Mobility Aware Blockchain Enabled Offloading and Scheduling in Vehicular Fog Cloud Computing

Abdullah Lakhan, Muneer Ahmad, Muhammad Bilal, *Senior Member, IEEE*, Alireza Jolfaei, *Senior Member, IEEE*, and Raja Majid Mehmood, *Senior Member, IEEE*

***Abstract*— The development of vehicular Internet of Things (IoT) applications, such as E-Transport, Augmented Reality, and Virtual Reality are growing progressively. The mobility aware services and network-based security are fundamental requirements of these applications. However, multi-side offloading enabling blockchain and cost-efficient scheduling in heterogeneous vehicular fog cloud nodes network become a challenging task. The study formulates this problem as a convex optimization problem, where all constraints are the convex set. The goal of the study is to minimize communication cost and computation cost of applications under mobility, security, deadline, and resource constraints. Initially, we propose a novel vehicular fog cloud network (VFCN) which consists of different components and heterogeneous computing nodes. The ensure mobility privacy, the study devises Mobility Aware Blockchain-Enabled offloading scheme (MABOS). It extends blockchain enable multi-side offloading (e.g., offline offloading and online offloading) with proof of work (PoW), proof of creditability (PoC) and fault-tolerant techniques. The purpose is to offload all tasks under the secure network without any violation. Furthermore, to ensure Quality of Service (QoS) of applications, this work suggests linear search based task scheduling (LSBTS) method, which maps all tasks onto appropriate computing nodes. The experimental results show that devise schemes outperform all existing baseline approaches to the considered problem.**

***Index Terms*— Task offloading, Internet of Things, task scheduling, VFCN, mobility, task deadline, system costs.**

I. Introduction

THESE days, mobility aware Internet of Things (IoT) applications have been growing progressively [1]. Due to resource-constraint (limited storage, battery, and CPU) issues of local devices, these applications rely on remote cloud resources to offload and execute requested complete tasks [2]. Such cloud systems are also able to support workloads by storing data and processing offloaded tasks [3]. In contrast, centralized cloud computing can cope with many delay-tolerant types of applications and tasks (e.g., file system, database type tasks). Due to multiple hops away from users, lower latency type applications cannot run on cloud computing. The fog computing is a new distributed computing prototype that leverage cloud capacity closer to users. The fog paradigm supports low latency applications with minimum execution cost. Furthermore, fog cloud-based vehicular network supports both delay-tolerant and delay-sensitive IoT applications with mobility features in the environment. Therefore, fog-cloud nodes could be used together to achieve applications goals.

Existing the mobile ad hoc network (MANET) and the Vehicular ad hoc network (VANET) are widely used to run mobility-based vehicle applications [4]. Furthermore, the high-speed access networks, such as 5G Dynamic Spectrum Access (DSA) and high-speed WiFi, offers seamless links in the [5] to boost the performance of the vehicle network. The DSA-based base station and VFCN have ubiquitous communication to run any program [6]. For E-Transport applications, the cloud services are required to integrate at the roadside unit of the network [7]. It is called Vehicle to Infrastructure (V2I).

However, VFCN distributed systems has a security issue which is an important challenge in the network. Trust aware resources in VFCN reflects users' decision on the authenticity, integrity, security and stability of services provided by various service providers. Authenticity refers to whether the fog and cloud nodes based services have real identity claims. Therefore, it is a challenge to how to ensure secure multi-side offloading and scheduling in the VFCN for E-Transport applications. Blockchain enables technology is a promising solution, which consists of the growing list of data, named blocks. These blocks are connecting through cryptographic hash values. Each block involves a cryptographic fingerprint of the previous block, includes timeframe and transaction data. These essential characteristics of blockchain can be employed in VFCN to address the security and trust issues.

The paper formulates blockchain-enabled and mobility aware task multi-side offloading and scheduling problem with vehicular fog cloud network. The objective is to minimize communication cost and process cost of vehicular applications. The problem constraints, i.e., network bandwidth, the deadline of tasks, resource capacity of nodes are taking into considering during formulation. The nodes are a combination of

Manuscript received February 19, 2020; revised August 15, 2020; accepted January 22, 2021. Date of publication February 17, 2021; date of current version July 12, 2021. This work was supported by the Xiamen University Malaysia Research Fund (XMUMRF) under Grant XMUMRF/2019-C3/IECE/0007. The Associate Editor for this article was S. Mumtaz. *(Corresponding authors: Muhammad Bilal; Raja Majid Mehmood.)*

Abdullah Lakhan is with the College of Computer Science and Artificial Intelligence, Wenzhou University, Wenzhou 325035, China (e-mail: abdullahrazalakhan@gmail.com).

Muneer Ahmad is with the Department of Information System, Faculty of Computer Science and Information Technology, University of Malaya, Kuala Lumpur 50603, Malaysia (e-mail: mmalik@um.edu.my).

Muhammad Bilal is with the Department of Computer and Electronic Systems Engineering, Hankuk University of Foreign Studies, Yongin 17035, South Korea (e-mail: m.bilal@ieee.org).

Alireza Jolfaei is with the Department of Computing, Macquarie University, Sydney, NSW 2113, Australia (e-mail: alireza.jolfaei@mq.edu.au).

Raja Majid Mehmood is with the Information and Communication Technology Department, School of Electrical and Computer Engineering, Xiamen University Malaysia, Sepang 43900, Malaysia (e-mail: rmeex07@ieee.org).

Digital Object Identifier 10.1109/TITS.2021.3056461

heterogenous decentralized fog nodes and centralized cloud to process generated requests by users. The Blockchain is a distributed decentralized network that provides immutability, privacy, security, and transparency. It is centralized to authenticate and validate the transactions, yet each transaction in the Blockchain is supposed to be effectively ensured and tested. It to be done via consensus protocol, that is a part of the Blockchain network. However, there are many challenges in VFCN which are to be investigated more.

Research Questions: There is a large room of challenges in VFCN. However, the study is considering these questions. (i) How to design a secure system for E-Transport applications? (ii) How to perform offloading and scheduling for E-Transport applications in the VFCN system which satisfies all constraints of problem during processing.

- The paper proposes a novel vehicular fog cloud network (VFCN) architecture with blockchain multi-side offloading (offline offloading and online offloading) with mobility, fault-tolerant and mobility constraints. However, all existing up to architectures, i.e., Vehicular Ad hoc Network (VANET) and Mobile ad hoc network (MANET) offers only single side offloading enabling with mobility features without fault-tolerant mechanism. The existing architectures exploited asymmetric load balancing and sharing of resource technique. Therefore network-based security cannot be implemented in this feature. We suggest asymmetric load balancing and heterogeneous computing nodes which offers network-based security and control based single master node. we devise Mobility Aware Blockchain Enabled offloading scheme (MABOS) which handles blockchain enable multi-side offloading (e.g., offline offloading and online offloading) of tasks with proof of work (PoW), proof of creditability (PoC) and fault-tolerant techniques.
- All existing scheduling methods rarely consider the mobility and network-based security features during resource allocation onto heterogeneous computing nodes. We are different than existing studies in the following way. We devise Linear Searching Based Task Scheduling (LSBTS) with sequencing, initial scheduling and task migration components in VFCN. However, existing studies [1]–[4] did not consider these components during resource allocation which has directly impact on application costs.

The rest of the paper is organized as follows. Section II is about Related Work of existing works. Section III explains the proposed description and formalizes the problem under study. The proposed schemes for the considered problem is explained in Section IV. Section V is the performance evaluation part. Section VI is about the conclusion of the proposed work.

II. Related Work

The mobility based offloading and scheduling in the fog cloud network is a widely investigated problem. Many practical applications utilize the aforementioned network for their executions. These constraints, e.g., deadline, security, nodes resources, are taken into consideration with different objectives such as system energy, latency, response time and costs. The costs determined by the communication cost and computation cost. Existing studies addressed multi-parameters based offloading and cost-aware scheduling based problem and implied architectures and frameworks.

Existing studies addressed multi-parameters based offloading and scheduling based on VANET and MANET architectures for, especially vehicular applications. Contract and sharing resources and time-efficient and Mobility-Aware Task Scheduling in Cloud-Fog IoT schemes advised in [1], [2]. For the offloading decision, deadline, resource (CPU) and bandwidth parameters are taken into consideration. The goal was to minimize service and communication costs during scheduling. These studies [3]–[5] formulated cost-efficient task scheduling problem under deadline constraint in MANET (i.e., only edge nodes). The objective was to minimize execution cost during scheduling.

A mobility aware secured device-based data communication scheme and mobile Drone Assisted task scheduling schemes in MANET investigated extensively in literature studies [6]–[8]. Based on MANET, these studies proposed FCFS, Opportunistic Load Balancing and min-Max schemes to solve offloading and scheduling problem with security constraints. The Software-Defined Network helps to geographically distribute services of computing nodes. The Software-Defined Network-based mobility aware dispersed fog nodes services aware task scheduling schemes investigated by these works [9]–[12]. The primary goal is to deploy computing and communication nodes at the road unit side for vehicular applications in MANET. The malicious activities, latency and resource-limitation, were elements taken into account. The goal was to minimize latency and response time of applications at the users level. However, the studies above solved either offloading or scheduling for IoT applications individually in their considered problems. Furthermore, joint offloading and scheduling problems are analyzing [13]–[16] with security constraint in vehicular edge network. The methods based on simulated annealing linear search, local search and global, and different earliest finish were suggested. However, certain methods offer centralized security at particular nodes (e.g., either on local nodes and computation/communication nodes) without considering the entire network situation.

Moreover, based on network security, especially with the introduction of blockchain technology, much decentralized aware security-aware mechanism were discussed. For instance, the works [17]–[20] introduced blockchain enable and mobility aware task scheduling models in the distributed vehicular network. The public blockchain-based security their particular methods were discussed. The decentralized nodes (e.g., cloudlet, edge, and fog) and centralized cloud were taken as the processing nodes for offloading tasks. Many objectives were obtained, i.e., minimize latency, communication and computation costs, and increase the authentication with proof of works, and proof of crediability techniques. The private blockchain aware methods and architectures with consensus and failure aware techniques were examined for applications [21]–[23].

To the best our knowledge, blockchain-enabled mobility aware multi-side offloading and scheduling problem in

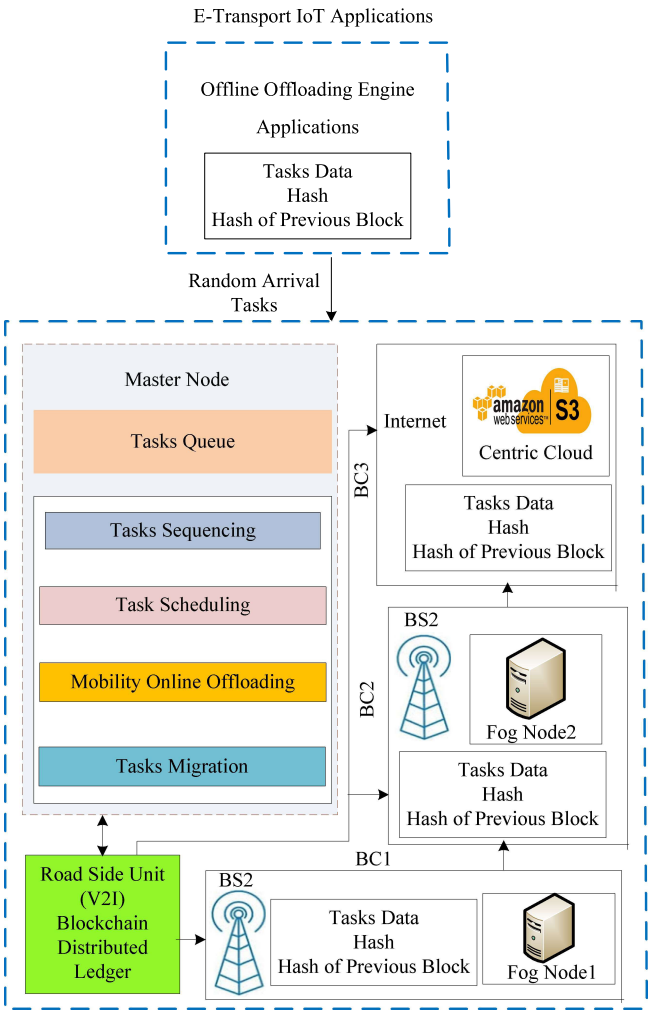

Fig. 1. Blockchain aware vehicular fog cloud network.

vehicular fog cloud network has not been studied yet. The significant constraints such as offloading time, security, resource and deadline constraints have widely ignored by existing studies. Recently, published [24] investigated Blockchain-Based Secure Spectrum Trading for Unmanned-Aerial-Vehicle-Assisted Cellular Networks. However, they only focused on communication security instead of computation security, and they did not consider multi-side offloading safety during mobility. We proposed a Mobility Aware Blockchain-Enabled offloading scheme (MABOS) to ensure privacy, offloading time constraints of tasks and different earliest finish time-based task scheduling algorithm under deadline, resource and security constraints. Due to mobility, an initial solution could not be optimal. Therefore we propose a linear Search-Based Task Scheduling (LSBTS) algorithm, which determines the allocations of tasks to the resources.

III. Problem Formulation

The proposed vehicular fog cloud network consists of different components as shown in Figure 1. The offloaded tasks arrive randomly to the system. The system consists of the master node and the tasks queue. The queue initially arranges all tasks into first-come-first-serve order. Furthermore, the master node consists of the following elements: Task Sequencing, Task Scheduling, Mobility Online Offloading, and Task Migration. We will explain these elements later in detail. The primary role of master node allocates all tasks to the appropriate nodes. All the assignment must satisfy tasks constraints requirements and ensure security during offloading and scheduling. Task Queue in VFCN implemented with multiple heterogeneous nodes which are handled by the Master Node. Load Balancing is the aspects which keep tasks evenly distributed across all nodes in VFCN. It is essential to keep the workload balanced among all nodes to fully utilize the benefits of having more than one node else one or more node will sit idle while other nodes have high workloads along with lists of node awaiting the CPU. We implement push Migration routinely checks the load on each processor. If it finds an imbalance, then it evenly distributes the weight on each node by moving the processes from overloaded to idle or less busy processors.

Road Side Unit enables Blockchain-Enable Distributed Ledger with different computing/communication nodes at Vehicle-to-infrastructure. Each node connected with the associated base station (BS) to facilitated applications. The fog nodes offer services at the edge of the network, and Amazon cloud services at the corner of the Internet. The VFCN presents the hierarchical, bi-directional computing infrastructure: IoT applications communicate with fog nodes communicate with Amazon public cloud. All fog nodes can also interact with each other to perform data and process management to support application requirements and to exchange data.

A. System Model

The paper considers A number of applications, i.e., $A\{a_1, \ldots, aA\}$. Each application has N number of tasks, they contain two types of tasks such as general tasks and security tasks. The tasks are denoting by $i \in \{1, 2, \ldots, N\}$. Each task has the following attributes, e.g., $\{SC_{a,i}, S_{a,i}, W_{a,i}, d_{a,i}, F_{a,i}\}$. Where $SC_{a,i}$ denotes a security annotated task, $S_{a,i}$ denotes data size, $W_{a,i}$ shows they require CPU instruction to process data size, $d_{a,i}$ illustrates a deadline, and $F_{a,i}$ denotes finish time. All requested workloads to be followed by Poisson process. Furthermore, the study shows K numbers of hybrid nodes, i.e., $K = \{k = 1, 2, \ldots, K\}$. We denote the initial location of an application when it is roaming among networks and indicated by l_a^0 when the requested tasks for offloading.

The VFCN architecture made up of fog and cloud nodes, and these nodes are distinct by their computing speeds depicts by $\zeta_j\{1, 2, \ldots, K\}$. We employed the variable $x_{a,i,k} \in \{0, 1\}$ which the assignment of task i of application to the k^{th} node. If $x_{a,i,k} = 1$ otherwise looking for assignment to any node. We denote execution time of tasks by $T^e_{\{a,i\}} = \sum_{k=1}^{K} \frac{W_{a,i}}{\zeta_j}$. The notation Φ_{comp} shows computing cost of k^{th} during execution.

We are considering random the offloading scheme for application a, which may offloads computation tasks to the VFCN architecture any time via the V2I communication network. Each server in the VFCN can migrate some workloads to

another neighbour server for further execution. The distance and transmission rate between applications and available services measures as follows.

$$\sum_{a=1}^{A} B_a = log_2\left\{1 + \frac{P_{tx} h_{ag} (d_{ag})^{-} r}{\overline{w}_o}\right\}, \quad (1)$$

whereas, B_a is available bandwidth of network of application a, P_{tx} denotes transmission power, h_{ag} illustrates channel gain, $\overline{w}_o$ shows level of inference-noise, d_{ag} denotes distance between requested tasks and available services. The optimal offloading time is a key constraint which allows the offloading engine to make the decision either to migrate tasks or not. The key parameters of the communication model measure as follows.

$$d_{ag} = \sqrt{l_g^2 + (D_g/2 - l_a^x - \upsilon_a - \tau_a)^2}, \quad if\ l_a^x \leq D_g/2, \quad (2)$$

l_g signifies base-station hop height which links nodes in VFCN, and D_g is optimal coverage area in base-stations. If the $x = 1$, notation $l_a^x \leq D_g/2$ shows lower distance between request and available and incur lower communication cost as defined in equation (2). And if $x = 0$, then $l_a^x > D_g/2$ tells due to high distance and communication cost is higher.

These symbols B_a is showing bandwidth cost and data sending time τ_u. We are measuring cost of bandwidth utilization B_a by i.e., Φ_{com}. The symbol F_a^{com} calculates communication as follows.

$$F_a^{com} = \Phi_{com} \times \tau_u \times B_a. \quad (3)$$

The data size of the task $S_{a,i}$ directly affects on the bandwidth cost when it offloads to any node for execution. It is critical to offload tasks when network has good enough bandwidth with minimum communication cost. We exploit a binary variable $\alpha_{a,i}$, if $\alpha_{a,i} = 1$ it means the task i is ready to offload, where $\alpha_{a,i} = 0$ shows the task is waiting for offloading. We measured the offloading time in the following way:

$$\tau_u = \sum_{i \in a}^{N} \alpha_{a,i} \frac{S_{a,i}}{B_a}. \quad (4)$$

We measure average execution time of all tasks of the particular application as follows:

$$T_{a,i}^e = \sum_{j \in k}^{K} \sum_{i \in a}^{N} x_{a,i,k} \times \frac{W_{a,i}}{\zeta_j}. \quad (5)$$

The computation cost of each application determines in the following way.

$$F_a^{comp} = \sum_{i=1}^{N} T_{a,i}^e \times \Phi_{comp}. \quad (6)$$

The total time of a task consists of offloading time, processing time and feedback time. The feedback time is fixed and denoted by notation t_B. We measured the total time of all applications as follows

$$T_{a,i} = \max \sum_{a=1}^{A} \sum_{i \in a}^{N} T_{a,i}^e + \tau_u + t_B. \quad (7)$$

The total costs (i.e., communication cost and computation cost) of all applications as

$$F_a = \sum_{a=1}^{A} F_a^{com} + F_a^{comp}. \quad (8)$$

The objective function of the paper is to minimize costs of all applications while satisfying their security and deadlines. The task offloading decision and task assignment $x_{a,i,k}$, security SC_i, bandwidth B_a, offloading time τ_a, and execution time $T_{\{a,i,k\}}^e$ constraints are mutually optimized as a convex function via Integer Linear Programming (ILP) method. Thus, the convex optimization function **P1** to the consider problem formulated as

$$\left\{ \min_{x_{a,i,k}, B_a, SC_{a,i}, \tau_a, T_{\{a,i,k\}}^e} F_a \quad \forall a \in A \right\} \quad (9)$$

$$subject\ \ to\ \ T_{a,i} \leq d_{a,i},$$

$$\tau_a, l_a^0, T_{j,0} = 0, \quad (10)$$

$$T_{j,i} = T_{j,k} - 1 + \sum_{i=1}^{N} x_{a,i,k}.T_{a,i}^e, \quad (11)$$

$$T_{a,i}^e = \sum_{j=1}^{K} x_{a,i,k}.\frac{W_{a,i}}{\zeta_j}, \quad (12)$$

$$F_{a,i} = \sum_{j=1}^{K} x_{a,i,k}.T_{j,i}, \quad (13)$$

$$B_a \leq B_{max}, \quad (14)$$

$$\sum_{a=1}^{A} W_{a,i} \leq \sum_{k=1}^{K} \epsilon_k, \quad (15)$$

$$\sum_{i=1}^{N} x_{a,i,k} = 1 \quad \forall k \in K, \quad (16)$$

$$\sum_{k=1}^{K} x_{a,i,k} = 1 \quad \forall i \in N, \quad (17)$$

$$x_{a,i,k} \in \{0, 1\}. \quad (18)$$

The equation (9) shows the objective function of the considered problem. For each cloud, the finish time, initial location and offloading time are initialized to 0 as defined in constraint (10). The initial setup time of a task to be started after finishing of the previous task on the same resource is described in equation (11) since equations (12) and (13) show the task execution and finish time during scheduling. The requested bandwidth and resources should be less than available channel bandwidth, and computing resources of the associated cloud is stated in equations (14) and (15). Equation (16) and equation (17) show exactly one cloud to each task and exactly one task to each cloud in the VFCN system. Since equation (18) denotes, assignment is done equal to 1 or 0 otherwise.

IV. Proposed Schemes

To solve the considered problem, we proposed different schemes which are explain in the following subsections.

A. Security Mechanism

For the security seeking tasks, i.e., $\sum_{i=1}^{N} SC_i$ we apply Asymmetric encryption SHA-256 RSA method. It exploits a private key and a public key. Public keys are handed out to everyone to use; you make general information about them. Anyone can use the public key to encrypt data, and then only those with the private key can decrypt the document. It also works in the opposite direction but keeping your private key secret is a rule. The Proof-of-Work (PoW) is the primary consensus mechanism in a network of Blockchain nodes. Through Blockchain, this process' primary aim is to validate data transactions and secure process data between nodes (e.g., fog and cloud nodes) through the network chain. During mobility, nodes communicate with each other with PoW to complete data offloading on the vehicle network. This article presents a mixture of algorithm approaches, i.e., Pow and Proof of Credibility (PoC) algorithm for detecting Daniel of Service (DoS) and attack issues. To support a resilient process without any service interruption (e.g., failure of a node or crash due to attack) the study exploits the consensus mechanism. It is a fault-tolerant tool that is used in nodes with Blockchain-Enable network to obtain inevitable compromise on a single data value in the distributed fog cloud nodes such as with cryptocurrencies.

B. Mobility Aware Blockchain Enabled Offloading Scheme (MABOS)

MABOS is a blockchain enable scheme which cares about the data transfer between different computing nodes during mobility. For instance, whenever the task schedules on the computing node, before encryption it computes the hash of data of task i and node k_1. When the task i moves to another node k_2, then the node k_2 decrypts the data, recalculate the hash on i to confirm the correctness, and take another hash plus it keeps previous hash as well. Previous hash also confirmed there was no change in data during migration. Also, in case of a security breach, the chain of hashes help to find out the compromised in the network.

$$\tau_a = \sum_{i\in a}^{N}\sum_{k=1}^{K} \alpha_{a,i}.\left(\frac{S_{a,i}}{B_a} + T^e_{a,i} + t_B\right) \leq \sum_{a=1}^{A}\sum_{i=1}^{N} d_{a,i}. \quad (19)$$

$$\tau'_a = \sum_{i\in a}^{N}\sum_{k=1}^{K} \alpha_{a,i}.\left(\frac{S_{a,i}}{B_a} + T^e_{a,i} + t_B\right) + 1 - \alpha_{a,i}.\left(\frac{S'_{a,i}}{B'_a} + (1 - T^e_{a,i}) + t_B\right) \leq \sum_{a=1}^{A}\sum_{i=1}^{N} d_{a,i}. \quad (20)$$

The Offline offloading engine makes a decision when to offload tasks. The offline offloading is also part of the blockchain network, and the encryption process should be locally first onto all security. (19). MABOS offloads all tasks to the VFCN when applications have an efficient communication channel with gain lower cost, i.e., $l^0_a \leq D_g/2$.

It is worth discussing that initially, all secure tasks are encrypted locally and offloaded to the system without considering migration, as shown in Figure 2. MABOS ensures safe task migration between computing nodes. Figure 2 shows that a task i migrates from node 1 to node 2, then server 2 proceeds as below:

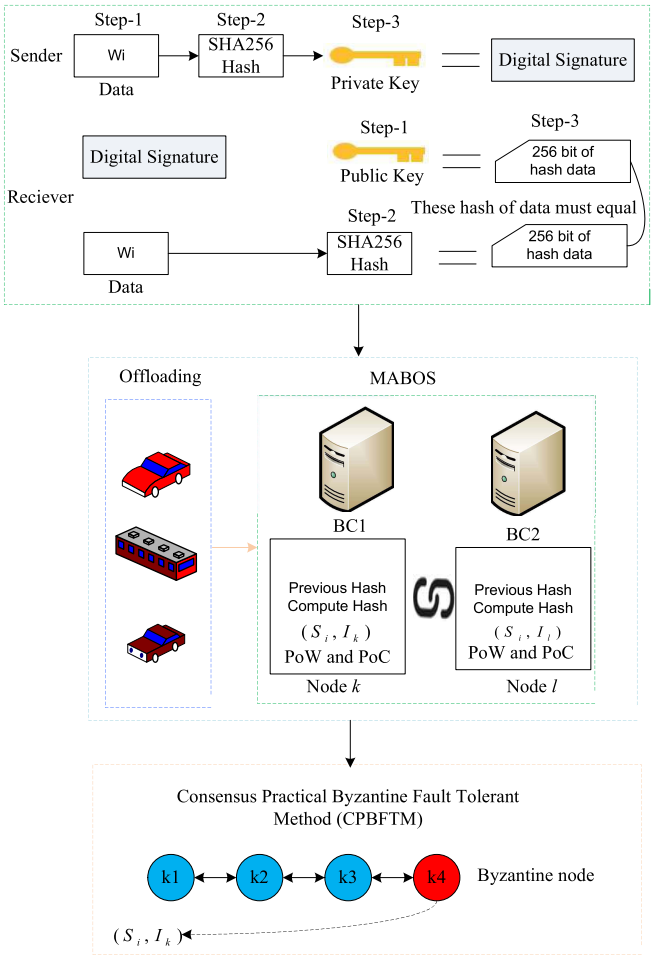

Fig. 2. Blockchain enabling mobility mechanism.

Whenever a security task to be scheduled to another node, the next node decrypt the data, recalculate hash on data, and the previous node identifies I_k, if both hash values are identical, then this verifies the correctness of data and confirms the migration process was safe. Further, the new node computes the hash on data, and current server identifies I_l, and encrypt the data. To node inserts the hash of the previous block in the current immutable block as well. In case of a security breach, the blockchain can help to find out the compromised VFCN with scenarios, as shown in Algorithm 1.

- Decrypt the task data S_i.
- recalculate the hash on decrypted task's data S_i and previous node recognises I_1.
- compare it with hash value included in task block; if both values are identical, then this verifies the correctness of data and confirms the migration process was safe.
- Further, node 2 once again computes the hash on data but this time with current server catalogues I_2.
- To build a blockchain, the node inserts the hash of the previous block in the current immutable block as well.

Algorithm 1: Blockchain Enable Mobility Task Offloading

Input: $\{a \in A, \tau_a, v_a, l_a^x, k \in K, i \in N\}, BC[]\}$

1 **begin**
2 **foreach** $(i \in a \in A \text{ as } N \ \&k \in K)$ **do**
3 **if** $(l_a^0 \leq D_g/2 \ \& \ \tau_a == true)$ **then**
4 $\alpha_{a,i}$=1;
5 Apply SHA256 Encryption on security annotated tasks $\sum_{i=1}^{N} SC_i$ SHA256 follows all steps as explains in the bullets;
6 $BC[\tau_a, S_i] \leftarrow hash(S_i, I_a)$;
7 Calculate system costs based on equation (19);
8 $BC[\tau_a, S_i] \leftarrow hash(S_i, I_k)$;
9 Optimize **P1**;
10 **if** $(1 - \tau_a')$ **then**
11 Calculate system costs based on equation (20);
12 $BC[1 - \tau_a', S_i, k] \leftarrow k$;
13 $BC[\tau_a, S_i] \leftarrow hash(S_i, I_l)$;
14 Optimize **P1**;
15 **if** $(k \in K = 0)$ **then**
16 $k = 0$ means crashed node and $k = 1$ means stable node;
17 Apply Consensus Practical Byzantine Fault Tolerant Method (CPBFTM);
18 $(S_i, k1) \leftarrow k2$;
19 End-Loop;

- Step 1: Offload tasks data to the node receiver from the sender applications. The recipient node has not been tampered with to ensure that the tasks data obtained from the sender have been received and need has only been obtained from the sender.
- Step 2: The sender collects the data from the applications by offloading it to be transmitted offline. The node sender uses the hash function of SHA256 to get the data computed into a 256-bit number.
- Step 3: The sender node then signs a private key with a 256-bit number, which encrypts the 256-bit number into a digital signature. Now the sender transmits the data of the assignments, the Digital Signature, and the public key node to the receiver node (remember, you can't use the public key to find out the corresponding private key, so it's safe to share). The receiver node must authenticate that the data sent to the tasks has not been altered and must have been posted by any computing node that has the private key to the public shared key.
- Step 1: The receiver node takes the 256-bit digital Signature number and decrypts it using the obtained public key. Applying the public key to the Digital Signature 'reverses' step3 above that of the sender.
- Step 2: The receiver then takes the data it receives and applies the SHA256 hash to it to obtain a 256-bit number the process is the same as step 1 and step 2 of the transmitter.
- Step 3: Then, the receiver checks to ensure that the two 256-bit numbers are identical. If wrong, then someone has tampered with the data or given a public key not corresponding to the sender's private key. If real, the receiver should be informed that the data is ready to go.
- Consensus The Byzantine Practical Fault-Tolerant Method (CPBFTM) follows the checkpointing method. The aim is to manage node failure or crash if any computing node fails, and without interruption, it will turn all tasks into another node. Checkpointing is a useful technique that lets functions migrate from the point of failure. CPBFTM ensure tasks are secured simultaneously with PoW and PoC.

We define the Algorithm 1 as follows.

- BC is an array variable which stores are hashing history of the current block.
- Line 1 to 8 denotes the process of the initial security process. Line-4 describes that, if a device gains optimal offloading time based on equation (19), and the condition is real. Then scheme encrypts the sensitive data locally. Line-5 shows that server k can compute the hash on the offloaded data.
- All the condition, as mentioned earlier, is met, then the optimization problem **P1** partially optimize during initial scheduling.
- Line 9-15 performs the aware mobility security between servers. Line-9 describes that if the workload of application a to be migrated to another server. The system calls MABOS Scheme and measures the costs based on equation (20) to support the aware mobility security. It process makes sure that data migration has not to impact on the system costs and data during the process.
- Line-14 verifies that the migration process in mobility is done in an efficient.

C. Task Sequencing

In VFCN, all tasks are saved initially in the queue based on the first come first order. The queue length is finite in the system. However, due to the different requirement of tasks, furthermore, we sort all tasks based on the following sequences.

- Shortest Processing First (SPF): Delay-sensitive tasks are those who have short execution time and required immediate execution. Therefore, we order all shortest process tasks with the highest at the processor for execution. Delay-tolerant tasks have a long deadline; therefore, they will get lower priority.
- Earliest Due Date (EDD): The deadline QoS constraint of a task is very crucial during processing. The task with the smallest deadline to be scheduled first at the processor. If two tasks have the same periods, then order based on first come first serve methods.
- Small Slack Time First (SSTF): Both SPF and EDD are proactive topological sorting methods of tasks. However, the migration of tasks due to the mobility require dynamic prioritization of tasks during scheduling. We accredit priority based on the slack time of a process. Slack time

is the sum of time left after the task if the request was started now. This method is also known as the least laxity first.

D. Linear Search Based Task Scheduling

The aim of linear search based task scheduling (LSBTS) is to schedule all sorted tasks based on their QoS requirements (e.g., deadline, security, and cost). LSBTS has two primary goals. Initially, it plans all requests onto appropriate nodes. Secondly, it reschedules all migrated tasks to another node without any violation of QoS requirements. This process is going iteratively until all tasks assigned or executed at the end. LSBTS relies on the technique of traversing a list from start to end by searching. To solve the optimization problem, the proposed LSBTS Algorithm 2 has following steps:

Algorithm 2: Linear Search Based Task Scheduling

```
   Input: i ∈ N ∈ A, k ∈ K;
1  begin
2      K ← Sort the each server k by the F_a;
3      k ← NULL;
4      foreach (i ∈ N as A) do
5          Call Algorithm 1;
6          foreach (k ∈ K) do
7              if (F_{a,i} ≤ d_{a,i}) then
8                  Must satisfy equation (14)(15);
9                  Assignment x_{a,i,k} based on
                    equation (16)(17)(18);
10                 Calculate the F_a of k ∈ K by the
                    equation(9);
11                 Initial Scheduling F_a ← x_{a,i,k} ;
12                 Obtain F_a^* by Algorithm 3;
13                 Improve initial Scheduling F_a^* ← F_a;
14                 break;
15             End-Loop;
16         End Conditions;
17     return F_a^*;
18 End Main;
```

We denote candidate space, i.e., $N(F_a)$ which consists of many solutions to the problem. There are three types of solutions, namely optimal, average, and worst solutions. It is hard to find an existing optimal solution. However, we can find a near-optimal solution from the solution space. However, the main focus of Algorithm 3 chooses an optimal solution from the candidate solution globally in the distributed computing nodes.

1) Inline 2, all servers are sorted by F_a. Each server has not cost before scheduling as defined as inline 3.
2) Initially, Algorithm 1 decides optimal tasks offloading to the system as defined inline 5. All requests to be scheduled based on their QoS requirements as set inline 7. Line 8 and 9 make sure initial assignment of tasks to the servers must satisfy all constraints. Line 10 and 11 calculate scheduling of all tasks based on cost-efficient servers which have an optimal allocation as compared to others.
3) Due to the mobility, different time zone, and fluctuation in resources, initial scheduling could be improved by time to time. Line 12 and 13 show that initial solution is improved by Algorithm 3 i.e., $F_a^* \leftarrow F_a$. to avoid from overhead, we do not search worst solution and only pick best solutions randomly instead of sequential searching.

E. Solution Improvement

This searching is finding a solution maximizing F_a among several candidate solutions. Many searching algorithms sway from solution to solution in the space of candidate solutions (i.e., F_a^*) by employing local search. This will continue until an optimal solution found. There are many existing search methods to improve the initial solution, Hill climbing, Tabu search, Late acceptance hill-climbing, Reactive search optimization, and simulated annealing. Based on existing simulated annealing searching method. However, based on simulated annealing, we devise a lightweight cost-efficient searching algorithm. The goal is to improve the initial solution to the problem. The algorithm assumes a small random displacement of an objective function resulting in the cost of changing the method. If the device cost difference is negative, the new objective function can be replaced by the old one. If the shift in system costs is positive, then the corresponding Boltzmann constant (i.e., $e^{\frac{\Delta}{t}}$) factor may still be adopted as a new solution. The proposed Cost Improvement Search Scheme Algorithm 3 defined as below.

Algorithm 3: Cost Improvement Search Scheme

```
   Input  : Σ_{a=1}^{A} F_a;
   Output: F_a^*;
1  t ← Initial Temprature;
2  f ← Final Temprature;
3  R ← Find round;
4  iter ← 0;
5  begin
6      while (t > f) do
7          while (iter ≤ R) do
8              F_a^* ← select a random solution F_a^* ∈ N(F_a);
9              Δ ← f(F_a^*) − f(F_a);
10             if (Δ ≤ 0) then
11                 F_a^* ← F_a;
12                 if (f(F_a^*) ≤ F_a)) then
13                     F_a ← F_a^*
14             else if (rand(0, 1) ≤ e^{Δ/t}) then
15                 F_a ← F_a^*;
16         iter ← iter + 1;
17         t ← t × (1 − α);
18     return F_a^*;
```

- Algorithm takes the objective function of all applications $\sum_{a=1}^{A} F_a$, and related parameters.

TABLE I
EXPERIMENT PARAMETERS

Simulation Parameters	Values
Windows OS	Linux Amazon GenyMotion
Centos 7 Runtime	X86-64-bit AMI
Languages	JAVA, XML, Python
Android Phone	Google Nexus 4, 5, and 7S
Experiment Repetition	160 times
Experiment Duration	12 hours
Experiment Monitoring	Every 1 hour
Evaluation Method	ANOVA Single and Multi-Factor
Amazon On Demand Service	EC2 t3
Android Operating System	GenyMotion
t_B	50 seconds
t	100
f	10
R	120
B_{max}	10 Megabyte
v_a	$\{80-120km/h\}$
$S_{a,i}$	1000-1500 MB
P_{tx}	0.1
h_{ng}	0.005
d_{ng}	200
$\overline{w}_o$	15
l_g	100m
a	$\{500-20000\}$ devices
SHA256	256-bits
x	1,2,3,4 speed and location co-ordinates

- We declare initial temperature t and final temperature f, iteration $iter$, and round R variables.
- Line 8-13 ensure that algorithm accepts all solutions randomly either worst or optimal based on Boltzmann constant.
- Line 14 shows that, if the solution does not improve further, it will run on the current solution.
- Line 18 return the final optimal solution i.e., F_a^*.

V. PERFORMANCE EVALUATION AND EXPERIMENTAL SETTING

To evaluate the effectiveness of proposed schemes we conducted experiment on different workloads with different schemes and architectures.

A. Master Node Setting

The Master Node contains different components such as Task Queue, Task Sequence, Task Scheduling, Online Mobility Offloading and Task Migration. The experimental parameters are implemented in the analysis defined in Table I. The configuration of VFCN resource specification illustrated in Table II with its characteristics and specifications. Cost value in Table II shows the function utilization cost per hour of different servers resources in the VFCN for IoT applications. The workload of each application implemented in the experiment is defined in Table III.

B. Offloading Time Schemes

We implemented all existing offloading schemes (e.g., single side offloading) with the proposed multi-side offloading (offline and online) plans in the experiment setup. The main goal of offloading is to decide whether to offload or not during the execution of applications. The baseline approaches are describing below.

- Random offloading (RO): The τ_a application randomly chooses to offload time between 0 and 1 that is equal to the probability distribution. Generally, device offloading time τ_a and assignment $x_{k,c,a,i}$ with device bandwidth B_a are required to optimized mutually.
- Direct offloading scheme (DOS): When offloading time of devices is $\tau_a = 1$ is optimal, and remaining values τ_a, $x_{k,c,a,i}$. B_a are optimized mutually.
- Random Offloading With Single Criteria (ROWSC): This offloading supports mobility aware offloading in the static environment, where network values and available resources are fixed.
- MABOS (Optimal Offloading Time (OOT)): It is a multi-side offloading and blockchain-enabling security technique, which is using for offloading of tasks to different nodes. It shows an offloading time, i.e., $\tau_a = 1$ or $\tau'_a = 1$ when a task migrates between different nodes.

C. Performance Metrics

We evaluate the performance of VANET, MANET and VFCN based on components calibration for instance, such as Task Queue, Task Sequence, Task Scheduling, Online Mobility Offloading and Task Migration. The paper exploits ANOVA RPD (Relative Percentage Deviation) statistical analysis in the experimental setup to evaluate the performances of experiments.

$$RPD(\%) = \frac{F_a^* - F_a}{F_a^*} \times 100\%. \tag{21}$$

F_a shows the initial solution and F_a^* optimal solution of the problem.

D. Task Queue

We implement a single Task Queue in VFCN internal system. The arrival of tasks to system determines by Poisson process. Therefore, the system accepts the random arrival of tasks and schedule them without any queue delay.

E. Task Sequence

The task sequencing rules (i.e., EDD, SPF, and SSTF) are the methods to sort out all tasks in topological order. We have chosen EDD for the task sequencing component in the VFCN for scheduling. In our paper, we exploit the EDD sequence method to compute the task priority. We choose the best γ, i.e., .8 for tightness of task deadline. The highest priority is to set to those tasks which have the smallest deadlines among others.

F. Online Mobility Aware Offloading and Task Migration Methods and Comparison

The mobility is a feature of the network which offers services during roaming among different networks. For experiment evaluation, the paper deployed real scenario of mobility

TABLE II
VFCN RESOURCE SPECIFICATIONS

Cloud	CORE	MIPS/CORE	RAM(GB)	Storage(GB)	Computation Cost	Communication Cost
k_1	i3	1000	800	1000	0.5Â£Â¤¥	5G: 1.5Â£Â¤¥
k_2	i5	3000	1000	2000	0.7Â£Â¤¥	4G: 1Â£Â¤¥
k_3	i7	5000	1200	4000	0.9 Â£Â¤¥	Cellular Data: 2.5Â£Â¤¥
k_4	i9	10000	3200	10000	0.05Â£Â¤¥	3G: 0.5Â£Â¤¥

TABLE III
WORKLOAD ANALYSIS OF IOT APPLICATIONS

Workload	General Tasks	Security Tasks
Healthcare	1700	300
Augmented Reality	1100	100
E-Transport	870	130
E-Business	1300	200

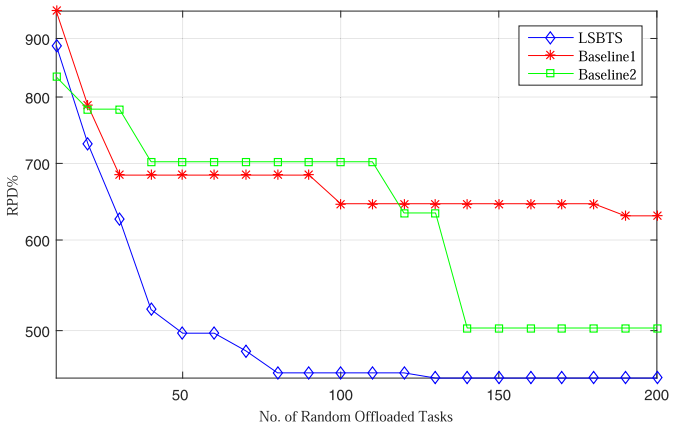

Fig. 3. Random offloading of tasks during mobility.

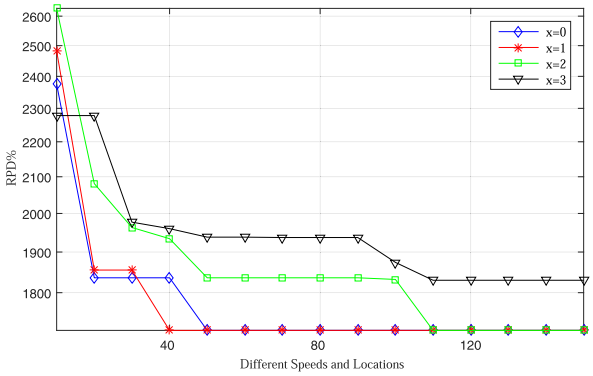

Fig. 4. l_a^x: Different locations of tasks offloading.

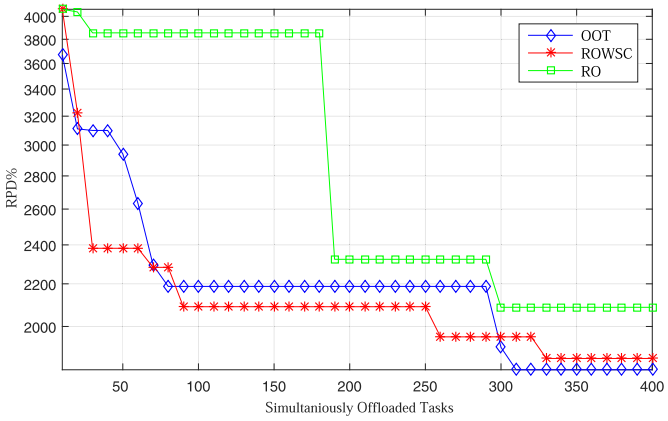

Fig. 5. Mobility aware offloading methods.

aware by implementing device movement trace mechanism Every-Where-lab dataset. Whereas, the trace method monitors device movement within the given range (i.e., the road network of southeast university). The entire road boundary is 17 $\times$ 17 km. Dataset provides 100,000 end-users in the one waypoint for invoking the cloud services during mobility, whereas the location of coordinates devices are tracing in every 25 seconds. We measure each coordinate of the equipment and VFCN resources by a density distribution function, which always return an optimal offloading time for task offloading.

We experimented with services of VANET, MANET and VFCN with different speed and locations of vehicular application during mobility. Figure 4 shows each application has different RPD% with different x values. Therefore, offloading time is critical when to call particular services when they have lowered execution costs. There are many reasons of impacting costs (i.e., communication cost and computation cost) such as availability of resource due to heavyweight requests, less distance between caller and callee, and QoS aware execution is a hard task during mobility. The paper considered these challenging tasks into considerations and proved that all mobility features obtained successfully. Figure 3 and Figure 5 illustrate RPD% performances of applications during mobility and Optimal Offloading Time (OOT) outperform existing mobility methods. There are many reasons behind that, ROWSC and RO offer single side offloading and single criteria (i.e., cost and delay threshold) process during mobility. The main limitation of these methods they support only single side offloading. During mobility, multi-side offloading with multiple parameters could not support by these methods. Therefore, MABOS (OOT) suggests multi-criteria (offloading time, communication cost, computation cost, security) and multi-side offloading for applications, and gain lower RPD% as compared to the existing offloading method.

The mobility-based services are very challenging when the applications are roaming among multiple stations: existing Waypoint and Nomadic mobility methods commonly used in a vehicular network. These methods are implementing with

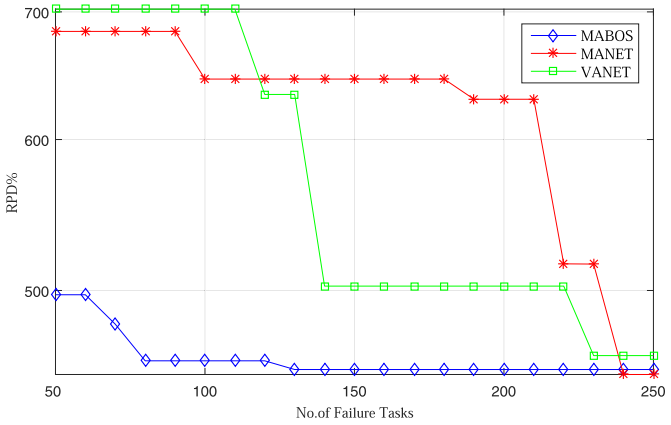

Fig. 6. Tasks failure performance of mobility methods.

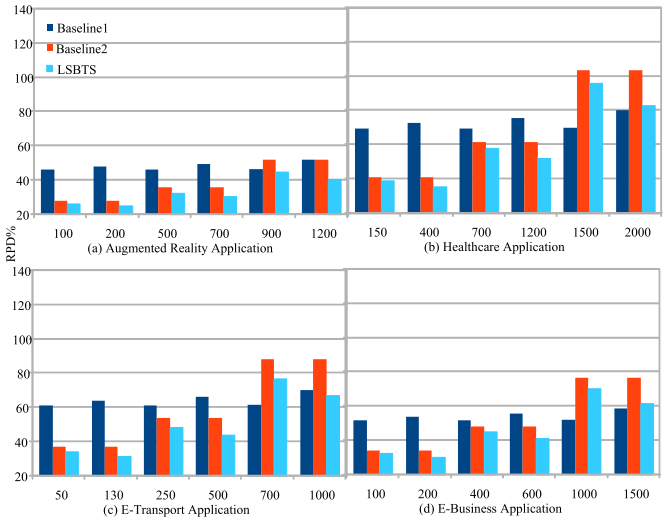

Fig. 7. System costs under 95% confidence interval Tukey HSD mean interval chart.

Software Defined Network at the edge of the net. Handoff technique allows the system to switch services and application requests among servers with handoff technique. However, they did not consider the security mechanism except Secure Server Level (SSL) authentication, which is not enough to handle data security. Due to the earlier situation, many tasks could fail during offloading and downloading between devices and servers. We experimented on random arrival tasks of different in the network during mobility. After evaluation, Figure 6 illustrates that with existing mobility methods, the recovery costs of failure tasks increase as the ratio of failure increase. These tasks failed due to the network failure, services failure and security reason. Therefore, the recovery cost is higher when the failure ratio of jobs increase in the network. The proposed MABOS offers secure offloading service and offloading time in to avoid any failure during untrusted or unavailability of services. Whereas existing mobility methods provided reliable services, but the failure cost of tasks is very high. Hence, optimal offloading time, the lightweight security mechanism of MABOS incur a lower price as compared to existing methods, as shown in Figure 6.

G. Task Scheduling Comparison Methods

The existing task scheduling methods, such as Round Robin (Baseline 1), Min-Max (Baseline 2) and HEFT (Baseline 3) are deploying as baseline approaches. The Linear Search-Based Task Scheduling (LSBTS) is the proposed task scheduling for the IoT applications, which iteratively tries to improve the objective with its constraints. Figure 7 shows that the LSBTS is improving objective function F_a^* continuously from its neighbourhood solutions and reduces the costs of the applications.

The task scheduling on the heterogeneous nodes of offloaded tasks is always a difficult problem. The latest implemented existing baseline approaches Baseline1, and Baseline2 applied rate monotonic based approaches are dynamic scheduling by setting the static task priorities. It implies an inclination of periodic and autonomous tasks with deadlines equal to their periods. The task with the earliest deadline is assigned the highest dynamic priority. However, there are many limitations to these methods. Firstly, the static priority will not work correctly when the adaptive scheduling performs in the environment. The runtime and dynamic priority of different rules could be beneficial and increase the efficiency of the schedule. Secondly, initial mappings which were implementing in existing studies did not consider the adaptive situation, where mobility, security and resource availability inconstantly and intermittently change among computing nodes. Therefore, the performance of scheduling with multi-tasks will degrade during allocation. Another hand, LSBTS is an efficient and aware mobility task dynamic scheduling algorithm which adapts any environment changes to minimize the system cost. The LSBTS algorithm gained lower RPD% of all IoT applications (a) (b) (c) (d) during offloading and scheduling in the VFCN system, as illustrated in Figure 7. Hence, the task sequencing with different rules and local search task scheduling is the most effective way to improve application performance and reduce the system costs.

A practical task scheduling strategy presents a promising approach to deliver better resource utilisation in VFCN. Many task scheduling strategies with optimisation and multi-criteria for IoT applications. These approaches ignored scheduling conflict among the tasks of different types and fluctuation in resources. The friction often directs to miss the deadlines of the assignments and compromises their data in the VFCN. The existing techniques in backfilling algorithms (e.g., baseline1) to execute single criteria (i.e., deadline-based tasks) in VFCN. In general, the jobs are picking as backfill tasks approach (Baseline3), whose purpose is to implement ideal resources to other functions in the backfilling method (baseline2). The selection of the backfill task is challenging one when there are different types of tasks. It creates conflict in the scheduling. We take the assumption of nine random criteria (0-9) to evaluate the performance of algorithms. Figure 8 (a) shows the experimental result of single based scheduling on workloads in VFCN. LSBTS gains lower RPD% as compared to existing approaches. There are a few reasons. LSBTS improves the initial solution with local search from solution space either

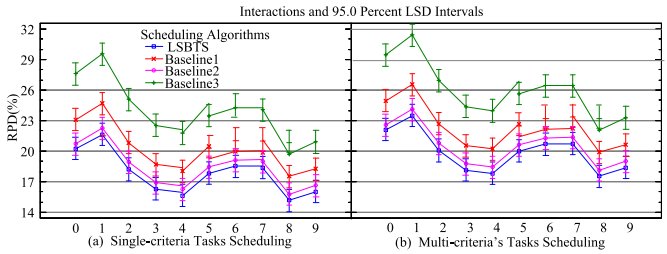

Fig. 8. Multi-criteria based scheduling.

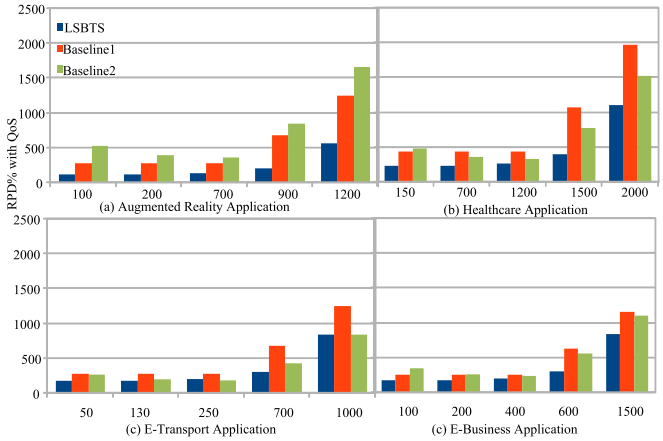

Fig. 9. RPD% under QoS of all IoT applications.

based on single criteria (e.g., deadline or cost). However, due to an adaptive environment, all existing methods incurred higher RPD% as compared to LSBTS. In multi-criteria based scheduling LSBTS considers convex sets (e.g., security, execution time, cost, deadline and required CPU, and mobility) to optimise convex function within polynomial time. However, existing methods optimise all factors individually instead of jointly maximise. Figure 8 (b) result shows together maximise LSBTS accumulations better RPD% as compared to all existing suggestions in the scheduling for IoT applications.

Each application has stringent QoS requirements. Therefore, it is necessary to satisfy all the QoS constraints of applications during scheduling. It is hard to optimize all constraints individually. However, the system can combine all constrain and optimize them together. In our case, the mobility, costs, security and deadline constraints are optimizing jointly as a convex set and improve overall convex objective function for all applications. Figure 9 (a) (b) (c) and (d) shows the LSBTS satisfy all QoS of applications as compared to existing baseline approaches. The main reason behind is that all existing methods make schedules based on execution time.

VI. Conclusion

This study formulates the scheduling problem with convex optimization in Vehicular Fog Cloud Network (VFCN), which includes service costs and security in which the arrival system has a general mobility dynamic distribution. The study proposes the local search aware task scheduling and mobility aware offloading schemes based on linear programming. The study devises Local Search Based Task Scheduling (LSBTS), Mobility Aware Blockchain Enabled offloading scheme (MABOS) scheme to optimize functions for applications. The experimental evaluation shows that the proposed scheme outperforms in term of cost, security, and deadline as compared to existing contemporary scheduling methods.

In future work, we will consider the budget and fault-tolerant aspects of IoT applications in the VFCN. These two constraints are critical when task offloading and scheduling problem are get done on the same for IoT applications.

References

[1] H. Sun, H. Yu, and G. Fan, "Contract-based resource sharing for time effective task scheduling in fog-cloud environment," *IEEE Trans. Netw. Service Manage.*, vol. 17, no. 2, pp. 1040–1053, Jun. 2020.

[2] R. M. Abdelmoneem, A. Benslimane, and E. Shaaban, "Mobility-aware task scheduling in cloud-fog iot-based healthcare architectures," *Comput. Netw.*, vol. 179, Oct. 2020, Art. no. 107348.

[3] X. Zhao and C. Huang, "Microservice based computational offloading framework and cost efficient task scheduling algorithm in heterogeneous fog cloud network," *IEEE Access*, vol. 8, pp. 56680–56694, 2020.

[4] T. Ying Wah *et al.*, "A novel cost-efficient framework for critical heartbeat task scheduling using the Internet of medical things in a fog cloud system," *Sensors*, vol. 20, no. 2, p. 441, 2020.

[5] A. Lakhan and X. Li, "Transient fault aware application partitioning computational offloading algorithm in microservices based mobile cloudlet networks," *Computing*, vol. 102, no. 1, pp. 105–139, 2020.

[6] M. Usman, M. A. Jan, X. He, and P. Nanda, "Qasec: A secured data communication scheme for mobile ad-hoc networks," *Future Gener. Comput. Syst.*, vol. 109, pp. 604–610, Aug. 2020.

[7] K. Kumar, S. Kumar, O. Kaiwartya, P. K. Kashyap, J. Lloret, and H. Song, "Drone assisted flying ad-hoc networks: Mobility and service oriented modeling using neuro-fuzzy," *Ad Hoc Netw.*, vol. 106, Sep. 2020, Art. no. 102242.

[8] A. Agarwal, R. Pal, and A. Prakash, "A scheduling algorithm including deadline of messages in vehicular ad hoc network," in *Advances in VLSI, Communication, and Signal Processing*. Singapore: Springer, 2020, pp. 115–123.

[9] B. Ko, K. Liu, S. H. Son, and K.-J. Park, "Rsu-assisted adaptive scheduling for vehicle-to-vehicle data sharing in bidirectional road scenarios," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 2, pp. 977–989, Feb. 2020.

[10] R. Vatambeti, "A novel wolf based trust accumulation approach for preventing the malicious activities in mobile ad hoc network," *Wireless Pers. Commun.*, vol. 113, pp. 1–26, Aug. 2020.

[11] C. Lin, G. Han, X. Qi, M. Guizani, and L. Shu, "A distributed mobile fog computing scheme for mobile delay-sensitive applications in SDN-enabled vehicular networks," *IEEE Trans. Veh. Technol.*, vol. 69, no. 5, pp. 5481–5493, May 2020.

[12] C. Tang, X. Wei, C. Zhu, Y. Wang, and W. Jia, "Mobile vehicles as fog nodes for latency optimization in smart cities," *IEEE Trans. Veh. Technol.*, vol. 69, no. 9, pp. 9364–9375, Sep. 2020.

[13] M. Zhu, Y. Hou, X. Tao, T. Sui, and L. Gao, "Joint optimal allocation of wireless resource and MEC computation capability in vehicular network," in *Proc. IEEE Wireless Commun. Netw. Conf. Workshops (WCNCW)*, Apr. 2020, pp. 1–6.

[14] Y. Wu, J. Wu, L. Chen, G. Zhou, and J. Yan, "Fog computing model and efficient algorithms for directional vehicle mobility in vehicular network," *IEEE Trans. Intell. Transp. Syst.*, early access, Feb. 10, 2020, doi: 10.1109/TITS.2020.2971343.

[15] K. Liu, K. Xiao, P. Dai, V. Lee, S. Guo, and J. Cao, "Fog computing empowered data dissemination in software defined heterogeneous VANETs," *IEEE Trans. Mobile Comput.*, early access, May 25, 2020, doi: 10.1109/TMC.2020.2997460.

[16] Q. Yuan, J. Li, H. Zhou, T. Lin, G. Luo, and X. Shen, "A joint service migration and mobility optimization approach for vehicular edge computing," *IEEE Trans. Veh. Technol.*, vol. 69, no. 8, pp. 9041–9052, Aug. 2020.

[17] A. A. Al-Habob, O. A. Dobre, A. G. Armada, and S. Muhaidat, "Task scheduling for mobile edge computing using genetic algorithm and conflict graphs," *IEEE Trans. Veh. Technol.*, vol. 69, no. 8, pp. 8805–8819, Aug. 2020.

[18] Q. Wu, X. Fan, W. Wei, and M. Wozniak, "Dynamic scheduling algorithm for delay-sensitive vehicular safety applications in cellular network," *Inf. Technol. Control*, vol. 49, no. 1, pp. 161–178, Mar. 2020.
[19] T. Wang, H. Luo, X. Zeng, Z. Yu, A. Liu, and A. K. Sangaiah, "Mobility based trust evaluation for heterogeneous electric vehicles network in smart cities," *IEEE Trans. Intell. Transp. Syst.*, early access, Jun. 19, 2020, doi: 10.1109/TITS.2020.2997377.
[20] Z. Guan, H. Lyu, D. Li, Y. Hei, and T. Wang, "Blockchain: A distributed solution to UAV-enabled mobile edge computing," *IET Commun.*, vol. 14, no. 15, pp. 2420–2426, Sep. 2020.
[21] J. Feng, F. R. Yu, Q. Pei, J. Du, and L. Zhu, "Joint optimization of radio and computational resources allocation in blockchain-enabled mobile edge computing systems," *IEEE Trans. Wireless Commun.*, vol. 19, no. 6, pp. 4321–4334, Jun. 2020.
[22] Y. Dai, D. Xu, K. Zhang, S. Maharjan, and Y. Zhang, "Deep reinforcement learning and permissioned blockchain for content caching in vehicular edge computing and networks," *IEEE Trans. Veh. Technol.*, vol. 69, no. 4, pp. 4312–4324, Apr. 2020.
[23] N. D. Janakbhai, M. J. Saurin, and M. Patel, "Blockchain-based intelligent transportation system with priority scheduling," in *Data Science and Intelligent Applications*. Singapore: Springer, 2021, pp. 311–317.
[24] J. Qiu, D. Grace, G. Ding, J. Yao, and Q. Wu, "Blockchain-based secure spectrum trading for unmanned-aerial-vehicle-assisted cellular networks: An Operator's perspective," *IEEE Internet Things J.*, vol. 7, no. 1, pp. 451–466, Jan. 2020.

Abdullah Lakhan received the Ph.D. degree in computer science and technology from Southeast University, China. He is currently an Assistant Professor with the College of Computer Science and Artificial Intelligence, Wenzhou University, Wenzhou, China. His research interests include mobile cloud computing, fog computing, edge computing, job shop scheduling, machine scheduling, metaheuristics, and artificial intelligence.

Muneer Ahmad received the Ph.D. degree in computer science from Universiti Teknologi PETRONAS, Malaysia, in 2014. His research interests include big data analysis, machine learning, bioinformatics, digital signal processing, and the medical IoT. He has 16 years of teaching, research, and administrative experience internationally.

Muhammad Bilal (Senior Member, IEEE) received the Ph.D. degree in information and communication network engineering from the Electronics and Telecommunications Research Institute (ETRI), Korea University of Science and Technology, in 2017. Since 2018, he has been an Assistant Professor with the Division of Computer and Electronic Systems Engineering, Hankuk University of Foreign Studies, Yongin, South Korea. Prior to joining the Hankuk University of Foreign Studies, he was a Post-Doctoral Research Fellow with the Smart Quantum Communication Center, Korea University, Seoul, South Korea. His research interests include design and analysis of network protocols, network architecture, network security, the IoT, named data networking, blockchain, cryptology, and future Internet.

Alireza Jolfaei (Senior Member, IEEE) received the Ph.D. degree in applied cryptography from Griffith University, Gold Coast, QLD, Australia, in 2015. He is currently a Lecturer with the Department of Computing, Macquarie University, Macquarie Park, NSW, Australia. Before this appointment, he was an Assistant Professor with Federation University Australia, Ballarat, VIC, USA, and Temple University, Philadelphia, PA, USA. He has authored more than 60 peer-reviewed articles on topics related to cybersecurity. His current research interests include cybersecurity, the Internet-of-Things security, human-in-the-loop CPS security, cryptography, AI, and machine learning for cybersecurity.

Raja Majid Mehmood (Senior Member, IEEE) received the Ph.D. degree in computer engineering from the Division of Computer Science and Engineering, Chonbuk National University, South Korea. He has served as a Research Professor with the Department of Brain and Cognitive Engineering, Korea University. He is currently an Assistant Professor with the School of Electrical and Computer Engineering, Information and Communication Department, Xiamen University Malaysia, Malaysia. His main research interests include affective computing, brain–computer interfaces, information visualization, image processing, pattern recognition, and multi-task scheduling.