Blockchain Leveraged Task Migration in Body Area Sensor Networks

Md Ashraf Uddin, *Member, IEEE*, Andrew Stranieri, Iqbal Gondal, Venki Balasubramanian
Internet Commerce Security Laboratory
School of Science, Engineering and Information Technology, Federation University
Ballarat, VIC 3350, Australia
mdashrafuddin@students.federation.edu.au, (a.stranieri, iqbal.gondal, v.balasubramanian)@federation.edu.au

***Abstract*—Blockchain technologies emerging for healthcare support secure health data sharing with greater interoperability among different heterogeneous systems. However, the collection and storage of data generated from Body Area Sensor Networks(BASN) for migration to high processing power computing services requires an efficient BASN architecture. We present a decentralized BASN architecture that involves devices at three levels; 1) Body Area Sensor Network- medical sensors typically on or in patient's body transmitting data to a Smartphone, 2) Fog/Edge, and 3) Cloud. We propose that a Patient Agent(PA) replicated on the Smartphone, Fog and Cloud servers processes medical data and execute a task offloading algorithm by leveraging a Blockchain. Performance analysis is conducted to demonstrate the feasibility of the proposed Blockchain leveraged, distributed Patient Agent controlled BASN.**

***Index Terms*—Internet of Things, Task Offload, Blockchain, Patient Agent, Fog/Edge, Cloud, Body Area Sensor Network, Assignment Algorithm.**

I. Introduction

Body Area Sensor Networks generate patients' physiological data with wearable or ingestible sensors and transfer data streams to remote applications. BASN has accelerated the development of remote patient monitoring systems which decrease health delivery costs for residents, particularly in developing countries [1]. BASN supports a wide range of applications including continuous vital signs monitoring, arrhythmia detection, fall detection, regulating oxygen therapy, monitoring of pregnant women, chemotherapy reaction and glucose monitoring [2].

Body area sensors upload their generated data to a central Cloud server through a Local Processing Unit(LPU) or Base station and patients share the health data in the Cloud with different stakeholders. Most of the conventional BASN architecture is unable to meet the exponential growth of medical sensor devices predicted [3] and further raise scalability and interoperability limitations. Centralized BASN architectures are vulnerable to a single point of failure that malware including ransomware and Denial of Services(DoS) can exploit [4]. In addition, Cloud based storage and processing create concerns about patient's privacy as third party Cloud Service Providers(CSP) belong to these storage systems. Conventional CSP(Cloud Service Provider) cannot ensure accountability, and traceability of patient's medical data [5] as health data is stored in different off-premise Cloud servers.

Further, Body Area Sensors have limited storage, processing and energy resources and cannot undertake the high computational power for processing Big health data. Mobile Cloud Computing(MCC) has emerged to expand the capabilities of Body Area Sensor Networks through data or task migration to Cloud servers. Task migration can overcome limitations of BASN such as limited memory, CPU power and battery life. Although the Cloud servers support very large storage and very high processing capacity, the excessive transmission delays and unstable connections degrade the quality of service(QoS). If medical sensors directly connected to the Cloud become prevalent, transmitting and retrieving data to/from the Cloud can be expected to cause higher latency and become intractable. Recent advances in healthcare, Edge computing has enabled extensive processing capabilities at the Edge of the network. Edge computing can reduce this latency and improve quality of service because Edge devices are located closest to medical devices [6]. Medical data produced in settings such as emergency or intensive care units rely on rapid, near real-time transmission of data to healthcare professionals [7]. In these situations, the Fog resources closest to the Patient's Smartphone can support the processing of streaming data from wearable sensors in real time. Edge servers are now capable of extracting meaningful analytics from medical sensors to ensure a precise healthcare services. Despite this ongoing advancement, there are growing concerns regarding the privacy and integrity of sensing and transmitting data to the Edge from its embedded medical sensors.

Blockchain which is a distributed, and tamper proof ledger operates on a peer to peer network. Each node in the Blockchain runs with a similar suite of protocols. Blockchain technology can establish a cooperative and trust relationship between BASN and Edge devices. This technology can be utilized to enable a comprehensive, interoperable, and secure exchange of patient's record between embedded sensors and Edge devices to address privacy and security concerns in task offloading. Further, a Blockchain Electronic Health Record(EHR) avoids the reliance on a single institution, which reduces the risk of patient's record keeping. Blockchain based EHR ensures that patient records are verified, managed by patients and remain unaltered.

However, Blockchain is an open system. The contents of

a transaction are revealed to the entities that participate in validating and processing data in the Blockchain. Therefore, processing sensitive health data in Edge network risks compromising privacy. In this article, we describe that a Patient Agent can be decentralized and replicated at the BASN, Fog and Cloud levels to facilitate the migration of patient's task to the Edge and Cloud while preserving patient's privacy and security. This replication of the Patient Agent enables medical data to be stored rapidly and securely without third party trusted authorities. Our contribution also includes a proposal of Blockchain leveraged Task Migration Algorithm executed in the decentralized BASN architecture.

We review related papers in Section II and describe the task migration in Section III. The performance of the proposed approach is presented in Section IV before concluding the paper.

II. Literature Review

Uddin et al. [1], [8] proposed a Patient Centric Agent residing in the Smart Gateway to determine storage, access control and privacy levels during the insertion of patient medical data into a customized Blockchain. The Patient Centric Agent also selects Blockchain providers to schedule medical data for processing and storage. In [1], the role of the Patient Agent is extended to manage multiple Blockchains and multiple storage mediums including the Local Computer, and Cloud database to preserve patient's privacy. Tuli et al. [9] presented FogBus that is a lightweight Blockchain based Fog computing framework. They introduced a universal broker software executing on the Fog device to merge Blockchain with Edge devices such as medical sensors. The broker schedules jobs among other devices in the Fog. However, a universal broker system in eHealth causes security and privacy threat for the patients. Rahman [10] proposed a secure framework including Blockchain at MEC(Mobile Edge Computing) and the Cloud. Therapy data from physicians and patients is processed by Cloud and MEC Blockchain nodes to ensure immutable, anonymous, secure and transparent sharing. The Blockchain stores only hashes of the therapy multimedia and the actual multimedia data containing images, audios, and videos are stored off-chain in a separate database. Although the framework includes MEC Blockchain to avoid shortcomings of high bandwidth and analytical processing required by the Cloud, the Ethereum consensus consumes high power at MEC and they did not focus on task migration.

Griggs [11] presented an architecture for automated remote patient monitoring using smart contract executed on Ethereum. A smart device such as a mobile phone or laptop collects and aggregates data transferred by body area sensors. The smart device sends the aggregated data to pre-specified smart contract stored on the Ethereum. The smart contract processes the data and sends the result and notification to smart devices and healthcare providers. The Blockchain only keeps a record of an event's occurrence and data is stored on the Electronic Health record. However, the smart devices can cause single point of failure and be vulnerable to Denial of Service attack. The architecture only ensures the secure processing of medical data. Dwivedi et al. [2] proposed a decentralized privacy preserving Blockchain based healthcare framework inspired by Ali et al.'s lightweight Blockchain for IoT. A new block is created and stored in the Cloud server. The cluster head of an overlay network verifies the block and confirms the integrity of the block. However, the avoidance of any consensus mechanism weakens the sustainability of Blockchain based healthcare. The authors focused on some lightweight security and privacy protocols to safeguard the system from malicious attacks but did not focus on task migration. Although some existing healthcare architectures include Fog and Cloud for the storage and processing of patient's data, studies in BASN did not advance the notion of developing Blockchain leveraged task migration among Fog devices. To bridge this research gap, we propose a Blockchain leveraged task migration algorithm that is executed by Fog devices. In MCC, many state-of-the-art task offloading algorithms have been proposed but cannot be directly applied in BASN architecture due to patient's security and privacy issues.

MAUI [12] and CloneCloud [13] solve linear optimization problems offline to define the location of a method (i.e. remote or local). The execution or energy cost required for every offloading task is assumed to be known to the system, however this not always practical. Solving a linear optimization problem to make offloading decision adds computation cost and static offline partitioning in the proposal is non-optimal. ThinkAir [14] and CADA [15] used average runtime cost of local device and remote device to decide whether to offload tasks. Both approaches assumed execution environment parameters remain unchanged regardless location or time. As a result, the approaches did not provide accurate result on dynamic execution environment. Khoda [16] solved the code offloading decision problem using non linear optimization with a Lagrange multiplier. The authors predicted the execution time for a task using linear regression model. We propose to categorize tasks on the basis of privacy and time sensitivity prior to deciding whether to offload a task. The tasks that need to be offloaded are assigned to remote devices optimally using the Hungarian assignment method(polynomial time solved algorithm).

III. The Decentralized Patient Agent for Migrating Tasks

The task migration approach proposed here executes on a hierarchical BASN architecture. The BASN architecture depicted in fig 1 consists of BASN, Fog and Cloud. A decentralized Patient Agent executes on the devices at these three levels to assist task migration and other operations such as managing Blockchain and Big health data. At least one instance of such a Patient Agent resides in each of the following platforms.

- **The BASN Level:** Various kinds of wearable sensor devices such as motion tracker, physiological sign measurement devices(EEG, ECG, BSC etc.) [17] constitutes

the BASN level. Patient's wearable sensors wirelessly connected to a nearby Smartphone via a star network form a BASN. Typically, these wearable devices transmit patient's data to the Smartphone using the Bluetooth or ZigBee protocol [18]. The Smartphone executing a Patient Agent aggregates patient's physiological data and transmits the data to Fog devices for further processing.

- **Fog Level:** The Fog also called edge computing network that is close to the wearable sensors consists of traditional switches, router, and low profile devices. A Fog device hosts a replica of the Patient Agent to process health data that requires quick response or real time processing. A device in the Fog level might host Patient Agents for many patients.
- **Cloud Level:** A replica of the Patient Agent also executes in the Cloud server. The Cloud server might house many patient or healthcare professionals' agents and can conduct intensive computations with high availability and flexibility. Cloud servers accommodate the distributed storage for the Blockchain.

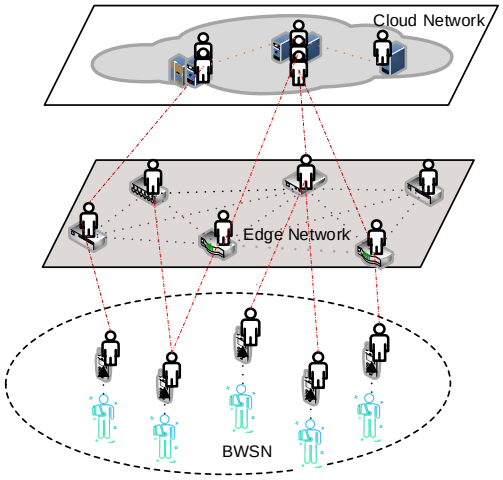


Fig. 1: The decentralized Patient Agent Architecture

The Patient Agent hosted in the Smartphone, Fog device and Cloud server has several components depicted in fig 2. Here, we focus on developing the Migration Handler component of the Patient Agent. The Patient Agent at the Fog device receives health data from body area sensors and decide to offload or execute the task. Every Fog device stores their resources information in the Fog Blockchain. The Profile Monitoring(PH) component collects this resources information through Blockchain Manager(BM) to assist the Migration Handler.

*1) **Blockchain Manager(BM):*** The Blockchain ledger is spread across the network among all peers in the network, and each peer holds a copy of the complete ledger. A Block in a Blockchain is a collection of transactions packed into a Merkle tree. A Block depicted in fig 3 typically contains the Nonce, Timestamps, Previous Block Hash in the header and Data. The Previous Block Hash field creates cryptographic

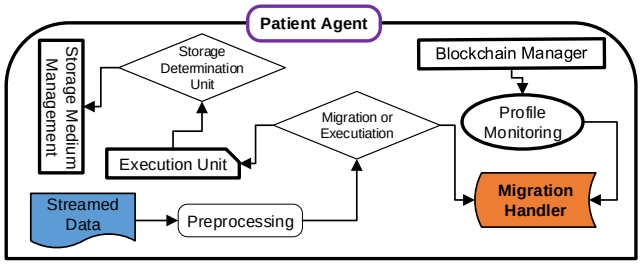


Fig. 2: The functionalities of the Patient Agent at three levels

links between the Blocks which makes Blockchains irreversible and tamper proof. A group of nodes in the Blockchain network participates in processing and validating the Block prior to adding Blocks in the Blockchain.

The Blockchain in this architecture executes in the upper two levels: Fog and Cloud. This component makes different kinds of health data transactions including Task Migration Transaction to transfer a task to a remote Fog Agent. The BM runs the consensus protocol to confirm the new Block into the Blockchain. The Fog devices store a chain of Block's header needed to validate the next Block whereas the Cloud provides the storage for the Blockchain data because Cloud servers have virtually unlimited memory and processing power.

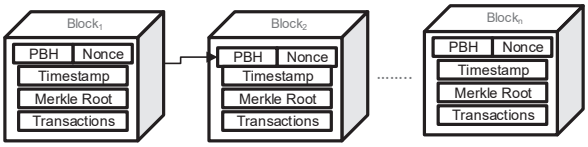


Fig. 3: The Block of a typical Blockchain

*2) **Profile Monitoring(PM):*** The Migration Handler described in section III-3 aims to outsource and distribute computing tasks to neighboring Patient Agents that have more computing resources. Each Patient Agent therefore needs to know the topology of its neighbor Agent. The Profile Monitoring module requires queue latency, CPU speed, availability, and bandwidth of the neighboring Fog devices to pass on to the Migration Handler. The PM broadcasts a Task Migration Request Transaction shown in fig 4(a) throughout the Blockchain network of miners. This transaction contains requestor address and task requirements such as deadline, CPU processing speed and Bandwidth. The Fog Agents with available resources reply to the PM by making a Response Transactions shown in fig 4(c) holding queue latency and other dynamic parameters. The requestor can retrieve Profile Transactions containing some static parameters(CPU speed, bandwidth) of the remote Patient Agents. All these transactions related to task migration are processed, validated by the Blockchain nodes and stored in the Blockchain.

*3) **Migration Handler(MH):*** The adoption of Blockchain in Body Area Sensor Networks becomes an increasing challenges as BASN involves a large number of smart devices or sensors with limited computational capacity. Offloading

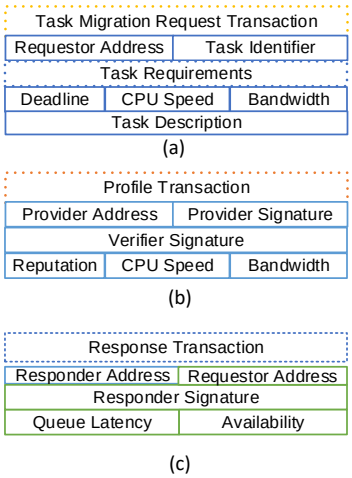


Fig. 4: Transactions for migrating tasks

resource requirements from medical sensors to Edge or Cloud processes can alleviate high computational cost and high bandwidth overhead that results in delays of data and significant processing power [4].

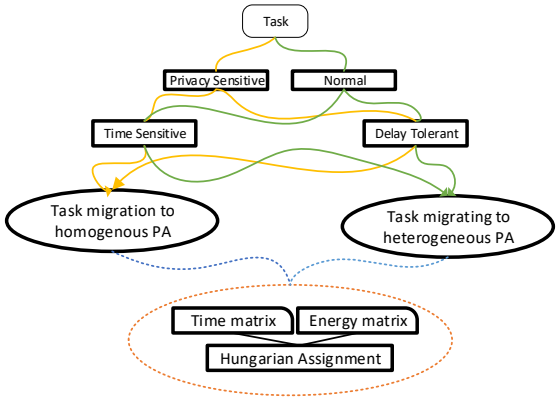


Fig. 5: Classification for task migration

Here, a migration handler determines the optimal location to execute a task. The execution time and processing power vary depending on the task's complexities and health data size. For instance, health data filtration, fusion, compression, and other data mining analyses often require high computing power and massive storage available only on Cloud servers. Some tasks cannot tolerate long delays. These kinds of tasks should not be uploaded to the Cloud server from the local processor even if offloading saves power consumption. For instance, an early warning module should be executed while streaming medical data from the BASN without delays inherent in Cloud processing. The Smartphone or Fog level is the most appropriate platform to run such modules to ensure better quality of service. The Migration Handler module is designed to distribute tasks among neighboring Patient Agents that have spare capacity. A Patient Agent on a local device will migrate tasks to higher processing computing devices only if the required time and power consumption to execute the tasks in the local device is less than that of the response time and power consumption required to transmit data to that higher computing device. The Patient Agent at any levels should not migrate medical data tagged as sensitive to other Fog devices. A task offloading algorithm is designed to protect patient's privacy and optimize the utilization of the host device. Our proposed task migration is described below.

Suppose, the Patient Agent at Smartphone has some tasks $t_1, t_2, ..., t_n$ to perform and responds to the Patient. The Smartphone needs to migrate some tasks to the Fog Agent for processing. A task is defined as privacy sensitive if it involves medical data particularly sensitive to the patient, key management and other cryptographic operations. As mentioned above, some tasks require quick response and some tasks can tolerate delay. Here we assume that Smartphone transmits all tasks to the Fog Agent and the Fog Agent assigns the tasks to the foreign Fog Agent considering their sensitivity.

- **Task Classification:** A task classification is illustrated in fig 5. First, the Smartphone Agent categorizes a task as a Privacy Sensitive Task(PST) or a Normal Task (NT) following the method proposed in [19]. Secondly, both PST and NT are further classified as Time Sensitive Task(TST) or Delay Tolerant Task(DTT). After that, PST(Privacy Sensitive Task) is scheduled among the replicated homogeneous Patient Agents. Time Sensitive Tasks(TST) under NT are transmitted to the remote Fog device only if the execution time required in the remote Fog device is less than that of the local Fog device. Likewise, Delay Tolerant Tasks(DTT) under NT are transmitted to the remote Fog devices only if the power consumption of the local device for transmitting the task to the remote device is less than that of executing the task by itself. Next, two matrices containing time and energy required to offloading the tasks are formed and fed into Hungarian algorithm to optimally assign those tasks to remote Fog Agents.
- **Time Based Cost Matrix Formation:** The local Fog Agent makes a cost matrix that is formed with the execution time of all the TST(Time Sensitive Tasks) satisfying the offloading condition described below.
  Task Execution in local Fog Agent: If μ_l is the MIPS(Million Instructions per Second) of the local Fog Agent and a task(j) involves i number of instructions including its execution environment, the time required for the local Fog Agent to finish the task(i^j) can be estimated as follows:
  Response Time = Execution Time + Queue Latency
  $t_l^j = \frac{i^j}{\mu_l} + \sum_{k=1}^{m} \tau_l^k$
  where $\sum_{k=1}^{m} \tau_l^k$ represents queue Latency which is the processing time for m number of tasks to be waiting for being executed by the local Fog Agent.
  Task Execution in foreign Fog Agent: The response

time for a task from a foreign Fog Agent is calculated below. The response time of a task includes the time required to upload the task to the foreign Fog Agent and time required for the foreign Fog Agent to execute the task. Uploading time is the summation of propagation and transmission time. Propagation time is the time for one bit to travel from one router or switch to next router or switch. This depends on the distance between the two entities and speed of the communication medium. Transmission time represents the time required to get out all the bits of a task from the host device to the transmission wire. The response time for the task(j) from a foreign Fog Agent is measured as follows.

Response Time = Transmission Time + Propagation Time + Execution Time + Queue Latency

$t_f^j = \frac{s^j}{B_{l,f}} + \frac{D_{l,f}}{\nu} + \frac{i^j}{\mu_f} + \sum_{k=1}^{m} \tau_f^k$

where s^j is the size of data in the task(j), $D_{l,f}$ is the distance between the local and foreign Fog Agent, ν presents the propagation speed of the link between these two devices and μ_f is CPU speed of the foreign Fog Agent in MIPS. $B_{l,f}$ is the transmission bandwidth of the communication link between local and foreign Fog Agent.

A cost matrix based on the response time is formed. The matrix include the response time for only those tasks that satisfy $t_l^j > t_f^j$(local response is greater than foreign response time). The first row of the matrix represents the response time(t) for the first task if it is offloaded to n number of foreign Fog Agent. The second row represents the response time for the second task from n number of foreign Fog Agent and so on. This matrix is input to the assignment algorithm for scheduling the time sensitive tasks and the algorithm maps the tasks to a foreign Fog Agent.

$$\begin{Bmatrix} t_1^1 & t_2^1 & \dots \\ \dots & \dots & \dots \\ \dots & \dots & t_n^m \end{Bmatrix}$$

- **Energy Based Cost Matrix Formation:** The local Fog Agent also forms another cost matrix that includes energy consumption of the delay tolerated tasks. The local Fog Agent offloads such tasks to a foreign only if it can save energy otherwise the task is executed locally. The energy required to execute a task(j) in the local Fog Agent is estimated as follows.

  $e_l = p_l \times \frac{i^j}{\mu_l}$. where p_l indicates the power consumption rate of the local Fog Agent.

  Task Execution in Foreign Fog Agent: The local Fog Agent consumes energy while offloading the task to a foreign Fog Agent due to network interfaces and idle mode in case local Fog Agent's queue is empty. The energy consumption to transfer a task(j) to the foreign Fog Agent is estimated as follows. *energy consumption = idle mode energy consumption + network interface energy consumption* $e_f = \rho \times t_f^j + \epsilon_{trans}$ Where ρ is power consumption rate of the local Fog Agent during idle mode. t_f^j is the response time of the task(j) from the foreign Fog Agent. The energy consumption for the network interface while transmitting a task to a foreign device is estimated as follows.

  $\epsilon_{trans} = p_l \times \frac{s^j}{B_{l,f}}$ where p_l, s^j, $B_{l,f}$ represents the power consumption of the local Fog Agent, the size of the task to be uploaded, and bandwidth of the communication link between the local and foreign Fog Agent respectively.

  Another cost matrix is formed where the matrix elements are energy consumption required to upload delay tolerated tasks that satisfy $e_l^j > e_f^j$(Local execution energy consumption for a task is greater than energy consumption required to transmit the task to a foreign device). The first row of the matrix represents the energy consumption required for the local Fog Agent to transmit the first task to n number of available foreign Fog Agent and the second row represents energy consumption to transmit the second task to n number of foreign Agent. For example, e_n^m indicates the energy consumption if m^{th} task is assigned to n^{th} remote device.

  $$\begin{Bmatrix} e_1^1 & e_2^1 & \dots \\ \dots & \dots & \dots \\ \dots & \dots & e_n^m \end{Bmatrix}$$

  This matrix is input to the assignment algorithm for scheduling the delay tolerated tasks to different foreign Fog Agent.

- **Representation of Hungarian Assignment:** Finally, the Hungarian assignment problem for time based cost matrix can be mathematically expressed as follows (1).

$$\begin{aligned} \min_{t,x} \quad & \sum_{i=1}^{n}\sum_{j=1}^{m} t_i^j x_i^j \\ \text{s.t.} \quad & \sum_{i=1}^{n} x_i^j = 1, \quad j = 1, \dots, m, \\ & \sum_{j=1}^{m} x_i^j = 1, \quad i = 1, \dots, n, \\ & \forall t_i^j \le t_l^j \quad i, j = 1, \dots, n, m \end{aligned} \tag{1}$$

  where

$$x_i^j = \begin{cases} 1 & if\, i^{th}\, device\, is\, assigned\, j^{th}\, task \\ 0 & if\, the\, i^{th}\, device\, is\, not\, assigned\, j^{th}\, task \end{cases}$$

  t_i^j indicates the response time of the task j^{th} from the remote device i^{th} and t_l^j indicates the response time of the task j^{th} from the local device l.

*4) **Execution Unit(EU)**:* The execution unit is responsible for processing medical data such as filtration, fusion, warning generation, and automatic diagnosis. A Patient Agent has the option to choose an EU among its own EU, other Patient

Agent's EU and Smart Contract based EU. A Smart Contract defines a set of rules coded by different kinds of programming language [20]. Every node in the Blockhain network contains coded rules for a Smart Contract. A Smart Contract is triggered when a transaction specified to that smart contract is issued in the Blockchain network. For instance, smart contract for task migration is triggered while migrating tasks to high computing devices.

5) ***Storage Determination(SD)****:* Health related data can be stored on diverse repositories including government managed repositories (eg. myGov electronic health record in Australia), Blockchain, on healthcare service provider servers, on Private Cloud servers, on a patient's personal computer or many other devices. Storage mechanisms have different security levels and patients have diverse privacy preferences. This module suggests an appropriate storage repository for large volume of data.

IV. Performance Analysis

We implemented a customized Blockchain using Java Programming to simulate the proposed task migration method. The simulation parameter for the Fog network is presented in TableI where PC and TPC stands for power consumption and transmission power consumption. Variable numbers of tasks(10, 20,30,...,100) are considered to simulate the Patient Agent based Task Migration(PATM). Data generated by medical sensors is collected by the Smartphone agent. The Smartphone agent transmitted the data to the Fog agent. EE, ET, TE and TT in the graph6 stands for Execution Energy, Execution Time, Transmission Energy and Transmission Time respectively.

TABLE I: The Simulation Parameters

Network Area	$1000\times1000m^2$
Fog device MIPS	9900M - 83000M
Smartphone MIPS	14000M
RAM	8 - 16
Fog device Bandwidth	600M - 300M
Smartphone Bandwidth	100M-50M
Fog device PC Rate(per Hour)	140-95W
Fog device TPC Rate(per Hour)	10W
Smartphone PC Rate(per Hour)	25-20W
Smartphone TPC Rate(per Hour)	2
Transaction Size	1024 bytes
Block Size	10$\times$ 1024 bytes
Task Size	10-5KB/MB
Instruction in Block Validation	10M
Instruction in Task	100-50M

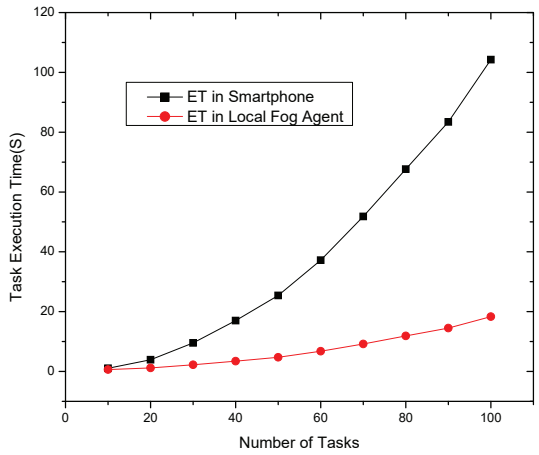
The local Fog Agent executes tasks locally using FCFS(First Come First Service) scheduling. The tasks are assigned to remote Fog Agent using Hungarian Assignment algorithm discussed in sectionIII-3. The performance of the assignment algorithm is analyzed in terms of execution energy consumption and time. **Energy Consumption** indicates the energy required to locally execute a task and transmit the task to a foreign/remote Fog Agent. **Execution Time** indicates the time required to execute a task locally or transmit the task to a foreign Fog Agent and the response from the foreign Agent.

The comparison of energy consumption and latency for the designed task assignment algorithm is depicted in fig 6. Fig 6(a) depicts the comparison of task's execution time by the Smartphone Agent and local Fog Agent. The Smartphone Agent requires longer to complete a set of tasks because the Smartphone Agent's MIPS is less than the Fog Agent and the Smartphone Agent experiences longer queue delays. In contrast, fig 6(b) shows that the Smartphone's energy consumption to transmit the tasks to the local Fog Agent is higher than the Smartphone's energy consumption to locally execute the tasks.

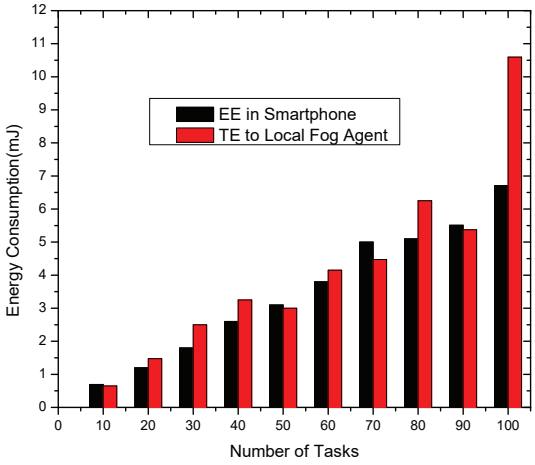
The energy consumption and time for data transmission to remote devices depends on the task's size. The local device can save significant power consumption for migrating tasks that involves small data size. The effect of task's data size is shown in fig 7(a) and (b). Fig 7(a) and (b) shows the transmission energy consumption and time for two different task's size(one large dataset, other small dataset). The local device benefits from lower transmission energy consumption and time if the transmitted task's size is small. The comparison of execution time and energy consumption between local Fog Agent and remote Fog Agent is depicted in fig 6 (c) and (d) respectively. Fig 6(c) shows that the local Fog Agent can optimize the execution time by distributing the tasks among foreign Fog Agents because the foreign Fog Agents parallel execute the assigned tasks but the transmission energy consumption for the local Fog Agent is higher than that of its local execution energy consumption because of large data size.

The energy consumption of five offloading approaches is depicted in fig 8(a). This energy consumption includes energy required for task's transmission and execution. The proposed Patient Agent based task migration (PATM) improves energy consumption over other methods when the number of tasks is few. The PATM consumed high energy for the larger number of tasks because the Hungarian assignment algorithm costs a great deal in terms of energy and time for a large number of tasks. Overall, the PATM saves 1.81% and 8.45% energy in comparison to ExTrade and MAUI approaches respectively.

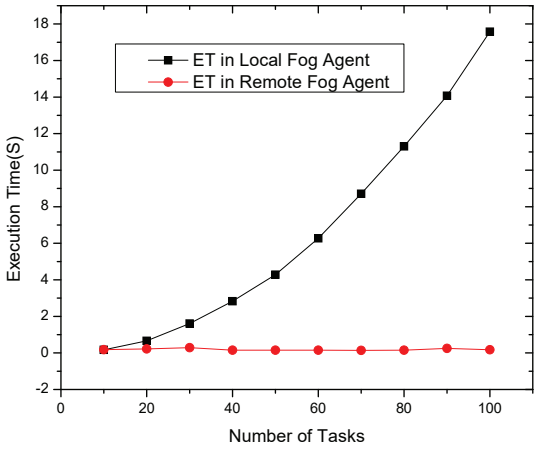
The comparison of the execution time among five offloading approaches is depicted in fig 8(b). The PATM improves the execution time over other approaches because the Hungarian method chooses some remote Fog devices to optimize the execution of all the tasks. Other migration approaches serially assign a task to a remote Fog device. Other approaches show higher execution time as the number of tasks increase whereas the PATM shows almost constant execution time for the increasing number of tasks. The PATM not only decides offloading but also optimally assign tasks to remote Fog Agents. Overall, the proposed tasks assignment improves execution time 38.28% over the ExTrade approach that shows the lowest execution time among the existing methods.



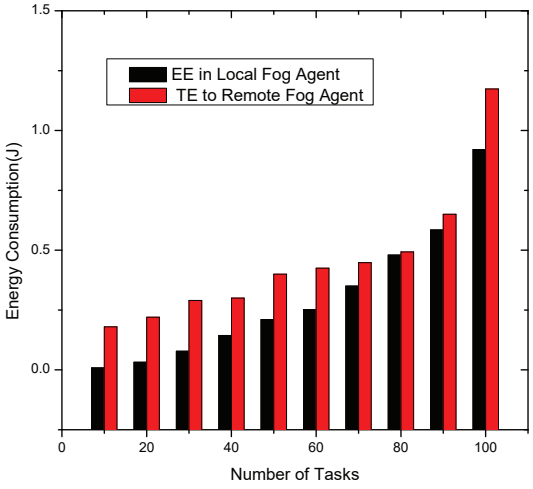
(a) Execution time in Smartphone



(b) Execution energy in Smartphone

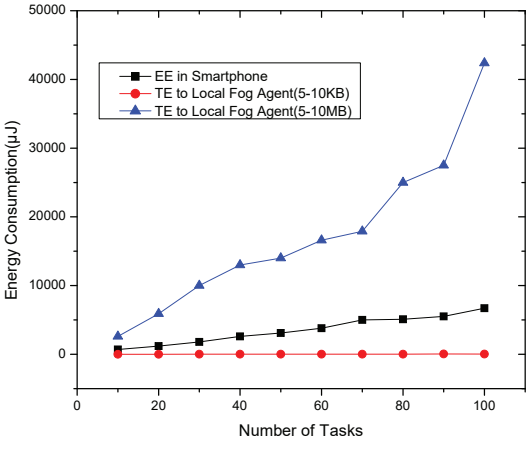


(c) Execution time in local Fog Agent

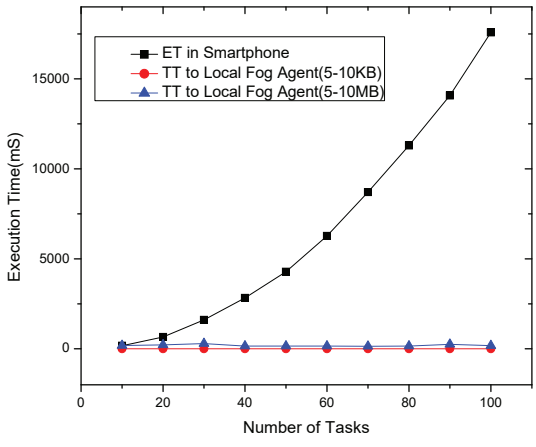


(d) Execution energy in local Fog Agent

Fig. 6: The time and energy for task execution and transmission

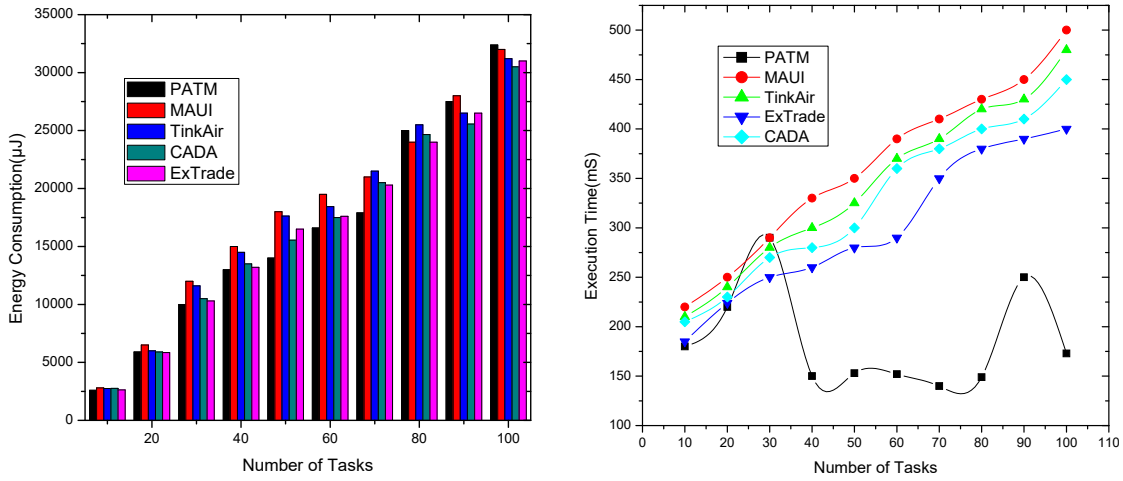


(a) Energy consumption in Smartphone



(b) Execution time in Smartphone

Fig. 7: The comparison of performance for two different size of data



(a) Energy consumption

(b) Execution time

Fig. 8: The comparison of performance between the proposed and existing offloading approaches

V. Conclusions

Like Cloud, heterogeneous Fog devices with diverse security methods or no security are normally deployed by different stakeholders. Sensing and processing of health records by Fog devices is susceptible to malicious attack. In our architecture, sensitive medical data is processed by the homogeneous replicated Patient Agent through migration that can preserve patient's privacy. Blockchain leveraged task migration enhances data confidentiality and integrity. In conventional task migration, the remote entities can lie to the local entity about their performance parameters and data is at risk of being intercepted by malicious attacker while uploading tasks. Our task migration approach stores a device's profile related information in the Blockchain and tasks are transmitted to the remote device through Blockchain technology.

References

[1] M. A. Uddin, A. Stranieri, I. Gondal, and Balasubramanian, "A patient agent to manage blockchains for remote patient monitoring," *Studies in health technology and informatics*, vol. 254, pp. 105–115, 2018.

[2] A. D. Dwivedi, G. Srivastava, S. Dhar, and R. Singh, "A decentralized privacy-preserving healthcare blockchain for iot," *Sensors*, vol. 19, no. 2, p. 326, 2019.

[3] B. Cao, Y. Li, L. Zhang, L. Zhang, S. Mumtaz, Z. Zhou, and M. Peng, "When internet of things meets blockchain: Challenges in distributed consensus," *IEEE Network*, 2019.

[4] M. M. H. Onik, S. Aich, J. Yang, C.-S. Kim, and H.-C. Kim, "Blockchain in healthcare: Challenges and solutions," in *Big Data Analytics for Intelligent Healthcare Management*. Elsevier, 2019, pp. 197–226.

[5] P. Zhang, J. White, D. C. Schmidt, G. Lenz, and S. T. Rosenbloom, "Fhirchain: applying blockchain to securely and scalably share clinical data," *Computational and structural biotechnology journal*, vol. 16, pp. 267–278, 2018.

[6] M. Aazam and E.-N. Huh, "Dynamic resource provisioning through fog micro datacenter," in *Pervasive Computing and Communication Workshops (PerCom Workshops), 2015 IEEE International Conference on*. IEEE, 2015, pp. 105–110.

[7] E. Baccarelli, P. G. V. Naranjo, M. Scarpiniti, M. Shojafar, and J. H. Abawajy, "Fog of everything: Energy-efficient networked computing architectures, research challenges, and a case study," *IEEE access*, vol. 5, pp. 9882–9910, 2017.

[8] M. A. Uddin, A. Stranieri, I. Gondal, and V. Balasubramanian, "Continuous patient monitoring with a patient centric agent: A block architecture," *IEEE Access*, vol. 6, pp. 32 700–32 726, 2018.

[9] S. Tuli, R. Mahmud, S. Tuli, and R. Buyya, "Fogbus: A blockchain-based lightweight framework for edge and fog computing," *arXiv preprint arXiv:1811.11978*, 2018.

[10] M. A. Rahman, M. S. Hossain, G. Loukas, E. Hassanain, S. S. Rahman, M. F. Alhamid, and M. Guizani, "Blockchain-based mobile edge computing framework for secure therapy applications," *IEEE Access*, vol. 6, pp. 72 469–72 478, 2018.

[11] K. N. Griggs, O. Ossipova, C. P. Kohlios, A. N. Baccarini, E. A. Howson, and T. Hayajneh, "Healthcare blockchain system using smart contracts for secure automated remote patient monitoring," *Journal of medical systems*, vol. 42, no. 7, p. 130, 2018.

[12] A. Cuervo, D.-k. Cho, A. Wolman, S. Saroiu, R. Chandra, and P. Bahl, "Making smartphones last longer with code offload," in *8th international conference on Mobile systems, applications, and services*, pp. 49–62.

[13] B.-G. Chun, S. Ihm, P. Maniatis, M. Naik, and A. Patti, "Clonecloud: elastic execution between mobile device and cloud," in *Proceedings of the sixth conference on Computer systems*. ACM, 2011, pp. 301–314.

[14] S. Kosta, A. Aucinas, P. Hui, R. Mortier, and X. Zhang, "Thinkair: Dynamic resource allocation and parallel execution in the cloud for mobile code offloading," in *2012 Proceedings IEEE Infocom*. IEEE, 2012, pp. 945–953.

[15] T.-Y. Lin, T.-A. Lin, C.-H. Hsu, and C.-T. King, "Context-aware decision engine for mobile cloud offloading," in *2013 IEEE Wireless Communications and Networking Conference Workshops (WCNCW)*. IEEE, 2013, pp. 111–116.

[16] M. E. Khoda, M. A. Razzaque, A. Almogren, M. M. Hassan, A. Alamri, and A. Alelaiwi, "Efficient computation offloading decision in mobile cloud computing over 5g network," *Mobile Networks and Applications*, vol. 21, no. 5, pp. 777–792, 2016.

[17] M. Haghi, K. Thurow, and R. Stoll, "Wearable devices in medical internet of things: Scientific research and commercially available devices," *Healthcare informatics research*, vol. 23, no. 1, pp. 4–15, 2017.

[18] S. B. Baker, W. Xiang, and I. Atkinson, "Internet of things for smart healthcare: Technologies, challenges, and opportunities," *IEEE Access*, vol. 5, pp. 26 521–26 544, 2017.

[19] H. Yang and J. M. Garibaldi, "Automatic detection of protected health information from clinic narratives," *Journal of biomedical informatics*, vol. 58, pp. S30–S38, 2015.

[20] Y. Zhang, S. Kasahara, Y. Shen, X. Jiang, and J. Wan, "Smart contract-based access control for the internet of things," *arXiv preprint arXiv:1802.04410*, 2018.