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/TSUSC.2019.2907110, IEEE Transactions on Sustainable Computing

EDCSuS: Sustainable Edge Data Centers as a Service in SDN-enabled Vehicular Environment

Gagangeet Singh Aujla*, *Member, IEEE* and Neeraj Kumar†, *Senior Member, IEEE*,
Sahil Garg‡, *Member, IEEE*, Kuljeet Kaur§, *Member, IEEE*, Rajiv Ranjan¶, *Senior Member, IEEE*
*Computer Science and Engineering Department, Chandigarh University, Mohali (Punjab), India
†Computer Science and Engineering Department, Thapar Institute of Engineering and Technology (Demmed to be University), Patiala (Punjab), India
‡§ École de Technologie Supérieure, Université du Québec, Montréal, Canada
¶School of Computer, Chinese University of Geosciences, Wuhan, School of Computing, Newcastle University, United Kingdom (UK), and Data61, Australia
(Email: gagi_aujla82@yahoo.com, neeraj.kumar@thapar.edu, garg.sahil1990@gmail.com, kuljeet0389@gmail.com, raj.ranjan@ncl.ac.uk)

Abstract—Cloud computing has emerged as a popular technologies which provide on-demand services to the end users. Such services are hosted by massive geo-distributed data centers (DCs). Nowadays, connected vehicles in a smart city can also avail cloud services through Internet using cellular technologies. But, the advent of 5G technology has posed challenges for DCs such as-low latency and higher data rate requirements. To handle these challenges, edge-DCs (EDCs) can be deployed across a smart city to provide low latency services to the connected vehicles. In lieu of this, in this paper, EDCSuS: Sustainable EDC as a service framework in software defined vehicular environment is proposed. In EDCSuS, firstly, a software defined controller handles the incoming requests and suggest an optimal flow path. Secondly, a multi-leader multi-follower Stackelberg game is presented for resource allocation. Thirdly, to improve the resource utilization, a cooperative resource sharing scheme is designed, thereby minimizing the energy consumption of servers in the EDCs. Lastly, a caching scheme is presented to avert excessive energy consumption for retracing the lost link due to vehicular mobility. The efficacy of the proposed scheme has been evaluated using extensive simulations with respect to various parameters. The results obtained prove the effectiveness of EDCSuS.

Index Terms—Cloud computing, energy management, edge data centers, renewable energy, software-defined networks, Sustainability.

◆

1 INTRODUCTION

Cloud Computing (CC) has emerged as a one of the most powerful technologies which provides location-independent, on-demand, and ubiquitous services to the end users. Such services are hosted by massive geo-distributed data centers (DCs) which realize the provisioning of the resources to the end users. Nowadays, the connected vehicles in a smart city can also avail the cloud services using Internet. Vehicles are equipped with communication, computing, and sensing devices to connect to Internet on the move using high-speed cellular networks. This convergence of mobile Internet with vehicles has laid the foundation for innovation of cutting-edge technologies such as-*mobile cloud computing* and *vehicular clouds*. For instance, mobile cloud computing has emerged as a recent platform to provide the services to mobile devices using Internet on-demand [1]. But, this platform has to face various challenges (such as-limited computing resources, limited battery, high processing time, and high cost) with respect to service provisioning to the vehicles. However, the advent of 5G is expected to overcome some of these challenges such as-higher battery consumption, complication of hardware, implementation challenges, and higher cost of equipment. Moreover, 5G is expected to provide a unique network to broadcast large amount of data in gigabits per second. 5G networks are also expected to provide ubiquitous connectivity and high-rate services to a large number of devices including human and machine type communications [2]. Therefore, smart vehicles are converging with 5G networks to connect with each other, with human end users and cloud services. This convergence is bound to reduce computational time significantly while providing longer lasting network mobility.

But, this has a strong impact on the energy consumption of DC infrastructure. The requirement of extremely higher data rates and lower latency elevates the use of energy consuming technologies [3]. Therefore, to overcome this challenge, *edge computing* can be an effective solution, wherein the location of the vehicles plays a significant role in selecting the edge-DC (EDC) or edge node to provision the required resources. In this scheme, the vehicular users are able to access cloud services from the EDCs located closer to their position. This helps to achieve lower latency and higher data rates [4]. However, due to the vehicular mobility, the link between the EDC and vehicle is often lost. In such a case, another EDC connects directly to the vehicle which requires to retrace the location of the previously lost EDC server. This leads to additional energy consumption for re-searching and re-routing the lost link [3].

After analysis of the above discussion, it is quite evident that energy consumption is one of the most fluctuating aspect that needs significant attention. Hence, to handle this aspect, in this paper, EDCSuS: Sustainable EDC as a service framework in software defined vehicular environment is proposed as a competent solution. In this framework, EDCs are geographically distributed in a smart city instead of a centralized DC to handle vehicular application closer to their location [4]–[6]. An exemplar layout of the proposed solution is shown in Fig. 1.

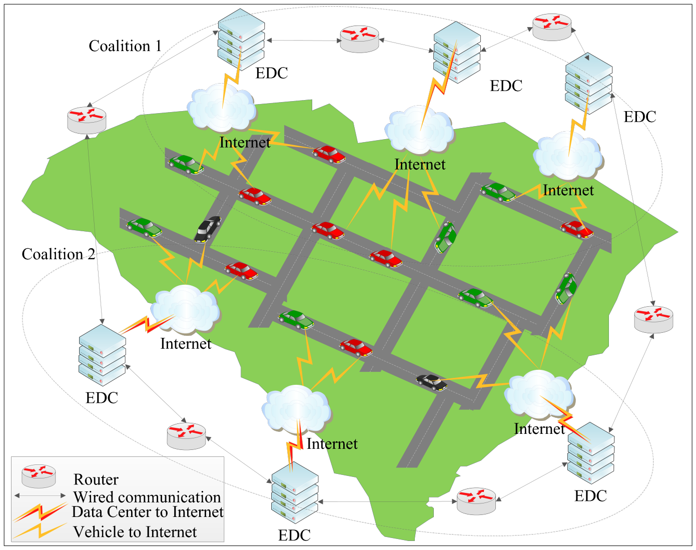


Fig. 1: Deployment of EDCs in a smart city

This would help the CSPs to provision vehicles with low latency and high data rates. These EDCs are connected to renewable energy sources (RES) like solar panels. The proposed framework is divided into following modules which work in tandem.

- Firstly, a software defined controller handles the vehicle services requests and suggest an optimal flow path for the same using flow management algorithm.
- In order to select an optimal EDC, a *multi-leader multi-follower Stackelberg game* is presented for renewable energy-aware resource allocation to 5G-enabled vehicles from geo-distributed EDCs.
- Moreover, to improve the resource utilization, a cooperative resource sharing scheme is designed, thereby minimizing the energy consumption of servers located in the EDCs. The mutual cooperation among these EDCs for utilization and sharing of resources and information in an optimized manner could help to reduce energy consumption.
- Lastly, to avoid additional energy consumption, an information sharing and caching scheme is also proposed to avoid researching and rerouting of lost link due to the mobility of vehicles.

Therefore, EDCSuS framework forms a multi-objective solution to realize the vision of sustainable EDCs while providing QoS satisfaction, low latency services, mutual cooperation among servers and EDCs, and efficient caching to cope with vehicular mobility. But, to cope with huge number of 5G-enabled vehicles connected to cloud services using EDCs, the underlying networks paradigm need to be more flexible and scalable to handle the dynamic requirements. Hence, software-defined networks (SDN) are used to control the EDC network by decoupling the data plane from control plane. Such a platform could provide an energy-efficient, resilient, scalable, flexible, and dynamic network for sustainable DCs [7], [8].

2 RELATED WORK

Existing proposals (as shown in Table 1) proposed different techniques which used geo-distributed cloud infrastructure for providing services to the end users. For example, Chen *et al.* [9] suggested that the DC infrastructure must be energy-efficient and sustainable in order to cope with the growing demand for data processing. To handle this challenge, the authors proposed a framework which integrates RES, storage units, along with adopting dynamic pricing for workload and energy management in DCs. Similarly, Guo *et al.* [10] proposed an optimization technique for opportunistic scheduling and load balancing among geo-distributed DCs. The authors also utilized thermal storage and distributed RES to optimize the cost and energy usage in DCs.

However, higher latency, lower data rates and increased energy consumption paved the path from centralized cloud DCs towards geo-distributed EDCs. Moreover, the conventional centralized cloud infrastructure provides limited resource sharing and uses high bandwidth for communications. Wang *et al.* [18] highlighted the need of edge computing to meet the low latency and higher data rate requirements of mobile services. The authors suggested that EDCs prove to be effective in handling lower latency and mobility support through geo-distribution. Similarly, Kaur *et al.* [15] highlighted that traditional cloud infrastructure may not be sufficient to process the large volume of information generated by smart devices due to long response time and higher bandwidth consumption. The authors suggested that edge computing can be used to handle these issues by providing service availability to end users at the edge of the network. In this direction, the authors proposed a task selection and scheduling architecture for edge computing using container-as-a-service. They solved the multi-objective problem using cooperative game theoretic approach.

In last few years, various studies have been performed where edge computing has played an effective role in providing low latency and high data rate services to cloud users. For example, Puthal *et al.* [13] proposed a load balancing scheme for EDCs to decrease the latency and network congestion for real time data processing. In an another work, Misra *et al.* [14] proposed a bi-objective (energy and makespan) optimization scheme for metaheuristic-based service allocation using fog servers. Tziritas *et al.* [16] designed a hyper graph based partitioning scheme to reduce network overhead for virtual machine migrations between DCs and micro-DCs. In a different work, Du *et al.* [17] proposed a differential privacy based query model for sustainable fog DCs in order to preserve quality of privacy.

Some of the existing proposals have considered a cooperative game play between edge and cloud resources for effective service provisioning to the end users. For example, Deng *et al.* [20] presented a workload allocation scheme in edge-cloud environment wherein the cooperation between edge and cloud resources helps to achieve an optimal trade-off between delay and power consumption. However, none

TABLE 1: Comparative analysis of existing proposals related to geo-distributed cloud and edge DCs

Author	Technique used	QoS	VW	EE	RU/RA/RO	Coop	GEDCs	Vehicles	SDN	Cach	Mob	Cost	RES	Sust
Chen *et al.* [9]	Robust optimization scheme using Lagrange dual decomposition method	✓	✓	✓	RA	×	×	×	×	×	×	×	✓	✓
Guo *et al.* [10]	Energy and network aware workload management scheme	✓	✓	✓	RA	×	×	×	×	×	×	×	✓	✓
Aujla *et al.* [4]	Support vector machine based workload classification and flow management scheme	✓	✓	✓	RA/RU	×	×	×	×	×	×	×	✓	✓
Li *et al.* [11]	Observable partial markov decision process based scheme	✓	✓	×	RO	×	✓	✓	✓	✓	✓	✓	×	×
Li *et al.* [12]	Energy management framework with distributed RES	×	✓	✓	RA	✓	✓	×	×	×	×	×	✓	✓
Puthal *et al.* [13]	Authentication and sustainable load balancing scheme	×	×	×	RA/RO	×	✓	×	×	×	×	×	×	✓
Misra *et al.* [14]	Meta-heuristic service allocation using particle swarm optimization and bat algorithm	×	×	✓	RA	×	×	×	×	×	×	×	×	✓
Kaur *et al.* [15]	Game theoretic approach for task selection and scheduling	×	×	✓	RA	✓	✓	×	×	×	×	×	×	×
Tziritas *et al.* [16]	Hyper graph partitioning scheme	×	×	×	RA	×	✓	×	×	×	×	×	×	×
Du *et al.* [17]	Privacy based query model	✓	✓	✓	–	×	✓	×	×	×	×	×	×	✓
Wang *et al.* [18]	ChachinMobile, a mobile caching network paradigm for energy-efficient edge nodes management	✓	×	✓	RA	×	✓	×	×	✓	✓	×	×	×
Borylo *et al.* [19]	Optical SDN-based energy-aware fog-cloud interplay	✓	✓	✓	RA	×	✓	×	✓	×	×	×	×	×
Deng *et al.* [20]	Energy aware workload allocation for edge and cloud nodes	✓	✓	✓	RA	×	✓	×	×	×	×	×	×	×
EDCSuS	Software defined EDC as a service framework for vehicular environment	✓	✓	✓	RA/RU/RO	✓	✓	✓	✓	✓	✓	✓	✓	✓

QoS: Quality of services, VW: Variable workload, EE: Energy efficiency, RU/RA/RO: Resource utilization/Resource allocation/Resource optimization, Coop: Cooperative strategy, GDEDCs: Geo-distributed EDCS, SDN: Software defined networks, Cach: Caching, Mob: Mobility, Cost: Cost minimization, RES: Renewable energy sources, Sust: Sustainability.

of the above discussed approaches have focused on the use of sustainable energy resources to design an energy-efficient and software defined edge computing framework. To handle the sustainability issue, Li *et al.* [12] proposed a renewable energy powered sustainable EDC approach for energy management. Moving a step ahead, Aujla *et al.* [4] proposed a software defined energy management scheme for sustainability in edge-cloud scenario. The authors proposed a software defined flow management scheme which works in tandem with a support vector machine-based workload classification approach to achieve energy efficiency and optimal utilization of network and computing resources. Borylo *et al.* [19] proposed a dynamic resource scheduling scheme, wherein an energy-aware interplay takes place between cloud DCs and EDCs. To handle the dynamic network requirements of such a scenario, the authors used SDN architecture for providing energy efficient traffic provisioning between edge and cloud resources.

However, the most challenging aspect that have not been addressed by any of the existing proposals is the device or vehicle mobility while providing the cloud services from EDCs deployed at various locations in smart city. In this direction, Li *et al.* [11] proposed an architecture for vehicular network in a smart city to mitigate the network congestion. The authors proposed a joint optimization scheme for networking, caching and computing resources in a geo-distributed edge computing scenario. The delay sensitive and delay tolerant vehicular traffic was scheduled as per the required QoS while considering the vehicle mobility. However, this proposal have not considered the energy efficiency or sustainable energy resources as a part of their study. Therefore, the proposed framework, EDCSuS is the one of the newest proposal which proposes a software defined edge as a service architecture for solving multi-objective problem related to energy efficiency (cooperative resource sharing and utilization), sustainability (RES), QoS (latency) and caching (link breakage).

3 SYSTEM MODEL

In this section, the network model for the software defined framework for geo-distributed EDCs in a smart city is introduced. Apart from network model, mobility, caching, computational and energy models are also presented.

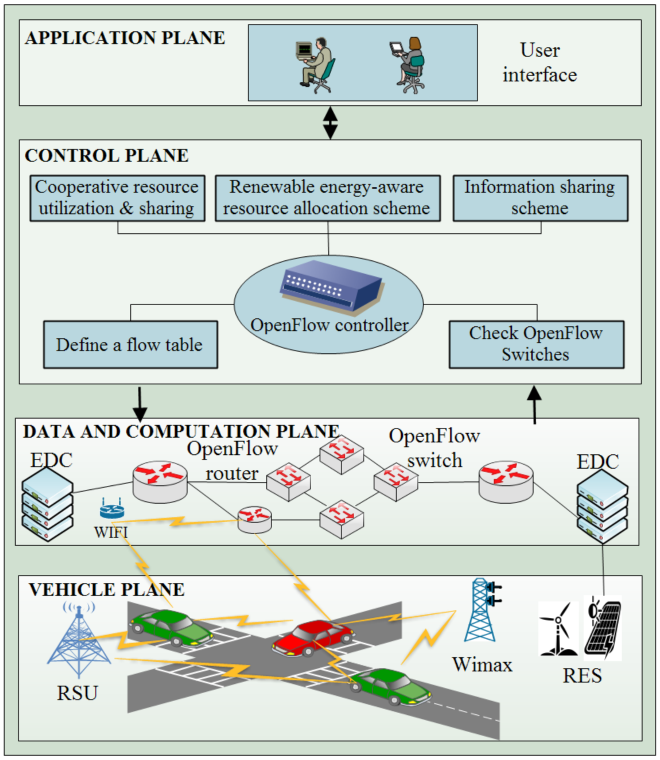


Fig. 2: SDN based controller for energy management

3.1 SDN-enabled Network Model

SDN is a software-centric paradigm which manage the network resources efficiently using decoupled planes. It provides a programmable platform which intelligently provides abstraction to the underlying hardware infrastructure from network resources. It uses open flow (OF) protocol to manage and reconfigure the network according to the dynamic changes in the network requirements [7]. Such underlying network platform improves the overall communication efficiency, energy-efficiency, and latency problems. The high data rate requirements for 5G-enabled vehicles could be effectively handled using scalable and efficient SDN-enabled EDCs. An SDN based controller is used for information flow between vehicles, CSPs, and EDCs. The SDN-enabled network model comprises of four distinct planes, 1) vehicular plane, 2) data and computational plane, 3) control plane and 4) application plane. Fig. 2 depicts the layered architecture of the proposed network model, wherein the EDCs are located at the data and computation plane. The detailed description of the network model is presented as follows.

3.1.1 *Vehicular Plane*

As shown in Figs. 1 and 2, it is considered that i vehicles represented as V_i move randomly on the road. The communication range between i^{th} vehicle and j^{th} EDC is denoted as . Now, whenever a vehicle has to run or access a remote based application, then it connect to cloud. In the proposed work, these vehicles connect to EDC instead of central cloud. The amount of resources required from an EDC to run the required application or provision the intended services depends on CPU (S_i), memory (M_i), storage (ST_i), bandwidth (B_i), time (τ_k), and energy (E_i) required. All these resources are modeled together to compute the required resources as below.

$$R_i^{rq} \approx (S_i, M_i, ST_i, B_i)\tau_k \tag{1}$$

Once the CSPs receive the vehicle request, they compute R_i^{rq} for handling the same. If R_i^{rq} are available with the EDCs connected to the CSPs, they compute price (P_i) to be charged on the basis of R_i^{rq}.

3.1.2 *Forwarding/Data and Computational Plane*

This plane consists of two distinct entities, 1) forwarding devices (OF switches and router) and 2) j EDCs. The OF forwarding devices follow the rules added in their flow tables to forward the data traffic to the destination node [8]. This plane use OF protocol as a communication standard to forward the acquired data [7]. The data acquired from the end devices in forwarded using forwarding devices located at data plane and then processed in an efficient way at EDCs. The EDCs are responsible for providing computing resources to the vehicles for running their applications. Moreover, they are also equipped with storage and cache capabilities. Therefore, two types of delays are associated with this plane, 1) transmission delay (τ_i^{tr}) and 2) computational delay (τ_i^{cp}).

The proposed scheme considers equal sized K time slots represented as τ_0, τ_1, ..., τ_k, ...,τ_{K-1}, wherein the duration of each time slot is given as below.

$$\delta\tau_k = \tau_k - \tau_{k-1} \tag{2}$$

The maximum permissible or tolerant delay (δ_{t_k}) must satisfy following condition.

$$\delta_{t_k} \leq \tau_{mx}^{tr} \tag{3}$$

where, $\tau_m^{tr}x$ denotes the maximum transmission delay.

Each vehicle request can be represented as a distinct computing task (T_i) depicted as below.

$$\mathbf{T_i} \equiv (\mathbf{D_i}, \mathbf{R_i^{rq}}) \tag{4}$$

where, D_i denotes the size of data to be computed.

For performing T_i at j^{th} EDC for i^{th} vehicle, the τ_i^{cp} is given as below.

$$\tau_i^{cp} = \frac{R_i^{rq}}{\zeta_j} \tag{5}$$

where, ζ_j represents the computing capability of j^{th} EDC.

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/TSUSC.2019.2907110, IEEE Transactions on Sustainable Computing

3.1.3 Control Plane

In the proposed software defined vehicular environment, OF controller is responsible for all the networking, computing and caching decisions. Apart from these, the controller provides guidelines to the OF forwarding devices for establishment of flow rules. The controller is also responsible for handling the migrations among EDCs and CSPs.

3.1.4 Application/Management Plane

All the network applications are present at this plane. The major objective of this plane is to act as provide a platform for meeting QoS requirements and latency needs.

3.2 Energy Model

The architecture of an EDC connected to RES is shown in Fig. 3. One or more EDCs are administrated by l CSPs. The CSP keeps regular updates regarding (i) available resources, (ii) level of utilization, and (iii) renewable energy. Using the above information, CSP decides the EDC that would host the request of users and the respective price of the resources.

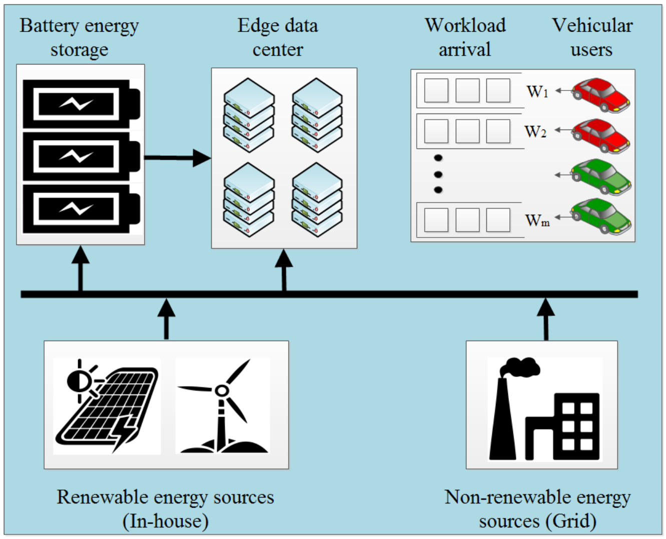


Fig. 3: Architecture of renewable-powered edge DC

The energy model is divided into two parts, 1) energy consumption model and 2) energy generation model. Both these parts are discussed as below.

3.2.1 Energy Consumption Model

The edge infrastructure consists of j geo-distributed EDCs that are responsible to host the applications to i 5G-enabled vehicles. Each EDC is equipped with sufficient number of servers, memory and storage units which consume energy ($\mathbf{E_i}$) to run its routine activities. The overall energy consumption of an EDC depends on the energy consumed by p servers ($\mathbf{E^P_j}$), q OF devices ($\mathbf{E^q_j}$), cooling equipment ($\mathbf{E^{cl}_j}$), and various other activities ($\mathbf{E^{oth}_j}$) and is given as below.

$$\mathbf{E}_j = \sum_p \mathbf{E}^s_j + \sum_q \mathbf{E}^q_j + \mathbf{E}^{cl}_j + \mathbf{E}^{oth}_j \tag{6}$$

The energy consumption of each server depends upon its level of utilization ($\mho^s_j$). The level of utilization is the ratio of resources allocated, $\mathbf{R}^{all}_{sj}(k)$ at k^{th} time slot and the maximum resource capacity ($\mathbf{R}^{mx}_{sj}$).

$$\mho^s_j = \frac{\mathbf{R}^{all}_{sj}(k)}{\mathbf{R}^{mx}_{sj}} \times 100 \tag{7}$$

Alternately, $\mho^s_j$ can be computed using the below mentioned equation:

$$\mho^s_j = \frac{(c_1 \times \mathbf{S}^p) + (c_2 \times \mathbf{S}^m)}{\mathbf{S}^p + \mathbf{S}^m} \times 100$$

where, $\mathbf{S}^p$ and $\mathbf{S}^m$ refer to the server's processing and memory capacities respectively. On the hand, the overall allocated portions of $\mathbf{S}^p$ and $\mathbf{S}^m$ are represented using the constants c_1 and c_2.

On the basis of level of utilization, the energy consumed by s^{th} server of j^{th} EDC is given as below.

$$\mathbf{E}^s_j = \mathbf{E}^{idl}_j + (\mathbf{E}^{mx}_j - \mathbf{E}^{idl}_j)\mho^s_j \tag{8}$$

where, where, $\mathbf{E}^{idl}_j$ is energy consumption of an idle server, $\mathbf{E}^{mx}_j$ represents maximum energy consumption of a server.

The above equation can be represented as below.

$$\mathbf{E}^s_j = \mathbf{E}^{idl}_j + (\mathbf{E}^{mx}_j - \mathbf{E}^{idl}_j)\left(\frac{\mathbf{R}^{all}_{sj}(k)}{\mathbf{R}^{mx}_{sj}} \times 100\right) \tag{9}$$

The energy consumption of the network devices depends on the fixed part ($\mathbf{E}^q_{sw}$), i.e., active switch components and variable part ($\mathbf{E}^q_{port}$), i.e., active ports. The energy consumption of a typical DC is given as below [21].

$$\mathbf{E}^q_j = \mathbf{E}^q_{sw} + \mathbf{E}^q_{port} \tag{10}$$

The above equation can be further expanded as below.

$$\mathbf{E}^q_j = \sum_{s \in \mathbf{S}} \mathbf{P}_s \times \mathbf{T}_s + \sum_{p \in \mathbf{P}_s} \mathbf{P}^s_p \times \mathbf{T}^s_p \tag{11}$$

where, $\mathbf{S}$ denotes set of switches, and $\mathbf{P}_s$ denotes set of ports in switch s. $\mathbf{P}_s$, $\mathbf{T}_s$, $\mathbf{P}^s_p$, and $\mathbf{T}^s_p$ is the fixed power consumed by s^{th} switch, working time of s^{th} switch, dynamic power consumed by p^{th} port of s^{th} switch, and working time of p^{th} port of s^{th} switch.

3.2.2 Energy Generation Model

Each EDC is powered by RES: solar energy ($\mathbf{E}^{sl}_j$) and wind energy ($\mathbf{E}^{wn}_j$). The photovoltaics (PV) panels are used to generate the solar energy using variable sunshine. Similarly, wind energy is generated using wind turbines using intermittent wind speed. The energy generated ($\mathbf{E}^{res}_j$) by RES connected to j^{th} EDC is given as below [22].

$$\mathbf{E}^{res}_j = \mathbf{E}^{sl}_j + \mathbf{E}^{wn}_j \tag{12}$$

The energy generated ($\mathbf{E}^{sl}_j$) by PV panels connected to j^{th} EDC is given as below [23].

$$\mathbf{E}^{sl}_j = ([1 - \mathbf{L}_{cor}]\ \eta\ \mathbf{S}_{pv}\ a\ \text{Ŗ}) \tag{13}$$

where, $\mathbf{S}_{pv}$ is the size of panel and Ŗ denotes solar radiation, η is the conversion efficiency of a PV panel, $a = cos(\alpha)$ depends on radiant angle (α) of sunlight on a PV panel, $\mathbf{L}_{cor}$ is the corridor temperature exedence loss.

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/TSUSC.2019.2907110, IEEE Transactions on Sustainable Computing

The energy generated ($\mathbf{E}_j^{wn}$) by wind turbine connected to j^{th} EDC is given as below [23].

$$\mathbf{E}_j^{wn} = \left(\frac{1}{2}\left[\mathbf{C}_p\ \rho\ \mathbf{A}\ v^3\right]\right) \tag{14}$$

where, $\mathbf{C}_p$ denotes rotar efficiency, $\mathbf{A}$ denotes the rotor swept area, ρ is the air density, and v is the wind speed.

Using Eq. 13 and 14, the Eq. 12 can be expanded as shown below.

$$\mathbf{E}_j^{res} = \left(\left[1 - \mathbf{L}_{cor}\right]\eta\ \mathbf{S}_{pv}\ a\ R\right) + \left(\frac{1}{2}\left[\mathbf{C}_p\ \rho\ \mathbf{A}\ v^3\right]\right) \tag{15}$$

EDCs are sensitive infrastructure and any power shortage situation could lead to economic and data losses. Hence, to avert such a situation, a battery energy storage system is connected to each EDC. EDCs are also connected to grid for handling situations when RES is in deficit [24].

3.3 Vehicle Mobility Model

The biggest concern for the proposed framework is the vehicle mobility. Initially, the vehicles are provisioned resources from an EDC for running their applications. However, due to the mobility of the vehicles, the initial location of V_i would change from $loc(x_1, y_1)$ to $loc(x_2, y_2)$. In this case, to avoid any QoS degradation or service level of agreement (SLA) violations, there may be a need to migrate from one EDC to another. Therefore, we consider a set $\varphi(k)$ which represents the direction of movement of vehicles at time t_k. There are following cases for movement, i.e., dynamic (D) and fixed (F). The dynamic case considers north (N), south (S), east (E), west (W) movements. The distance from current EDC to the destination EDC where the vehicle applications would be migrated is represented as $d(k)$.

3.4 Caching Model

In the proposed scheme, another important issue is caching. Whenever a service is to be migrated from one EDC to another EDC, they successful QoS maintenance depends on prior caching of data at destination EDC. It is assumed that each EDC has sufficient physical cache ($\mathbf{C}_j$) availability. It may be noted that EDCs have limited storage in contrast to central cloud storage. Therefore, the content to be cached must satisfy following condition.

$$\sum_{j=1}^{J}\sum_{l=1}^{L}(\mathbf{C}_i^{mig}) \leq \mathbf{C}_j^{cap} \tag{16}$$

4 PROBLEM FORMULATION

Definition 1: (SLA adherence) In cloud sector, SLA plays a very significant role in defining the relationship between CSPs and end users. SLA refers to agreement regarding the level of QoS, availability and reliability for providing services to end users. In the proposed work, SLA adherence ($\mathbf{SLA}(S_{sj})$) is decided on the basis of computing and memory resources allocated ($\mathbf{R}_{sj}^{all}$) to task, T_i and the resources required ($\mathbf{R}_i^{rq}$) for successful task execution.

$$\mathbf{SLA}(S_{sj}) = \left(\frac{\mathbf{R}_{sj}^{all}}{\mathbf{R}_i^{rq}}\right) \times 100 \tag{17}$$

$$\begin{aligned} s.t.\ \mathbf{R}_{sj}^{all} &= (c_1 \times r_{sj}^{p}) + (c_2 \times r_{sj}^{m}) \\ \mathbf{R}_i^{rq} &= r_{sj}^{p} + r_{sj}^{m} \end{aligned}$$

where, r_{sj}^{p} and r_{sj}^{m} are the processing and memory requirements for handling task T_i on the s^{th} server of the j^{th} EDC.

Definition 2: (SLA violation) SLA violation ($\mathbf{SLA}_v$) refers to the non-adherence of SLA by CSP for provisioning the services. Herein, $\mathbf{SLA}_v$ is defined as compliment of $\mathbf{SLA}$, which is defined as follows:

$$\mathbf{SLA}_v(S_{sj}) = \left(\frac{\mathbf{R}_{sj}^{rq} - \mathbf{R}_{sj}^{all}}{\mathbf{R}_{sj}^{rq}}\right) \times 100 \tag{18}$$

Definition 3: (Energy utilization cost) The cost of server energy utilization ($\mathbf{C}$) depends on the cost of total energy utilization by the IT devices ($\mathbf{C}_{IT}$) and the cost of energy overhead due to migrations ($\mathbf{C}_{MIG}$). This cost can be represented as below.

$$\mathbf{C}(S_{sj}) = \Big(\mathbf{C}_{IT}(S_{sj}) + \mathbf{C}_{MIG}(S_{j_1-j_2})\Big) \tag{19}$$

$$\begin{aligned} s.t.\ \mathbf{C}_{IT}(S_{sj}) &= \sum_{\rho\in\{RES\}} \mathbf{E}_j^{s} \times \mathbf{C}_\rho^{E} \\ \mathbf{C}_{MIG}(S_{j_1-j_2}) &= \sum_{\rho\in\{RES\}} \mathbf{E}_{j_1-j_2}^{MIG} \times \mathbf{C}_\rho^{E} \end{aligned}$$

where, $\mathbf{E}_{j_1-j_2}^{MIG}$ represents the migration overhead energy consumption and $\mathbf{C}_\rho^{E}$ is the energy cost and ρ represents the energy source, i.e., solar, wind or non-RES.

Definition 4: (Task to Server mapping variable) A binary decision variable, i.e, α_{sjl} is used to represent task, T_i to S_{sj} mapping as shown below.

$$\alpha_{sjl} = \begin{cases} 1: & \text{If } T_i \text{ is mapped on the } s^{th} \text{ server of } j^{th} \text{ EDC} \\ 0: & \text{Otherwise} \end{cases}$$

where, α_{sjl} is set to the value of 1 if T_i is mapped on the s^{th} server of j^{th} EDC. Otherwise, it is set to the value of 0.

Definition 5: (EDC to CSP mapping variable) A binary decision variable (β_{jl}) for EDC to CSP mapping variable is represented as below:

$$\beta_{jl} = \begin{cases} 1: & \text{If } j^{th} \text{ EDC is mapped with } l^{th} \text{ CSP} \\ 0: & \text{Otherwise} \end{cases}$$

where, β_{jl} is set to the value of 1 if j^{th} EDC is mapped with l^{th} CSP. Otherwise, it is set to the value of 0.

4.1 List of Objectives

The proposed framework involves multiple objectives which are discussed as below.

4.1.1 Maximum SLA adherence and QoS guarantee

The topmost priority of the proposed framework is to meet the required SLA and guarantee the desired QoS while scheduling the task to geo-distributed EDCs. The mathematically expression for the same is shown as objective function $\mathbf{F}_1(\alpha_{sjl}, \beta_{jl})$.

$$\begin{aligned} \mathbf{F}_1(\alpha_{sjl}, \beta_{jl}) = \sum_{i\in I}\sum_{j\in J}\sum_{l\in L}\Big(\mathbf{SLA}(S_{sj})\Big) \\ \times\ \alpha_{sjl} \times \beta_{jl} \end{aligned} \tag{20}$$

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/TSUSC.2019.2907110, IEEE Transactions on Sustainable Computing

4.1.2 Minimum cost of energy

This objective function ($\mathbf{F}_2(\alpha_{sjl}, \beta_{jl})$) is related to the energy cost associated to hardware resources while providing desirable QoS. It is represented mathematically as below.

$$\mathbf{F}_2(\alpha_{sjl}, \beta_{jl}) = \sum_{i\in I}\sum_{j\in J}\sum_{l\in L}\Big(\mathbf{C}(S_{sj})\Big) \times \alpha_{sjl} \times \beta_{jl} \tag{21}$$

4.1.3 Maximum sustainability

The final objective of EDCSuS, represented as objective function ($\mathbf{F}_3(\alpha_{sjl}, \beta_{jl})$) deals with maximizing the use of sustainable energy. This is represented as below.

$$\mathbf{F}_3(\alpha_{sjl}, \beta_{jl}) = \mathbf{E}_i - \mathbf{E}_j^{ren} \tag{22}$$

4.2 List of Constraints

In lieu of the aforementioned objective functions, the set of constraints are defined as below.

$$\textbf{C1: } \sum_{s\in S}\sum_{j\in J} r_{sj}^p \times \alpha_{sjl} \leq S^p$$

$$\textbf{C2: } \sum_{s\in S}\sum_{j\in J} r_{sj}^m \times \alpha_{sjl} \leq S^m$$

$$\textbf{C3: } \alpha_{sjl} \in \{0,1\}; \forall s, j, l$$

$$\textbf{C4: } \beta_{jl} \in \{0,1\}; \forall j, l$$

wherein, constraints C1 and C2 apply upper limitation on the amount of the resources (processing, S^p and memory units, S^m) that can be allocated by a server for handling the incoming tasks, C3 and C4 are used to employ integrality restriction on α_{sjl} and β_{jl}.

4.3 Multi-objective optimization problem

Now, in order to achieve an optimal solution, the multi-objective optimization problem is formulated as below mentioned objectives ($\mathbf{F}(\alpha_{sjl}, \beta_{jl})$) in the multi-constraint setup.

$$\mathbf{F}(\alpha_{sjl}, \beta_{jl}) = \min f(-\mathbf{F}_1, \mathbf{F}_2, \mathbf{F}_3) \tag{23}$$
$$s.t. \quad \text{Constraints } \mathbf{C1} - \mathbf{C4} \text{ hold}$$

5 EDCSUS: THE PROPOSED FRAMEWORK

The major objective of the proposed scheme is to balance the energy generated by RES connected to edge-DCs and energy consumption of edge-DCs, i.e., ($E_{dc} = E_{ren}$). The objective is achieved using the proposed scheme which works in three parts as described below.

5.1 Energy-aware Flow Scheduling Algorithm

The proposed framework, EDCSuS has to face a tough challenge to handle the incoming service requests of vehicles from geo-distributed EDCs. For this purpose, efficient flow path provisioning is one of the most prerequisite. However, balancing the load over OF devices while maintaining flow rate and energy consumption is a tough challenge. Therefore, an energy-aware flow scheduling algorithm is proposed to handle the above mentioned challenges. For this purpose, a load balancing factor is used to limit the incoming load on OF devices, which is given as below [5].

$$\phi = \frac{1/q \times \sum_0^q \mathbf{L}^q(k)}{\mathbf{L}_q^{mx}} \tag{24}$$

where, $\mathbf{L}^q(k)$ and $\mathbf{L}_q^{mx}$ denotes load on q^{th} OF device and the maximum bearable load of an OF device.

The above defined load balancing factor range is considered between 0 and 1. The load is evenly distributed if ϕ is close to the value of 1, else it is otherwise.

An incoming flow is designated as task, $\mathbf{T}_i$ which must be scheduled at an optimal EDC. Each flow is represented as f having size ($\mathbf{P}_{size}$) with a deadline time ($\mathbf{T}_i^{deadline}$) and release time ($\mathbf{T}_i^{release}$). Using the above factors, the guaranteed flow rate ($\mathbf{F}_{i-j}^{gr}$) from i^{th} vehicle to j^{th} EDC is given as below.

$$\mathbf{F}_{i-j}^{gr} = \frac{\mathbf{P}_{size}}{\mathbf{T}_i^{deadline} - \mathbf{T}_i^{release}} \tag{25}$$

The proposed flow scheduling Algorithm 1 is as below.

Algorithm 1 Energy-aware flow scheduling algorithm

Input: $\mathbf{T}_i^{deadline}$, $\mathbf{T}_i^{release}$, f, $\mathbf{F}_{rate}$, $\mathbf{P}_{size}$
Output: $\mathbf{F}_{i-j}^{gr}$, F_{path}

```
1:  Calculate $\mathbf{F}_{i-j}^{gr} = \frac{\mathbf{P}_{size}}{\mathbf{T}_i^{deadline} - \mathbf{T}_i^{release}}$
2:  $f \leftarrow$ FindPath($\mathbf{F}_{i-j}^{gr}$, $\mathbf{P}_{size}$, $\mathbf{F}_{rate}$)
3:  if $F_{path}$ === TRUE then
4:      Schedule $f$ over $\mathbf{F}_{path}$
5:      for Each $\mathbf{F}_{path}$ do
6:          Divide into flow sets $\mathbf{F}_{set}$ having distinct links
7:          for $\mathbf{F}_{set}$ do
8:              Calculate $\tau_{act}$ = activetime($\mathbf{F}_{set}$)
9:              Compute $\mathbf{E}_j^q = \sum_{s\in\mathbf{S}} \mathbf{P}_s \times \mathbf{T}_s + \sum_{p\in\mathbf{P}_s} \mathbf{P}_p^s \times \mathbf{T}_p^s$
10:             if ($\tau_{act}$ is minimum) then
11:                 $\mathbf{F}^{queue} \leftarrow \mathbf{F}^{queue} + f$
12:             end if
13:         end for
14:     end for
15: else
16:     Report $\rightarrow$ OF controller
17:     OF controller $\rightarrow$ rebuilds new $\mathbf{F}_{path}$
18:     Repeat steps 3-11
19: end if
20: if $f$ == SUCCESS then
21:     Update $\mathbf{F}^{active} \leftarrow \mathbf{F}^{active}$ - $f$
22:     Move $\rightarrow$ next flow in queue to the top
23: end if
```

In the proposed algorithm, initially a guaranteed flow rate ($\mathbf{F}_{i-j}^{gr}$) is computed on the basis of incoming flow (f). After this, the OF controller finds a valid flow path (F_{path}) on the basis of $\mathbf{F}_{i-j}^{gr}$, packet size ($\mathbf{P}_{size}$) and flow rate ($\mathbf{F}_{rate}$). If F_{path} is valid, then f is scheduled. But, in order to find energy-aware path, F_{path} is divided into F_{set} having distinct links. After this, the active time (τ_{act}) for each F_{set} is computed. Now, f is scheduled on the path having minimal energy consumption and added to F^{queue} for scheduling. However, if in a case no valid F_{path} is available, then OF controller is reported for the same. The OF controller rebuilds a new F_{path} and install the same on the OF devices and thereon f is scheduled accordingly. If the scheduling of f is successful, then it is removed from the F^{active} list of incoming flows and the same is updated accordingly. The flow on the top of the list is scheduled thereafter.

5.2 Multi-leader Multi-follower Stackelberg Game for Renewable Energy-aware EDC selection

Stackelberg game is a two-stage game where leader initiates the game and the follower revert back accordingly [25]. In this scheme, the Stackelberg game moves in two stages where *i* vehicles act as multiple leaders and *l* CSPs administrating various edge-DCs act as multiple followers. Both the players play their moves in a sequential manner. The utility functions of vehicles and CSPs play an important role in the decision making. A concave revenue function ($\mathbf{R}_i$) and $\mathbf{P}_i$ is considered to calculate the utility function ($\mathbf{U}_i$) for vehicular users. The price announced by CSPs is computed using price coefficient ($\wp$) which varies for CPU, memory, storage, and bandwidth for a specific period of time. The utility functions for vehicles is given as below.

$$\mathbf{U}_i = \mathbf{R}_i - \mathbf{C}_i \tag{26}$$

where, cost involved with respect to different resources ($\mathbf{C}_i$) is given as below.

$$\mathbf{C}_i = (\wp\mathbf{S}_i + \wp\mathbf{M}_i + \wp\mathbf{S}_i + \wp\mathbf{B}_i + \wp\mathbf{E}_i)\, t_k \tag{27}$$

The utility function ($\mathbf{U}_l$) for CSPs is drafted using revenue function and a cost function. The revenue function ($\mathbf{R}_l$) for CSP consists of revenue generated by selling the cloud resources. The cost incurred ($\mathbf{C}_l$) consists of operational ($\mathbf{C}_{opr}$), maintenance ($\mathbf{C}_{mn}$), energy ($\mathbf{C}^E_\rho$), and migration ($\mathbf{C}_{MIG}(S_{j_1-j_2}$) costs. Hence, The $\mathbf{U}_l$ is given as below.

$$\mathbf{U}_l = \mathbf{R}_l - \mathbf{C}_l \tag{28}$$

After expanding Eq. 28, we get

$$\mathbf{U}_l = \sum_i (\mathbf{P}_i) - \Big(\mathbf{S}_j + \mathbf{C}^E_\rho + \mathbf{C}_{MIG}(S_{j_1-j_2} + \mathbf{C}_{opr}\Big) \tag{29}$$

The working of the two stage multi-leader multi-follower Stackelberg scheme for EDC selection is shown in Algorithm 2. In stage 1, i vehicular users who need to access resources ($\mathbf{R}^{rq}_i$) send a request to the available *l* CSPs. The CSPs map the required resources with the resources available at the EDCs. If the EDCs have sufficient resources, it check for the level of utilization of various resources. If the level of utilization is less than the specified threshold level of utilization, then the concerned server is stored in available server list (ASL). After this, the CSPs checks for the amount of renewable energy available with the EDCs. If the EDCs have sufficient amount of renewable energy, then the value of utility for concerned *sj* pair is calculated and compared with all other pairs. The best *sj* pair is selected and accordingly price is computed and announced to $\mathbf{V}_i$. After this the concerned server is stored in overloaded server list (OSL). However, in case of energy deficit CSPs select any other EDC. In stage 2, the vehicles compute their payoff using the utility function and compares it with the payoff received with respect to payoff computed for other available CSPs and selects the CSP which offers maximum benefit to it. Once the user pays the accepted price, the requested resources are allocated to it.

5.3 Optimal resource utilization and cooperative resource sharing scheme

This scheme is divided into two parts, (1) optimal resource utilization, and (2) cooperative resource sharing and migration. Both the schemes are discussed as below.

Algorithm 2 Stackelberg scheme for EDC selection

```
Input: $\mathbf{R}^{rq}_i$, $\mathbf{R}^{all}_{sj}$, $\mathbf{R}^{mx}_{sj}$
Output: EDC, sj pair
procedure FUNCTION(Stage 1)
    $\mathbf{R}^{rq}_i \approx (\mathbf{S}_i, \mathbf{M}_i, \mathbf{S}_i, \mathbf{B}_i)\tau_k$
    Request $\rightarrow l$
    for $(l = 1; l \leq n; l++)$ do
        for $(j = 1; j \leq m; j++)$ do
            for $(s = 1; s \leq p; s++)$ do
                Compute $\mathbf{R}^{all}_{sj} = (c_1 \times r^p_{sj}) + (c_2 \times r^m_{sj})$
                Compute $\mathbf{R}^{avl}_{sj} = \mathbf{R}^{mx}_{sj} - \mathbf{R}^{all}_{sj}$
                if $(\mathbf{R}^{rq}_i < \mathbf{R}^{avl}_{sj})$ then
                    Compute $\mho^s_j$;
                    if $(\mathbf{U}^s_j < \mathbf{U}^{mx}_{sj})$ then
                        Add $s^{th}$ server in ASL
                        Compute $(\mathbf{E}^j_{res})$
                        Compute $(\mathbf{E}^s_j)$
                        if $(\mathbf{E}^j_{res} > \mathbf{E}^s_j)$ then
                            Compute $\mathbf{U}_l = \mathbf{R}_l - \mathbf{C}_l$
                            if $\mathbf{U}_l(sj) > \mathbf{U}_l(s^*j^*)$ then
                                Select sj pair
                                sj pair can host $\mathbf{R}^{rq}_i$
                                Announce price $\mathbf{P}_i$ to $\mathbf{V}_i$
                            else
                                Select another pair from $s^*j^*$ set
                            end if
                            Select another pair from $s^*j^*$ set
                        end if
                        Add $s^{th}$ server in OSL
                    end if
                    $s^{th}$ server == FULL
                end if
            end for
        end for
    end for
end procedure
procedure FUNCTION(Stage 2)
    Compute $\mathbf{U}_i = \mathbf{R}_i - \mathbf{C}_i$
    if $\mathbf{U}_i(l) > \mathbf{U}_i(l^*)$ then
        Select best option
        Converge to result
        SET $\alpha_{sjl}$ == 1
        SET $\beta_{jl}$ == 1
    end if
end procedure
```

5.3.1 *Optimal resource utilization scheme*

The optimal resource utilization aims to utilize the resources, i.e. servers and network infrastructure in such a way that they consume less energy. The working of the utilization scheme is shown in Algorithm 3. If s^{th} server of j^{th} EDC at time slot *k* is idle, then it is shifted to sleep mode. Such servers in sleep mode are added to a sleep list (SL) so as to be used at later stages. If the level of utilization of s^{th} server of j^{th} EDC is below threshold level of utilization, then it is added to list active list (AL). The servers listed in AL are utilized for allocation of resources. But, when AL is empty, then the servers in list SL are shifted into active mode and added into list AL. Now such server could be utilized for resource allocation. Similarly, the network infrastructure is utilized in an optimal manner on the basis of OF devices. If an OF device is inactive, then it is shifted into sleep mode and stored in list SL. But, if it is active and its level of utilization at time slot *k* is less than the threshold level of utilization, then it is stored in list AL. Now, the OF devices listed in AL are utilized initially. However, if the list AL becomes empty, then the OF devices listed in SL are shifted into active mode and stored in AL and utilized accordingly. Hence, this scheme utilizes the resources optimally so as to reduce the energy consumption of resources.

Algorithm 3 Optimal resource utilization algorithm

Input: $\mathbf{R}_{sj}^{all}$, $\mathbf{R}_{sj}^{mx}$
Output: AL, SL

```
procedure FUNCTION(server)
    if ($\mho_j^s(k) == IDLE$) then
        $s^{th}$ server $\rightarrow$ $sleep\_mode$;
        Store $s^{th}$ server $\rightarrow$ SL;
    else if ($\mho_j^s(k) < THRESHOLD$) then
        Store $s^{th}$ server $\rightarrow$ AL;
        Allocate servers $\rightarrow$ AL;
    else
        $s^{th}$ server $\rightarrow$ FULL;
    end if
    if (AL == NULL) then
        Restart server $\leftarrow$ SL;
        Shift $s^{th}$ server in SL $\rightarrow$ AL;
        Allocate servers $\rightarrow$ AL;
    else
        All servers == UTILIZED;
    end if
end procedure
procedure FUNCTION(network)
    if ($\mho_j^q(k) == IDLE$) then
        $q^{th}$ OF device $\rightarrow$ $sleep\_mode$;
        Store $q^{th}$ OF device $\rightarrow$ SL;
    else if ($\mho_j^q(k) < THRESHOLD$) then
        Store $q^{th}$ OF device $\rightarrow$ AL;
        Allocate OF devices $\rightarrow$ AL;
    else
        $q^{th}$ OF device $\rightarrow$ FULL;
    end if
    if (AL == NULL) then
        Restart OF devices $\leftarrow$ SL;
        Shift $q^{th}$ OF device in SL $\rightarrow$ AL;
        Allocate OF devices $\rightarrow$ AL;
    else
        All OF devices == UTILIZED;
    end if
end procedure
```

5.3.2 *Cooperative resource sharing and migration scheme*

In the cooperative resource sharing and migration scheme, each CSP provides two types of resource allocation such as- local resource allocation, and remote resource allocation. Local resource allocation involves allocation of resources to a limited range or area, i.e., small traveling radius of vehicles. However, remote resource allocation covers a wide range or area and it is used when a vehicle exceeds the range of local resource allocation. Inter-EDC resource migration is used to bridge the two kinds of allocations which involves migration of applications to the EDCs located in the region closer to the moving vehicles. By adopting this mechanism, the vehicles can efficiently utilize local as well as remote resources on the move. The working of the cooperative resource sharing scheme is shown in Algorithm 4. If the utilization level of available resources with an EDC is a CSP is below utilization threshold, then such an EDC can join the local coalition (*lcol*). However, it has to verify the gain expected to *lcol* after becoming its part. For this purpose, a utility function of *lcol*, $\mathbf{U}_{lcol}(j)$ is computed as below.

$$\mathbf{U}_{lcol}(j) = \frac{1}{|\mathbf{J}, \mathbf{L}|} \sum_{\mathbf{J}, \mathbf{L}} [v(\mathbf{W}_{lcol}\ U\ j_1) - v(\mathbf{W}_{lcol})] \tag{30}$$

where, W_{lcol} is the worth of *lcol*, j_1 is the new EDC that intends to join *lcol*.

Now, the utility function is computed after adding new EDC (j_1) to the *lcol*. If the overall utility of *lcol* shows a gain after the joining of j_1, then j_1 is allowed to join *lcol*. If the resources required by CSPs is not available with connected EDC, then they search for resources with local and global EDCs that have joined the cooperative scheme. If the resources are available with multiple cooperative EDCs, then the Stackelberg game for resource allocation is followed to select an optimal EDC. Similarly, if the utilization level of available resources with a CSP is below utilization threshold level, then such a CSP can join the remote coalition (*rcol*). However, it has to verify the gain expected to *rcol* after becoming its part. For this purpose, a utility function of *rcol*, $\mathbf{U}_{rcol}(l)$ is computed which is given as below.

$$\mathbf{U}_{rcol}(l) = \frac{1}{|\mathbf{L}|} \sum_{\mathbf{L}} [v(\mathbf{W}_{rcol}\mathbf{U}j_1) - v(\mathbf{W}_{rcol})] \tag{31}$$

where, W_{rcol} is the worth of *rcol*, l_1 is the new CSP that intends to join *lcol*.

Now, the utility function is computed after adding new CSP (l_1) to the *rcol*. If the overall utility of *rcol* shows a gain after the joining of l_1, then l_1 is allowed to join *rcol*.

Algorithm 4 Cooperative resource sharing algorithm

Input: $\mathbf{R}_s^{avl}j$, $\mho_j^s$,
Output: Coalition (local and remote)

```
procedure FUNCTION(coalition-formation)
    for ($l = 1; l \leq n; l++$) do
        for ($j = 1; j \leq m; j++$) do
            if $\mathbf{R}_s^{idl}j < \mathbf{R}_s^{avl}j \leq \mathbf{R}_s^{cap}j$ then
                Join $\rightarrow$ local collision (icol);
                $\mathbf{U}_{lcol}(j) \rightarrow$ Eq.();
                $\mathbf{U}_{lcol}(j + j_1) \rightarrow$ Eq.();
                if ($\mathbf{U}_{lcol}(j + j_1) \geq \mathbf{U}_{lcol}(j)$) then
                    Allow to join;
                    Update $j + j_1 \rightarrow j$;
                else
                    Not allowed
                end if
            end if
        end for
        if ($\mho_{l_1} \leq \mho_{l_1}^{thr}$) then
            Join $\rightarrow$ remote collision (rcol);
            $\mathbf{U}_{rcol}(l) \rightarrow$ Eq.();
            $\mathbf{U}_{rcol}(l + l_1) \rightarrow$ Eq.();
            if ($\mathbf{U}_{rcol}(l + l_1) \geq \mathbf{U}_{rcol}(l)$) then
                Allow to join;
                Update $l + l_1 \rightarrow l$;
            else
                Not allowed
            end if
        end if
    end for
end procedure
```

However, the mobility of vehicles act as a hindrance in meeting the SLA, QoS and latency commitments. In order to resolve this issue, the allocated resources are migrated to other EDCs which are located closer to the position of vehicles. Hence, a migration scheme is designed to meet the latency requirements. The working of the migration scheme is shown in Algorithm 5. In this algorithm, the current requirement of resources, $R_i^{rq}(current)$ are computed on the basis of $R_i^{rq}(k)$ and computations performed at current

EDC. Now, if the movement of vehicle, $\varphi(k)$ is dynamic (D), then the location of vehicle is checked. On the basis of this location, the distance, *d(k)* of V_i from all available EDCs is computed. After this, the migration is performed to the EDC which is located at shortest distance from current EDC. However, if $\varphi(k)$ is fixed (F), then the distance of all available EDCs from the current EDC is calculated and the migration is performed on the basis of shortest distance.

Algorithm 5 Inter-EDC migration algorithm

```
Input: $R_i^{rq}(k)$, $R_i^{rq}(k-1)$, $\mathbf{R}_s^{all} j$
Output: $R_i^{rq}(current)$, EDC

procedure FUNCTION(migration)
    for $(i = 1; i \leq n; i++)$ do
        $R_i^{rq}(current) = R_i^{rq}(k)$ - $R_i^{rq}(k-1)$
        if $\varphi(k)$ == D then
            Check $loc(x_i, y_i)(k)$
            for $(j = 1; j \leq m; j++)$ do
                Check $loc(x_j, y_j)(k)$
                Compute d(k)
                Arrange in order of d(k)
                if then $R_i^{rq}(current) \rightarrow lcol$
                    Migrate to EDC with lowest d(k)
                else
                    Check $rcol$
                    if then $R_i^{rq}(current) \rightarrow rcol$
                        Migrate to EDC with lowest d(k)
                    end if
                end if
            end for
        else if $\varphi(k)$ == F then
            Compute d(k)
            if then $R_i^{rq}(current) \rightarrow lcol$
                Migrate to EDC with lowest d(k)
            else Check $rcol$
                if then $R_i^{rq}(current) \rightarrow rcol$
                    Migrate to EDC with lowest d(k)
                end if
            end if
        end if
    end for
end procedure
```

The above discussed schemes helps EDCs in following ways (1) maintain the optimal level of utilization, (2) provide lower latency requirements, and (3) reduce energy consumption. However, EDCs have to face a challenge of link loss due to mobility of vehicles. In such situation, additional energy is consumed to retrace the lost link. To overcome such a situation, an information sharing and caching scheme is proposed in the next subsection.

5.4 Energy-efficient information sharing scheme

As the vehicles are moving on the road, so they may go far away from the EDC serving their request. This may lead to higher latency. So, to overcome this situation, the allocated services are migrated to another EDC closer to the location of vehicles. However, during this migration, a link loss and migration delay may occur. This is due to re-searching and re-routing performed to trace the location of the lost link. Such an activity bears additional energy consumption. However, it can be avoided by sharing the content information among EDCs, which can avoid re-searching and re-routing. The information regarding running applications is shared among all the EDCs participating in the cooperative scheme at regular intervals. A trace of running application is saved in the cache of each participating EDC.

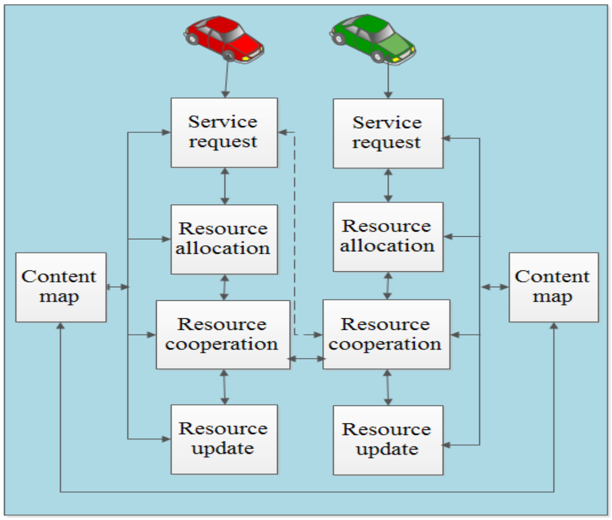


Fig. 4: Inter-DC information sharing scheme

Different aspects of proposed scheme as shown in Fig. 4 are discussed as follows.

- *Service request* (SR) is the procedure in which vehicular users sends a request to CSPs to access the services. The CSPs add the service requests in different queues with respect to priority and type of workload.
- *Resource allocation* (RA) follows two steps; Stackelberg game in the first step and cooperative scheme is the next step.
- *Resource cooperation* (RC) refers to sharing of resources among EDCs when deficit occurs. DCs migrate resources among each other to deal with the dynamic latency requirements of the vehicles.
- *Resource update* (RU) is the procedure adopted by EDCs to update the status of the resource after each allocation or sharing. The updated information is stored in the database repository or a content map.
- *Content map* (CM) is used to store the trace of the applications presently running at all cooperating EDCs. CM maintains a flow and content table of EDCs. CMs of each EDC share information at regular intervals and updates accordingly. When a EDC joins the cooperative scheme, its CM is empty. With each migration, the CM updates the information related to it.

For synchronization of cache located with in the same EDC or at different EDC with respect to data updates, a variable $\psi_{T_i D_a D_b}$ is considered. Here, D_a is the data located on the host EDC and D_b represents the data located on destination EDC where T_i is to be migrated. All the EDC caches must be synchronized by the controller and a task is considered as completed only when all the data updates are complete. The following condition needs to be satisfied.

$$\sum_{1} T <= T_i = T_i \times \psi_{T_i D_a D_b} \tag{32}$$

6 EVALUATION AND DISCUSSION

The proposed approach is analyzed on the basis of two case studies which are discussed as below.

6.1 Case Study I: Resource cooperation

A case study for description of resource cooperation is shown in Fig. 5. Let us consider four EDCs located at different geo-distributed locations. Let us say, that CSP_1 allocates resources from (EDC_1 and EDC_2), CSP_2 allocates resources from (EDC_1 and EDC_3), CSP_3 allocates resources from (EDC_2 and EDC_3), and CSP_4 allocates resources from (EDC_3 and EDC_4). Each EDC is allocated on the basis of the level of utilization of their resources. Level of utilization can be calculated using Eq. (8). In the above context, the level of utilization for all EDCs is considered as 90%. The resources are allocated from these EDCs with respect to their threshold level of utilization (U_{thr}). In this case, the order of the level of utilization for each edge-DCs; ($EDC_1 > EDC_4 > EDC_3 > EDC_2$) shows that EDC_1 and DC_4 has achieved their threshold level of utilization. The vehicle (V_1) sends a request for resources (R_1) to EDC_1 connected to CSP_1. However, EDC_1 has already achieved threshold level of utilization. So, vehicle (V_1) is allocated required resources from EDC_2 using remote resource allocation from same CSP. The vehicle (V_2) and (V_3) are served resources from EDC_2 and EDC_3, respectively using local resource allocation. However, in case of vehicle (V_4), the resources are provisioned from EDC_2 using remote resource allocation from a different CSP.

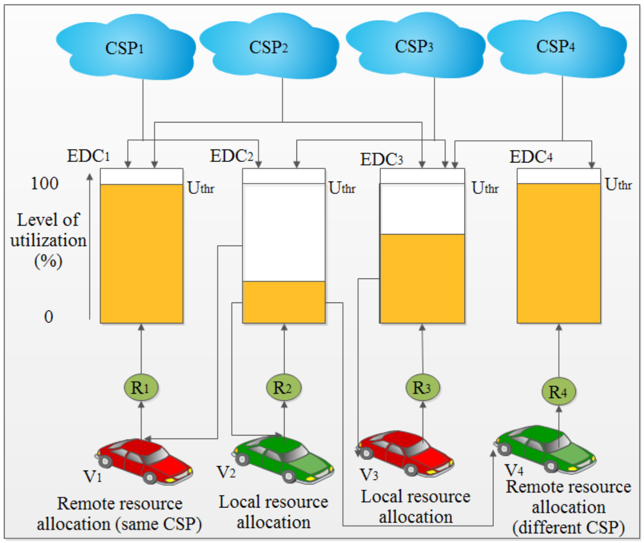


Fig. 5: Case study I

6.2 Case Study II: Resource migration

A case study for resource migration scheme is illustrated in Fig. 6. A vehicular user (V) located at loc(x,y) send a request for resources R_m to k CSPs. Using the proposed scheme, EDC_1 with highest utility (0.891) is selected to serve the request. Now, to show the effectiveness of the proposed migration scheme, three cases have been considered. V may reach three possible directions after time t as shown in Fig. 6. If V reaches location $loc(x_1, y_1)$, then the EDC_1 with highest utility (0.921) is retained to serve the request. However, if V reaches location $loc(x_2, y_2)$, the resources are migrated to EDC_2 with highest utility (0.923). Similarly, if V reaches location $loc(x_3, y_3)$, the resources are migrated to EDC_3 with highest utility (0.926).

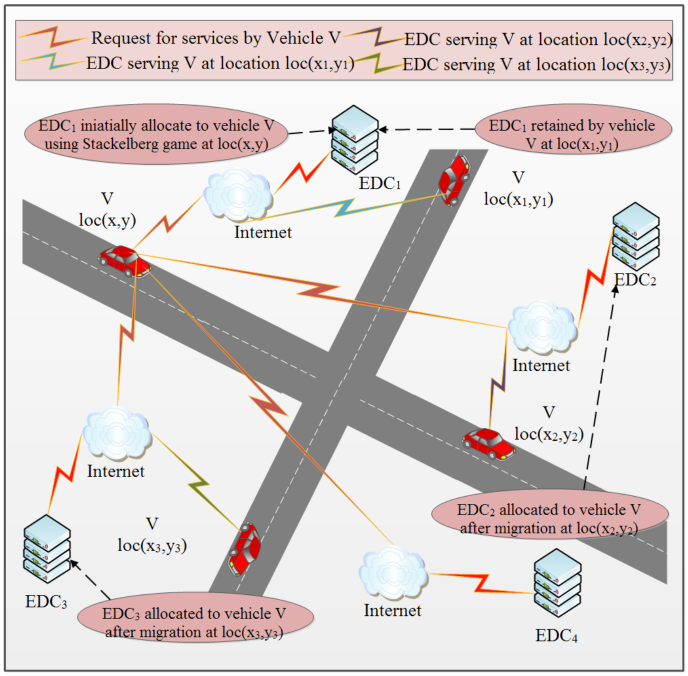


Fig. 6: Case study II

Table (I)-(IV) depicts the parameters (utility of V, distance of V from EDC, renewable energy availability with EDCs) used to select the EDC from various possible locations.

Table I

loc(x,y)	**Utility**	**Distance (m)**	**Energy (kWh)**
EDC 1	0.891	800	5
EDC 2	0.782	1400	4
EDC 3	0.813	900	5
EDC 4	0.421	1800	Nil

Table II

loc(x_1, y_1)	**Utility**	**Distance (m)**	**Energy (kWh)**
EDC 1	0.921	400	5
EDC 2	0.882	700	4
EDC 3	0.619	1900	5
EDC 4	0.221	1700	Nil

Table III

loc(x_2, y_2)	**Utility**	**Distance (m)**	**Energy (kWh)**
EDC 1	0.804	1100	5
EDC 2	0.923	700	5
EDC 3	0.527	1400	4
EDC 4	0.519	600	Nil

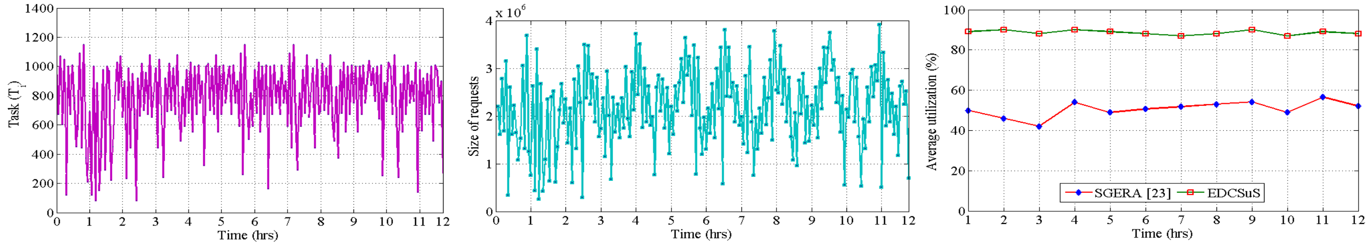
Table IV

loc(x_3, y_3)	**Utility**	**Distance (m)**	**Energy (kWh)**
EDC 1	0.590	1800	4
EDC 2	0.582	1900	4
EDC 3	0.926	500	5
EDC 4	0.492	900	Nil

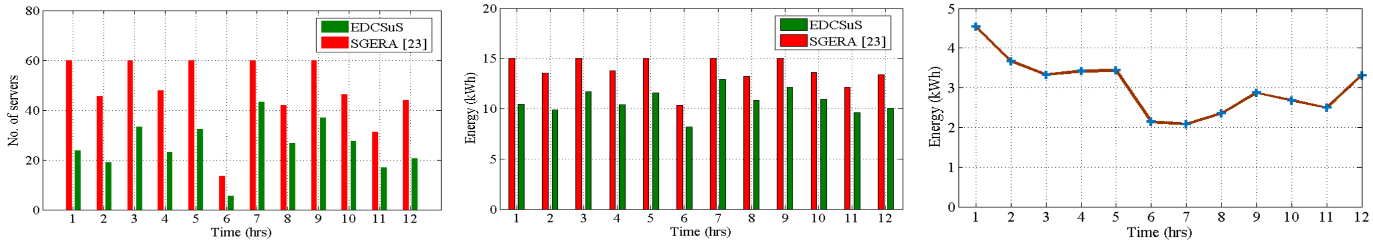
6.3 Evaluation results

This section investigates the impact of EDCSuS on the different performance metrics such as energy consumption, resource utilization, energy cost, SLA violations, and latency.

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/TSUSC.2019.2907110, IEEE Transactions on Sustainable Computing



(a) Tasks to be allocated (b) Size of tasks (c) Average level of utilization



(d) Number of servers provisioned by EDC (e) Mapping of generation-demand of energy (f) Energy saved

Fig. 7: Evaluation results

EDCSuS framework is implemented on EdgeCloudSim [26], which provides an edge environment for simulation and modeling. The proposed scheme is compared with an existing proposal for Stackelberg game for energy aware resource allocation (SGERA) for cloud data centers in [23]. The experiments were performed repeatedly 20 times for a 12-hr simulation period. The configurations for servers and Virtual machine (VM) used to perform extensive simulations are shown in Table 2. To evaluate the proposed scheme, the Google workload traces were considered [27]. The renewable energy generated by PV panels and wind turbine were taken from [10]. The threshold level of utilization is fixed at 90%. The energy consumption of idle server is considered as 50% of maximum power consumption of a server.

TABLE 2: Server and VM Configurations

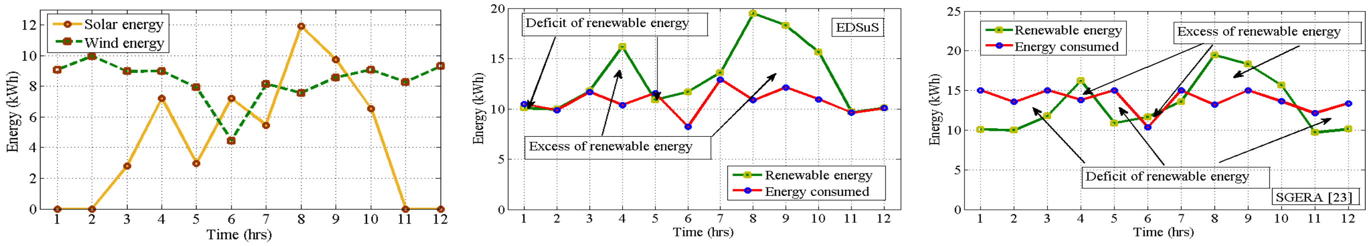
Server Configuration					VM Configuration		
Server type	CPU	Memory (GB)	E^p_{id} (kW)	E^p_{mx} (kW)	VM type	CPU MIPS	Memory (GB)
1	4 cores	64	100	150	1	2	128
2	8 cores	128	120	200	2	4	256
3	16 core	256	150	250	3	8	512

Using EDCSuS, vehicle request are allocated the required resources from EDCs having sufficient amount of renewable energy. Fig. 7(a) shows the task requests considered from Google workload traces [27]. The size of these task are shown in Fig. 7(b). The optimal resource utilization scheme helps EDCs to reduce the number of servers that are provisioned for handling the incoming tasks. Using EDCSuS framework, the average level of utilization achieved is shown in Fig. 7(c). It shows that the utilization level of resources is maintained closer to 90%, which is far more than the existing scheme, SGERA [23]. In this way, the number of servers provisioned to serve the requests of vehicles also reduce. Fig. 7(d) shows the number of servers provisioned by EDC, which are less than SGERA [23]. Hence, the energy consumed by the servers is also reduced. The energy consumption of EDCs to serve the requests of vehicular users is shown in Fig. 7(e), which is again lower in contrast to SGERA. Therefore, it may be concluded that EDCSuS helps to save a significant amount of energy and sustain the same using RES. The large amount of energy is saved by using EDCSuS is shown in Fig. 7(f).

The renewable energy generated by PV panels and wind turbine is shown in Fig. 8(a) [10]. If we use the existing scheme SGERA [23], then the EDCs would have been in deficit or excess of energy at various time-slots as shown in Fig. 8(b). However, using EDCSuS, the mapping of energy consumed and generated by RES is shown in Fig. 8(c). The result depicts that only a negligible deficit of renewable energy at 0100 hrs and 0500 hrs. Hence, it is clearly evident from the results that the energy consumption of EDCs serving the requests of vehicles is sustained using energy generated by RES. Moreover, Case study I and Table I clearly depicts that the EDC having sufficient renewable energy is selected to serve the request of vehicular users. The EDC_1 is selected at loc(x,y), EDC_1 is selected at $loc(x_1, y_1)$, EDC_2 is selected at $loc(x_2, y_2)$, and EDC_3 is selected at $loc(x_3, y_3)$. At $loc(x_2, y_2)$, the EDC_4 is located at a shorter distance as compared to the EDC_2 which is selected. This is due to deficit of renewable energy at EDC_4. Hence, the objective of maximizing the RES sustainability is achieved successfully.

Now, moving on to another objective function, i.e., energy cost, Fig. 8(d) shows the energy cost incurred for handling the allocated tasks. It is quite evident that the energy cost incurred using EDCSuS is far less than in contrast to SGERA. Therefore, the objective of minimizing energy cost is successfully achieved. The final objective function relates to SLA violations and QoS enhancement. Using EDCSuS, the SLA violations are reduced to a great extent. Fig. 8(e) shows the SLA violations witnessed with respect to tasks allocated. Moreover, the proposed scheme maintains a lower latency level with respect to an increase in velocity of the vehicles as shown in Fig. 8(f). The proposed energy-aware flow management scheme using SDN architecture provides dynamic capabilities to enhance the route configurations,

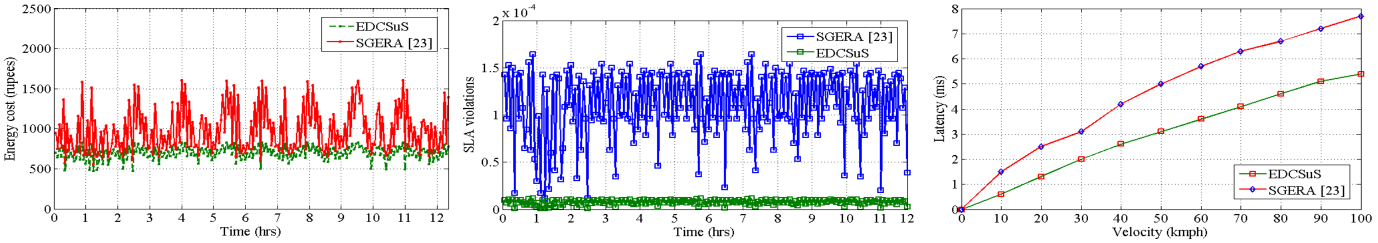
This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSUSC.2019.2907110, IEEE Transactions on Sustainable Computing



(a) Renewable energy generated

(b) Deficit/excess of energy: SGERA [23]

(c) Deficit or excess of energy: EDCSuS



(d) Cost saved

(e) SLA violations

(f) Latency

Fig. 8: Energy mapping, SLA violations, Latency results

thereby ending up in lower latency. Moreover, provisioning tasks closer to the location of vehicles also reduced latency to a great extent. therefore, the final objective of maximal SLA adherence and QoS enhancement is also achieved.

7 CONCLUSION

In this paper, EDCSuS: a software defined framework for providing sustainable EDCs as a service to vehicles for handling their service requests is proposed. EDCSuS handles multiple objectives such as-maximizing SLA adherence, minimizing energy costs and maximizing RES sustainability. To achieve these objectives, a multi-objective optimization problem is formulated, which is solved using different algorithms. Initially, incoming task requests are forwarded using energy-aware flow path selected by OF controller. After this, the task request from vehicles are provisioned to an EDC selected using multi-leader multi-follower Stackelberg game. In order to improve resource utilization and minimize energy consumption, a resource sharing and utilization scheme is proposed. Finally, to provision services closer to the location of vehicles, a service migration scheme is proposed to handle the mobility of vehicles, which is followed by a caching scheme designed to reduce link breakage. The proposed scheme has been evaluated using extensive simulations. The results obtained clearly shows depicts the superiority of EDCSuS in contrast to existing SGERA scheme. The results show that EDCSuS maximizes the latency and RES sustainability, minimizes SLA violations and energy cost and achieve better resource utilization in contrast to existing SGERA scheme.

In our future work, the proposed framework would be compared with more existing variants. Moreover, blockchain technology would be incorporated to enhance the data integrity while migrations between EDCs.

REFERENCES

[1] N. Kumar, S. Zeadally, N. Chilamkurti, and A. Vinel, "Performance analysis of bayesian coalition game-based energy-aware virtual machine migration in vehicular mobile cloud," *IEEE Network*, vol. 29, no. 2, pp. 62–69, Mar 2015.

[2] M. K. Arjmandi, "5G overview: Key technologies," *Opportunities in 5G Networks: A Research and Development Perspective*, p. 19, 2016.

[3] N. D. Han, Y. Chung, and M. Jo, "Green data centers for cloud-assisted mobile ad hoc networks in 5G," *IEEE Network*, vol. 29, no. 2, pp. 70–76, March 2015.

[4] G. S. Aujla and N. Kumar, "Mensus: An efficient scheme for energy management with sustainability of cloud data centers in edge–cloud environment," *Future Generation Computer Systems*, vol. 86, pp. 1279 – 1300, 2018.

[5] G. S. Aujla, N. Kumar, A. Y. Zomaya, and R. Ranjan, "Optimal decision making for big data processing at edge-cloud environment: An sdn perspective," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 2, pp. 778–789, Feb 2018.

[6] K. Kaur, S. Garg, G. S. Aujla, N. Kumar, J. J. P. C. Rodrigues, and M. Guizani, "Edge computing in the industrial internet of things environment: Software-defined-networks-based edge-cloud interplay," *IEEE Communications Magazine*, vol. 56, no. 2, pp. 44–51, Feb 2018.

[7] K. J. Kerpez, J. M. Cioffi, G. Ginis, M. Goldburg, S. Galli, and P. Silverman, "Software-defined access networks," *IEEE Communications Magazine*, vol. 52, no. 9, pp. 152–159, Sep 2014.

[8] G. S. Aujla, R. Chaudhary, N. Kumar, J. J. P. C. Rodrigues, and A. Vinel, "Data offloading in 5G-enabled software-defined vehicular networks: A stackelberg game-based approach," *IEEE Communication Magazine*, vol. 55, no. 8, pp. 100–108, Aug 2017.

[9] T. Chen, Y. Zhang, X. Wang, and G. B. Giannakis, "Robust workload and energy management for sustainable data centers," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 3, pp. 651–664, March 2016.

[10] Y. Guo, Y. Gong, Y. Fang, P. P. Khargonekar, and X. Geng, "Energy and network aware workload management for sustainable data centers with thermal storage," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 8, pp. 2030–2042, Aug 2014.

[11] M. Li, P. Si, and Y. Zhang, "Delay-tolerant data traffic to software-defined vehicular networks with mobile edge computing in smart city," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 10, pp. 9073–9086, Oct 2018.

[12] W. Li, T. Yang, F. C. Delicato, P. F. Pires, Z. Tari, S. U. Khan, and A. Y. Zomaya, "On enabling sustainable edge computing with renewable energy resources," *IEEE Communications Magazine*, vol. 56, no. 5, pp. 94–101, May 2018.

[13] D. Puthal, M. S. Obaidat, P. Nanda, M. Prasad, S. P. Mohanty, and A. Y. Zomaya, "Secure and sustainable load balancing of edge data centers in fog computing," *IEEE Communications Magazine*, vol. 56, no. 5, pp. 60–65, May 2018.

[14] S. K. Mishra, D. Puthal, J. J. P. C. Rodrigues, B. Sahoo, and

E. Dutkiewicz, "Sustainable service allocation using a metaheuristic technique in a fog server for industrial applications," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 10, pp. 4497–4506, Oct 2018.

[15] K. Kaur, T. Dhand, N. Kumar, and S. Zeadally, "Container-as-a-service at the edge: Trade-off between energy efficiency and service availability at fog nano data centers," *IEEE Wireless Communications*, vol. 24, no. 3, pp. 48–56, June 2017.

[16] N. Tziritas, M. Koziri, A. Bachtsevani, T. Loukopoulos, G. Stamoulis, S. U. Khan, and C. Xu, "Data replication and virtual machine migrations to mitigate network overhead in edge computing systems," *IEEE Transactions on Sustainable Computing*, vol. 2, no. 4, pp. 320–332, Oct 2017.

[17] M. Du, K. Wang, X. Liu, S. Guo, and Y. Zhang, "A differential privacy-based query model for sustainable fog data centers," *IEEE Transactions on Sustainable Computing*, 2018, doi: 10.1109/TSUSC.2017.2715038.

[18] S. Wang, X. Huang, Y. Liu, and R. Yu, "Cachinmobile: An energy-efficient users caching scheme for fog computing," in *2016 IEEE/CIC International Conference on Communications in China (ICCC)*, July 2016, pp. 1–6.

[19] P. Borylo, A. Lason, J. Rzasa, A. Szymanski, and A. Jajszczyk, "Energy-aware fog and cloud interplay supported by wide area software defined networking," in *2016 IEEE International Conference on Communications (ICC)*, May 2016, pp. 1–7.

[20] 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, Dec 2016.

[21] S. Chen, S. Irving, and L. Peng, "Operational cost optimization for cloud computing data centers using renewable energy," *IEEE Systems Journal*, vol. 10, no. 4, pp. 1447–1458, Dec 2016.

[22] R. P. Mukund, "Wind and solar power systems," *CRC press*, 1999.

[23] G. S. Aujla, M. Singh, N. Kumar, and A. Zomaya, "Stackelberg game for energy-aware resource allocation to sustain data centers using res," *IEEE Transactions on Cloud Computing*, 2017, doi: 10.1109/TCC.2017.2715817.

[24] G. S. Aujla, A. Jindal, N. Kumar, and M. Singh, "SDN-based data center energy management system using RES and electric vehicles," in *IEEE Global Communications Conference (GLOBECOM)*, Dec 2016, pp. 1–6.

[25] Y. Wang, X. Lin, and M. Pedram, "A stackelberg game-based optimization framework of the smart grid with distributed PV power generations and data centers," *IEEE Transactions on Energy Conversion*, vol. 29, no. 4, pp. 978–987, Dec 2014.

[26] C. Sonmez, A. Ozgovde, and C. Ersoy, "Edgecloudsim: An environment for performance evaluation of edge computing systems," in *Second International Conference on Fog and Mobile Edge Computing (FMEC), 2017*. IEEE, 2017, pp. 39–44.

[27] C. Reiss, J. Wilkes, and J. L. Hellerstein, "Google cluster-usage traces: format+ schema," *Google Inc., White Paper*, pp. 1–14, 2011.

Gagangeet Singh Aujla (S'15, M'18) is working as an Associate Professor in Computer Science and Engineering Department, Chandigarh University, Mohali, Punjab, India. He received his Ph.D. from Thapar Institute of Engineering and Technology, Patiala, Punjab, India in 2018. He received the B.Tech and M.tech. degrees in Computer Science and Engineering from Punjab Technical University, Jalandhar, Punjab, India, in 2003 and 2013, respectively. He is recipient of 2018 IEEE TCSC outstanding dissertation award at Guangzhou China. He is also a part of Indo-Austria Research project. Some of his research findings are published in top cited journals such as IEEE TKDE, IEEE TII, IEEE TCC, IEEE Communication Magazine, IEEE Network, IEEE CE Magazine, FGCS, Computer Networks., Information Science and JPDC and top-tier conferences (ACM MobiHoc, IEEE Globecom, IEEE ICC, IEEE WiMob, etc). He is member of the IEEE and ACM.

Neeraj Kumar (M'17, SM'17) received his Ph.D. in CSE from Shri Mata Vaishno Devi University, Katra (J & K), India in 2009, and was a postdoctoral research fellow in Coventry University, Coventry, UK. He is working as an Associate Professor in the Department of Computer Science and Engineering, Thapar Institute of Engineering and Technology (Deemed to be University), Patiala (Pb.), India. He has published more than 200 technical research papers in top cited journals such as IEEE TKDE, IEEE TIE, IEEE TDSC, IEEE TITS, IEEE TCE, IEEE TII, IEEE TVT, IEEE ITS, IEEE SG, IEEE Netw., IEEE Comm., IEEE WC, IEEE IoTJ, IEEE SJ, Computer Networks, Information sciences, FGCS, JNCA, JPDC and ComCom. He has guided many research scholars leading to Ph.D. and M.E./M.Tech. His research is supported by funding from UGC, DST, CSIR, and TCS. He is an Associate Technical Editor of IEEE *Communication Magazine*. He is an Associate Editor of *IJCS*, Wiley, *JNCA*, Elsevier, and *Security & Communication*, Wiley. He is senior member of the IEEE.

Sahil Garg (S'16) is working as a Post Doctrol Research Fellow at École de Technologie Supérieure, Université du Québec, Montréal, Canada. He received the Ph.D. degree from Thapar Institute of Engineering and Technology, Patiala (Punjab), India in 2018. He received the B.Tech Degree from Maharishi Markandeshwar University, Mullana, Ambala, India, in 2012, and M.Tech degree from Punjab Technical University, Jalandhar, India, in 2014, both in computer science and engineering. Some of his research articles are published in top-tier journals such as-IEEE Network, IEEE TII, IEEE ComMag, IEEE IoTJ, IEEE CEMag, FGCS and CAEE. He is member of the IEEE and ACM. He received the Best Paper Award in IEEE ICC 2018.

Kuljeet Kaur (S'15) is working as a Post Doctrol Research Fellow at École de Technologie Supérieure, Université du Québec, Montréal, Canada. She received the Ph.D. degree from Thapar Institute of Engineering and Technology, Patiala (Punjab), India in 2018. She received B. Tech. degree in computer science and M.E. degree in Information Security. Her main research interests include radio-frequency identification, cloud computing, and vehicle-to-grid. She has secured a number of research articles in top tier journals (IEEE WCM, IEEE TII, IEEE TVT, IEEE TSG, IEEE SJ, IEEE CM, IEEE TPDS, IEEE PS, Springer PPNA, etc.) of repute. He received the Best Paper Award in IEEE ICC 2018.

Rajiv Ranjan is a Full professor in Computing Science at Newcastle University, United Kingdom. Before moving to Newcastle University, he was Julius Fellow (2013-2015), Senior Research Scientist and Project Leader in the Digital Productivity and Services Flagship of Commonwealth Scientific and Industrial Research Organization (CSIRO C Australian Government's Premier Research Agency). Prior to that he was a Senior Research Associate (Lecturer level B) in the School of Computer Science and Engineering, University of New South Wales (UNSW). Dr. Ranjan has a PhD (2009) from the department of Computer Science and Software Engineering, the University of Melbourne.

IEEE Transactions On Sustainable Computing

Ref. No.: TSUSC-2018-06-0072

Title: EDCSuS: Sustainable Edge Data Centers as a Service in SDN-enabled Vehicular Environment

Authors: Gagangeet Singh Aujla, Neeraj Kumar, Sahil Garg, Kuljeet Kaur, and Rajiv Ranjan

EDITORS COMMENTS

Dear Editors and reviewers

Thank you for your useful comments and feedback on our paper which helped us to improve its presentation and quality. We have carefully addressed all of your comments in the revised manuscript. We hope that you will be satisfied with the response provided by us.

Sincerely,
The authors.

Response to Associate Editor's comments

Comment: I urge authors to make sure all comments are properly addressed, especially those related to ambiguity in algorithms and results.

Reply: The authors thanks the associate editor for the positive response to the manuscript. The manuscript has been reworked thoroughly on the basis of the comments received from the reviewers. The revised manuscript contains all the required algorithms in contrast to the sketchy figures in the earlier version. Moreover, the results have been represented in a better way and they have been compared with existing proposals also.

Reviewer 1's comments

This work proposed a renewable energy aware resource allocation scheme. It is to provide low latency cloud services to vehicular users.

RESPONSE: The authors want to thanks the reviewer as the comments provided have allowed us to transform the entire manuscript to improve the quality and presentation. The authors have reworked on each and every section, re-framed them and added suitable and significant content to improve the quality of the manuscript.

Comment 1: What is the purpose of Fig. 1? The materials in Fig 1 are not explained in the content.
Response: The purpose of Fig. 1 was to depict some energy related facts for data centers. However, Fig. 1 has been removed from the revised manuscript as it was creating confusion.

Comment 2: Where are the Figure 1 from? Specify the data sources.
Response: Fig. 1 was misleading the flow of proposed work, therefore it has been removed from the revised manuscript.

Comment 3: What is EDCSuS? This term simply appears in the title once and is never used again in the content.
Response: In this paper, EDCSuS: Sustainable EDC as a service framework in software defined vehicular environment is proposed. In this framework, EDCs connected to renewable energy sources are geographically distributed in a smart city instead of a centralized DC to handle vehicular applications closer to their location. EDCSuS framework is divided into four modules which work in tandem to achieve the multi-objective solution. Firstly, a software defined controller handles the incoming vehicular requests and suggest an optimal flow path for the same. Secondly, a multi-leader multi-follower Stackelberg game is presented for renewable energy-aware resource allocation to 5G-enabled vehicles. Thirdly, to improve the resource utilization, a cooperative resource sharing scheme is designed, thereby minimizing the energy consumption of servers located in the EDCs. Lastly, a mutual information sharing and caching scheme is presented to avert excessive energy consumption for retracing the lost link due to mobility of the vehicles.

As per the suggestion of the reviewer, the role of EDCSuS has been highlighted at various occasions throughout the manuscript.

Comment 4: Figure 2 is a schematic layout without sufficient scientific definitions. This figure does not reflect the actual considerations about the vehicular users, cloud infrastructures and energy sources. Please consider re-designing the figure.
Response: Figure 2 (Fig. 1 in revised manuscript) is simple a layout of edge data centers in the proposed framework. The same has been shifted to the introduction part (Section 1, page 2). Fig. 2 (which was Figure 4 in previous version) depicts the actual considerations about the vehicular users, cloud infrastructures and energy sources (Section 3, page 4). Moreover, Fig. 3 is also added to show the architecture of renewable energy powered edge data centers (Section 3.2, page 5).

Comment 5: The proposed approach is described too sketchy (Section 4).
Response: The authors agree with the reviewer regarding this aspect. Therefore, the entire manuscript has been transformed. Five algorithms (Algorithm 1-5), mathematical model and problem formulation has been added in the revised manuscript. Algorithm 1 presents the energy aware flow scheduling (Section 5.1, page 7), algorithm 2 depicts Stackelberg game for EDC selection (Section 5.2, page 8), algorithm 3 shows optimal resource utilization (Section 5.3.1, page 9), algorithm 4 presents cooperative resource sharing (Section 5.3.2, page 9) and algorithm 5 depicts inter-EDC migration (Section 5.3.2, page 10) schemes.

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/TSUSC.2019.2907110, IEEE Transactions on Sustainable Computing

Comment 6: The routing algorithm for requests is not presented.
Response: As requested by the reviewer, an energy aware flow scheduling algorithm which selects the optimal flow route has been added in the revised manuscript (Section 5.1, page 7).

Comment 7: The optimization model for energy resource is not presented, not to mention the optimization algorithms.
Response: The authors agree with the reviewer that the optimization model for energy resource is not presented in the paper. Therefore, a problem formulation (Section 4, page 6-7) followed by a multi-objective optimization model (Section 4.3, page 7) has been added in the revised manuscript .

Comment 8: There appears neither optimization procedure nor algorithm proposed in this section towards efficient use of said resources.
Response: The optimization procedure (Section 4, page 6-7) and five different algorithm proposed (Section 5, page 7-10) have been added in the revised manuscript.

Comment 9: Experiments are not convincing. How are these experiments conducted? How are these simulation results obtained? What is the "Without proposed scheme" method?
Response: As suggested by the reviewer, more experiments have been performed and added in the revised manuscript (Figs. 7(a) and 7(b), Figs. 8(d) and 8(e)). The complete detail of experimental setup, simulator and configurations are also provided in the revised manuscript.

The impact of EDCSuS on the different performance metrics such as energy consumption, resource utilization, energy cost, SLA violations, and latency is evaluated using extensive simulations. EDCSuS framework is implemented on EdgeCloudSim [26], which provides an edge environment for simulation and modeling. The proposed scheme is compared with an existing proposal for Stackelberg game for energy aware resource allocation (SGERA) for cloud data centers in [23]. The experiments were performed repeatedly 20 times for a 12-hr simulation period. The configurations for servers and Virtual machine (VM) used to perform extensive simulations are shown in Table 2. To evaluate the proposed scheme, the Google workload traces were considered [10]. The renewable energy generated by PV panels and wind turbine were taken from [27].

The "Without proposed scheme" method refereed to an existing proposal, which has now been specifically mentioned as SGERA [23] in the revised manuscript.

Comment 10: There is no comparison with other works.
Response: The comparison of the proposed scheme with an existing scheme of its category is presented in the revised manuscript. The performance metrics, average utilization level (Fig. 7(c), page 12), number of servers provisioned (Fig. 7(d), page 12), energy consumption (Fig. 7(e), page 12), mapping of energy consumption with RES generated (Fig. 8(b-c), page 13), energy cost (Fig. 8(d), page 13), SLA violations (Fig. 8(e), page 13) and latency (Fig. 8(f), page 13) have been compared with existing scheme SGERA [23] in the revised manuscript.

Reviewer 2's comments

Response to minor comments

Authors in the manuscript titled: "EDCSuS: Sustainable Edge Data Centers as a Service in SDN-enabled Vehicular Environment" propose a multi-follower Stackelberg game for renewable energy aware resource allocation in order to sustain geo-distributed edge-DCs. Furthermore, they proposed a cooperative resource sharing and migration schema in order to avoid energy consumption for re-searching and re-routing. The manuscript is well written and well organized, it fits well the topics of the Special Issue, also abstract and key words are appropriate. Number of references are adequate, however a part of them are from 2015 or earlier. In order to improve the work I suggest to:

RESPONSE: The authors want to thanks the reviewer as the comments provided have allowed us to transform the entire manuscript to improve the quality and presentation. The authors have reworked on each and every section, re-framed them and added suitable and significant content to improve the quality of the manuscript.

Comment 1: Page 2 row 53. The table analizes works from 8 to 14, but into the text you refer till 13, please check it.
Response: The table has been revised to overcome the issue raised by the reviewer. Additional proposals have been added in the table in the revised manuscript (Table 1, page 3).

Comment 2: Furthermore, into the paper could be interesting to have more details about of these works. Please, add a specific section in which you discuss them.
Response: A separate section for related work has been added in the revised manuscript which describes the details about of the works (Section 2, page 2-3). Moreover, an updated and detailed comparative table has also been added in revised manuscript (Table 1, page 3).

Comment 3: Page 3 row 36 there is a double ',' Please, remove "Such a platform could provide an energy-efficient, resilient, scalable, flexible,, and dynamic network for sustainable DCs [18]."
Response: The issue highlighted by the reviewer has been corrected. Moreover, the entire manuscript has been recheck to remove all such occurrences.

Comment 4: Page 4 row 6 the acronym RES is never explained
Response: The issue highlighted by the reviewer has been corrected in the revised manuscript.

Comment 5: Change the background color to white of figures 5, 6 and 7
Response: All the concerned figures have been removed in the revised manuscript and they have been replaced by separate algorithms (Section 5, page 7-10).

Comment 6: Enlarge the size of Figure 10
Response: The text size in the concerned figure 10 (Figure 6 in revised manuscript) has been enlarged in the revised manuscript (Fig. 6, page 11).

Comment 7: Add more updated references. Ref 14 authors are missing, please verify it.
Response: The references have been updated as suggested by the reviewer. More updated and latest references have been added in the revised manuscript.

Reviewer 3's comments

Response to minor comments

RESPONSE: The authors want to thanks the reviewer as the comments provided have allowed us to transform the entire manuscript to improve the quality and presentation. The authors have reworked on each and every section, re-framed them and added suitable and significant content to improve the quality of the manuscript.

Comment 1: The topics proposed in this paper are very interesting, but I would like to suggest you to improve the abstract of the paper, making it more concise and clear.
Response: The abstract has been improved to make it more concise and clear in the revised manuscript.

Comment 2: In order to better understand where the present work is placed respect to the others just recently done I suggest you to reference more recent papers in the state of the art.
Response: As per the suggestion of the reviewer, more recent papers have been added in the the state of the art. A separate section for related work has been added in the revised manuscript which describes the details about of the works (Section 2, page 2-3). Moreover, an updated and detailed comparative table has also been added in revised manuscript (Table 1, page 3).

Comment 3: I would like to suggest you to introduce a section where you analyze the proposed solution in terms of pros/cons and compare with the existing ones.
Response: A table has been added in the revised manuscript to analyze the proposed solution in terms of pros/cons in comparison of the existing solutions (Table 1, page 3).