivered to tier-$\beta$ according to a descending-index-based order. The higher the response latency requirement and complexity of a workload, the higher the index it obtains. For convenience, a summary of key notations are described in Table I. It is worth noting that theoretical analysis can involve an upper tier to represent the central cloud. However, extending the system to cover one more tier may generate a significant complexity to the optimization problem. Instead, we can transform the extension into two-stage optimization, which consists of 2-tier edge computing (e.g., the MRC considered in our paper) in the first stage and edge–cloud computing orchestration in the second stage. Fortunately, the edge–cloud computing orchestration problem has been investigated thoroughly in many studies in the literature [31]. Here, we investigated the multitier in the first stage to complement the extended scenarios.

From a computing perspective, the $i$-th workload is characterized by a three-parameter tuple of $\langle u_i, c_i, r_i \rangle$, where $u_i$,

TABLE I: Key notation description.

| Notation | Description |
|---|---|
| $\langle u_i, c_i, r_i \rangle$ | Parameter tuple of $i$-th workload, where $u_i$, $c_i$, and $r_i$ are the workload size in bits, complexity, and response latency in ms, respectively. |
| $\lambda^t$ | Arrival rate at tier-$\alpha$ in timeslot $t$. |
| $\mu^t_{\alpha_i}$ and $\mu^t_{\beta_j}$ | Departure rate of $i$-th and $j$-th RSUs in tier-$\alpha$ and tier-$\beta$ in timeslot $t$, respectively. |
| $\kappa_\alpha$ and $\kappa_\beta$ | Coefficient factors of the RSUs in tier-$\alpha$ and tier-$\beta$, respectively. |
| $f_\alpha$ and $f_\beta$ | CPU frequencies of the RSUs in tier-$\alpha$ and tier-$\beta$, respectively. |
| $\boldsymbol{E}^t$ | Energy consumption of the MRC system in timeslot $t$. |
| $\boldsymbol{B}^t$ | Workload buffer (in bits) of the MRC system in timeslot $t$. |
| $Q^t_i$ | Virtual workload buffer (in cycles) of $i$-th RSU in tier-$\alpha$ in timeslot $t$. |
| $V$ | Lyapunov control factor. |

$c_i$, and $r_i$ are the workload size in bits, complexity, and response latency in ms, respectively. It is worth noting that the workload complexity parameter represents the difficulty of workload execution; hence, it is calculated by the average number of central processing unit (CPU) cycles required to process a bit of the workload, i.e, in Hz/b. Accordingly, a specific workload can be identified by its *virtual computing size*, which is determined by $u_i \times c_i$ in Hz. As a result, the arrival rate ($\lambda^t$) of the tier-$\alpha$ at timeslot $t$ is given by

$$\lambda^t \triangleq \sum_{\forall i \in \Lambda^t} \lambda^t_i = \sum_{\forall i \in \Lambda^t} (u_i \times c_i) \text{ in Hz}, \tag{1}$$

where $\Lambda^t$ is the arrived workload set at timeslot $t$. A timeslot is defined as a given duration. During this time, the system performs configured algorithms to obtain optimal operation parameters. Depending on real environmental implementations, a timeslot can be selected as several hundreds of ms (e.g., 100 ms) or several seconds (e.g., 3 or 5 s). Selecting the duration of a timeslot should consider how much the system changes in time. For instance, if environmental conditions and user traffic volume are fluctuated highly, a short timeslot should be used for timely adaptation to the changes.

In contrast, energy consumption for workload execution during each CPU cycle is assumed to be $\kappa f^2$ in Joule [14], [15], where $\kappa$ and $f$ are the coefficient factor and CPU frequency, respectively. Note that $\kappa$ varies depending on the CPU category. Hence, the MRC system consumes energy ($\boldsymbol{E}^t$) during timeslot $t$ owing to workload execution at both the two tiers

$$\boldsymbol{E}^t = \kappa_\alpha \sum_{i=1}^{N_\alpha} f^2_{\alpha_i} \mu^t_{\alpha_i} + \kappa_\beta \sum_{j=1}^{N_\beta} f^2_{\beta_j} \mu^t_{\beta_j}, \tag{2}$$

where $N_\alpha$ and $N_\beta$ are the number of RSUs at tier-$\alpha$ and tier-$\beta$, respectively. In addition, $\mu^t$ denotes the departure rate of the RSU at timeslot $t$. $\alpha_i$ and $\beta_j$ indicate the $i$-th and $j$-th RSUs at tier-$\alpha$ and tier-$\beta$, respectively. Consequently, the total workload buffer ($\boldsymbol{B}^t$) of the MRC system at timeslot $t$ is given by

$$\boldsymbol{B}^t = \boldsymbol{B}^{(t-1)} + \sum_{\forall i \in \Lambda^t} u_i - \left(\gamma_\alpha + \gamma_\beta\right), \tag{3}$$

where $\gamma_\alpha$ and $\gamma_\beta$ are the total workload processed by tier-$\alpha$ and tier-$\beta$, respectively, during timeslot $t$.

### B. Problem Clarification

(2) and (3) show that a minimization of both energy consumption $\boldsymbol{E}^t$ and system workload buffer $\boldsymbol{B}^t$ at timeslot $t$ is unachievable because of their dependence on the departure rates and processed workloads. Therefore, trade-off approaches have been utilized in order to address this situation. Typically, the trade-off function at timeslot $t$ is expressed as

$$\min : g(\boldsymbol{E}^t) + h(\boldsymbol{B}^t), \tag{4}$$

where $g(\cdot)$ and $h(\cdot)$ are the functions of $\boldsymbol{E}^t$ and $\boldsymbol{B}^t$, respectively. By extending equation (4), cumulative minimization of the trade-off during [0, $t$] is given by

$$\min : \sum_{\tau=0}^{t} \left(g(\boldsymbol{E}^\tau) + h(\boldsymbol{B}^\tau)\right). \tag{5}$$

Equation (5) obtains its minimum solutions if and only if the following time-average expression is minimized

$$\min : \frac{1}{t} \sum_{\tau=0}^{t} \left(g(\boldsymbol{E}^\tau) + h(\boldsymbol{B}^\tau)\right). \tag{6}$$

The MRC system is considered stable if equation (6) is achieved when $t \longrightarrow \infty$. To this end, a feasible strategy ($\mathcal{P}$) is used to minimize the time-average energy consumption, while keeping the workload buffer stabilized under its maximum threshold ($W$) as

$$(\mathcal{P}) \quad \min : \lim_{t \to \infty} \frac{1}{t} \sum_{\tau=0}^{t} \boldsymbol{E}^\tau \tag{7}$$

$$\text{s.t.} \quad \lim_{t \to \infty} \frac{1}{t} \sum_{\tau=0}^{t} \|\boldsymbol{Q}^\tau\|_1 \leq \mathrm{C}, \tag{7a}$$

$$\mathrm{C} \leq \mathrm{W}, \quad \mathrm{C} \text{ is constant}, \tag{7b}$$

$$\left\|\boldsymbol{Q}^t\right\|_1 \leq \mathrm{W}, \forall t \in [0, \infty). \tag{7c}$$

## IV. Energy-Efficient Workload Scheduling

### A. Joint Platform Stability and Energy Efficiency Optimization

As aforementioned in Section III-A, the MRC system has a stochastic property. The problem $\mathcal{P}$ represents a stochastic optimization of the MRC system, which aims at time-average minimization of energy consumption subject to workload buffer stabilization support. In this view, Lyapunov-theoretic optimization has been proven to be a potential solution [32]–[34]. In particular, studies have shown that the Lyapunov DPP policy provides a dynamic adjustment of energy consumption following a stochastic change in the workload buffer size. The outcomes of the Lyapunov DPP policy include a minimal energy consumption achievement and an assurance of an upper bound of the workload buffer. The Lyapunov DPP expression is given by

$$(\mathcal{P}) \quad \min : V\boldsymbol{E}^t + Q^t \left( \lambda^t - \left( \sum_{i=1}^{N_\alpha} \mu^t_{\alpha_i} + \sum_{j=1}^{N_\beta} \mu^t_{\beta_j} \right) \right), \tag{8}$$

where $V$ is the control factor that balances the ratio between energy consumption and workload buffer size. By using an appropriate $V$, the constraints (7b) and (7c) are ensured. In addition, $Q^t$ is the virtual workload buffer size in computing cycles, which is a scalar product of the workloads in bits and their complexities, i.e., $Q^t \propto \boldsymbol{B}^t, \{c_i\}_{\forall i \in \Lambda^t}$. Moreover, $Q^t = Q^{(t-1)} + \lambda^t - \left(\sum_{i=1}^{N_\alpha} \mu^t_{\alpha_i} + \sum_{j=1}^{N_\beta} \mu^t_{\beta_j}\right)$. Therefore, the problem $\mathcal{P}$ is equal to

$$\begin{aligned}\min : \mathcal{F}(f,\mu) = \sum_{i=1}^{N_\alpha}\Bigg(&Q^t_i\Bigg(\lambda^t_i - \Bigg(\mu^t_i + \sum_{j=1}^{N_\beta}\delta_{ij}\mu^t_{ij}\Bigg)\Bigg)\\ &+V\Bigg(\kappa_{\alpha_i}\left(f^t_{\alpha_i}\right)^2\mu^t_i + \sum_{j=1}^{N_\beta}\delta_{ij}\kappa_{\beta_j}\left(f^t_{\beta_j}\right)^2\mu^t_{ij}\Bigg)\Bigg)\end{aligned} \tag{9}$$

$$\text{s.t. } 0 \le \mu^t_i + \sum_{j=1}^{N_\beta}\delta_{ij}\mu^t_{ij} \le Q^t_i + \lambda^t_i, \quad \forall i, \tag{9a}$$

$$\mu^t_i, \mu^t_{ij} \ge 0, \quad \forall i, j, \tag{9b}$$

$$0 \le f^t_{\alpha_i} \le F^{\max}_{\alpha_i}, \quad \forall i, \tag{9c}$$

$$0 \le f^t_{\beta_j} \le F^{\max}_{\beta_j}, \quad \forall j. \tag{9d}$$

It is observed that the data processing rate of each RSU $f^t_{\alpha_i} \ge \mu^t_i$ and $f^t_{\beta_j} \ge \sum_{i=1}^{N_\alpha}\delta_{ij}\mu^t_{ij}$. In order to minimize the energy consumption of each RSU for processing tasks, the optimal data processing rate $\overline{f}^t_{\alpha_i} = \mu^t_i$ and $\overline{f}^t_{\beta_j} = \sum_{i=1}^{N_\alpha}\delta_{ij}\mu^t_{ij}$ [14], [21]. Thus, the problem (9) is equivalent to

$$\begin{aligned}\min : \mathcal{F}(\mu) = \sum_{i=1}^{N_\alpha}\Bigg(&Q^t_i\Bigg(\lambda^t_i - \Bigg(\mu^t_i + \sum_{j=1}^{N_\beta}\delta_{ij}\mu^t_{ij}\Bigg)\Bigg)\\ &+V\Bigg(\kappa_{\alpha_i}\left(\mu^t_i\right)^3 + \sum_{j=1}^{N_\beta}\delta_{ij}\kappa_{\beta_j}\Bigg(\sum_{i=1}^{N_\alpha}\delta_{ij}\mu^t_{ij}\Bigg)^2\mu^t_{ij}\Bigg)\Bigg)\end{aligned} \tag{10}$$

s.t. (9a), (9b),

$$0 \le \mu^t_i \le F^{\max}_{\alpha_i}, \quad \forall i, \tag{10a}$$

$$0 \le \sum_{i=1}^{N_\alpha}\delta_{ij}\mu^t_{ij} \le F^{\max}_{\beta_j}, \quad \forall j. \tag{10b}$$

Problem (10) can be considered a nonlinear optimization problem with constraints. It is observed that $\mathcal{F}(\mu)$ is convex since its second derivative $\mathcal{F}''(\mu) \ge 0$ for all $\mu \ge 0$ (see (9b)). Therefore, Karush-Khun-Tucker (KKT) conditions can be used to find the optimal solution of the optimization problem [35]. Accordingly, the Lagrange multiplier technique is exploited to find the optimal value of $\mu$. Here, the equivalent Lagrange function for the objective function is expressed as

$$\begin{aligned}\mathcal{L}(\mu,\varepsilon) = \mathcal{F}(\mu) &+ \sum_{i=1}^{N_\alpha}\varepsilon_i\Bigg(\mu^t_i + \sum_{j=1}^{N_\beta}\delta_{ij}\mu^t_{ij} - Q^t_i - \lambda^t_i\Bigg)\\ &+ \sum_{i=1}^{N_\alpha}\varepsilon_{N_\alpha+i}\left(\mu^t_i - F^{\max}_{\alpha_i}\right) + \sum_{j=1}^{N_\beta}\varepsilon_{2N_\alpha+j}\Bigg(\sum_{i=1}^{N_\alpha}\delta_{ij}\mu^t_{ij} - F^{\max}_{\beta_j}\Bigg)\\ &+ \sum_{i=1}^{N_\alpha} -\varepsilon_{2N_\alpha+N_\beta+i}\mu^t_i + \sum_{i=1}^{N_\alpha}\sum_{j=1}^{N_\beta} -\varepsilon_{3N_\alpha+N_\beta+(i-1)N_\beta+j}\mu^t_{ij},\end{aligned} \tag{11}$$

where $\varepsilon_i(\forall i = 1, 2, \ldots, 3N_\alpha + N_\beta + N_\alpha N_\beta)$ denotes KKT multipliers. The optimal solution of the equivalent problem satisfies the KKT conditions, derived by

$$\nabla_\mu \mathcal{L}(\widehat{\mu}, \varepsilon) = 0, \tag{12a}$$

$$\widehat{\mu}^t_i + \sum_{j=1}^{N_\beta}\delta_{ij}\widehat{\mu}^t_{ij} \le Q^t_i + \lambda^t_i, \quad \forall i, \tag{12b}$$

$$\widehat{\mu}^t_i \le F^{\max}_{\alpha_i}, \quad \forall i, \tag{12c}$$

$$\sum_{i=1}^{N_\alpha}\delta_{ij}\widehat{\mu}^t_{ij} \le F^{\max}_{\beta_j}, \quad \forall j, \tag{12d}$$

$$\widehat{\mu}^t_i, \widehat{\mu}^t_{ij} \ge 0, \quad \forall i, j, \tag{12e}$$

$$\varepsilon_i \ge 0, \quad \forall i, \tag{12f}$$

$$\sum_{i=1}^{N_\alpha}\varepsilon_i\Bigg(\sum_{j=1}^{N_\beta}\delta_{ij}\widehat{\mu}^t_{ij} - Q^t_i - \lambda^t_i\Bigg) = 0, \quad \forall i, \tag{12g}$$

$$\sum_{i=1}^{N_\alpha}\varepsilon_{N_\alpha+i}\left(\widehat{\mu}^t_i - F^{\max}_{\alpha_i}\right) = 0, \quad \forall i, \tag{12h}$$

$$\sum_{j=1}^{N_\beta}\varepsilon_{2N_\alpha+j}\Bigg(\sum_{i=1}^{N_\alpha}\delta_{ij}\widehat{\mu}^t_{ij} - F^{\max}_{\beta_j}\Bigg) = 0, \quad \forall j, \tag{12i}$$

$$\sum_{i=1}^{N_\alpha} -\varepsilon_{2N_\alpha+N_\beta+i}\widehat{\mu}^t_i = 0, \forall i, \tag{12j}$$

$$\sum_{i=1}^{N_\alpha}\sum_{j=1}^{N_\beta} -\varepsilon_{3N_\alpha+N_\beta+(i-1)N_\beta+j}\widehat{\mu}^t_{ij} = 0, \forall i, j. \tag{12k}$$

where $\nabla_\mu \mathcal{L}$ is the gradient of the $\mathcal{L}(\mu, \varepsilon)$ function with respect to $\mu$, and $\widehat{\mu}$ is the optimal value of $\mu$. The equivalent problem is now a constrained optimization problem.

By using the barrier method [36], the constrained problem is transformed to an unconstrained problem as follows

$$\begin{aligned}\widetilde{\mathcal{L}}(\mu,\varepsilon) = \mathcal{F}(\mu) &- \sum_{i=1}^{N_\alpha}\varepsilon_i \ln\Bigg(Q^t_i + \lambda^t_i - \mu^t_i - \sum_{j=1}^{N_\beta}\delta_{ij}\mu^t_{ij}\Bigg)\\ &- \sum_{i=1}^{N_\alpha}\varepsilon_{N_\alpha+i}\ln\left(F^{\max}_{\alpha_i} - \mu^t_i\right) - \sum_{j=1}^{N_\beta}\varepsilon_{2N_\alpha+j}\ln\Bigg(F^{\max}_{\beta_j} - \sum_{i=1}^{N_\alpha}\delta_{ij}\mu^t_{ij}\Bigg)\\ &- \sum_{i=1}^{N_\alpha}\varepsilon_{2N_\alpha+N_\beta+i}\ln\mu^t_i - \sum_{i=1}^{N_\alpha}\sum_{j=1}^{N_\beta}\varepsilon_{3N_\alpha+N_\beta+(i-1)N_\beta+j}\ln\mu^t_{ij},\end{aligned} \tag{13}$$

where $\widetilde{\mathcal{L}}(\mu, \varepsilon)$ is the equivalent unconstrained problem. We define

$$\mathcal{G}_n(\mu) = \begin{cases} Q_i^t + \lambda_i^t - \mu_i^t - \sum_{j=1}^{N_\beta} \delta_{ij}\mu_{ij}^t, \begin{cases} \forall i = n, \\ n = 1, 2, \dots, N_\alpha, \end{cases} \\ F_{\alpha_i}^{\max} - \mu_i^t, \begin{cases} \forall i = n - N_\alpha, \\ n = N_\alpha + 1, \dots, 2N_\alpha, \end{cases} \\ F_{\beta_j}^{\max} - \sum_{i=1}^{N_\alpha} \delta_{ij}\mu_{ij}^t, \begin{cases} \forall j = n - 2N_\alpha, \\ n = 2N_\alpha + 1, \dots, 2N_\alpha + N_\beta, \end{cases} \\ \mu_i^t, \begin{cases} \forall i = n - 2N_\alpha - N_\beta, \\ n = 2N_\alpha + \beta + 1, \dots, 3N_\alpha + N_\beta, \end{cases} \\ \mu_{ij}^t, \begin{cases} \forall i = (n - 3N_\alpha - N_\beta) \bmod N_\beta, \\ \forall j = n - 3N_\alpha - N_\beta - (i-1)N_\beta, \\ n = 3N_\alpha + N_\beta + 1, \dots, 3N_\alpha + N_\beta + N_\alpha N_\beta. \end{cases} \end{cases} \tag{14}$$

Accordingly, the above problem is represented as

$$\widetilde{\mathcal{L}}(\mu, \varepsilon) = \mathcal{F}(\mu) - \sum_{i=1}^{3N_\alpha + N_\beta + N_\alpha N_\beta} \varepsilon_i \ln \mathcal{G}_i(\mu). \tag{15}$$

As $\varepsilon$ converges to zero, the minimum of $\widetilde{\mathcal{L}}(\mu, \varepsilon)$ should converge to a solution of problem (10). Similarly, the solution of problem (15) can be derived by using KKT conditions

$$\nabla_\mu \widetilde{\mathcal{L}}(\mu, \varepsilon) = \mathcal{H}(\mu) - \sum_{i=1}^{3N_\alpha + N_\beta + N_\alpha N_\beta} \varepsilon_i \frac{\nabla_\mu \mathcal{G}_i(\mu)}{\mathcal{G}_i(\mu)} = 0, \tag{16}$$

where $\mathcal{H}(\mu)$ is the gradient of the original function $\mathcal{F}(\mu)$ and $\nabla_\mu \mathcal{G}_i(\mu)$ is the gradient of $\ln \mathcal{G}_i(\mu)$.

In addition to the original primal variable $\mu$, we define a Lagrange multiplier-inspired dual variable $\omega$ subjects to

$$\mathcal{G}_i(\mu)\omega_i = \varepsilon_i, \forall i = 1, 2, \dots, 3N_\alpha + N_\beta + N_\alpha N_\beta. \tag{17}$$

(17) is the complementary slackness in KKT conditions. Substituting (17) to problem (16), an equivalent problem is derived as

$$\mathcal{H}(\mu) - \mathrm{J}^T \omega = 0, \tag{18}$$

where the matrix J is the $\mathcal{G}_i(\mu)$ Jacobian. The gradient of $\mathcal{F}(\mu)$ should lie in the subspace spanned by the constraint gradients. The complementary slackness with a small $\omega$ can be realized as the condition where the solution should either lie near the boundary $\mathcal{G}_i(\mu) = 0$ or that the projection of the gradient $\mathcal{H}(\mu)$ on the constraint component $\mathcal{G}_i(\mu)$ normal should be almost 0.

By using Newton's method [37], we can obtain the near optimal solution of problem (16). The parameters $\mu$ and $\omega$ will converge to the optimal value after each iteration. These values are updated at the *k+1*-th step by

$$\begin{cases} \mu^{k+1} = \mu^k - \nabla\mu, \\ \omega^{k+1} = \omega^k - \nabla\omega, \end{cases} \tag{19}$$

where $\nabla\mu$ and $\nabla\omega$ are obtained by

$$\begin{pmatrix} \mathrm{W} & -\mathrm{J}^T \\ \mathrm{J}\Omega & \mathrm{G} \end{pmatrix} \begin{pmatrix} \nabla\mu \\ \nabla\omega \end{pmatrix} = \begin{pmatrix} -\mathcal{H} + \omega \mathrm{J}^T \\ \mu 1 - \omega \mathrm{G} \end{pmatrix}, \tag{20}$$

where W is the Hessian matrix of $\widetilde{\mathcal{L}}(\mu, \varepsilon)$, $\Omega$ is a diagonal matrix of $\omega$, and G is a diagonal matrix, where $\mathrm{G}_{ii}$ is $\mathcal{G}_i(\mu)$.

**Algorithm 1** E2W Scheduling Optimization.

```
Require: $\lambda, Q^t, V, \kappa_\alpha, \kappa_\beta$
Ensure: Optimal $\mu^*$
Initialization
$\mathcal{H}(\mu)$, J, W, G
$\mu^k$, $\omega^k$ is the $\mu$ and $\omega$ at k-th step, respectively, $\mu^0 = 0$, $\omega^0 = 0$, and $k = 0$
$\xi$ is the tolerance
repeat
    Given $\mu^k$ and $\omega^k$, calculate the $\mathcal{H}(\mu^k)$, J($\mu^k$), W($\mu^k, \omega^k$), and G($\mu^k$)
    Find the $\nabla\mu$ and $\nabla\omega$ by equation (20)
    Update $\mu^{k+1}$ and $\omega^{k+1}$ by equation (19)
    $k = k + 1$
until $||\widetilde{\mathcal{L}}(\mu^k, \omega^k)|| \leq \xi$
$\mu^* = \mu^k$ and $\omega^* = \omega^k$
```

Algorithm 1 summarizes the steps for determining the optimal value $\mu^*$. At each step $k$, $\mathcal{H}(\mu^k)$, J($\mu^k$), W($\mu^k, \omega^k$), and G($\mu^k$) are calculated with the given $\mu^k$ and $\omega^k$. Accordingly, $\nabla\mu$ and $\nabla\omega$ at step $k$ are determined by equation (20). Then, $\mu^{k+1}$ and $\omega^{k+1}$ for step $k + 1$ are obtained by (19). Parameters $\mu$ and $\omega$ will converge to optimal values after iterations. The iteration stops when $||\widetilde{\mathcal{L}}(\mu^k, \omega^k)|| \leq \xi$. Finally, the optimal value $\mu^* = \mu^k$ is derived.

### *B. Computational Complexity Analysis*

As described in Section IV-A, the problem $\mathcal{P}$ for finding the optimal processing workload $\mu^*$ is transformed to the equivalent problem (18) by using the barrier method and KKT conditions. The near optimal solution of problem (18) can be achieved by using an E2W scheduling optimization algorithm based on Newton's method. Because of quadratic convergence to the optimal value of Newton's method, the proposed algorithm can obtain the solution rapidly and effectively. The computational complexity of the proposed algorithm is $O(\xi^{-2})$ [38], where $\xi$ is the tolerance used for the stop condition of the iteration. It is observed that problem (10b) can also be solved by using the ellipsoid method or the cutting plane method [39]. The complexity of these approaches is $O(n^4)$, where $n$ is number of $\mu$ variables. Because $n$ is relatively large, these algorithms have a much higher complexity than the proposed algorithm.

## V. Performance Evaluation

### *A. Simulation Settings*

**System parameter setup**: To evaluate the performance of the MRC system, we developed a network model including ten and five RSUs in tier-$\alpha$ and tier-$\beta$, respectively. These RSUs are designed to operate on various CPU frequencies in ranges of $\{1.5, 2.0, 2.5, 3.0, 3.5\}$ and $\{12.0, 14.0, 16.0, 18.0, 20.0\}$ GHz [40]. Without loss of generality, we assumed different ranges of CPU frequencies for two tiers in our model. It is worth noting that these assumptions are selected randomly and they are used consistently among simulated algorithms. Because the RSUs in tier-$\beta$ have been assumed to be equipped with a higher computational capacity owing to flexible virtualization, the coefficient factor of tier-$\beta$ RSUs is smaller than that of tier-$\alpha$ RSUs. In particular,

TABLE II: Simulation parameters.

| Parameter | Value |
|---|---|
| Number of RSUs in tier-$\alpha$ | 10 |
| Number of RSUs in tier-$\beta$ | 5 |
| $f_\alpha$ | $\{1.5, 2.0, 2.5, 3.0, 3.5\}$ GHz |
| $f_\beta$ | $\{12.0, 14.0, 16.0, 18.0, 20.0\}$ GHz |
| $\kappa_\alpha$ | 5.0E-09 |
| $\kappa_\beta$ | 4.0E-09 |
| Average arrival rate ($\bar{\lambda}$) | $\{50–300\}$ Mbps |
| Workload complexity | $\{100, 200, 500, 1000\}$ cycle/bit |
| Simulation time | 500 timeslots |

$\kappa_\alpha$ and $\kappa_\beta$ are set as $5.0E-09$ and $4.0E-09$, respectively. The offloaded workload of the vehicles is deployed as follows. During each timeslot, an offloading traffic from the vehicles arrives at tier-$\alpha$ RSUs with average arrival rate ($\bar{\lambda}$) varying from 50 to 300 Mbps in each simulation. Each simulation lasts 500 timeslots. A timeslot is 100 ms. In this context, to represent various services such as navigation, in-vehicle infotainment applications, vehicle social services, and virtual reality gaming, a complexity set of $\{100, 200, 300, 1000\}$ cycle/bit, which can be obtained through practical experiences as done in [26], [41], is used to map to the offloaded workload. Details of the simulation parameters are provided in Table II.

**Competitor description**: To demonstrate the advantages of the proposed E2W scheme, three typical schemes are additionally simulated to draw a comparison, including *self-calibrating (SC)*, *zero-buffering (ZB)*, and *energy-aware (EA)* schemes [16], [21].

- *Self-calibrating (SC) scheme:* The SC scheme aims at self-balancing between computational energy consumption and local buffer in each RSU separately. Each RSU maintains its operation based on the arrived offloaded data without considering the external entities' status.
- *Zero-buffering (ZB) scheme:* The ZB scheme prioritizes response latency to the vehicles by mitigating waiting time in the buffer. To this end, tier-$\alpha$ RSUs handle the arrived data with their maximal CPU frequencies. Then, the remaining offloaded data is delivered to tier-$\beta$ RSUs to release the buffer.
- *Energy-aware (EA) scheme:* The EA scheme focuses on minimizing energy consumption of the offloading services. Therefore, the offloaded data is directly streamed to tier-$\beta$ RSUs to use their maximum computational capacities because they have a high energy efficiency. The remaining data are stored in tier-$\alpha$ RSUs for processing and possibly buffering.

**Evaluation methodology**: As our focuses are to resolve the system stability and energy efficiency in MRC platforms. Hence, performance evaluation paid much attention on system aspects. Accordingly, the simulation was performed in two parts.

- First, the evaluation investigates appropriate control factor selection and reactive behavior of the proposed E2W scheme against different scenarios with varying offloaded arrival rate.
- Second, the proposed E2W scheme is compared to the competitors to demonstrate its superior performance metrics in terms of energy efficiency and system stability.

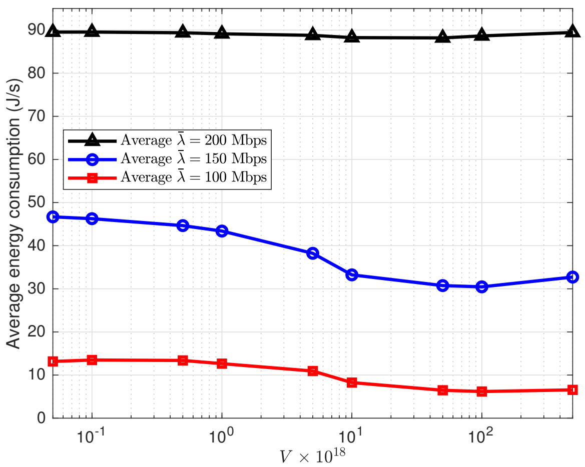


Fig. 2: Average energy consumption within various arrival rates.

**Simulation platform**: *MATLAB R2018a*.
**Numerical analysis tool**: *IBM SPSS Statistics 20*.

### *B. Control Factor Selection*

As aforementioned in Section IV-A, the control factor $V$ plays a key role in balancing energy consumption and system stability (measured by the buffer size fluctuation). Equation (8) shows that an increase in $V$ minimizes the energy consumption (i.e., energy efficiency prioritization) but increases the buffer size fluctuation, and vice versa. Figs. 2 and 3 visualize the effectiveness of $V$ on these mentioned metrics under three typical traffic patterns: 100-Mbps, 150-Mbps, and 200-Mbps average arrival rates ($\bar{\lambda}$), which require approximate 50%, 75%, and 100% of computational capacity of the MRC system, respectively. When $\bar{\lambda} = 200$ Mbps, a change in $V$ has an insignificant impact on the energy consumption because the system mostly operates at the maximum CPU frequency to process the offloaded traffic as shown in Fig. 2. This circumstance leads to an uncontrollable state of the buffer. Therefore, the buffer size fluctuates following the chaotic arrival rates as shown in Fig. 3.

The impacts of $V$ on the energy consumption and buffer size can be clearly seen at a lower average arrival rate $\bar{\alpha}$. Fig. 2 shows that energy consumption decreases as $V$ increases. The energy consumption significantly decreases when $V$ increases from $10^0 \times 10^{18}$ to $10^1 \times 10^{18}$. During this scale of $V$, the buffer size as well as its fluctuation increase proportionally. With the observed results, given a maximum buffer size of 5 Mb, $10^0 \times 10^{18}$ is considered to be an optimal selection for $V$ as the buffer size fluctuation is controlled under the 5-Mb threshold. Similarly, $V$ should be $10^1 \times 10^{18}$ if the given maximum buffer size is set to 10 Mb. Details on statistical indexes of buffer sizes in RSUs are provided in Table III. The numerical results reveal that the variation metric of buffer size, which represents the system stability, increases when $V$ and/or average arrival rate increase. Because there is no universally optimal $V$ for

TABLE III: Statistical indexes of buffer size in RSUs.

| **V ($\times$10$^{18}$)** | **Buffer size in RSUs (Mb)** | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | **$\bar{\lambda}$ = 200Mbps** | | | | **$\bar{\lambda}$ = 150Mbps** | | | | **$\bar{\lambda}$ = 100Mbps** | | | |
| | **Mean** | **Min** | **Max** | **SD** | **Mean** | **Min** | **Max** | **SD** | **Mean** | **Min** | **Max** | **SD** |
| 0.05 | 1.1071 | 2.86E-10 | 4.5775 | 0.7523 | 0.0726 | 1.53E-09 | 1.3665 | 0.0721 | 0.0727 | 1.73E-09 | 0.3776 | 0.0686 |
| 0.1 | 1.122 | 2.62E-10 | 4.4507 | 0.7587 | 0.0696 | 1.62E-09 | 0.4993 | 0.0632 | 0.0597 | 1.89E-09 | 0.3776 | 0.0548 |
| 0.5 | 1.119 | 7.12E-11 | 4.7383 | 0.7576 | 0.0727 | 1.29E-09 | 1.0421 | 0.0742 | 0.0603 | 1.83E-09 | 0.3736 | 0.0557 |
| 1 | 1.1094 | 2.32E-10 | 4.5933 | 0.7554 | 0.1184 | 1.19E-09 | 1.4581 | 0.1204 | 0.0567 | 2.22E-09 | 0.3543 | 0.0529 |
| 5 | 1.1356 | 2.66E-10 | 4.64 | 0.7652 | 0.373 | 1.45E-09 | 2.0409 | 0.3198 | 0.0433 | 4.27E-09 | 0.4966 | 0.0490 |
| 10 | 1.1653 | 2.29E-10 | 4.6562 | 0.7823 | 0.5433 | 1.27E-09 | 2.0765 | 0.4399 | 0.0716 | 3.54E-09 | 0.9746 | 0.0900 |
| 50 | 1.2319 | 2.47E-10 | 5.3397 | 0.8131 | 0.6144 | 3.90E-10 | 2.123 | 0.4846 | 0.1601 | 2.36E-09 | 1.2563 | 0.1825 |
| 100 | 1.356 | 2.15E-10 | 5.4949 | 0.8585 | 0.6355 | 1.20E-09 | 2.293 | 0.4957 | 0.2472 | 1.84E-09 | 1.3944 | 0.2540 |
| 500 | 1.6252 | 2.16E-10 | 5.6355 | 0.9616 | 0.6723 | 1.19E-09 | 2.3889 | 0.5222 | 0.3485 | 1.40E-09 | 1.6683 | 0.3266 |

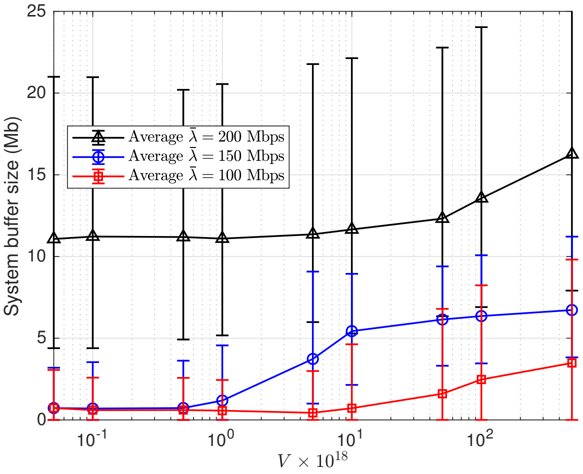


Fig. 3: System buffer size fluctuation with various arrival rates.

all scenarios [14], [42], [43], hereafter we assume a 10-Mb buffer size and select a $V$ of $10^1 \times 10^{18}$ for further simulation and comparison.

### C. Workload Distribution

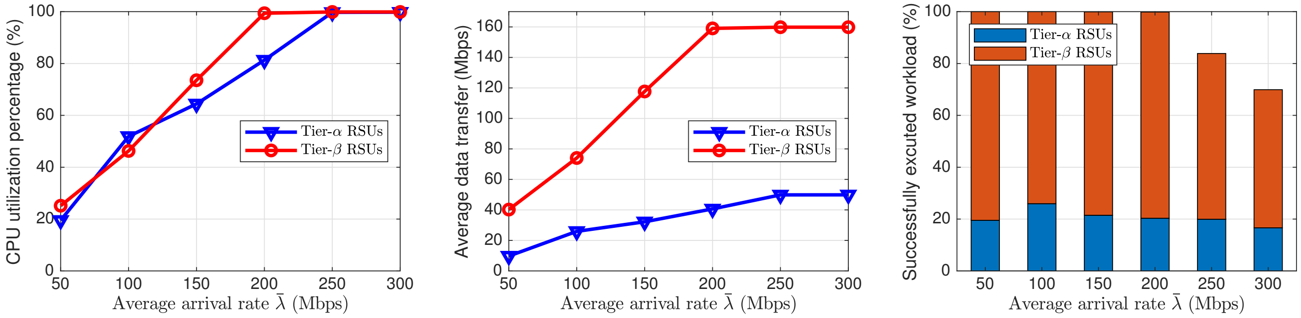
This section investigates workload distribution between tier-$\alpha$ and tier-$\beta$ in the MRC system within an arrival rate range of 50 to 300 Mbps. Fig. 4 depicts the results in terms of CPU utilization (Fig. 4a), offloaded data transfer (Fig. 4b), and successfully executed workload (Fig. 4c).

Fig. 4a illustrates the dynamic adjustment of CPU utilization between tier-$\alpha$ and tier-$\beta$ based on arrival rate observations. Because tier-$\beta$ RSUs have a higher energy efficiency ($\kappa_\beta < \kappa_\alpha$), their CPUs are highly utilized. However, as energy consumption is a cube function of CPU frequencies as shown in Equation (10), a harmonization of both tiers is required to achieve optimal results. Because of the stochasticity of arrival rate $\lambda$, CPU utilization has fluctuated accordingly. Numerical results expose that the standard deviation of CPU utilization percentage is high (approximate 36.77%) at a low arrival rate and low (approximate 11.93%) at a high arrival rate. The reason behind this phenomenon is when the arrival rate was low, available CPU capacity is high and hence flexible to be controlled. Whilst the arrival rate was reaching saturation condition, i.e., most of the CPU capacity is occupied, the flexibility of the algorithm is limited owing to a small room of available CPU capacity. It is worth noting that in all cases, system buffers are maintained not exceeding the maximum threshold of 10 Mb. Obviously, the CPU utilization percentage of all RSUs increases following the increase in arrival rate. In particular, tier-$\beta$ RSUs reach approximate 100% CPU utilization when the average arrival rate is 200 Mbps and tier-$\alpha$ RSUs meet the same condition with an average arrival rate of 250 Mbps or more. Similar to the CPU utilization behavior, Fig. 4b shows the amount of data transferred to each tier for offloading execution. The offloaded data is distributed between the two tiers based on the computational capability, and the data proportionally increase depending on the arrival rate.

Fig. 4c depicts the successfully executed workload in the MRC system. It is observed that tier-$\alpha$ RSUs dynamically contribute 20–25% of successful workload execution in the entire network according to the arrival rate. When the average arrival rate is under 200 Mbps, 100% workload is successfully executed. The network overloads as the average arrival rate increases to 250 and 300 Mbps. In these saturated traffic environments, the CPU of both tiers mostly operate within 100% capability. The overloaded amount of the 250-Mbps and 300-Mbps traffic results in losses of 16.13% and 30.10%, respectively.

### D. Performance Comparison

Fig. 5 shows a comparison of the competing schemes in terms of average energy consumption in the entire system. It is observed that the energy consumption of all schemes increases proportionally with the arrival rate. Among these schemes, the proposed E2W scheme exhibits a significant improvement in energy efficiency. In particular, the proposed E2W scheme reduces approximately 55.04%, 65.64%, and 61.40% of energy as compared to the results of the SC, ZB, and EA schemes, respectively, in a stable environment wherein the arrival rate is lower than the maximum capacity of the MRC system. This is because the E2W scheme has a sufficient buffer to dynamically adjust the optimal amount of workload to be processed and temporally stored in the buffer. Similar to the fluctuation of CPU utilization percentage, energy consumption is proportional to the amount of CPU usage, which subsequently depends on the stochasticity of arrival rate $\lambda$ (see Eq. (2)). Numerical results show the standard deviation of the system energy consumption is approximate 13.34, 18.08, 16.96, and 16.55 J/s in the E2W, SC, ZB, and EA schemes, respectively. Although the E2W scheme flexibly adjusts the CPU utilization



(a) CPU utilization in RSUs.

(b) Offloaded data transferred to RSUs.

(c) Successfully executed workload in RSUs.

Fig. 4: Reactive behavior of computational tiers in the MRC system.

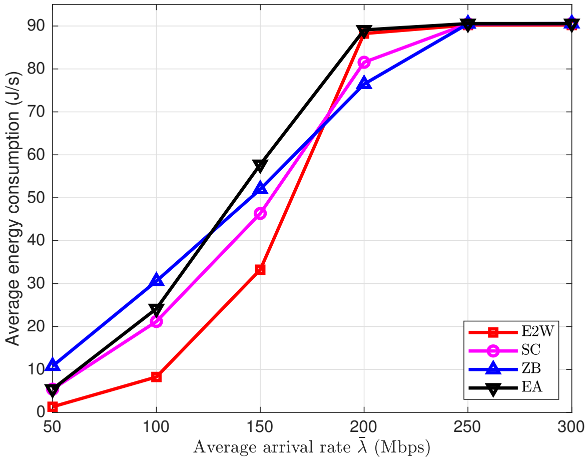


Fig. 5: Average energy consumption.

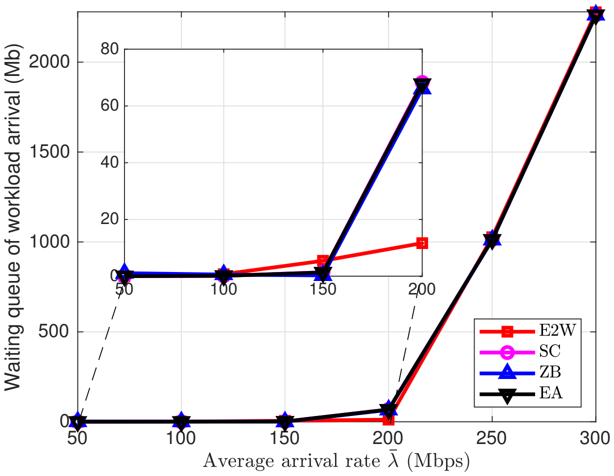


Fig. 6: Waiting queue of workload arrival.

at both tiers of the system resulting in a high fluctuation of the CPU usage, the minimization of energy consumption leads to a small standard deviation compared to those of the others. In case the arrival rate increases and reaches the limit of the system (e.g., $\bar{\lambda} = 200$ Mbps), all RSUs operate within their maximum CPU frequencies. Owing to the dynamic workload distribution between the two tiers, the E2W scheme can utilize its entire system capacity while the others cannot. Therefore, the average energy consumption generated by the E2W scheme is higher than the SC and ZB schemes. Note that the SC and ZB schemes consume lower energy than the E2W scheme because a significant amount of workload is not processed yet and stored in their buffer as shown in Fig. 6 and the following analysis of the waiting queue in the system.

Fig. 6 illustrates the average waiting queue of workload arrival at the MRC system. It is observed that the system maintains its stability when the arrival rate is under approximate 75% of the maximum system capability (i.e., $\bar{\lambda} = 150$ Mbps). At this threshold, the statistical metric standard deviation of simulation results exhibits that all schemes controlled the system buffer to be fluctuated not exceeding its maximum capacity of 10 Mb. Under these circumstances, the proposed E2W scheme leads to a higher buffer along with a high standard deviation as compared to those of other schemes. However, this phenomenon is not disadvantage; it proves the dynamic harmonization between energy consumption and the system buffer of the E2W scheme, which optimally minimizes the energy while keeping the buffer under the maximum size. It shows the advantage of the Lyapunov optimization as applied in the objective function $\mathcal{P}$. The dynamic harmonization is clearly depicted when the average arrival rate $\bar{\lambda} = 200$ Mbps. As described in the analysis of Fig. 5, when $\bar{\lambda} = 200$ Mbps, the E2W scheme consumes more energy than the SC and ZB scheme and approximately the same energy as the EA scheme. On the other hand, Fig. 6 shows the reason for this, i.e., the E2W scheme can significantly reduce the system buffer size to ensure system stability while the others cannot. It is clear that the system is in overload under the saturated conditions when $\bar{\lambda} > 200$ Mbps (e.g., $\bar{\lambda} = 250|300$ Mbps).

## VI. Conclusion

In this study, an energy-efficient workload scheduling scheme, namely E2W, is proposed for vehicle communications in MRC systems. The E2W scheme provides a dynamic balance between energy consumption minimization and system stability by using Lyapunov optimization. Moreover, the advantages of the proposed E2W scheme can be utilized in various heterogeneous networking scenarios such as smart manufacturing and smart cities. Compared to existing studies, the peculiarities of our work can be highlighted as: (*i*) Different from existing studies of generic edge computing, the MRC platform representing a multitier edge computing and it is modeled as a multitier queuing system from a computing perspective, (*ii*) Whilst existing studies mainly resolved the problems incorporated with latency, our paper targets the system stability in a scalable network. To extend the application of the proposed scheme on large-scale systems, a collaborated and cluster-based approach based on deep learning methods will be considered in future research.

## References

[1] B. Ji, X. Zhang, S. Mumtaz, C. Han, C. Li, H. Wen, and D. Wang, "Survey on the Internet of vehicles: Network architectures and applications," *IEEE Communications Standards Magazine*, vol. 4, no. 1, pp. 34–41, 2020.

[2] Z. Ning, K. Zhang, X. Wang, L. Guo, X. Hu, J. Huang, B. Hu, and R. Y. Kwok, "Intelligent edge computing in Internet of vehicles: a joint computation offloading and caching solution," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 4, pp. 2212–2225, 2020.

[3] S. Wan, R. Gu, T. Umer, K. Salah, and X. Xu, "Toward offloading Internet of vehicles applications in 5G networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 7, pp. 4151–4159, 2021.

[4] W. Na, N.-N. Dao, and S. Cho, "Mitigating WiFi interference to improve throughput for in-vehicle infotainment networks," *IEEE Wireless Communications*, vol. 23, no. 1, pp. 22–28, 2016.

[5] H. Kim, J. Ben-Othman, S. Cho, and L. Mokdad, "A framework for IoT-enabled virtual emotion detection in advanced smart cities," *IEEE Network*, vol. 33, no. 5, pp. 142–148, 2019.

[6] D.-N. Vu, N.-N. Dao, W. Na, and S. Cho, "Dynamic resource orchestration for service capability maximization in fog-enabled connected vehicle networks," *IEEE Transactions on Cloud Computing*, 2020.

[7] Y. Lee, S. Jeong, A. Masood, L. Park, N.-N. Dao, and S. Cho, "Trustful resource management for service allocation in fog-enabled intelligent transportation systems," *IEEE Access*, vol. 8, pp. 147 313–147 322, 2020.

[8] ETSI, "Multi-access edge computing," Available: https://www.etsi.org/technologies/multi-access-edge-computing, Accessed on August 27, 2021.

[9] H. Xiang, W. Zhou, M. Daneshmand, and M. Peng, "Network slicing in fog radio access networks: Issues and challenges," *IEEE Communications Magazine*, vol. 55, no. 12, pp. 110–116, 2017.

[10] H. Zhang, Y. Qiu, K. Long, G. K. Karagiannidis, X. Wang, and A. Nallanathan, "Resource allocation in NOMA-based fog radio access networks," *IEEE Wireless Communications*, vol. 25, no. 3, pp. 110–115, 2018.

[11] H. Xiang, M. Peng, Y. Sun, and S. Yan, "Mode selection and resource allocation in sliced fog radio access networks: A reinforcement learning approach," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 4, pp. 4271–4284, 2020.

[12] L. Liu, C. Chen, Q. Pei, S. Maharjan, and Y. Zhang, "Vehicular edge computing and networking: A survey," *Mobile Networks and Applications*, vol. 26, no. 3, pp. 1145–1168, 2021.

[13] A. Boukerche and V. Soto, "Computation offloading and retrieval for vehicular edge computing: algorithms, models, and classification," *ACM Computing Surveys (CSUR)*, vol. 53, no. 4, pp. 1–35, 2020.

[14] N.-N. Dao, D.-N. Vu, W. Na, J. Kim, and S. Cho, "SGCO: Stabilized green crosshaul orchestration for dense IoT offloading services," *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 11, pp. 2538–2548, 2018.

[15] D.-N. Vu, N.-N. Dao, Y. Jang, W. Na, Y.-B. Kwon, H. Kang, J. J. Jung, and S. Cho, "Joint energy and latency optimization for upstream IoT offloading services in fog radio access networks," *Transactions on Emerging Telecommunications Technologies*, p. e3497, 2018.

[16] L. Cui, C. Xu, S. Yang, J. Z. Huang, J. Li, X. Wang, Z. Ming, and N. Lu, "Joint optimization of energy consumption and latency in mobile edge computing for Internet of things," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4791–4803, 2018.

[17] M. Neely, *Stochastic Network Optimization with Application to Communication and Queueing Systems*. Morgan & Claypool Publishers, 2010.

[18] O. Muñoz, A. P. Iserte, J. Vidal, and M. Molina, "Energy-latency trade-off for multiuser wireless computation offloading," in *Proc. of IEEE Wireless Communications and Networking Conference Workshops (WCNCW)*, 2014, pp. 29–33.

[19] L. Pu, X. Chen, J. Xu, and X. Fu, "D2D fogging: An energy-efficient and incentive-aware task offloading framework via network-assisted D2D collaboration," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 12, pp. 3887–3901, 2016.

[20] J. Dolezal, Z. Becvar, and T. Zeman, "Performance evaluation of computation offloading from mobile device to the edge of mobile network," in *Proc. of IEEE Conference on Standards for Communications and Networking (CSCN)*, 2016, pp. 1–7.

[21] J. Liu, Y. Mao, J. Zhang, and K. B. Letaief, "Delay-optimal computation task scheduling for mobile-edge computing systems," in *Proc. of IEEE International Symposium on Information Theory (ISIT)*, 2016, pp. 1451–1455.

[22] W. Labidi, M. Sarkiss, and M. Kamoun, "Joint multi-user resource scheduling and computation offloading in small cell networks," in *Proc. of IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, 2015, pp. 794–801.

[23] S. Jošilo and G. Dán, "Decentralized algorithm for randomized task allocation in fog computing systems," *IEEE/ACM Transactions on Networking*, vol. 27, no. 1, pp. 85–97, 2019.

[24] L. Yang, H. Zhang, M. Li, J. Guo, and H. Ji, "Mobile edge computing empowered energy efficient task offloading in 5G," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 7, pp. 6398–6409, 2018.

[25] C. Wang, C. Liang, F. R. Yu, Q. Chen, and L. Tang, "Computation offloading and resource allocation in wireless cellular networks with mobile edge computing," *IEEE Transactions on Wireless Communications*, vol. 16, no. 8, pp. 4924–4938, 2017.

[26] X. Chen, L. Jiao, W. Li, and X. Fu, "Efficient multi-user computation offloading for mobile-edge cloud computing," *IEEE/ACM Transactions on Networking*, no. 5, pp. 2795–2808, 2016.

[27] K. Zhang, Y. Mao, S. Leng, Q. Zhao, L. Li, X. Peng, L. Pan, S. Maharjan, and Y. Zhang, "Energy-efficient offloading for mobile edge computing in 5G heterogeneous networks," *IEEE Access*, vol. 4, pp. 5896–5907, 2016.

[28] H. Yu, Q. Wang, and S. Guo, "Energy-efficient task offloading and resource scheduling for mobile edge computing," in *Proc. of IEEE International Conference on Networking, Architecture and Storage (NAS)*, 2018, pp. 1–4.

[29] R. Deng, R. Lu, C. Lai, T. H. Luan, and H. Liang, "Optimal workload allocation in fog-cloud computing toward balanced delay and power consumption," *IEEE Internet of Things Journal*, vol. 3, no. 6, pp. 1171–1181, 2016.

[30] V. G. Kulkarni, *Modeling and analysis of stochastic systems*. CRC Press, 2016.

[31] J. Ren, D. Zhang, S. He, Y. Zhang, and T. Li, "A survey on end-edge-cloud orchestrated network computing paradigms: Transparent computing, mobile edge computing, fog computing, and cloudlet," *ACM Computing Surveys (CSUR)*, vol. 52, no. 6, pp. 1–36, 2019.

[32] Q. Shi, L. Zhao, Y. Zhang, G. Zheng, F. R. Yu, and H.-H. Chen, "Energy-efficiency versus delay tradeoff in wireless networks virtualization," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 1, pp. 837–841, 2018.

[33] C. Qiu, Y. Hu, and Y. Chen, "Lyapunov optimized cooperative communications with stochastic energy harvesting relay," *IEEE Internet of Things Journal*, vol. 5, no. 2, pp. 1323–1333, 2018.

[34] J. M. Sanz Serna and K. C. Zygalakis, "The connections between Lyapunov functions for some optimization algorithms and differential equations," *SIAM Journal on Numerical Analysis*, vol. 59, no. 3, pp. 1542–1565, 2021.

[35] Z.-Q. Luo and W. Yu, "An introduction to convex optimization for communications and signal processing," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 8, pp. 1426–1438, 2006.

[36] A. Forsgren, P. E. Gill, and M. H. Wright, "Interior methods for nonlinear optimization," *SIAM Review*, vol. 44, no. 4, pp. 525–597, 2002.

[37] C. T. Kelley, *Solving nonlinear equations with Newton's method*. SIAM, 2003.

[38] C. Cartis, N. I. Gould, and P. L. Toint, "On the complexity of steepest descent, Newton's and regularized Newton's methods for nonconvex unconstrained optimization problems," *SIAM Journal on Optimization*, vol. 20, no. 6, pp. 2833–2852, 2010.

[39] S. Bubeck *et al.*, "Convex optimization: Algorithms and complexity," *Foundations and Trends in Machine Learning*, vol. 8, no. 3-4, pp. 231–357, 2015.

[40] Y. Lin, Y. Zhang, J. Li, F. Shu, and C. Li, "Popularity-aware online task offloading for heterogeneous vehicular edge computing using contextual clustering of bandits," *IEEE Internet of Things Journal*, 2021.

[41] A. P. Miettinen and J. K. Nurminen, "Energy efficiency of mobile clients in cloud computing." in *Proc. USENIX Conf. Hot Topics Cloud Comput. (HotCloud)*, 2010, pp. 1–7.

[42] Y. Mao, J. Zhang, and K. B. Letaief, "A Lyapunov optimization approach for green cellular networks with hybrid energy supplies," *IEEE Journal on Selected Areas in Communications*, vol. 33, no. 12, pp. 2463–2477, 2015.

[43] J. Kim, G. Caire, and A. F. Molisch, "Quality-aware streaming and scheduling for device-to-device video delivery," *IEEE/ACM Transactions on Networking*, vol. 24, no. 4, pp. 2319–2331, 2016.

**Nhu-Ngoc Dao** (Senior Member, IEEE) is an Assistant Professor at the Department of Computer Science and Engineering, Sejong University, Seoul, Republic of Korea. He received his M.S. and Ph.D. degrees in computer science at the School of Computer Science and Engineering, Chung-Ang University, Seoul, Republic of Korea, in 2016 and 2019, respectively. He received the B.S. degree in electronics and telecommunications from the Posts and Telecommunications Institute of Technology, Hanoi, Viet Nam, in 2009. Prior to joining the Sejong University, he was a visiting researcher at the University of Newcastle, NSW, Australia, in 2019 and a postdoc researcher at the Institute of Computer Science, University of Bern, Switzerland, from 2019 to 2020. He currently serves as an Editor of the *Scientific Reports*. His research interests include network softwarization, mobile cloudization, intelligent systems, and the Intelligence of Things.

**Duc-Nghia Vu** is a senior engineer at Innowireless Inc., Seongnam, South Korea. He received his B.S. degree in Electronics and telecommunications from Hanoi University of Science and Technology, Viet Nam, in 2015. He also received the M.S. degree in computer science from Chung-Ang University, South Korea, in 2018. He is a PhD candidate in computer science from Chung-Ang University, South Korea from 2021. His research interests include wireless network and fog computing.

**Anh-Tien Tran** received his BS degree in electronics and telecommunications from the Danang University of Science and Technology, Da Nang, Vietnam, in 2018. Since 2018, he has been working toward hid PhD degree in computer science from Chung-Ang University, South Korea. His research interests include wireless network communication, video streaming, fog computing, and machine learning.

**Trung V. Phan** (Member, IEEE) received the B.S degree in electronics and telecommunications from the Hanoi University of Science and Technology (HUST), Vietnam, in 2015, and the M.S degree in information communication technology from Soongsil University, Seoul, South Korea, in 2017. He is currently a Research Staff with the Faculty of Electrical Engineering and Information Technology, Technische Universitt Chemnitz, Germany. His research interests include deep reinforcement learning, federated learning, software defined networks, network functions virtualization, the Internet of Things, and network security.

**Schahram Dustdar** (Fellow, IEEE) is currently a Professor of computer science with the Distributed Systems Group, TU Wien, Vienna, Austria. From 2004 to 2010, he was an Honorary Professor of information systems with the University of Groningen, Groningen, The Netherlands, from 2016 to 2017, he was a Visiting Professor with the University of Sevilla, Seville, Spain, and in 2017, he was a Visiting Professor with the University of California at Berkeley, Berkeley, CA, USA. He is an elected member of the Academia Europaea, where he is Chairman of the Informatics Section. He was recipient of the ACM Distinguished Scientist Award (2009), the IBM Faculty Award (2012), and the IEEE TCSVC Outstanding Leadership Award (2018). He is the Co-Editor-in-Chief of the ACM Transaction on Internet of Things and the Editor-in-Chief of Computing (Springer). He is also an Associate Editor of the IEEE Transaction on Services Computing, the IEEE Transaction on Cloud Computing, the ACM Transaction on the Web, and the ACM Transaction on Internet Technology. He serves on the Editorial Board of IEEE Internet Computing and the IEEE Computer Magazine.

**Sungrae Cho** is a professor with the school of computer sciences and engineering, Chung-Ang University (CAU), Seoul. Prior to joining CAU, he was an assistant professor with the department of computer sciences, Georgia Southern University, Statesboro, GA, USA, from 2003 to 2006, and a senior member of technical staff with the Samsung Advanced Institute of Technology (SAIT), Kiheung, South Korea, in 2003. From 1994 to 1996, he was a research staff member with electronics and telecommunications research institute (ETRI), Daejeon, South Korea. From 2012 to 2013, he held a visiting professorship with the national institute of standards and technology (NIST), Gaithersburg, MD, USA. He received the B.S. and M.S. degrees in electronics engineering from Korea University, Seoul, South Korea, in 1992 and 1994, respectively, and the Ph.D. degree in electrical and computer engineering from the Georgia Institute of Technology, Atlanta, GA, USA, in 2002.

His current research interests include wireless networking, ubiquitous computing, and ICT convergence. He has been a subject editor of IET Electronics Letter since 2018, and was an area editor of Ad Hoc Networks Journal (Elsevier) from 2012 to 2017. He has served numerous international conferences as an organizing committee chair, such as IEEE SECON, ICOIN, ICTC, ICUFN, TridentCom, and the IEEE MASS, and as a program committee member, such as IEEE ICC, MobiApps, SENSORNETS, and WINSYS.