EdgeSafe: Dynamic Load Balancing Among Edge Nodes for Provisioning Safety-as-a-Service in Vehicular IoT Applications

Chandana Roy[†], *Student Member, IEEE*, Sudip Misra[§], *Senior Member, IEEE*, Jhareswar Maiti [¶], and Soumi Nag[‡]
[†¶] Department of Industrial & Systems Engineering, [§]Department of Computer Science and Engineering,
[‡] Summer Intern, Department of Computer Science and Engineering,
[†§¶‡]Indian Institute of Technology Kharagpur, India
{[†]chandanaroy, [§]sudipm}@iitkgp.ac.in, [¶]jmaiti@iem.iitkgp.ernet.in, [‡]mnp.sinu@gmail.com

***Abstract*—In this paper, we propose a dynamic load balancing scheme, EdgeSafe, for provisioning *Safety-as-a-Service* (Safe-aaS) [1]. Typically, in a Safe-aaS infrastructure, the sensor nodes are either static or mobile in nature. With the variation in the geographical location of the vehicles, the sensor nodes attached to them attain mobility. Consequently, the distance between the mobile sensor node and the edge nodes present within their vicinity changes. Further, the average number of processes executed at the edge nodes varies with the type of edge nodes and time. As the data is time-critical, it is necessary to be primarily processed at the edge nodes. Considering road transportation as the application scenario of Safe-aaS, we perform load balancing in two stages. In the first stage, we calculate the preferred capacity ratio of the edge nodes present within the communication range of the sensor nodes. We apply *Markowitz Portfolio Selection Theory* in the second stage to select the appropriate edge node. The profit return and risk incurred in their selection is calculated to design the portfolio of the edge nodes. Thereafter, the utility of each edge node is computed. We formulate an optimization function to obtain the minimum value of the utility of the edge nodes, considering their risk incurred and remaining memory. Further, we apply Lagrangian Multiplier to solve and reformulate the optimization function. Existing research works on provisioning safety services consist of shortcomings in consideration of dynamic load balancing among the edge nodes. Extensive simulation results show that the proposed scheme, EdgeSafe, is capable of improving the data rate by 41.37%, 35.64%, and 21.05% respectively, compared to the existing schemes, HO [2], MLB [3], and Honeybee [4].**

***Keywords*—Road transportation, Safety-as-a-Service, Ingress component, Egress component, Load balancing, Markowitz Portfolio Selection Theory.**

I. Introduction

In order to provide safety services across different industries such as manufacturing, transportation, and power generation, the integration of Internet of Things (IoT)-based technologies is an essential aspect of concern [5]. For example, in the transportation industry, road safety applications include intimation of road safety information, detection of drowsiness of drivers [6], detection of manhole covers [7], and assessment of road safety threats.

A Safe-aaS [1] infrastructure provides customized safety-related decisions to multiple end-users simultaneously. The end-users request the decision parameters through a Web portal, according to their requirement. In a Safe-aaS architecture, heterogeneous types of static, and mobile sensor nodes are present in the *device layer*. The static sensor nodes are deployed at a particular geographical location and the mobile sensor nodes are deployed into the vehicles. With the mobility of the vehicles, the number and type of edge nodes present within their vicinity vary. As a result, the load/volume of time-critical sensed data at these edge nodes fluctuates. Additionally, the volume of time-critical data generated from these sensor nodes may vary with time. Therefore, the dynamic distribution of load/time-sensitive data among the edge nodes is necessary. Further, the edge nodes may vary from one another with their type, storage capacity, and process execution capability. Considering road transportation as the application scenario of Safe-aaS, we proposed a dynamic load balancing scheme for provisioning safety services.

With the increase in the number of on-road vehicles, various safety measures are adopted in the existing literature for the safety of both the drivers and vehicles. In Safe-aaS, the edge nodes primarily processes the time-critical data. Hence, timely delivery of the requested decisions to the end-users is necessary. In a critical scenario, such as a severe road accident and any natural calamity, colossal volume of time-sensitive data is available for processing at the edge nodes. Considering this facts, we design the load sharing mechanism for Safe-aaS platform. As the data are time-sensitive in nature, it is important to process them quickly. Further, the volume of data present in the system also fluctuates with time. Therefore, there is a possibility of variable load among the edge nodes. The major restriction to the situation is that each edge node does not compute beyond their maximum capacity. If any of the neighboring edge nodes of a sensor node attains its maximum capacity, then the sensor nodes search for other neighboring edge nodes with the allowable capacity. Further, Safe-aaS provides customized decisions to multiple end-users, founded on the concept of decision virtualization. None of the existing load balancing schemes provides customized decisions and considers the time-criticality of data. Therefore, we propose a scheme for dynamic load balancing among the edge nodes for provisioning safety-related decisions.

The primary contribution of this work is to design a scheme for dynamic load balancing among the edge nodes. The proposed scheme performs load balancing in two levels – (a) computation of preferred capacity ratio of the edge nodes within the proximity of each of the sensor nodes, and (b) load balancing among the edge nodes based on the optimal value of memory utilized by the edge nodes. More specifically, the detailed *contributions* of this work are:

- We consider the *edge flow-rate*, *cloud flow-rate*, *ingress*

component, and *egress component* of the edge nodes to compute the preferred capacity ratio of the edge nodes. We introduce a *mapping rule* to map the *ingress* and *egress* components of the edge nodes.

- We derive the maximum capacity of the edge nodes by solving the modeled optimization function. We use the Lagrangian multiplier to reformulate the optimization function and apply KKT conditions to find the maximum capacity of the edge nodes.
- We use *Markowitz Portfolio Selection Theory* (MPST) to uniformly share the load among the edge nodes based on the process execution capability, reputation, profit return, and risk incurred the selection of the edge nodes. The justification for the choice of MPST is given in Subsection III-C. We design an optimization function to find the minimum value of the utility of the edge nodes. We apply a Lagrangian multiplier to reformulate the optimization function and use KKT conditions to obtain the optimal value of the utilized memory of the edge device.
- Extensive simulation results illustrate that the proposed scheme, EdgeSafe, performs better in terms of the variation in datarate, capacity, and average residual energy of the edge nodes compared to the load balancing methods for wireless networks [2], downlink LTE networks [3], and work sharing algorithm for mobile edge nodes [4].

II. Related Work

In this section, we discuss the existing research works on the problems of load balancing in mobile networks and safety in road transportation. Load balancing among the edge nodes is an emerging concept in the road transportation industry. The edge nodes process the data, which are time-critical in nature. We discuss the existing researches on load balancing in wireless networks and cloud considering the transfer of load, the energy consumption of electric vehicles, and the throughput [2], [8]–[15].

Park *et al.* [2] proposed a method for self-organizing wireless networks based on mobility load balancing (MLB). The authors considered that an overloaded cell transfers the load to the least loaded neighboring node without evaluating the amount of load acceptable by the neighboring node. However, in this proposed work, we estimate the maximum capacity of any edge node. Thereafter, the available volume of time critical data is shared with the neighboring edge nodes depending upon their preferred capacity ratio. Considering the middleware nodes in the IoT application layer as brokers, Sun and Ansari [8] designed a method for traffic load balancing. They proposed a mechanism to cache the popular resources from the highly loaded to the lightly loaded brokers. Additionally, the authors formulated the resource re-caching problem to minimize the average delay and proved the reallocation problem is NP-hard. Similarly, Toumpis and Gitzenis [10] proposed a load balancing approach in wireless sensor network depending on the *Wireless Minimum Cost Problem*. However, the authors did not consider the quantitative analysis of communication and computation overhead because of memory constraints.

On the other hand, Han and Ansari [11] designed a traffic load balancing framework with the primary aim of stabilizing the network utility. Additionally, the simulation results depict that the on-grid power consumption and average delivery latency reduces. In another recent work, Takulder *et al.* [12] proposed a *Dual Threshold Load Balancing* approach to select the best machine for migration of the process in the SDN-based system. The authors demonstrated that their scheme yields energy-efficient load balancing to improve the throughput and response time. Jijin *et al.* [9] proposed the concept of *Virtual Fog Access Points* (FAPs), where FAPs act as service nodes for processing the task of end-users. The authors modeled the end-user's tasks as a task graph and service nodes as an edgeless service graph. Dai *et al.* [15] proposed a solution to the joint load balancing as well as offloading problem in the vehicular edge computing environment. Further, the authors modeled the offloading problem as a mixed integer non-linear program to maximize the system utility. Roy *et al.* [1], [16], [17] proposed a unique platform for provisioning customized safety-related decisions to the end-users on pay-per-use basis. Based on the time-criticality of the data sensed by the sensor nodes deployed at various geographical locations and into the vehicles, the primary processing is done at the edge nodes or cloud. On the other hand, the sensor nodes present in the vehicles become mobile with the change in the location of vehicles. Therefore, an appropriate selection of edge nodes is necessary.

Synthesis: In the existing literature, the authors proposed different solution approaches on load balancing in wireless sensor networks and IoT. However, there is research lacuna on dynamic load balancing among the edge nodes, to impart safety-related decisions to the end-users in Safe-aaS. The existing approaches do not consider the mobile nature of sensor nodes, preferred capacity ratio of the edge nodes, and dynamic varying distance between the mobile sensor nodes and the edge nodes within their proximity. Therefore, dynamic sharing of load among the edge nodes in Safe-aaS, is challenging in nature. We propose a load balancing scheme based on the preferred capacity and process execution capability of the edge nodes.

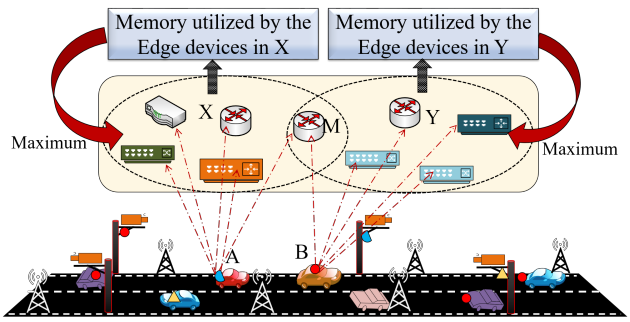


Fig. 1: Dynamic load balancing

III. EdgeSafe: Dynamic Load Balancing among the Edge Nodes for Safe-aaS

A. Problem Scenario

We consider an Intelligent Transportation System (ITS) environment where Safe-aaS infrastructure [1] is implemented. A Safe-aaS architecture provides customized safety-related decisions to the registered end-users. The end-users select certain decision parameters, as per their requirement, through the Web portal. On the other hand, based on the decision parameters selected, the safety service provider (SSP) delivers safety-related decisions to the end-users. Heterogeneous types

of static and mobile sensor nodes are either deployed at a particular geographical location or are placed on the vehicles in the *device layer*, as depicted in Fig. 1. The number and type of edge nodes present within the vicinity of the static sensor nodes remain constant, while they vary in mobile sensor nodes. The reason behind this is that the sensor nodes attached to these vehicles attain mobility with the change in the geographical location of the vehicles. These sensor nodes sense and transmit data to the edge nodes/cloud, based on the time-critical nature of the data. The raw sensed data are primarily processed at the edge nodes/cloud. Thereafter, the primarily processed data is transmitted to the decision layer, where the decision is generated. The volume of time-critical data present in the system may vary with time. Depending on the storage capacity and process execution capability, the edge nodes are categorized into different types. On the other hand, each type of edge nodes possesses a restricted amount of storage capacity. A probable solution to this problem is sharing the load among the edge nodes present within the proximity of the sensor nodes. In such a scenario, when the neighboring edge nodes of a sensor node attain their maximum capacity, the sensor node searches for the nearest neighboring edge node with the permissible capacity for sharing the load. We apply *Markowitz Portfolio Selection Theory* to select the appropriate edge node and dynamically share the load among them from the multiple edge nodes, present within the proximity of the sensor node.

B. Problem Formulation

We consider the presence of heterogeneous types of static and mobile sensor nodes in a Safe-aaS infrastructure which is represented as, $\mathbb{S} = \{S^{st}, S^{mo}\}$, where S^{st} and S^{mo} denote the set of static and mobile sensor nodes respectively. Further, $\mathbb{S} = \{\mathcal{S}_j | 1 \leq j \leq p\}$ and $\mathcal{S}_j \implies (S^{st} \vee S^{mo})$. We consider $\mathcal{E} = \{e_1, e_2, \cdots, e_n\}$ as the set of edge nodes. The capacity of these edge nodes is represented as $\mathcal{C} = \{c_1, c_2, \cdots, c_n\}$. We consider that the static and mobile sensor nodes transmit beacons to the edge nodes. The edge nodes present within the communication range of these sensor nodes, which receive this beacon, are considered to be within the *neighborhood* of that sensor node. Each data packet consists of - (i) id of the sensor node, (ii) distance between the sensor node and the edge nodes within its proximity, and (iii) sensed data. Therefore, the edge nodes present within the communication range of static/mobile sensor nodes are referred to as the "neighboring edge nodes" of that sensor node. In case of a static sensor node, the corresponding neighboring edge nodes of the sensor node always remains same. However, the list of neighboring edge nodes of a mobile sensor node vary with the mobility of the vehicle.

Efficacious Distance ($D_{ij}^{t,eff}$): We compute the distance between the edge node, e_i and the sensor node, $\mathcal{S}_j$ within a particular geographical region at any given time instant, t using the *Euclidean distance formula*. Suppose, the position of the j^{th} sensor node, $\mathcal{S}_j$ is denoted as $pos_{sen}^{j} = \{lat_j^{sen}, lng_j^{sen}\}$ and the position of the i^{th} edge node, e_i is represented as $\phi_i = \langle lat_i, lng_i \rangle$. The distance between $\mathcal{S}_j$ and the edge nodes present within its vicinity is denoted as D_{ij}^{t}. Suppose, $\mathbb{D} = \{D_{1j}^{t}, D_{2j}^{t}, \cdots, D_{kj}^{t}\}$ denotes the set of euclidean distances of the neighboring edge nodes of the j^{th} sensor node. Therefore, the efficacious distance is mathematically represented as, $D_{ij}^{t,eff} = \frac{D_{ij}^{t}}{max(\mathbb{D})}$, where $max(\mathbb{D})$ returns the maximum value of distance between the sensor node and its neighboring edge node. To balance the load or available volume of time-critical data among the edge nodes present within the vicinity of mobile sensor nodes, we consider their capacity. We calculate the input capacity of the edge nodes in terms of the ingress component and edge flow rate. Similarly, the output capacity of the edge nodes is computed in terms of their egress component and cloud flow rate.

***Definition* 1.** *Edge flow-rate* (EFR_{ij}^{t})*: The edge flow-rate of an edge device is expressed as a function of the distance,* $D_{ij}^{t,eff}$ *between the sensor node,* $\mathcal{S}_j$ *and the edge node,* e_i *within its proximity, and the average packet forwarding rate from the* j^{th} *sensor node to the* i^{th} *edge node,* $pf_{ij}^{avg,t}$*. Mathematically,*

$$EFR_{ij}^{t} = \frac{pf_{ij}^{avg,t}}{D_{ij}^{t,eff}} \tag{1}$$

Similarly, we compute the cloud flow-rate, (CFR_{iC}^{t}) considering the data transfer between the i^{th} edge device and cloud, C as a single hop.

***Definition* 2.** *Cloud flow-rate* (CFR_{iC}^{t})*: Cloud flow-rate is expressed as a function of the efficacious distance between the edge node and the cloud and the average packet forwarding rate from the* i^{th} *edge node to cloud,* $pf_{iC}^{avg,t}$*. Therefore, the cloud flow-rate is represented as,*

$$CFR_{iC}^{t} = \frac{pf_{iC}^{avg,t}}{D_{iC}^{eff}} \tag{2}$$

Efficacious distance between the edge node and the cloud: We consider that each of these edge nodes, e_i, is deployed over a particular geographical position, ϕ_i. Therefore, $\phi_i = \langle lat_i, lng_i \rangle$, where lng_i and lat_i are the longitude and latitude of the edge node, e_i, respectively. Due to the spherical nature of the Earth's surface, we compute the distance between any two points using the concept of haversine distance [18]. The efficacious distance between the i^{th} edge node and cloud, C as the *Haversine* distance between them.

$$\mathcal{D}_{iC}^{eff} = \mathcal{H}_i(\phi_a, \phi_b) \frac{\pi}{180^{\circ}}, \forall a, b \in \{1, 2, \cdots, n\} \tag{3}$$

where $\mathcal{H}_i(\phi_a, \phi_b)$ is the Haversine distance between the edge node, a and the cloud server, b. The Haversine distance, $\mathcal{H}_i(\phi_a, \phi_b)$ is computed as:

$$\mathcal{H}_i(\phi_a, \phi_b) = 2\mathbb{E}_r \sin^{-1} \sqrt{hav\left(\frac{\mathcal{H}_i(\phi_a, \phi_b)}{2\mathbb{E}_r}\right)} \tag{4}$$

In Equation (4), $\mathbb{E}_r$ represents the radius of the Earth. Further, $hav(\mathcal{H}_i(\phi_a, \phi_b))$ is represented in Equation (5).

$$\begin{aligned} hav(\mathcal{H}_i(\phi_a, \phi_b)) = \sin^2\left(\frac{\Delta lat_{ab}}{2}\right) + \\ \cos(lat_a)\cos(lat_b)\sin^2\left(\frac{\Delta lng_{ab}}{2}\right) \end{aligned} \tag{5}$$

where $\Delta(lat_{ab}) = (lat_a - lat_b)$, $\forall a, b \in \{1, 2, \cdots, k\}$ and $\Delta(lng_{ab}) = (lng_a - lng_b)$, $\forall a, b \in \{1, 2, \cdots, k\}$.

To estimate the mean flow-rate of the overall network, we consider the average value of edge and cloud flow-rate. In order to provide an overview of the average volume of the

time-sensitive data flow in the network, we took the mean of the data flow-rate from the sensor nodes to the edge nodes and edge nodes to the cloud.

Average flow-rate (Ψ_i^t): Average flow-rate is the mean of the edge flow-rate, EFR_i^t and the cloud flow-rate, CFR_i^t, which is expressed as $\Psi_i^t = \frac{1}{2}(CFR_{iC}^t + (EFR_{ij}^t))$. In other words, the average flow rate of the i^{th} edge node as Ψ_i^t denotes the mean of the volume of data entering and leaving the i^{th} edge node.

Ingress component of edge device (IC_i) : We represent the ingress component as a tuple of data input ports, d_p^{in} input queue, $e_{q_{in}}$, and control input ports, c_p^{in} [19]. We consider that each of these parameters possesses a real value. Considering this fact, the parameter ingress component is also a real number. IC_i is expressed as a 3-tuple $IC_i = \langle e_{q_{in}}, d_p^{in}, c_p^{in} \rangle$.

Egress component of edge device(EC_i) : The egress component of an edge device comprises the packet output queue, $e_{q_{out}}$, data output ports, d_p^{out}, and a control output port, c_p^{out} [19], and is expressed as a 3-tuple $EC_i = \langle e_{q_{out}}, d_p^{out}, c_p^{out} \rangle$.

Further, to map the arrival and transmission rates of the data packets at the edge nodes present within the vicinity of the sensor nodes, we introduce the mapping rule.

***Definition* 3.** *Mapping rule: Mapping rule of any edge node is the rule based on which the ingress component, IC_i of the i^{th} edge node is mapped with the egress component, EC_i of that edge node. It is expressed as a 2-tuple – the arrival rate of data, AR_d^{ij} from the j^{th} sensor node to the i^{th} edge node and the transmission rate, TR_{iC}^t of the i^{th} edge node to the cloud server. Mathematically, $e_{mt} = \langle AR_d^{ij}, TR_{iC}^t \rangle$.*

Typically, in a Safe-aaS infrastructure, different types of edge nodes are present. Therefore, ingress component, IC_i, egress component, EC_i, edge flow-rate, EFR^t, and cloud flow-rate, CFR^t differ for each type of edge nodes. To compute the capacity, c_{ij}^t of an edge node, we consider the ingress component, egress component, edge flow-rate, and cloud flow-rate of the edge node.

Capacity (c_{ij}^t): We express the capacity of the i^{th} edge node as a function – $c_i^t = f(IC_i, EC_i, (EFR_{ij}^t), CFR_i^t)$, where $c_i^t \in \mathcal{C}$. The input capacity of an edge node is expressed as directly proportional to the ingress component (IC) as well as edge flow rate (EFR). c_i^t is mathematically expressed as – $cap_{in}^{ij,t} \propto IC_i$ and $cap_{in}^{ij,t} \propto EFR_{ij}^t$. Therefore, $(cap_{in}^{i,t})^2 \propto (IC_i \times EFR_{ij}^t)$, and $cap_{in}^{i,t} \propto \sqrt{IC_i \times EFR_{ij}^t}$. Similarly, the output capacity function is also proportional to the egress component (EC) and cloud flow rate (CFR and is expressed as – $cap_{out}^{i,t} \propto EC_i$ and $cap_{out}^{i,t} \propto CFR_i^t$. Therefore, $(cap_{out}^{i,t})^2 \propto (EC_i \times CFR_{ij}^t)$, and $cap_{out}^{i,t} \propto \sqrt{EC_i \times CFR_{ij}^t}$. The capacity of the i^{th} edge node, e_i is expressed as:

$$cap_{in}^{ij,t} = k_1 \left(\frac{IC_i \times (EFR_{ij}^t)}{EFR_{ij,max}^d} \right)^{\frac{1}{2}} \tag{6a}$$

$$cap_{out}^{i,t} = k_2 \left(\frac{EC_i \times CFR_i^t}{CFR_{i,max}^t} \right)^{\frac{1}{2}} \tag{6b}$$

where k_1 and k_2 are the proportionality constants. $(EFR_{ij,max}^d)$ and $CFR_{i,max}^t$ denote the maximum value of the edge flow-rate and cloud flow-rate of the i^{th} edge device respectively. Therefore, the average capacity of the edge device is represented as $c_{ij}^t = \frac{\left(cap_{in}^{ij,t} + cap_{out}^{i,t}\right)}{2}$.

The set of capacity of the edge nodes present within the vicinity of the j^{th} sensor node is represented as, $\mathcal{C}_j = \{f_1(IC_1, (EFR_{1j}^t), EC_1, CFR_1^t), f_2(IC_2, (EFR_{2j}^t), EC_2, CFR_2^t), \cdots, f_n(IC_n, (EFR_{nj}^t), EC_n, CFR_n^t)\}$.

Theorem 1. *The input and output capacity function of an edge node is concave in nature, provided that the ingress component, egress component, edge flow-rate, and cloud flow-rate vary with time.*

Please refer to the supplementary file for the proof.

Due to the concave nature of the input and output capacity functions, we design the optimization function to maximize the average capacity value. As the average flow-rate, Ψ_i^t of the i^{th} edge device varies with time, therefore, to compute the maximum capacity of the edge node, e_i, at any time instant, t, we formulate the maximization function as,

$$\text{Maximize } c_{ij}^t \tag{7}$$

subject to, $EC_i > IC_i$, $EFR_{ij}^t < EFR_{ij,max}^d$, $CFR_i^t < CFR_{max}^d$, and $CFR_i^t > EFR_{ij}^t$. In order to solve and reformulate the optimization problem, we apply *Lagrangian* function, which is expressed as,

$$\mathcal{L} = -c_{ij}^t + \mu_1(EC_i - IC_i) + \mu_2(EFR_{ij}^t - EFR_{ij,max}^d) + \mu_3(CFR_i^t - CFR_{max}^d) + \mu_4(EFR_{ij}^t - CFR_i^t) \tag{8}$$

where μ_1, μ_2, μ_3, μ_4, μ_5, and μ_6 denote the *Lagrangian* constants. Further, we constrain the optimization function and compute the maximum capacity of the edge node, using *Karush-Kuhn-Tucker* (KKT) conditions. Equation (9a) and (9b) represent the dual feasibility and complementary slackness conditions.

$$\nabla_x \mathcal{L} = -\nabla c_{ij}^t + \nabla \sum_i \mu_i y_i = 0 \tag{9a}$$

$$\mu_i y_i = 0 \text{ and } \mu_i \geq 0, \quad \forall i = \{1, 2, \cdots, 6\} \tag{9b}$$

where x represent the parameters such as $IC_i, EC_i, k_1, k_2, EFR_{ij}^t$, and CFR_i^t of the Equation (7). The constraints of the Equation (7) are denoted as y_i. On solving the Equations (9a) and (9b), we obtain the maximum value of the capacity. Further, from Equation (7), we obtain the maximum value of capacity of the edge nodes, which are present within the neighborhood of any sensor node. We represent the maximum capacity of the edge nodes present within the proximity of the j^{th} sensor node as a set, $\mathcal{S}_{j,cap}^{max,t} = \{c_{1j}^{max,t}, c_{2j}^{max,t}, \cdots, c_{nj}^{max,t}\}$. From the set $\mathcal{S}_{j,cap}^{max,t}$, we find the maximum capacity value and compute the effective maximum capacity, $c_{max}^{eff,t}$. Therefore, $c_{max}^{eff,t} = \max\left(\mathcal{S}_{j,cap}^{max,t}\right)$.

Preferred Capacity Ratio ($C_i^{pre,t}$): Preferred capacity ratio of each edge device is represented as the ratio of the difference between the maximum capacity, $c_i^{max,t}$ and the present capacity of the i^{th} edge device, c_i^t to the effective maximum capacity, $c_{max}^{eff,t}$ of the edge devices present within the vicinity of the j^{th} sensor node. Therefore, the preferred capacity ratio of each edge device is computed as $C_{ij}^{pre,t} = \frac{\left(c_i^{max,t} - c_{ij}^t\right)}{c_{max}^{eff,t}}$. We consider $\mathbb{S}_{j,cap}^{pre,t} = \{C_1^{pre,t}, C_2^{pre,t}, \cdots, C_n^{pre,t}\}$ as the set of computed preferred capacity ratio of the neighborhood edge nodes of the j^{th} sensor node. Further, we select the

appropriate edge node in terms of their maximum preferred capacity ratio. Preferred capacity ratio is one of the essential inputs required to calculate the profit return of the edge nodes. In order to share the load among the edge nodes, we apply *Markowitz Portfolio Selection* method [20]. Finally, we find the utility of these edge nodes. The edge node which possess minimum utility at an optimal value of utilised memory is selected for load sharing.

Algorithm 1 EdgeSafe: Computation of Preferred capacity ratio

INPUTS:

1: $\mathcal{NL}_j$ - Neighborhood of the j^{th} sensor node

OUTPUTS:

1: C_{max}^{eff} and C_{ij}^{pre}

PROCEDURE:

1: **for** $\forall e_i \in \mathcal{NL}_j$ **do**
2: EFR_{ij}^t, CFR_i^t, IC_i, and EC_i are estimated for the i^{th} edge node.
3: c_{ij}^t is computed for each e_i.
4: **end for**
5: $c_{max}^{eff,t}$ is computed.
6: $C_{ij}^{pre,t}$ of the i^{th} edge node is computed.

Algorithm 1 provides insights regarding the computation of the preferred capacity ratio of the edge nodes present within the proximity of a sensor node. Based on the location of the sensor node (static or mobile), the neighboring edge nodes of the sensor node are computed. Thereafter, in Steps 2 and 3, the capacity of each of these edge nodes is estimated based on their EFR, CFR, IC, and EC. Step 6 computes the preferred capacity ratio of each of these edge nodes, as per the expression derived in Section III-B. Further, the utility is computed from the preferred capacity ratio, process execution capability, risk incurred in selecting that edge node, and spare memory of the edge node in Algorithm 2. Algorithm 1 runs $\mathcal{NL}_i$ times to estimate the effective capacity of the edge nodes present within the neighborhood of the i^{th} sensor node. Further, the utility of these edge nodes are estimated and the appropriate edge node is selected for sharing the load using Algorithm 2. Therefore, the running time of Algorithm 2 is $\mathcal{NL}_i$. The total running time complexity of Algorithms 1 and 2 are $\mathcal{O}(\mathcal{NL}_i) + \mathcal{O}(\mathcal{NL}_i) = \mathcal{O}(\mathcal{NL}_i)$. As there are p sensor nodes present in the scenario, the overall running time complexity of the proposed scheme is $\mathcal{O}(p\mathcal{NL}_i)$. Therefore, the proposed dynamic load balancing approach is scalable.

C. Solution Approach

In our problem scenario, we consider the presence of a huge volume of time-sensitive data in the Safe-aaS infrastructure. Therefore, the selection of the appropriate neighboring edge node of the sensor node to dynamically share the load among them is necessary. We assume that the edge nodes receive an incentive from the Safety Service Provider (SSP) in the form of profit whenever the edge nodes are selected. We apply *Markowitz Portfolio Selection Theory* [20]–[22] to select the appropriate edge node for sharing the load. Thereafter, we compute the utility for each of the neighboring edge nodes of the sensor node. Finally, based on the optimum value of the memory utilized by the edge node, we select the edge node for load balancing.

Justification for Markowitz Portfolio Selection Theory: The primary aim of using Markowitz Portfolio Selection Theory is to dynamically share the available volume of time-critical data among the edge nodes, such that the risk incurred in selecting them is minimized. In case any delay is incurred in sharing the time-critical data among the edge nodes or the selected edge node is not capable for processing the data, then the decision generation time increases. Therefore, the selection of the appropriate edge node among the other edge nodes present within the communication range of the sensor node is essential. We map this scenario with the portfolio selection theory. We design the portfolio of each of these edge nodes in terms of their process execution capability, available remaining memory, profit return, and risk incurred. Further, to estimate the risk incurred in the selection of an edge node, we consider the relative variation in the profit over consecutive time slots, process execution capability, and effective spare memory. The amount of memory utilized by an edge device acts as an essential factor for edge node selection. Therefore, the risk incurred in selection of the edge node increases with the increase in their amount of utilized memory.

We consider the set of end-users as $\mathcal{U} = \{u_1, u_2, \cdots, u_y\}$, where $\forall u_x \in \mathcal{U}$. We divide each day into q equal time-intervals, assuming the duration of Safe-aaS services provided to the end-users. Therefore, the set of time intervals $DT_d = \{t_1, t_2, \cdots, t_q\}$.

Lemma 1. *There are at least $\lceil \frac{\mathbb{N}}{K} \rceil$ end-users, who have requested for the safety services if $\mathbb{N}$ number of end-users request for Safe-aaS services during the K different time-slots in a day.*

Please refer to the supplementary file for the proof.

We assume that the SSP possesses the information concerning the average packet delivery ratio, the number of data packets successfully delivered, the number of times load shared with other edge nodes, and the incentive received by the edge node.

***Definition* 4.** *Process execution capability of the edge device ($\mathbb{PEC}_i^{t_k}$): The process execution capability of the i^{th} edge device is expressed as the ratio of the number of processes executed ($\mathbb{P}_i^{t_k}$) per unit time on the d^{th} day till the time instant, t_k, to the maximum number of processes executed, $\mathbb{P}_i^{max,\mathcal{N}}$ per unit time during the past $\mathcal{N}$ days. Therefore,*

$$\mathbb{PEC}_i^d = \sum_{k=1}^{q} \mathbb{PEC}_i^{t_k} = \sum_{k=1}^{q} \frac{\mathbb{P}_i^{t_k}}{\mathbb{P}_i^{max,\mathcal{N}}} \tag{10}$$

where $\mathbb{P}_i^{max,\mathcal{N}} = \max(\mathbb{P}_i^x)$, where $\mathbb{P}_i^x$ denotes the maximum number of processes executed by the i^{th} edge node on the x^{th} day. The process execution capability of the i^{th} edge device is represented as $\mathbb{PEC}_i^d$.

Effective Spare Memory ($ESM_i^{t_k}$): We represent the effective spare memory of the i^{th} edge node as the ratio of the unused memory, M_i^{rem,t_k} of the i^{th} edge node at the t_k^{th} time instant to the total memory, M_i^T of that edge device.

$$ESM_i^d = \sum_{k=1}^{q} ESM_i^{t_k} = \sum_{k=1}^{q} \frac{M_i^{rem,t_k}}{M_i^T} = \sum_{k=1}^{q} \frac{M_i^T - M_i^{ut,t_k}}{M_i^T} \tag{11}$$

where M_i^{ut,t_k} is the amount of memory utilized at the t_k^{th} time instant and ESM_i^d represents the effective spare memory

of the i^{th} edge node.

***Definition* 5.** *Reputation of edge node ($\mathbb{R}_{ij}^{t_k}$): Reputation of the i^{th} edge node is the ratio of the probability of the number of data packets, $\mathcal{P}_i^{t_k}$ successfully delivered by the i^{th} edge node at the time instant, t_k, and the response transmitted from the end-user/vehicle, to the number of times the node failed, $\mathbb{F}_i^{t_k}$ till the d^{th} day. Therefore, $\mathbb{R}_{ij}^{d} = \sum_{k=1}^{q} \mathbb{R}_{ij}^{t_k}$*

$$\mathbb{R}_{ij}^{t_k} = \begin{cases} \frac{\alpha \mathcal{P}_i^{t_k} N_{ij}^{t_k}}{\mathbb{F}_i^{t_k}}, & \text{node has failed} \\ \alpha \mathcal{P}_i^{t_k} N_{ij}^{t_k}, & \text{node has not failed} \end{cases} \tag{12}$$

where α is the proportionality constant and $0 \leq \alpha \leq 1$. In case the i^{th} edge node has not failed once till the t_k^{th} time instant, then we calculate reputation according to the second case. Thereafter, we estimate the reputation of the i^{th} edge node for serving the j^{th} end-user during each of the t_k^{th} time instants. Based on the value of $\mathcal{P}_i^{t_k}$, α varies. The number of data packets transmitted as response from the j^{th} end-user to the i^{th} edge node is represented as $N_{ij}^{t_k}$. We consider the total number of data packets transmitted by the edge device as N^{t_k} and the number of data packets which successfully reaches the cloud as $n_s^{t_k}$. The probability of transferring the data packets from the edge node to the cloud is denoted by $p^{t_k} = \frac{pf_i^{t_k}}{pf_i^{max,d}}$, where $pf_i^{t_k}$ and $pf_i^{max,d}$ represent the packet forwarding rate at the t_k^{th} time instant and the maximum packet forwarding rate of the i^{th} edge device on the d^{th} day. Therefore, we compute the probability of successful transfer of $n_s^{t_k}$ data packets using *Binomial distribution*. Mathematically,

$$\mathcal{P}_i^{t_k} = \frac{N^{t_k}!}{n_s^{t_k}!(N^{t_k} - n_s^{t_k})!} \times p^{t_k, n_s^{t_k}} (1 - p^{t_k})^{(N^{t_k} - n_s^{t_k})} \tag{13}$$

***Definition* 6.** *Profit return of the edge node ($P_{ij}^{t_k}$): The profit return of the edge node at any time instant, t_k is expressed as the chances of the edge node to get selected for load balancing and the incentive received by the edge node from the Safety Service Provider (SSP).*

$$P_{ij}^{d} = \sum_{k=1}^{q} P_{ij}^{t_k} = \sum_{k=1}^{q} \left(S_i^{t_k} \times I_i^{t_k} + \left(\mathbb{R}_{ij}^{t_k} \times C_{ij}^{pre,t_k} \times \lambda_i^{t_k} \right) \right) \tag{14}$$

where C_{ij}^{pre,t_k} is the preferred capacity ratio of the i^{th} edge node. Practically, each of the edge nodes may be present within the proximity of multiple sensor nodes. In case the selected edge node processes these data, then the neighboring edge nodes with the permissible capacity shares the load among them. Considering these facts, we introduced $\lambda_i^{t_k}$ while designing the profit return formula. $S_i^{t_k}$ is the number of times the i^{th} edge node was selected for load balancing till the last time instant, $(t_k - 1)$. The incentive received by the i^{th} edge node for being selected at the t_k^{th} time instant is denoted as $I_i^{t_k}$.

***Definition* 7.** *Risk Incurred ($RI_i^{t_k}$): The expected risk incurred by the i^{th} edge node for sharing load is expressed in terms of the process execution capability, $\mathbb{PEC}_i^{t_k}$, effective spare memory, $ESM_i^{t_k}$, and the relative variation in the profit return, $\sigma_{ij}^{t_k}$, of the i^{th} edge node during the last $\mathcal{N}$ days.*

Therefore,

$$RI_i^{d} = \sum_{k=1}^{q} RI_i^{t_k} = \sum_{k=1}^{q} \frac{\min(\sigma_{xy}^{t_k})}{\mathbb{PEC}_i^{t_k}} \times e^{ESM_i^{t_k}} \tag{15}$$

The co-variance of an edge node over the past $\mathcal{N}$ days is expressed as the sum of the deviation in the profit of that edge node from the mean profit value, between any x^{th} and y^{th} time slots on the d^{th} day is mathematically defined as,

$$\sigma_{xy}^{t_k} = \frac{1}{\mathcal{N}} \left(\sum_{d=1}^{\mathcal{N}} \left(\left(P_{ij}^{t_x,d} - P_{ij}^{d,mean} \right) \left(P_{ij}^{t_y,d} - P_{ij}^{d,mean} \right) \right) \right) \tag{16}$$

where the mean value of profit return during the d^{th} day upto the t_k^{th} time instant, is denoted as $P_{ij}^{d,mean}$. $\sigma_{xy}^{t_k}$ is a matrix of order $q \times q$, where q denotes the number of time-slots in a day.

To distribute the load among the edge nodes present within the vicinity of any sensor node, we calculate the preferred capacity, effective spare memory, profit return, and risk incurred in selecting the edge node. Further, we compute the preferred capacity in terms of the volume of available time-critical data present at the input ports and the volume of primarily processed data to be transmitted to the cloud/server. Finally, we formulate the utility of these edge nodes in terms of the above-mentioned parameters and design an optimization function.

Utility ($U_i^{t_k}$): The utility function is composed of the risk incurred and effective spare memory. On the other hand, the edge node with the minimum value of utility possesses the slightest risk in their selection. The least deviation in the profit return (co-variance) per unit process execution capability of an edge node is the expected risk incurred in selecting them. Moreover, deducting the effective value of spare memory from the risk incurred in choosing that edge node determines their utility. Further, utility denotes the usefulness in terms of the processing of time-critical data. Considering these facts, we formulate the utility function as,

$$U_i^{t_k} = RI_i^{t_k} - \beta \log(ESM_i^{t_k})^m \tag{17}$$

Algorithm 2 EdgeSafe: Dynamic edge node selection

INPUTS:
1: $\mathcal{DT}_d$ and $c_{e_i}^{pre}$
OUTPUT:
2: $U_i^{t_k}$- Minimum utility of e_i
PROCEDURE:
1: $\mathbb{PEC}_i^{t_k}$, $\mathbb{PEC}_i^{d}$, $ESM_i^{t_k}$, ESM_i^{d}, $\mathbb{R}_{ij}^{t_k}$, $\mathbb{R}_{i}^{d}$, $\mathbb{P}_i^{t_k}$, $\mathbb{P}_i^{d}$, and $RI_i^{t_k}$ are computed for the i^{th} edge device.
2: The minimum value of $U_i^{t_k}$ of the i^{th} edge device and optimal value of utilized memory is computed, according to Equation (18).

Lemma 2. *There exist real roots for the optimal memory utilized by the edge node, such that the utility of the edge node is minimum.*

Proof: In our problem scenario, we use the *Markowitz Portfolio Selection Theory* to select the appropriate edge node for load balancing, based upon the optimal value of the amount of memory used by that edge node. The optimization

function is mathematically represented as,

$$\underset{M_i^{ut,t_k}}{argmin} \quad U_i^{t_k} \tag{18}$$

subject to, $M_i^{ut,t_k} < M_i^T$, $\mathbb{F}_i^{t_k} > 0$, $1 > \mathcal{P}_i^{t_k} > 0$, $S_i^{t_k} \geq 0$, $0 < C_i^{pre,t_k} < 1$, $I_i^{t_k} > 1$, $\lambda_i^{t_k} \geq 1$.

The *Lagrangian* form of Equation (18) is mathematically represented in Equation (19). In order to solve Equation (18), we apply KKT conditions.

$$\mathcal{L} = U_i^{t_k} + \mu_1(M_i^T - M_i^{ut,t_k}) - \mu_2\mathbb{F}_i^{t_k} + \mu_3(1 - C_{ij}^{pre,t_k}) - \mu_4 S_i^{t_k} + \mu_5(1 - \mathcal{P}_i^{t_k}) + \mu_6(1 - I_i^{t_k}) + \mu_7(1 - \lambda_i^{t_k}) \tag{19}$$

The *dual feasibility* and *complementary slackness* of the KKT conditions are given in Equations (20a) and (20b).

$$\nabla_{M_i^{ut,t_k}} \mathcal{L} = 0 \tag{20a}$$

$$\mu_i(X) = 0, \text{ and } \mu_i \geq 0, \quad \forall i = \{1, 2, \cdots, 8\} \tag{20b}$$

where $\mu_i(X)$ represent the *Lagrangian* multipliers. In the Equation (20b), X represents the constraints of Equation (18). On solving the KKT conditions, we obtain the optimal value of M_i^{ut,t_k} for which the utility of the edge node is minimum. Further, the condition, $\left(\left(\frac{min(\sigma_{xy}^{t_k})}{\mathbb{PEC}_i^{t_k} \times M_i^T}\right)^2 - 4\beta m\mu_1\right) \geq 0$ depicts the presence of real roots. Therefore, the optimal value of the utilized memory obtained is represented in Equation (21).

$$M_i^{ut,t_k*} = \frac{1}{2\mu_1}\left(\sqrt{\left(\frac{min(\sigma_{xy}^{t_k})}{\mathbb{PEC}_i^{t_k} \times M_i^T}\right)^2 - 4\beta m\mu_1} - \left(\frac{min(\sigma_{xy}^{t_k})}{\mathbb{PEC}_i^{t_k} \times M_i^T}\right)\right) \tag{21}$$

■

IV. Performance Evaluation

A. Simulation Design

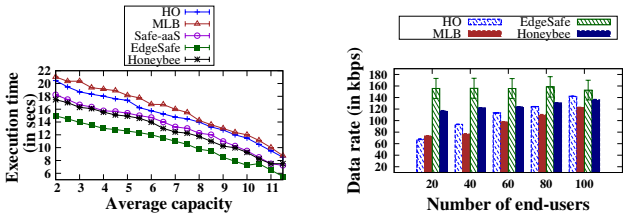
To evaluate the performance of the proposed scheme, EdgeSafe, we consider the presence of 100–1000 heterogeneous static and mobile sensor nodes and 45–75 edge nodes randomly deployed over the simulation area of $10km \times 10km$. We assume that the cloud/server is placed near the i^{th} edge node. In Safe-aaS, the mobility of the mobile sensor nodes is an important causal factor for load balancing among the edge nodes present within their proximity. We consider that the cloud is centrally placed in the simulation environment. We consider that 10 time instants are present in each day for a period of 20 days. We estimate the speed of the sensor node using the *Gauss-Markov mobility* model [23]. The speed of any mobile sensor node is computed as:

$$s_n = \alpha s_{n-1} + (1 - \alpha)\bar{s} + \sqrt{(1 - \alpha \times \alpha)} \times s_{x_{n-1}} \tag{22}$$

where α is the tuning parameter and $\bar{s}$ is the mean speed. $s_{x_{n-1}}$ represents the random variable from a Gaussian distribution that assigns randomness to the speed of the sensor node. The different parameters considered for simulation are listed in Table I. Further, in each iteration, we estimate the number of edge nodes present within the communication range of each edge node.

B. Benchmark

We evaluate and compare the performance of the proposed scheme, EdgeSafe with three existing schemes proposed by Lobinger *et al.* [3], Park *et al.* [2], and Fernando [4]. Park *et al.* [2] proposed a self-organizing mobility load balancing method to stabilize the network as well as improve the efficiency of the resources. On the other hand, Lobinger *et al.* [3] considered a long-term-evolution (LTE) communication



(a) Average capacity (b) Data rate

Fig. 2: Variation of average capacity and datarate

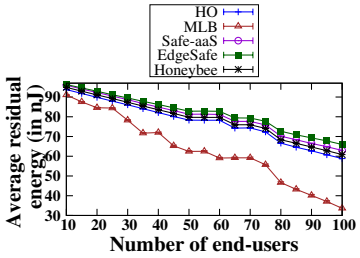
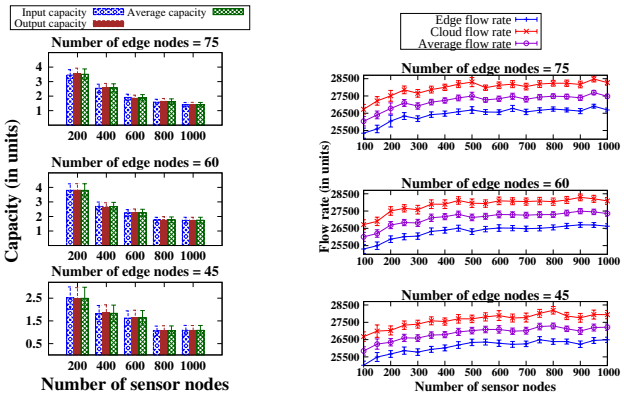


Fig. 3: Variation of average residual energy



(a) Capacity (b) Flow-rate

Fig. 4: Variation of capacity and flow-rate with time

TABLE I: Simulation Parameters

Parameter	Value
Type of sensor nodes	5
Number of end-users	10-100
Communication Range	$25 - 80m$
Bandwidth of the data [24]	$10 - 20MHz$
Packet forwarding rate [25]	$4 - 8$ kbps
Frequency of the data packet [24]	$800 - 2600$ MHz
Velocity of mobile sensor node [26]	$25 - 105$ Kmph
Incentive received ($I_i^{t_k}$)	$20 - 30$ units

system and proposed a self-optimizing load balancing algorithm. Similarly, Fernanda *et al*. [4] proposed a load sharing algorithm among the nearby heterogeneous mobile devices through the adoption of the work stealing algorithm. However, none of these existing schemes consider the capacity of the devices before sharing/distributing the load among them. We label the algorithm proposed by Lobinger *et al.* [3] as HO, Park *et al.* [2] as MLB, and Fernando *et al*. as Honeybee. Fig. 2(a) illustrates the variations in the execution time of the edge nodes with the increase in their capacity from 2–11.5 with an interval of 0.5 along the x-axis. We consider the presence of 45 edge nodes in the simulation environment. We observe that the execution time reduces by 7.95%, 8.35%, and 11.32% respectively, compared with the HO, MLB, and Honeybee algorithms. Additionally, we observe that the execution time reduces by 6.01% with the integration of our algorithm into the existing Safe-aaS architecture. Further, we observe that the execution time follows a decreasing trend with the increase in the capacity of the edge nodes.

In Fig. 2(b), we analyze the variations of data rate in case of EdgeSafe, HO [3], MLB [2], and Honeybee [4] algorithm. We vary the number of end-users from 20–100 with an interval of 20. We observe that the average data rate increases by 53.13% and 36.4%, in the case of HO and MLB. However, the average data rate is much higher in amplitude in our proposed algorithm, EdgeSafe. In addition to this, the packet forwarding rate of the edge nodes varies with the increase in the number of end-users in the simulation area.

We represent residual energy of a sensor node at any time instant, t_k as the remaining energy of that node. We compare the average residual energy of the proposed algorithm, EdgeSafe, with the existing algorithms, HO, MLB, and Honeybee. In Fig. 3, we observe that the residual energy reduces with the increase in the number of end-users. We increase the number of end-users from 10–100 along the x-axis. The probable reason behind this decreasing pattern is that the amount of energy consumed to provide safety-related decisions increases with the increase in the number of end-users. Therefore, the average residual energy decreases.

C. Results

We study and analyze our load balancing scheme among the edge nodes through several experimental conditions, which are discussed considering the following performance metrics:

Variation of capacity and flow-rate of edge nodes: Fig. 4(a) illustrates the variation of capacity of the edge nodes present in the neighborhood of the selected sensor nodes. Along the x-axis, we vary the number of sensor nodes starting from 200 up to 1000, with an interval of 200. Fig. 4(a) demonstrates the variations in the input, output, and average of the edge nodes. We observe that the input, output, and average capacity of the edge nodes improves by 5.77%, 12.89%, and 6.05% with the increase in the number of edge nodes from 45–75. The probable reason behind such a trend is that the input capacity function and output capacity function depends on the edge flow rate, ingress component, cloud flow rate, and egress component. Moreover, with the increase in the number of sensor nodes, the data arrival rate and the number of data packets transmitted from the sensor nodes to the edge nodes within their proximity vary.

Fig. 4(b) illustrates the variations of the average flow rate of edge nodes present in the neighborhood of the active sensor nodes. Along the x-axis, we increase the number of sensor nodes starting from 100 upto 1000, in steps of 100. We observe that the edge, cloud, and average flow-rate increases by 0.5%, 1.87%, and 1.96% with the increase in the number of edge nodes from 45–75. Interestingly, we notice that the rate of increase in the flow-rate drops after a certain number of sensor nodes. One of the factors behind such a trend is the distance between the edge node and the mobile sensor node varies with the mobility of the vehicles. Consequently, we also observe that there exists more randomness in the cloud flow-rate in case of CFR_{e_i}, rather than in the case of EFR_{e_i} in the presence of 45, 60, and 75 edge nodes in the scenario. The probable reason behind this is the average packet forwarding rate from the edge nodes to the cloud is comparatively higher than the packet forwarding rate of the sensor nodes to the edge nodes.

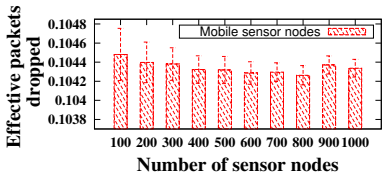


Fig. 5: Variation of packet dropped by mobile sensor nodes

Effective packets dropped: Fig. 5 illustrates the effective packet dropped by the mobile sensor nodes in the presence of 100–1000 sensor nodes in the scenario. We define effective packets dropped as the ratio of the number of data packets dropped by the sensor node to the number of data packets transmitted by that node. We observe that the effective packet drop ratio does not fluctuate significantly with the increase in the number of sensor nodes. Therefore, the effective packet drop ratio is independent of the number of sensor nodes present in the scenario.

Profit return by the edge device: Fig. 6 demonstrates the profit return of the edge devices in the presence of 100, 200, and 300 sensor nodes. We vary the number of edge nodes from 45–180, in steps of 15, along the x-axis. We observe that the profit of the edge nodes fluctuates randomly. The probable reason behind this is that the incentive received by the edge nodes depends upon the number of times the edge node was selected for load balancing. Further, the probability of the number of data packets successfully delivered by any edge node also varies with time. Therefore, we conclude that the profit of the edge nodes is independent of the number of sensor nodes present in the simulation area.

Risk incurred in selecting the edge node: Fig. 7 demonstrates the analysis of the risk incurred in selecting the edge node for load balancing. Along the x-axis, we represent the number of edge nodes present in the simulation environment. We vary the number of edge nodes from 45 to 180 with an interval of 15. We observe that the risk incurred in selecting the edge nodes for sharing load vary randomly in the presence of 100, 200, and 300 sensor nodes. The amount of memory utilized by the edge nodes depend on the number of processes executed at any time instant, which may vary randomly. We compute the risk incurred in selecting the edge nodes using Equation 15. The profit return of the edge nodes also fluctuate randomly, therefore, the variation in profit return changes. Consequently, the risk incurred in selecting the edge nodes vary.

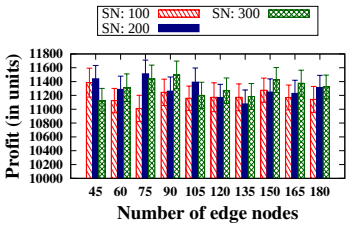


Fig. 6: Profit return

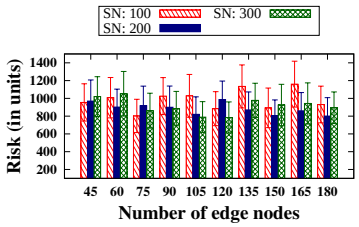


Fig. 7: Risk incurred

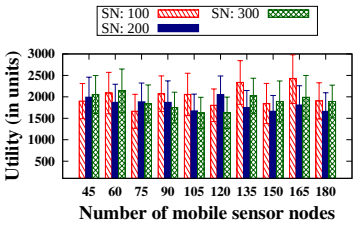


Fig. 8: Utility

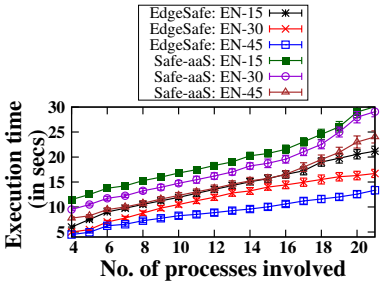


Fig. 9: Execution time of edge nodes

Utility of the edge device: Fig. 8 illustrates the utility of the edge nodes in the presence of 100, 200, and 300 sensor nodes. We observe a random pattern in the profit return of the edge nodes in Fig. 6 and risk incurred in selection of the edge nodes in Fig. 7. Further, we compute the utility of the edge nodes depending upon the risk incurred in selecting them for load balancing and their effective spare memory as given in Equation 17. Therefore, the utility of the edge nodes do not depend upon the number of edge nodes present in the simulation area.

Execution time: Fig. 9 demonstrates the variations in the execution time of our proposed scheme with the increase in the number of processes being executed at the edge nodes. We define execution time as the time required to distribute the available time-critical data dynamically among the edge nodes. We estimate the execution time in the presence of 15, 30, and 45 edge nodes in traditional Safe-aaS and EdgeSafe (after load sharing in Safe-aaS). In this figure, we observe that the execution time in EdgeSafe with 15 edge nodes in the simulation environment is almost similar to that of 45 edge nodes in Safe-aaS. We observe that the execution time decreases by 55.24%, 59.98%, and 63.496% in our proposed scheme, EdgeSafe, compared to the existing Safe-aaS architecture, in the presence of 15, 30, and 45 edge nodes, respectively.

V. Conclusion

In this work, we propose a novel scheme of dynamic load distribution among the edge nodes for provisioning Safe-aaS. With the change in the location of the vehicles, the sensor nodes deployed in the vehicles attain mobility. Therefore, the distance between the mobile sensor nodes and the neighboring edge nodes varies at every time instant. The load in the edge nodes also varies with time. Additionally, the edge nodes are heterogeneous in nature and the capacity of the various edge nodes differ. Thus, dynamic sharing of the resources among the edge nodes, based on their capacity, is necessary. As per our knowledge, this is the one of the first attempt in road transportation industry, where load distribution among edge nodes is considered.

As per the existing research work, the theoretical model of the Safe-aaS architecture and cost analysis is proposed. We plan to implement the Safe-aaS infrastructure with a dynamic load balancing scheme, EdgeSafe in the real-life scenario. As multiple steps are involved in the registration process, the type of end-users, and their demand may vary, we plan to design the microservice architecture in the Safe-aaS infrastructure to meet the users' demand. Safe-aaS platform is proposed for providing safety-related decisions to the end-user. Therefore, we plan to incorporate safety and security in the different layers of the architecture from the users' and Safety Service Provider's perspective.

References

[1] C. Roy, A. Roy, S. Misra, and J. Maiti, "Safe-aaS: Decision Virtualization for Effecting Safety-as-a-Service," *IEEE IoT J.*, vol. 5, no. 3, pp. 1690–1697, June 2018.

[2] J. Park, Y. Kim, and J. R. Lee, "Mobility Load Balancing Method for Self-Organizing Wireless Networks Inspired by Synchronization and Matching With Preferences," *IEEE Trans. Veh. Techol.*, vol. 67, no. 3, pp. 2594–2606, March 2018.

[3] A. Lobinger, S. Stefanski, T. Jansen, and I. Balan, "Load Balancing in Downlink LTE Self-Optimizing Networks," in *IEEE* 71^{st} *VTC*, May 2010, pp. 1–5.

[4] N. Fernando, S. W. Loke, and W. Rahayu, "Computing with nearby mobile devices: A work sharing algorithm for mobile edge-clouds," *IEEE Transactions on Cloud Computing*, vol. 7, no. 2, pp. 329–343, 2019.

[5] S. Lee, C. Oh, and S. Hong, "Exploring Lane Change Safety Issues for Manually Driven Vehicles in Vehicle Platooning Environments," *IET Intell. Transp. Syst.*, vol. 12, no. 9, pp. 1142–1147, 2018.

[6] U. Budak, V. Bajaj, Y. Akbulut, O. Atila, and A. Sengur, "An Effective Hybrid Model for EEG-Based Drowsiness Detection," *IEEE Sensors J.*, vol. 19, no. 17, pp. 7624–7631, Sep. 2019.

[7] Y. Yu, H. Guan, and Z. Ji, "Automated Detection of Urban Road Manhole Covers Using Mobile Laser Scanning Data," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 6, pp. 3258–3269, Dec 2015.

[8] X. Sun and N. Ansari, "Traffic Load Balancing Among Brokers at the IoT Application Layer," *IEEE Trans. Netw. Service Manag.*, vol. 15, no. 1, pp. 489–502, March 2018.

[9] J. Jijin, B. C. Seet, P. H. J. Chong, and H. Jarrah, "Service Load Balancing in Fog-Based 5G Radio Access Networks," in *IEEE* 28^{th} *Annual Int. Symposium on PIMRC*, Oct 2017, pp. 1–5.

[10] S. Toumpis and S. Gitzenis, "Load Balancing in Wireless Sensor Networks using Kirchhoff's Voltage Law," in *IEEE INFOCOM*, April 2009, pp. 1656–1664.

[11] T. Han and N. Ansari, "A traffic load balancing framework for software-defined radio access networks powered by hybrid energy sources," *IEEE/ACM Trans. Netw.*, vol. 24, no. 2, pp. 1038–1051, April 2016.

[12] A. Talukder, S. F. Abedin, M. S. Munir, and C. S. Hong, "Dual Threshold Load Balancing in SDN Environment using Process Migration," in *in Proc. of ICOIN*, Jan 2018, pp. 791–796.

[13] A. Khaligh and M. D'Antonio, "Global Trends in High-Power On-Board Chargers for Electric Vehicles," *IEEE Trans. Veh. Technol.*, vol. 68, no. 4, pp. 3306–3324, April 2019.

[14] A. Desreveaux, A. Bouscayrol, R. Trigui, E. Castex, and J. Klein, "Impact of the Velocity Profile on Energy Consumption of Electric Vehicles," *IEEE Trans. Veh. Technol.*, vol. 68, no. 12, pp. 11 420–11 426, Dec 2019.

[15] Y. Dai, D. Xu, S. Maharjan, and Y. Zhang, "Joint load balancing and offloading in vehicular edge computing and networks," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4377–4387, 2019.

[16] C. Roy, S. Misra, J. Maiti, and M. S. Obaidat, "DENSE: Dynamic Edge Node Selection for Safety-as-a-Service," in *IEEE Global Communications Conference (GLOBECOM)*, 2019, pp. 1–6.

[17] C. Roy, S. Misra, and S. Pal, "Blockchain-enabled safety-as-a-service for industrial iot applications," *IEEE Internet of Things Magazine*, vol. 3, no. 2, pp. 19–23, 2020.

[18] M. F. E. David A. Brannan and J. J. Gray, *Geometry*. Cambridge University Press, 1999.

[19] B. Ho, C. Pham-Quoc, T. N. Thinh, and N. Thoai, "A Secured OpenFlow-Based Switch Architecture," in *Int. Conf. ACOMP*, Nov 2016, pp. 83–89.

[20] H. Markowitz, "Portfolio Selection," *The Journal of Finance*, vol. 7, no. 1, pp. 77–91, 1952.

[21] S. Bera, P. Gupta, and S. Misra, "D2s: Dynamic demand scheduling in smart grid using optimal portfolio selection strategy," *IEEE Transactions on Smart Grid*, vol. 6, no. 3, pp. 1434–1442, 2015.

[22] J. Xu, P. B. Luh, F. B. White, E. Ni, and K. Kasiviswanathan, "Power portfolio optimization in deregulated electricity markets with risk management," *IEEE Transactions on Power Systems*, vol. 21, no. 4, pp. 1653–1662, 2006.

[23] T. Camp, J. Boleng, and V. Davies, "A survey of mobility models for ad hoc network research," *Wireless communications and mobile computing*, vol. 2, no. 5, pp. 483–502, 2002.

[24] D. Calabuig, D. Martín-Sacristán, J. F. Monserrat, M. Botsov, and D. Gozálvez, "Distribution of Road Hazard Warning Messages to Distant Vehicles in Intelligent Transport Systems," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 4, pp. 1152–1165, April 2018.

[25] B. Das, S. Misra, and U. Roy, "Coalition Formation for Cooperative Service-Based Message Sharing in Vehicular Ad Hoc Networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 27, no. 1, pp. 144–156, Jan 2016.

[26] N. Kumar, S. Misra, J. J. P. C. Rodrigues, and M. S. Obaidat, "Coalition Games for Spatio-Temporal Big Data in Internet of Vehicles Environment: A Comparative Analysis," *IEEE IoT J.*, vol. 2, no. 4, pp. 310–320, Aug 2015.

Dr. Sudip Misra received the bachelor's degree from the Indian Institute of Technology, Kharagpur, India; the master's degree from the University of New Brunswick, Fredericton, NB, Canada; and the Ph.D. degree in computer science from Carleton University, Ottawa, ON, Canada. He is a Professor with the Department of Computer Science and Engineering, Indian Institute of Technology, Kharagpur. He has several years of experience working in the academia, government, and the private sectors in research, teaching, consulting, project management, architecture, software design, and product engineering roles.

Currently, Dr. Misra is the Associate Editor of the IEEE Transactions Mobile Computing and IEEE Systems Journal. He is the Editor of the IEEE Transactions on Vehicular Computing. He is the author of over 290 scholarly research papers (including 160 journal papers). His current research interests include algorithm design for emerging communication networks and Internet of Things. He was awarded the Canadian Government's prestigious NSERC Post Doctoral Fellowship and the Humboldt Research Fellowship in Germany.

Dr. Jhareswar Maiti received the bachelors degree from the Department of Mining Engineering, Indian Institute of Engineering Science And Technology, Shibpur, India; the masters and the Ph.D. degree in Mining Engineering, Indian Institute of Technology, Kharagpur, India. He is a Professor with the Department of Industrial & Systems Engineering, Indian Institute of Technology, Kharagpur. He has several years of teaching and research experience on safety analytics, quality analytics, engineering ergonomics, and virtual reality applications.

Currently Dr. Maiti is the Associate Editor of the Journal of Safety Science. He is the Editorial Board Member of the International Journal of Injury Control and Safety Promotion and Safety and Health at Work. His current research interests include occupational safety and health analytics, quality analytics and virtual reality based accident modeling and simulation. He has established a unique, first of its kind laboratory "Safety Analytics and Virtual Reality Laboratory" in IIT Kharagpur. The details about the laboratory and the current profile of Prof Maiti can be found at www.savr.iitkgp.ac.in.

Chandana Roy is a Institute Scholar and is pursuing her PhD from the Department of Industrial and Systems Engineering, Indian Institute of Technology Kharagpur, India. She received her M.Tech and B.Tech degree from National Institute of Technology Durgapur, India in 2012, and West Bengal University of Technology, India in 2010, respectively. The current research interests of Ms. Roy include Industrial Internet of Things, Wireless Body Area Networks, and Cloud Computing. She is a student member of the IEEE.

Soumi Nag is working as a Software Engineer in Advanced Analytics in the Abzooba India Pvt. Ltd., India. Prior to that, she completed her B. Tech in Computer Science from the National Institute of Technology Durgapur, India. This work was done during her internship at IIT Kharagpur. Her research interest includes Wireless Sensor Networks and Internet of Things.