Received: Added at production | Revised: Added at production | Accepted: Added at production

DOI: xxx/xxxx

ARTICLE TYPE

T-RESIN: Throughput-Aware Dynamic Resource Orchestration for IoE-Enabled Software-Defined Edge Networks

Lalita Agrawal[1] | **Ayan Mondal, Senior Member, IEEE**[1] | **Mohammad S. Obaidat, Life Fellow, IEEE**[2]

[1]Department of Computer Science and Engineering, Indian Institute of Technology Indore, Madhya Pradesh, India

[2]Distinguished Professor, The King Abdullah II School of Information Technology, The University of Jordan, Amman 11942, Jordan and School of Computer and Communication Engineering, University of Science and Technology Beijing, Beijing 100083, China, Department of Computational Intelligence, School of Computing, SRM University, SRM Nagar, Kattankulathur 603203, TN, India, and School of Engineering, The Amity University, Noida, UP 201301, India. Email: m.s.obaidat@ieee.org

Correspondence

Corresponding author: Ayan Mondal. Email: ayanm@iiti.ac.in

Abstract

In this work, we address the problem of resource allocation in Internet of everything (IoE)-enabled software-defined edge networks. In the existing literature, the researchers considered optimizing the performance of the software-defined networking (SDN) platform using a single-tier architecture, where the Internet of Things (IoT) devices are in the same tier. However, with the advent of edge computing, we can explore the two-tier architecture of edge networks - local tier and edge tier - in the presence of SDN, which has not been explored. Hence, we propose an evolutionary game-based resource allocation scheme for software-defined edge networks. Additionally, we aim to optimize the QoS of the edge-based IoE services while optimizing the throughput of the system. The IoT devices use the proposed scheme in the local tier to identify the optimized mapping to the SDN switches. On the other hand, in the edge tier, the proposed scheme aims to optimize the throughput while allocating the IoE service to the optimal subset of edge devices. We evaluated the performance of the proposed scheme using the Python3-based Mininet platform in the presence of Ryu Controller and Open vSwitches. Ryu is an element-based software defined networking framework, which offers software elements with well described API that enables developers to produce new network control functions. We observed that by using T-RESIN, the network throughput increases by 27.98-31.84% than using the existing schemes.

KEYWORDS

Software-Defined Edge Networks,Internet of Everything, Throughput, Evolutionary Game

1 | INTRODUCTION

With the advent of Internet of Things (IoT), the number of devices increased significantly. The International Data Corporation (IDC) estimates that the number of IoT devices will increase to 55.7 billion and these IoT devices will generate almost 80 zettabytes of data every year by 2025[1]. Moreover, Internet of Everything (IoE) considers not only the collection of raw data and interconnecting things, but also the decision-making process and people[2]. Hence, IoE is a superset of the Internet of Things (IoT) by providing multiple forms of communication to IoT. In IoE, the process analyses the real-time data and decides the activity to make IoE an automated intelligence system. We envision that the processing of these huge amount of data by the cloud, i.e., the centralized architecture, will incur high latency and low bandwidth utilization in provisioning IoE-based services[3,4]. Hence, we plan to use the distributed edge network for processing the IoT-based applications[5,6,7]. We observe that the edge devices are vendor specific and cannot be integrated seamlessly. Therefore, we introduce software-defined networking-enabled edge architecture, named as Software-Defined Edge Networks (SDENs). Similar to the traditional SDN architecture, SDEN also separates the traditional edge networks in two plane — control and data edge planes[8,9,10,11].

In the existing literature, researchers focused on the task offloading in the edge networks and resource allocation in SDN. However, the task offloading in the SDEN remains a challenging task due to several factors such as limited bandwidth and flow space of the SDN edge switches and processing capacity of the edge devices. Additionally, we wish to highlight that the

Abbreviations: SDEN, Software-Defined Edge Networks; IoE, Internet of Everything; SDN, Software-Defined Networks.

data generated by IoT devices are heterogeneous in terms of the volume and variety. Therefore, we argue that there is a need for resource allocation scheme for SDENs in the presence of heterogeneous IoT devices and services, while maximizing the bandwidth utilization of the edge networks.

Motivation Scenario: Edge computing platform follows a distributed architecture and provides resource utilization and computation on edge nodes. Edge computing has many advantages such as less computational latency, networking load, and low response time. We aim to allocate optimal resources to the processes of the IoT devices based on their requirements while ensuring a high network bandwidth utilization.

In this work, we propose a throughput-aware dynamic resource allocation scheme, named T-RESIN, for SDENs. We consider a three-tier architecture, as shown in Figure 1. To model the interactions between the IoT devices and the Open vSwitches, and Open vSwitches and the edge computing nodes, i.e., through access and edge tiers, respectively, we use evolutionary game theoretic approach separately. Initially, the IoT devices interact with the Ryu SDN controller to decide the association among the IoT devices and the Open vSwitches. Thereafter, the Ryu SDN controller decides the association among the Open vSwitches and the edge computing nodes. We consider that the T-RESIN algorithms are to be executed by the Ryu SDN controller and ensure maximizing the overall throughput of the network. In summary, the specific contributions of this paper are as follow:

1. We propose T-RESIN, a throughput-aware dynamic resource allocation scheme, for IoE-enabled SDENs.
2. The interactions among the entities — IoT devices, Ryu SDN controller/Open vSwitches, and the edge computing nodes — are modelled using evolutionary game-theoretic approach.
3. We evaluated the performance of the proposed scheme, T-RESIN, theoretically and evaluated the performance of T-RESIN while comparing with the existing schemes.

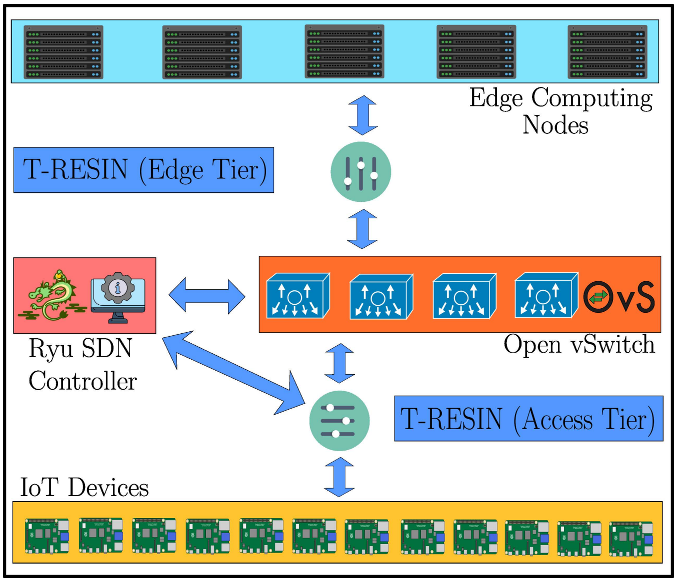

FIGURE 1 Schematic Diagram of Software-Defined Edge Networks

2 | RELATED WORKS

In this literature survey, we focus on two broad areas. Firstly, we study the internet of everything, how it differs from IoT, and the components and enabling techniques for IoE. Secondly, we survey existing work related to resource allocation techniques for IoT and IoE; and bandwidth distribution among SDN switches. So, we can analyze the network's overall performance and quality of service (QoS) using different approaches.

Enabling IoE: Langleya *et al.*[12] studied the business model for IoE at three-level that are micro, meso, and macro. The authors described the potential impact of IoE on current business models and the value creation processes by firms and their customers. Liu *et al.*[13] presented the Unmanned aerial vehicle (UAV) enabled IoE to enhance the critical aspect of IoE that are scalability, intelligence, and diversity. Liyanage *et al.*[14] presented Multi-Access Edge Computing (MEC) enabled 5G wireless system and explained the integration of SDN, Network Function Virtualization (NFV) in MEC enabled IoT network. Singh and Gill[15] surveyed the importance of edge computing techniques for supporting AI-enabled IoE platform. Iannacci *et al.*[16] focused on the hardware perspective of IoT, IoE, 5G, and tactile internet features and worked on Micro Electro Mechanical Systems for the application of energy harvesting (EH-MEMS) and Radio Frequency (RF-MEMS).

Resource Allocation for IoT and IoE: Misra *et al.*[17] proposed pricing based resource allocation model, FogPrime, and clustered the fog nodes. Authors used dynamic coalition-formation game approach for resource allocation locally within a cluster. Gurung and Mondal[18] proposed a multi-hop data transmission scheme for SDEN. Misra *et al.*[19] studied the data broadcasting approach for SDN while incorporating the SDN-enabled edge and core networks. In another work, Yuan *et al.*[20] proposed the network management scheme for software-defined networking in terrestrial and non-terrestrial networks. Mondal and Misra[21] proposed the FlowMan scheme for heterogeneous data flow management. Authors used the Nash bargaining game approach to achieve a sub-optimal problem further bounded in 0,1 knapsack problem and ensured a good quality of service. Jain *et al.*[22] proposed meta-heuristic with blockchain based resource allocation technique (MWBA-RAT) for IoE infrastructure. Authors used 6G enabled blockchain technique for management, and observation of shared resources. Cybertwin has been used for virtual representation of IoE end nodes (cyberspace) so its architecture has cyber-twin along with edge nodes. Their proposed method reduced the power consumption and communication cost for optimal resource allocation. For the same purpose, Manogaran *et al.*[23] analysed resource utilization numerically for optimal solution and used two algorithms that are profitable resource allocation and optimal neighbour replacement. In another work, Deb *et al.*[24] proposed a distributed load management scheme using edge platforms for IoT-enabled smart grid environment. Sami *et al.*[25] presented IScaler for dynamic resource allocation in the context of mobile edge computing with the vision of 6G and IoE services. The authors have used the Deep Reinforcement Learning approach to overcome the challenges of the proposed method.

Synthesis: In the existing literature, most researchers focused on various resource allocation methods in contrast to IoT networks. Few researchers have proposed IoE based resource allocation technique. We present a comparative analysis of the existing resource allocation approach based on software-defined IoE network features in Table 1. No work has been done for dynamic resource allocation for the IoE network using the evolutionary game theoretic approach. Here, we present an evolutionary game-based dynamic resource allocation technique for IoE-enabled SD-edge networks. Additionally, we optimize bandwidth utilization for SDN switches.

TABLE 1 Comparison of Existing Resource Allocation Methods

Existing Work for Resource Allocation	IoE-Enabled SDEN Features				
	Heterogeneous	Dynamic Configuration	Fog	Cloud	IoE Network
Misra *et al.*[17]	✓	✓	✓	✗	✗
Mondal *et al.*[21]	✓	✓	✗	✓	✗
Jain *et al.*[22]	✗	✓	✓	✗	✓
Manogaran *et al.*[23]	✗	✗	✓	✓	✓
Sami *et al.*[25]	✗	✓	✓	✗	✓

3 | SYSTEM MODEL

We consider an SDEN architecture with a single controller, multiple SDN switches, and multiple IoE devices computing edge nodes, as shown in Figure 1. The IoT devices generates the data from the IoE process and applications that are to be processed by the edge nodes[‡]. However, the devices are connected through the SDN switches at the access tier. $\mathcal{N}$ and $\mathcal{S}$ represent the sets of IoE devices and SDN switches, respectively. Bandwidth associated with each SDN switch $s \in \mathcal{S}$ is represented as B_s. Therefore, the total bandwidth $\mathcal{B}$ distributed among all the switches is as follows:

[‡] We clarify the use IoT for the devices and IoE for the associated services/processes.

$$\mathcal{B} = \sum_{s \in \mathcal{S}} B_s \tag{1}$$

Each IoE device $n \in \mathcal{N}$ generates f_n number of data flows and the data generation rate of each data flow i is denoted as d_i, where $i \in (\mathbb{Z}^+ \cap [0, f_n], \forall n)$. Flows are heterogeneous in terms of bandwidth depending on flow type — scalar or multimedia, and the volume generated by flow. We consider that the SDN switches are capable to forward the generated data and support the network requirement of the IoE applications. However, the flows might need to distributed among the multiple SDN switches as these switches have physical limitations in terms of flow rule capacity and bandwidth. Accordingly, each switch needs to satisfy the following constraints.

$$V_s = \sum_{n \in \mathcal{N}} \sum_{i \in \mathbb{Z}^+ \cap [0, f_n]} x_{i,n,s} d_i \leq B_s \tag{2}$$

$$F_s = \sum_{n \in \mathcal{N}} \sum_{i \in \mathbb{Z}^+ \cap [0, f_n]} x_{i,n,s} \leq R_s^{max} \tag{3}$$

where V_s and F_s denote the volume of data and the number of flow-rules to be handled by switch s. We consider that each switch $s \in \mathcal{S}$ is capable of installing maximum R_s^{max} number of flow rules in its ternary content-addressable memory (TCAM) memory. Here, $x_{i,n,s}$ is a binary variable and defines the association among flow i, where $i \in (\mathbb{Z}^+ \cap [0, f_n])$, and switch s and defined as follows:

$$x_{i,n,s} = \begin{cases} 1, & \text{if flow } i \text{ of } n \in \mathcal{N} \text{ is associated switch } s \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

In other words, we consider that the set of P_s process is associated with switch $s \in \mathcal{S}$, where each process $q \in P_s$ requires f_q number of flows. Hence, we get:

$$\sum_{q \in P_s} f_q \equiv \sum_{n \in \mathcal{N}} \sum_{i \in \mathbb{Z}^+ \cap [0, f_n]} x_{i,n,s} \tag{5}$$

In the edge tier, we consider that there is $\mathcal{E}$ set of edge nodes. Each edge node $e \in \mathcal{E}$ has computational and memory capacities of C_e and M_e, respectively. Hence, while allocating the processes to the edge nodes, we need to satisfy the following constraints.

$$\sum_{s \in \mathcal{S}} \sum_{q \in P_s} \sum_{j \in \mathbb{Z}^+ \cap [0, f_q]} y_{j,q,e} m_q \leq M_e \tag{6}$$

$$C_{use}^e = \sum_{s \in \mathcal{S}} \sum_{q \in P_s} \sum_{j \in \mathbb{Z}^+ \cap [0, f_q]} y_{j,q,e} c_p \leq C_e \tag{7}$$

where m_q and c_q represent the required memory and CPU resources for each process $q \in P_s$, respectively. We denote the memory and computational capacities of edge node $e \in \mathcal{E}$ using M_e and C_e, respectively. Here, $y_{j,q,e}$ is a binary variable and is evaluated as follows:

$$y_{j,q,e} = \begin{cases} 1, & \text{if process } q \in P_s \text{ of switch } s \in \mathcal{S} \text{ is allocated to edge node } e \\ 0, & \text{otherwise} \end{cases} \tag{8}$$

We consider that the overall delay for each process q is denoted as λ_q and evaluated as follows:

$$\lambda_q = \lambda_{q,s} + \lambda_{q,e} \tag{9}$$

where $\lambda_{q,s}$ and $\lambda_{q,e}$ represent the incurred delay at the access and edge tiers, respectively. We consider that each IoE process is to be served within a threshold delay λ_q^{th}. Hence, we need to satisfy the following constraint:

$$\lambda_q \leq \lambda_q^{th} \tag{10}$$

We also consider that the edge devices have limited energy, i.e., maximum energy E_e^{max} for each edge device e. Considering that process q requires E_q amount of energy, the edge node e needs to ensure that it satisfies the following constraint for serving process q.

$$\sum_{s\in\mathcal{S}} \sum_{q'\in P_s\cup\{q\}} \sum_{j\in\mathbb{Z}^+\cap[0,f_{q'}]} y_{j,q',e}E_{q'} \leq E_e^{max} \tag{11}$$

Therefore, considering the incoming process q is allocated to edge node e, the remaining amount of energy, E_e^{res}, is represented as follows:

$$E_e^{res} = E_e^{max} - \sum_{s\in\mathcal{S}} \sum_{q'\in P_s\cup\{q\}} \sum_{j\in\mathbb{Z}^+\cap[0,f_{q'}]} y_{j,q',e}E_{q'} \tag{12}$$

Table 2 summarizes the main symbols and their corresponding description which are frequently used in the paper.

TABLE 2 List of Symbols

Symbol	Description
$\mathcal{N}$	Set of IoE devices/things
$\mathcal{S}$	Set of SDN switches
$\mathcal{E}$	Set of Edge nodes
f_n	Set of data flow generated by each IoE device $n \in \mathcal{N}$
B_s	Bandwidth associated with each switch $s \in \mathcal{S}$
F_s	Number of lows associated with switch s
$P_n(t)$	Set of process required by IoE end-device $n \in \mathcal{N}$
d_i	Data generation rate of each flow $i \in (\mathbb{Z}^+ \cap [0, f_n])$
C_e	Computational capacity of edge node $e \in \mathcal{E}$
M_e	Memory capacity of edge node $e \in \mathcal{E}$
P_s	Set of process associated with switch $s \in \mathcal{S}$
R_s^{max}	Maximum no. of flow rules in TCAM of switch $s \in \mathcal{S}$
λ_p	Total delay for each process $p \in P_n(t)$
λ_n	Total processing delay for each $n \in N$ IoE end-device
P_e	Set of process associated with edge node $e \in \mathcal{E}$
E^e	Total energy associated to an edge node $e \in \mathcal{E}$
$y_s(\omega)$	Population share of each switch $s \in \mathcal{S}$
$U_s(\omega)$	Utility function for each switch $s \in \mathcal{S}$
$\dot{y_s}(\omega)$	Replicator dynamics for SDN Switches
$x_e(\phi)$	Population share of each edge node $e \in \mathcal{E}$
$W_e(\phi)$	Utility function for each edge node $e \in \mathcal{E}$
$\dot{x_e}(\phi)$	Replicator dynamics for edge nodes
α, β	Evolutionary control factor
ω	Evolutionary iteration for SDN switches
ϕ	Evolutionary iteration for edge nodes

4 | T-RESIN: THE PROPOSED RESOURCE ORCHESTRATION SCHEME

For modelling the interactions to ensure dynamic resource allocation in SDEN, i.e., a three-layer architecture, we use the evolutionary game theoretic approach[26]. In the subsequent sections, we discuss the use of the evolutionary game-theoretic approach for the proposed scheme, T-RESIN.

4.1 | Justification for Using Evolutionary Game

For optimal resource allocation of IoE processes in SDEN, we rely on Equations (2), (3), (6) and (7). These equations are a function of binary variables $x_{i,n,s}$ and $y_{j,q,e}$ defined in Equations (4) and (8), respectively. Hence, we argue that the problem mentioned above is a *multiple binary integer programming problem*[27] that is mapped to *0-1 knapsack problem*[27,28]. It is an NP-complete problem. Hence, to get a sub-optimal resource allocation in polynomial time, we use an evolutionary game-theoretic approach.

4.2 | Game Formulation

We design the proposed scheme, T-RESIN, as a two-stage game — access and edge tiers. In the access tier, the IoT devices or the IoE processes act as the players and choose the set of forwarding SDN switches, i.e., strategies to forwarding generated data from the processes at the user end to the computing edge nodes, with the help of the SDN controller. The generated data by the IoE processes is considered as the population share at access tier in T-RESIN. Hence, the population share $y_s(\omega)$ of each switch $s \in \mathcal{S}$, where ω is the evolutionary iteration, is defined as follows:

$$y_s(\omega) = \frac{V_s(\omega)}{\sum\limits_{s\in\mathcal{S}} V_s(\omega)} \tag{13}$$

On the other hand, the population share $x_e(\phi)$ of each edge node $e \in \mathcal{E}$ is defined as follows:

$$x_e(\phi) = \frac{C^e_{use}(\phi)}{\sum\limits_{e\in\mathcal{E}} C^e_{use}} \tag{14}$$

where ϕ is the evolutionary iteration for edge nodes. Here, $x_e(\phi)$ signifies the computation power contribution of edge node e. C^e_{use} represents the used computation power of edge node e. We identify the selection of edge nodes for processing the data generated by the IoE devices and forwarded through the SDN switches. Based on the population shares, we define the utility functions of each SDN switch and edge node as given in the subsequent sections.

Utility Function of Each SDN Switch: Utility function $U_s(\omega)$ signifies the fitness function for switch s for evolutionary iteration ω. We consider that $U_s(\omega)$ varies linearly with the population share of switch s. Similarly, with the increase in the processes P_s associated with the switch s, the utility value increases with the increase in the number of flow rules installed in the switch. We define utility function $U_s(\omega)$ for each switch as follows:

$$U_s(\omega) = \frac{y_s(\omega)F_s(\omega)}{R^{max}_s} \tag{15}$$

where $F_s(\omega)$ signifies the number of flows associated with switch s at evolutionary iteration ω.

Therefore, the average payoff $\overline{U}(\omega)$ of the SDN switches is evaluated as follows:

$$\overline{U}(\omega) = \sum_{s\in\mathcal{S}} y_s(\omega)U_s(\omega) \tag{16}$$

Utility Function of Each Edge Node: Utility function $W_e(\phi)$ signifies the fitness function for edge node e. We consider that $W_e(\phi)$ varies linearly with the population share and total energy consumed of edge node e. Additoinally, we consider that $W_e(\phi)$ decreases with the increase of residual energy E^e_{res} of each edge node $e \in \mathcal{E}$. The utility function $W_e(\phi)$ for each edge node is designed as follows:

$$W_e(\phi) = x_e(\phi)\left(1 - \frac{E^e_{res}(\phi)}{E^e}\right) \tag{17}$$

We define the average payoff $\overline{W}(\phi)$ of the edge nodes $\mathcal{E}$ as follows.

$$\overline{W}(\phi) = \sum_{e\in\mathcal{E}} x_e(\phi)W_e(\phi) \tag{18}$$

Replicator Dynamics: As evolutionary games are dynamic and have two mechanisms that are *mutation* and *selection*. The mutation mechanism modifies the characteristics of the population whenever new players come into the population and choose different strategies. The selection mechanism determines the strategy with high fitness and promotes that strategy in the population. In an evolutionary game, players replicate themselves by changing the strategies, called as *replicator dynamics*. Based on the significance of replicator dynamics in T-RESIN, we define replicator dynamics for SDN switches $\dot{y_s}(\omega)$ and edge nodes $\dot{x_e}(\phi)$ as follows:

$$\dot{y_s}(\omega) = \alpha y_s(\omega)\left(U_s(\omega) - \overline{U}(\omega)\right) \tag{19}$$

$$\dot{x_e}(\phi) = \beta x_e(\phi)\left(W_e(\phi) - \overline{W}(\phi)\right) \tag{20}$$

Algorithm 1 Algorithm for SDN switches

INPUTS:
$\mathcal{N}, \mathcal{S}, f_n, F_s, d_i, R_s^{max}, \alpha$

OUTPUT:
$\boldsymbol{y^*}, \boldsymbol{U^*}$

PROCEDURE:
$\omega \leftarrow 0$
Randomly assign each flow $0 \leq i \leq f_n, \forall n$, to switch $s \in \mathcal{S}$
do
    $\omega \leftarrow \omega + 1$
    for Each $s \in \mathcal{S}$ **do**
        Calculate $V_s(\omega)$ using Equation (2)
        Calculate $F_s(\omega)$ using Equation (3)
        Calculate population share $y_s(\omega)$ using Equation (13)
        Calculate utility value $U_s(\omega)$ using Equation (15)
    end for
    Calculate average utility value $\overline{U}(\omega)$ using Equation (16)
    for Each $s \in \mathcal{S}$ **do**
        Calculate replicator dynamic $\dot{y_s}(\omega)$ using Equation (19)
        $y_s(\omega + 1) \leftarrow y_s(\omega) + \dot{y_s}(\omega)$
    end for
while $\dot{y_s}(\omega) \not\approx 0$
$y_s^* \leftarrow y_s(\omega)$
Calculate utility value U_s^* using Equation (15) at evolutionary iteration ω
return $\boldsymbol{y^*}, \boldsymbol{U^*}$

where $\{\alpha, \beta\} > 0$ and act as the evolutionary control factors.

Evolutionary Equilibrium: Through evolution, the population adapts higher utility value strategies, eventually leading to an evolutionary stable strategy or evolutionary equilibrium. The fractions of the population choosing different strategies cease to change at evolutionary equilibrium[26]. In T-RESIN, evolutionary equilibrium is determined at $\dot{y_s}(\omega)$ for SDN switches and $\dot{x_e}(\phi)$ for edge nodes.

4.3 | Theoretical analysis

In T-RESIN, the SDN controller executes Algorithms 1 and 2, on behalf of the SDN switches and the edge nodes, respectively, to achieve the evolutionary equilibrium among the IoE devices, SDN switches, and edge computing nodes. We analyze the existence of evolutionary equilibrium in T-RESIN in this section.

Evolutionary equilibrium for SDN switches: At evolutionary equilibrium, we get.

$$\dot{y_s}(\omega) = \alpha y_s(\omega) \left(U_s(\omega) - \overline{U}(\omega) \right) = 0 \tag{21}$$

Considering that the population share of each switch $s \in \mathcal{S}$ is $y_s(\omega) \geq 0$ and evolutionary control factor α is a positive constant i.e. $\alpha > 0$. we get:

$$U_s(\omega) - \overline{U}(\omega) = 0 \tag{22}$$

By solving Equation (22), we get:

$$\left(y_s^*\right)^2 - y_s^* + \frac{\sum\limits_{s' \in \mathcal{S}/\{s\}} \left(y_{s'}^*\right)^2 \left(\frac{F_{s'}}{R_{s'}^{max}}\right)}{\frac{F_s}{R_s^{max}}} = 0 \tag{23}$$

At evolutionary equilibrium, we yield optimal population share y_s^* for each switch $s \in \mathcal{S}$ as follows:

Algorithm 2 Algorithm for edge nodes

INPUTS:

$\mathcal{S}, \mathcal{E}, C^e_{use}, E^e, P_e, W, \beta$

OUTPUT:

$\boldsymbol{x}^*, \boldsymbol{W}^*$

PROCEDURE:

$\phi \leftarrow 0$
Randomly map processes $P_s, \forall s$, to edge node $e \in \mathcal{E}$ for computational resources
do
    $\phi \leftarrow \phi + 1$
    for Each $e \in \mathcal{E}$ **do**
        Calculate $C^e_{use}(\phi)$ using Equation (7)
        Calculate $E^e_{res}(\phi)$ using Equation (12)
        Calculate population share $x_e(\phi)$ using Equation (14)
        Calculate utility value $W_e(\phi)$ using Equation (17)
    end for
    Calculate average utility value $\overline{W}(\phi)$ using Equation (18)
    for Each $e \in \mathcal{E}$ **do**
        Calculate replicator dynamic $\dot{x}_e(\phi)$ using Equation (20)
        $x_e(\phi + 1) \leftarrow x_e(\phi) + \dot{x}_e(\phi)$
    end for
while $(\dot{x}_e(\phi) \not\approx 0)$
$x^*_e \leftarrow x_e(\phi)$
Calculate utility value W^*_e using Equation (17) at evolutionary iteration ϕ
return $\boldsymbol{x}^*, \boldsymbol{W}^*$

$$y^*_s = \frac{1 \pm \sqrt{1 - 4\psi}}{2} \tag{24}$$

where $\psi = \left[\frac{\sum_{s' \in \mathcal{S}/\{s\}} (y^*_{s'})^2 \left(\frac{F_{s'}}{R^{max}_{s'}} \right)}{\frac{F_s}{R^{max}_s}} \right]$.

Evolutionary equilibrium for edge nodes: At evolutionary equilibrium, the change in the population share of the edge nodes reaches zero. Hence, we get.

$$\dot{x}_e(\phi) = \beta x_e(\phi) \left(W_e(\phi) - \overline{W}(\phi) \right) = 0 \tag{25}$$

Considering that $x_e > 0, \forall e$, and $\beta > 0$, we get.

$$W_e(\phi) - \overline{W}(\phi) = 0 \tag{26}$$

At evolutionary equilibrium, we get that the optimal population share x^*_e for each edge node e is as follows:

$$(x^*_e)^2 - x^*_e + \frac{\sum_{e' \in \mathcal{E}/\{e\}} (x^*_{e'})^2}{\left(1 - \frac{E^{e'}_{res}}{E^{e'}} \right)} \left(1 - \frac{E^e_{res}}{E^e} \right) = 0 \tag{27}$$

At evolutionary equilibrium, we yield optimal population share x^*_e for each edge node $e \in \mathcal{E}$ as follows:

$$x^*_e = \frac{1 \pm \sqrt{1 - 4\kappa}}{2} \tag{28}$$

where $\kappa = \left[\frac{\sum_{e' \in \mathcal{E}/\{e\}} (x^*_{e'})^2 \left(1 - \frac{E^{e'}_{res}}{E^{e'}} \right)}{\left(1 - \frac{E^e_{res}}{E^e} \right)} \right]$

5 | PERFORMANCE ANALYSIS

We emulated the proposed scheme, T-RESIN, using the Mininet network emulator [§]. The performance of T-RESIN is evaluated while comparing with the existing competing schemes. The detailed experimental setup with emulation parameters and the yield results are discussed in the subsequent sections.

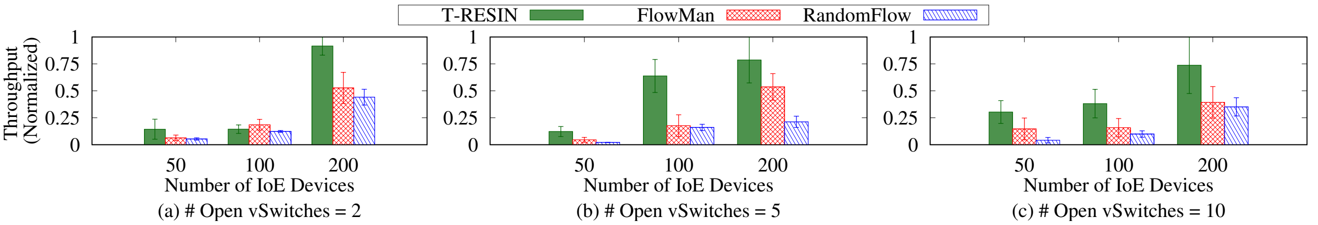

FIGURE 2 Network Throughput

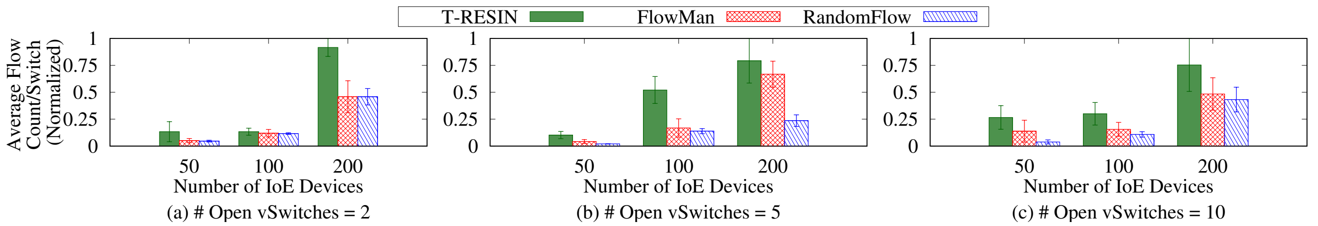

FIGURE 3 Flow Count per Open vSwitch

5.1 | Experimental Setup

To evaluate the performance of T-RESIN, we emulated a SDN-enabled edge platform in Mininet. We consider that SDEN is equipped with a Ryu SDN controller[¶] and Open vSwitch (SDN switches)[#]. The detailed experimental setup is mentioned in Table 3.

We consider that there is a single Ryu SDN controller for SDEN Mininet topology in T-RESIN in the presence of multiple SDN switches and edge nodes. The detailed emulation parameters are shown in Table 4.

TABLE 3 Experimental Setup

Hardware	Intel® Core™ i7-9700 CPU @3.00GHz × 8
Operating System	Ubuntu 20.04.6 LTS
Network Emulator	Mininet (Version 2.31b1)
SDN Controller	Ryu Controller (Version ryu 4.34)
SDN Switch	Open vSwitch (Version ovs-vsctl 2.13.8)
Programming Language	Python3 (Version 3.8.10)

[§] https://mininet.org/
[¶] https://ryu-sdn.org/
[#] https://www.openvswitch.org/

TABLE 4 Emulation Parameters

Parameter	Value
Number of Open vSwitches in Mininet topology	2, 5, 10
Number of Edge Nodes in Mininet topology	10, 20, 30
Number of IoE Devices in Mininet topology	50, 100, 200
Energy Consumption at Transmitter side (Tx)	50 nJ/bit [29]
Maximum Energy of each Edge node	20 Joule [29]
Evolutionary Control Factor	α = 0.01, β = 0.1

5.2 | Benchmarks

We compare the performance of T-RESIN with two schemes – RandomFlow and FlowMan [21]. In RandomFlow, we consider that the resource allocation for the switches and the edge nodes is random. To ensure unbiased result, we took 50 runs of the emulated platform for each topology and evaluated the 95% confidence interval result. On the other hand, in FlowMan, Mondal and Misra considered the presence of heterogeneous flows and flow-rules are placed to ensure high throughput. We argue that these schemes consider resource allocation in SDN-enabled platforms. However, the optimal edge resource allocation in provisioning IoE-enabled services are not considered in the existing literature.

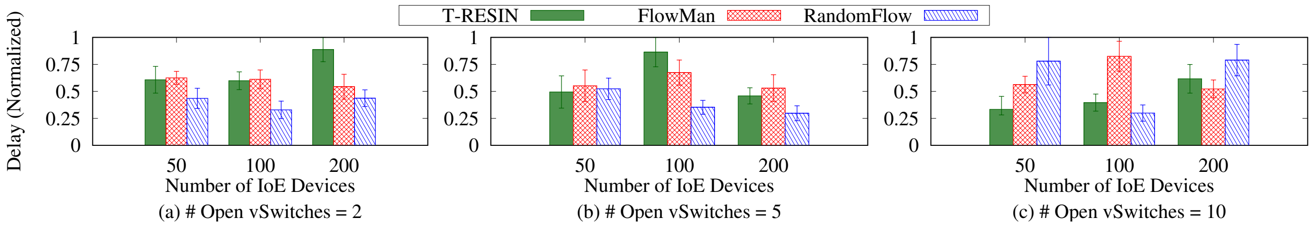

FIGURE 4 Switch Delay at Access Tier

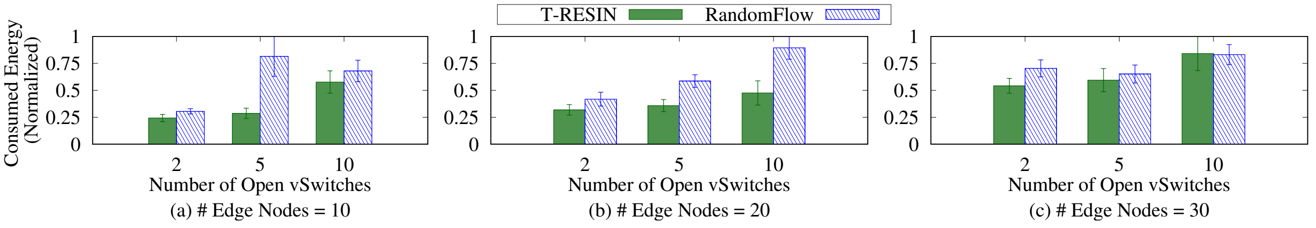

FIGURE 5 Energy Consumption of the Edge Nodes

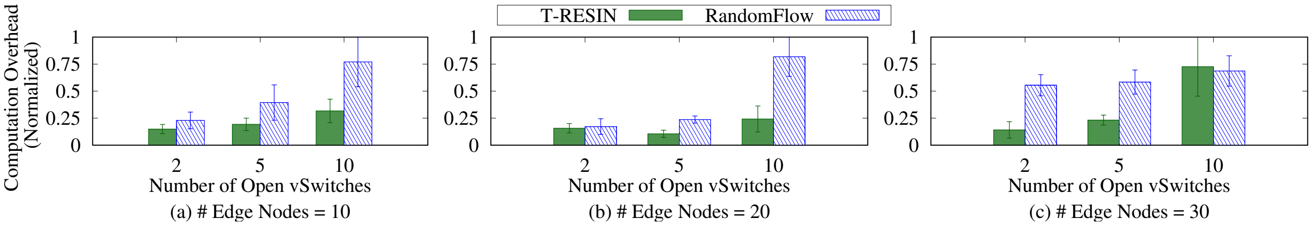

FIGURE 6 Computation Overhead of the Edge Nodes

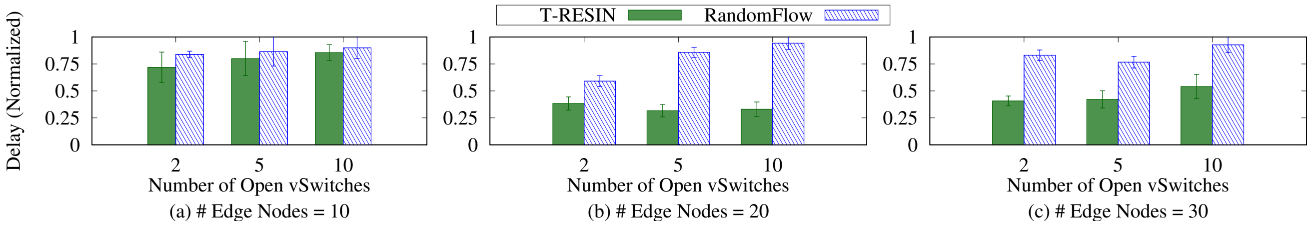

FIGURE 7 Computation Delay at the Edge Nodes

5.3 | Performance Metrics

To evaluate the performance of the proposed scheme, T-RESIN, we considered the following performance metrics.

- *Network Throughput:* It signifies the amount of data delivered and processed by the SDEN. We aim to achieve high network throughput.
- *Flow Count at Switch:* It is calculated as the total number of flow rules associated with TCAM memory for each switch.
- *Switch Delay:* It is calculated as the latency incurred by the flows at the SDN switches in Mininet.
- *Average Energy Consumption at Edge Node:* It signifies the amount of energy consumed to provision the IoE-enabled services/processes.
- *Average Computation Overhead at Edge Node:* We aim to achieve moderate utilization of the edge nodes and ensure low failure probability while provisioning the IoE-based services.
- *Computation Delay at Edge Node:* It is calculated as the processing delay the edge devices in provisioning the IoE-enabled services.

5.4 | Results and Discussions

From Figures 2, we observe that the average throughput per IoE devices increases by 27.98-31.84% using T-RESIN than using other schemes — RandomFlow and FlowMan. This is due to the fact that T-RESIN ensures that the flows are optimally distributed among the available SDN switches. Additionally, in FlowMan, the flow association is decided based on the one hop networks and in RandomFlow, the flows are allocated randomly. Moreover, we observe that with the increase in the number of SDN switches the network throughput outperforms the other existing competing schemes while ensuring a high throughput. Moreover, using T-RESIN, average flow count per Open vSwitch increases by 10.73-13.03% than using other competing schemes — RandomFlow and FlowMan, as observed in Figure 3. This eventually helps in distributing the network load optimally at the SDEN and ensures in achieving high throughput. However, in Figure 4, we observe that delay in the provisioned IoE-enabled services are mostly random using T-RESIN, as we did not consider delay parameters while modeling the game theoretic model. Though we argue that T-RESIN ensures the delay threshold values while provisioning services in IoE-enabled SDEN.

On the other hand, Figure 5 depicts that using T-RESIN, energy consumption at edge tier decreases by 4.59-14.57% using other scheme — RandomFlow. This is due to the fact that the requested processes/applications are allocated optimally among the available edge nodes. Hence, we also observe that computation overhead of the edge nodes reduces by 9.5-22.29% using T-RESIN than using RandomFlow, as shown in Figure 6. However, we observe from Figure 7 that the delay incurred at the edge nodes varies randomly using T-RESIN, as we did not consider the delay parameters while designing the mathematical model of T-RESIN, as mentioned earlier.

Hence, we argue that T-RESIN ensures a high throughput while reducing the energy consumption and computation overhead than using the competing existing schemes. We plan to extend this work and optimize the delay performance for IoE-enabled SDEN.

6 | CONCLUSION

In this paper, we studied the problem of optimal resource allocation in an IoE-enabled SDEN. We designed a multi-tier dynamic resource allocation scheme, named T-RESIN, using evolutionary game theory. In the bottom tier, the volume of data generated by the IoE devices defined the population. Population share of each SDN switch is evaluated as the volume of data associated with the corresponding switch. On the other tier, the population share of each edge node is evaluated as amount of data processed by it. While following the replicator dynamics principle, we theoretically analyze the existence of evolutionary equilibrium in T-RESIN. We also evaluated the performance of T-RESIN while emulating on Ryu controller-based Mininet platform and observed that T-RESIN outperforms the competing existing schemes in terms of achieved network throughput.

This work can be extended while optimizing the energy consumption of the SDEN network and design schemes to ensure less carbon footprint and sustainable. This work also can be included while considering hierarchical controller architecture in the presence of SDEN and cloud. We also plan to extend this work while optmizing the overall delay at IoE-enabled SDEN.

ACKNOWLEDGEMENT

Ayan Mondal acknowledges that this work was partially supported by the IIT Indore Young Faculty Research Seed Grant (YFRSG) Scheme (Grant No: IITI/YFRSG/2022-23/12).

References

1. Hojlo J. Future of Industry Ecosystems: Shared Data and Insights. Online; 2021. Accessed on December 15, 2023.

2. Evans D. The Internet of Everything: How More Relevant and Valuable Connections Will Change the World. *Cisco Internet Business Solutions Group (IBSG), Cisco Systems, Inc., San Jose, CA, USA.* 2012.

3. Botta A, Donato dW, Persico V, Pescapé A. Integration of Cloud computing and Internet of Things: A survey. *Future Generation Computer Systems.* 2016;56:684–700.

4. Stergiou C, Psannis KE, Kim BG, Gupta B. Secure integration of IoT and Cloud Computing. *Future Generation Computer Systems.* 2018;78:964–975.

5. Dizdarevic J, Carpio F, Jukan A, Masip-Bruin X. A Survey of Communication Protocols for Internet of Things and Related Challenges of Fog and Cloud Computing Integration. *ACM Computing Survey.* 2019;51:1-29. doi: https://doi.org/10.1145/3292674

6. Laroui M, Nour B, Moungla H, Cherif MA, Afifi H, Guizani M. Edge and fog computing for IoT: A survey on current research activities & future directions. *Computer Communications.* 2021;180:210-231. doi: https://doi.org/10.1016/j.comcom.2021.09.003

7. Tefera G, She K, Shelke M, Ahmed A. Decentralized adaptive resource-aware computation offloading & caching for multi-access edge computing networks. *Sustainable Computing: Informatics and Systems.* 2021;30.

8. Alam I, Sharif K, Li F, et al. A Survey of Network Virtualization Techniques for Internet of Things Using SDN and NFV. *ACM Computing Survey.* 2020;53. doi: https://doi.org/10.1145/3379444

9. Jazaeri SS, Jabbehdari S, Asghari P, Javadi HHS. Edge computing in SDN-IoT networks: a systematic review of issues, challenges and solutions. *Cluster Computing.* 2021;24:3187–3228.

10. Shah SDA, Gregory MA, Li S, Fontes RDR. SDN Enhanced Multi-Access Edge Computing (MEC) for E2E Mobility and QoS Management. *IEEE Access.* 2020;8:77459-77469.

11. Baktir AC, Ozgovde A, Ersoy C. How Can Edge Computing Benefit From Software-Defined Networking: A Survey, Use Cases, and Future Directions. *IEEE Communications Surveys and Tutorials.* 2017;19.

12. Langleya DJ, Doorn vJ, Ng IC, Stieglitz S, Lazovik A, Boonstra A. The Internet of Everything: Smart things and their impact on business models. *Journal of Business Research.* 2021;122:853-863. doi: https://doi.org/10.1016/j.jbusres.2019.12.035

13. Liu Y, Dai HN, Wang Q, Shukla MK, Imran M. Unmanned aerial vehicle for internet of everything: Opportunities and challenges. *Computer Communications.* 2020;155:66-88. doi: https://doi.org/10.1016/j.comcom.2020.03.017

14. Liyanagea M, Porambageb P, Dingc AY, Kalla A. Driving forces for Multi-Access Edge Computing (MEC) IoT integration in 5G. *ICT Express.* 2021;7:127-137. doi: https://doi.org/10.1016/j.icte.2021.05.007

15. Singh R, Gill SS. Edge AI: A survey. *Internet of Things and Cyber-Physical Systems.* 2023;3:71-92. doi: https://doi.org/10.1016/j.iotcps.2023.02.004

16. Iannacci J. Internet of things (IoT); internet of everything (IoE); tactile internet;5G – A (not so evanescent) unifying vision empowered by EH-MEMS(energy harvesting MEMS) and RF-MEMS (radio frequency MEMS). *Sensors and Actuators A: Physical.* 2018;272:187-198. doi: https://doi.org/10.1016/j.sna.2018.01.038

17. Misra SC, Mondal A. FogPrime: Dynamic Pricing-Based Strategic Resource Management in Fog Networks. *IEEE Transactions on Vehicular Technology.* 2021;70.

18. Gurung S, Mondal A. TiME: Time-Sensitive Multihop Data Transmission in Software-Defined Edge Networks for IoT. In: Casteleyn S, Mikkonen T, García Simón A, Ko IY, Loseto G., eds. *Current Trends in Web Engineering*Springer Nature Switzerland 2024; Cham:44–54.

19. Misra S, Mondal A, Khajjayam S. Dynamic Big-Data Broadcast in Fat-Tree Data Center Networks With Mobile IoT Devices. *IEEE Systems Journal.* 2019;13(3):2898-2905. doi: 10.1109/JSYST.2019.2899754

20. Yuan S, Peng M, Sun Y, Liu X. Software defined intelligent satellite-terrestrial integrated networks: Insights and challenges. *Digital Communications and Networks.* 2023;9(6):1331-1339. doi: https://doi.org/10.1016/j.dcan.2022.06.009

21. Mondal A, Misra S. FlowMan: QoS-Aware Dynamic Data Flow Management in Software-Defined Networks. *IEEE Journal on Selected Areas in Communications.* 2020;38.

22. Jain DK, Tyagi SKS, Neelakandan S, Prakash M, Natrayan L. Metaheuristic Optimization-Based Resource Allocation Technique for Cybertwin-Driven 6G on IoE Environment. *IEEE Transactions on Industrial Informatics.* 2022;18.

23. Manogaran G, Rawal BS. An Efficient Resource Allocation Scheme With Optimal Node Placement in IoT-Fog-Cloud Architecture. *IEEE Sensors Journal.* 2021;21.

24. Deb PK, Mondal A, Misra S. AuGrid: Edge-Enabled Distributed Load Management for Smart Grid Service Providers. *IEEE Transactions on Green Communications and Networking.* 2022;6(1):437-446. doi: 10.1109/TGCN.2021.3121877

25. Sami H, Otrok H, Bentahar J, Mourad A. AI-Based Resource Provisioning of IoE Services in 6G: A Deep Reinforcement Learning Approach. *IEEE Transactions on Network and Service Management.* 2021;18.

26. Han Z, Niyato D, Saad W, Basar T, Hjørungnes A. *Game Theory in Wireless and Communication Networks*. Cambridge University Press, New York, 2012.

27. Williams HP. *Logic and Integer Programming*. International Series in Operations Research & Management ScienceBoston, MA: Springer, US, Mar. 2009.

28. Drexl A. A simulated annealing approach to the multiconstraint zero-one knapsack problem. *Computing.* Mar. 1988;40:1-8.

29. Misra S, Mondal A, Bhavathankar P, Alouini MS. M-JAW: Mobility-Based Jamming Avoidance in Wireless Sensor Networks. *IEEE Transactions on Vehicular Technology.* 2020;69(5):5381-5390. doi: 10.1109/TVT.2020.2982966