A Dynamic Scalable Fast Blockchain-Based Framework for Smart Cities: The Case Study of Intelligent Transportation System

Mohammad Bagher Moradi*,Siamak Najjar Karimi,Amir hossein Jalali
Department of Electrical Engineering, Shabestar Branch, Islamic Azad University, Shabestar, Iran.
Email: mb.moradi@gmail.com, a.h.jalali@iaushab.ac.ir, siyamak.karimi@gmail.com
Receive Date: 7 May 2023, Revise Date: 19 June2023, Accept Date:16 July2023

Abstract

With the emergence of smart cities vision, its large distributed applications such as intelligent transportation systems demand scalable low-latency trusted data exchange architecture with high storage and computational resources for storing the high-volume of IoT data and providing real-time services. In recent years, blockchain technology has gained extensive attention to fulfil the requirements of such highly distributed large systems. However, there are a number of technical challenges in the integration of blockchain and IoT applications. Firstly, Bitcoin blockchain with low scalability and throughput is not able to provide fast services. Secondly, there are limitations like constrained spaces for establishing big blockchain nodes storing a massive volume of data generated by numerous smart IoT devices or sensors inside the streets of cities. This paper argues that solving both issues in one large blockchain network is infeasible. Therefore, we prioritize this two weakness of blockchain in relation to such systems and propose two separate level of blockchain networks cooperating with each other asynchronously to address them. One network called Fast BlockChain (FBC) composed of multiple scalable sub-blockchain networks responsible for fast services. Another network, CityBC, supports the networks of FBC through the long-term storing of their data and providing their smart manager with knowledge for dynamic autonomous partitioning of them in order to decrease network-to-network communications and avoid wasting storage resources and network bandwidth. Furthermore, this paper evaluates the ideal size of sub-blockchain and then proposes a novel idea for an initial partitioning technique before using collected data by blockchain nodes for dynamic partition of network.

Keywords: decentralized management systems, interoperable blockchain framework, Internet of Things, pervasive systems, dynamic partitioning.

1- Introduction

G The high rate of migration form rural area to cities has resulted in environmental and social challenges, such as traffic jams, air pollution and poor public services that demands novel approaches taking advantage of new technologies to supply the requirements of the citizens. Internet and social networks like Facebook have connected many people across the world that allows them to share and use data. In recent years, with the widespread expansion of smart devices and advancements in the communication paradigm, there has been a new trend towards interlinked devices[1] enabling the vision of smart cities[2], Intelligent Transportation systems (ITS)[3], pervasive healthcare, smart energy trading and Internet of Everything (IoE) and so forth.

With the emergence of ubiquitous computing, everyday objects are able to sense their surrounding area, gather information and communicate to other things. Internet of things (IoT) [4, 5] enables real-world objects to create a large distributed network of connected smart objects and sensors providing a new range of pervasive services. Accordingly, IoT applications and services are rapidly gaining popularity leading to the expansion of Internet of things at a fast pace. It is expected that IoT devices will grow to

more than 24 million by 2020 [6] and it is also anticipated more than 3.3 billion Machine-to-Machine communications by 2021[7].

Although the recent advancements in the Internet of things helps to realize the concept of smart cities, deploying it in such a large distributed systems has challenges, such as scalability, security, privacy, reliability, high latency, bandwidth bottlenecks[8, 9]. Centralized approach, such as cloud computing, has scalability issue and can be vulnerable to single point of failure that make IoT things susceptible to denial-of-service attacks[10]. In this sense, highly distributed nature of the smart objects, resource constrained sensors and time-sensitive applications demand scalable low-latency trusted data exchange architecture[11].

Blockchain is a distributed tamper-resistant auditable ledger with potential to address security, privacy and centralization challenges. Blockchain provides a distributed trusted sharing network with reliable data that increases the autonomy of IoT devices and able them to administer their own security role as well as interaction rules with others. It provides trust in decentralized approach and eliminates a large number of intermediaries that authenticate the data to establish trust between transacting participants [9, 12]. In contrast to the centralized approach, in blockchain network, data is widely distributed where each node is able to store and even verify information and any single node cannot control it, tackling the single point of failure and centralized points of vulnerability challenges [10].

Taking advantages of blockchain in the IoT network can pave the way for creating a secure data communication framework for large distributed IoT applications, such as smart city, Intelligent Transportation System (ITS) and healthcare. However, the exponential increasing number of smart devices and sensors in these applications will grow the size of blockchain considerably because of storing such a massive amount of data generated by them. Furthermore, sharing such a growing blockchain in the network for updating distributed ledger demands high storage and computational resources in blockchain nodes and pose high overhead to the network. Clearly, a novel approach in order to address the growing size of blockchain and its overall data replication is needed. This paper envisions multiple blockchain networks for large distributed applications in which each network provides services locally, but it is connected with other blockchain networks to supply services demanding network-to-network communications. In this sense, the partitioning strategies of large smart applications and communication between blockchain networks seems to be utmost important in the near future.

2. Related Work

It is believed that blockchain technology with the potential to provide trust and security for storing and managing data across the internet enables a new way to write and deploy applications and also allows to create a new kind of organization without hierarchy and centralized decision making. However, since it has not been devised originally for IoT environments, some weaknesses such as heavy consensus algorithm (POW) and poor scalability (of Bitcoin) in relation to IoT systems should be optimized. Although several authors [13, 14] analysed a number of influential aspects related to performance in different scenarios, their main focus is the optimization of consensus algorithms. In addition to them, other blockchain factors should be adapted to provide the requirements of IoT systems. Next subsection describes different types of architectures in relation to blockchain based IoT applications.

2.1. Optimized architecture for BIoT application

In blockchain-Based IoT (BIoT) systems, an appropriate architecture should be adaptable to the conditions of the environment, such as the amount of data traffic generated by smart devices or sensors. In [15] three types of architectures is proposed that one depicts the problems of traditional architecture like cloud computing and two other types, fog and edge computing, going towards solving such issues to support BIoT systems and also reach to the huge growth of IoT in the near future.

In the cloud-based architecture, IoT gateways forward collected data in the node layer to the cloud and they also have capabilities like sensor fusion [16] and protocol conversion (in case it is required). However, it suffers inherent vulnerabilities like a point of failure, i.e. downing the cloud server to any reason like cyberattacks or software problems leads to crashing the whole system. It is also noteworthy that a malicious IoT device might disrupt the whole system via Denial of Service (DoS) attacks, leak private data and even alter data.

Edge/ fog computing bring computational and storage resources to the edge layer close to end nodes, where the massive data of devices/sensors can be processed, filtered and compressed locally before transmitting to the centralized cloud or blockchain nodes, thereby conserving network bandwidth, minimizing data flow, reducing processing burden from cloud servers (or blockchain nodes), real-time communication with minimum latency and energy efficiency [17, 18].

Given that Fog computing is highly distributed and able to support resource constrained IoT devices or sensors, it can play a key role in the architecture of blockchain and DAG [19] applications [20]. Fog layer, a filtering layer between sensor/node layer and centralized cloud or blockchain nodes, composed of a number of local gateways providing real-time response to end nodes. They are also able to communicate with each other and even with the cloud or blockchain (in case it is needed).

Precisely, regarding the highly distributed nature of IoT, decentralized blockchain can be an appropriate technology. In this context, Fog computing follows a distributed approach like blockchain through distributing computing resources and bringing them close to end devices, easing IoT and blockchain integration[12].

2.2. Blockchain and smart city

With regard to urban challenges, the recent progression of IoT technologies as a new approach of sustainable development paves the way for developing the concept of smart cities in order to improving the quality of life, increasing operational efficiencies, providing better citizen services, reducing traffic congestion and maximising environmental sustainability efforts[21, 22].

Authors in [8] define Smart cities as a framework for creating a modern urbanization which is based on distributed and autonomous infrastructure including ubiquitous sensing, intelligent processing information and network with heterogeneous infrastructure. They also mention some problems for current architectures like scalability, privacy, security, network bandwidth and high latency caused by increasing IoT devices and their growing data. To address these challenges, they introduce a hybrid architecture composed of edge and core network. In this architecture, they utilize software defined networking in the edge layer and blockchain in core network. Regarding trust management, it uses proposed architecture in [23] introducing locally centralized and globally distributed trust management.

SpeedyChain [11] proposes a data exchange framework based on blockchain for smart cities including roadside infrastructure units (RSI), service providers (SP) and smart vehicles . It allows smart vehicles to share their data to store in blockchain network while maintaining privacy and data integrity. Authors argue that smart city and intelligent transportation system demands a low-latency trusted data exchange platform. In order to fast appending data in blockchain, they eliminate proof of work (of bitcoin) for creating new block and adding it to blockchain. In addition, appended transactions are decoupled from the block header. Regarding trust and validating data sent by car, this framework uses cryptographic keys (public and private keys).

3. Architecture Overview

We propose a blockchain-based framework for the transportation network of a smart city in which blockchain nodes are connected in a distributed mode inside multiple sub-networks, which allows the development of the scalable decentralized network to provide the fast services on a large scale. It is composed of three layers: fog nodes, Fast Blockchain (FBC) and CityBC.

3.1. Fog computing layer

In large scale systems, such as smart city or Traffic monitoring systems, sending large volumes of data generated by IoT devices and sensor networks directly to cloud computing or processing nodes demands high network bandwidth, storage and computation resources. To tackle this problem, Fog nodes (FNs) pre-process data at the edge of network and then send results to the processing nodes, thereby minimizing traffic volumes in the network and bringing energy efficiency. Regarding blockchain networks in which nodes broadcast received data to other nodes, utilizing fog nodes as a filtering layer between sensor networks and processing nodes in blockchain layer reduces network traffic and the demand for computation and storage resources considerably.

In proposed architecture, the data of smart devices or sensors located in streets are processed locally by Fog Nodes that prevents from sending massive amount of data to smart Routing Nodes (RNs) in blockchain layer. While processing data, FNs detect occurred events, such as changing max-speed in a street due to raining, snowing or car incident, and then sends it to related RN in blockchain layer. FN also notifies related cars located in that street about latest condition or rules of street, such as accident or updated max-speed.

For secure communication between Smart devices and Fog layer, FNs only accept the data of smart devices which were already registered in related FNs. Precisely, a FN has the PK list of its own permissioned devices and authentication is based on predefined PK for each device. While receiving data, FN first checks the PK of device and then validate the signature of transaction. If the validation process is successful, then FN reads its data. Once an event is detected via processing the data, FN create a transaction including event belongs with the PK of devices which their sending data involved. Afterwards, FN signs it using its own private key and sends to the nearest Routing Node (RN) in the blockchain layer. Therefore, based on proposed method, malicious devices are recognized by fog computing layer and ignored their data, thereby preventing from transmitting them to blockchain. In addition, they are able to detect device associated to suspicious or incomplete data through its PK embedded in the transaction. Once discovered, it sends an alarm including the PK of device as well as type of error to an operation and maintenance centre. Until receiving the confirmation

message demonstrating that the device is repaired and safe from Repair Technicians, FN does not accept the data of such devices.

In sending data form FN to RN, every FN have the predefined PK list of RNs corresponding to its own region of city. Based on the distance of RNs, these PKs were already prioritized for every FN. In the first step, FN sends data to the first PK of its own list, which is belong to the closest RN. If data delivering is not successful to any reason, it will send to second RN via its PK in the list. As a result, the proposed method eliminates single point of failure as well as delivering issues in large distributed systems.

3.2. Fast Blockchain layer

Based on our studies in several big cities, such as Tabriz, with high rate of traffic congestion and vehicle air pollution, more than 50% of vehicle journeys are short, inside a region of a city, and around 35% of them are between two neighboured regions of a city and others (around 10%) are long journeys, which is passed through several regions. By taking these kinds of journeys into account, this paper proposes a light blockchain based architecture to provide a secure communication platform for managing traffic and supporting the routing for all 3 types of journeys, thereby paving the way for establishing a fast distributed routing system with minimum cost (i.e. network traffic, computation and storage resource).

In this architecture, a city is partitioned to some regions. This division is flexible and depends on the condition of a city, such as geographical features and management policy of municipality. Fig. 1 demonstrates a standard partition of a real big city, Tabriz, which is composed of 10 municipalities (regions).

In every region, there is a blockchain network composed of Routing Nodes showing the optimal routes to vehicles towards their destinations. The number of Routing Nodes is associated to the condition of a region, such as traffic congestion, nature conditions and the financial budget. In this architecture, at least two Routing Nodes is considered for the network of every region, thus overcoming single point of failure issue in every region and increasing data security as data is replicated in nodes.

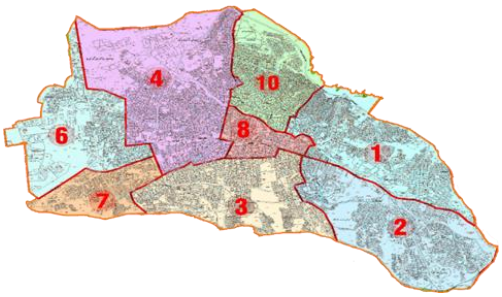

Fig. 1.A big city, Tabriz, with several partitions based on the management condition

In addition, any changes in the data of a RN can be discovered automatically based on the consistency concept in the Blockchain context as a distributed global ledger. It is noteworthy that increasing the number of RN inside a region decrease the distance between RNs and FNs, thereby reducing communication latency and in turn improving response time.

As shown in Fig. 2, every region has a blockchain network composed of at least 2 Routing Nodes (connected with green lines), storing the condition of streets in its own region and showing optimal routes to vehicles. In addition, there are connection (dashed black line) between the blockchain networks of regions which is described in subsections 3.2.2.

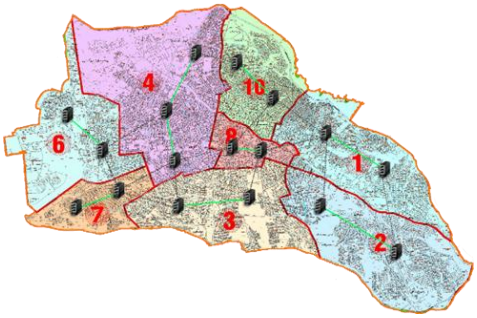

Fig 21. Partitioned big city with interconnected blockchain based IoT nodes

3.2.1. Storing transactions sent by fog nodes

As mentioned in previous section 3.1, FNs pre-process the data sent by smart devices or sensors located in streets and then sends events to related Routing Node (RN) in blockchain layer. Undoubtedly, the fast storing of data in the blockchain is the utmost importance as they are used for exploring the optimal routes for vehicles. To do this, in proposed architecture, a Genesis block was created for every street in related RNs. A Main Smart Manager (MSM) was also defined in every RN which verifies received transactions from fog nodes based on Algorithm 1(shown in Fig. 3) before appending to block, paving the way for eliminating consensus algorithm cost as FN and MSM act as filtering layer for verifying data and transaction, respectively. By doing this, the time and cost of creating block and consensus algorithm is omitted in runtime.

Additionally, depends on the policy of municipalities to manage, study or transmit data to other storing resources such as cloud computing, MSM creates a new block for every street and joins to previous block at every specified time slice such as every 24 hours (t = 00:00 o'clock, at the begin of everyday). In other words, there is a chain of blocks for every street that every block includes the transactions of a street during a day (24hours).

Transactions sent from FNs to the MSM of a RN have two parts, header and data. Header includes the ID of street, timestamp, Public Key of FN and Public keys of smart devices which their data involved. Data is the result of pre-processed data sent by smart devices (sensors) in street. On the ground of having private key, only corresponding MSM can decode transaction. Once decoded it, MSM verifies the identity of sender (FN) based on the algorithm 1 before appending transaction in the related block.

Algorithm 1 demonstrates the steps of fast storing of a transaction in a block (without the time and cost of creating block and consensus algorithm). Once received data from FN, MSM validates the PK of FN with its own list including the Predefined PK of allowed FNs. If it found the PK, it would locate the block of street and then verify the signature of transaction. Finally, transaction is appended in block, provided that it is valid.

Algorithm 1: Appending new transaction

\\Validate the PK of FN which sends a transaction
If PK is in the valid list of RN **then**
Locates the block of corresponding street using ID in transaction.
If verifying the signature in the transaction is successful **then**
Append transaction to block;
Sent to the peer RNs of the same region;
Else
Ignore appending transaction;

Fig. 3. Appending new transaction

3.2.2. Fast Smart Routing

This section describes 3 ways of routing covering all type of journeys mentioned in section 3.2.

A. Vehicle journey inside region

RNs utilize the proposed method in previous section to store transactions associated with street events in its own regions. For example, in Fig. 2, region 10 has 2 RNs, connected with green line, storing and sharing the events of streets in region 10. When receiving a routing request from a vehicle, which both origin and destination are inside a same region, RN process the condition of streets towards destination and then sends the optimal route to vehicle in real-time.

B. Vehicle journey between two regions

For routing the destination of a vehicle which is out of a region, Routing Nodes need

the condition of streets in neighboured regions. Obviously, storing the numerous street transactions of those regions in RNs require large storage. To deal with this problem, in proposed architecture, a new role, Light RN (LRN), is assigned to RNs which keep the light blockchain of neighboured regions including only the latest condition of streets in those regions. In addition, a Smart Manager (SM) is defined for every LRN which is responsible to update its own light blockchain based on following algorithm. For instance, in Fig.2, every RN of region 10 has three LRNs including the light blockchain associated with regions 1, 8, 4, respectively.

Once received a transaction containing a new event or condition changing related to a street in neighboured regions, the SM of related LRN should find corresponding street in its own light blockchain and then updates it. For example, in Fig. 2, once happened an event in a street in region 4, closest RN of region 4 to region 10 sends a transaction including its own PK as well as signature, the ID of street and event (data) to the corresponding SM of nearest LRN in region 10. It should be noted that, RNs in region 4 send also this event to SM of related LRN in region 8, 3, 7 and 6. Since each SM has its own unique address, sender knows where to send transaction exactly. Since having private key to decode truncations is essential, only the SM of LRN in regions 10, 8, 3, 7 and 6 are able to decode the transaction and then update the data related to that street in their own light blockchain. To be precisely, once decoded received transaction, SM based on following algorithm verify the identity of senders (closest RN of region 4) encoded in transaction. If it is valid, then SM updates its own light blockchain and then sends it to SM of another LRN in its own region.

To demonstrate the advantageous of proposed method, Smart Manager (SM) for updating each LNR associated with every neighboured region, this paper defines two algorithms: a) without SM (Algorithm 2 shown in Fig. 4) and b) with SM (Algorithm 3 shown in Fig. 5) and then compares them.

In Algorithm 2, to help Main Smart Manager (MSM) of RN to distinguish between the transaction of updating LRN sent by RN of neighboured region and transaction sent by FN in its own region, the header of update LRN transaction should include update flag. Once found update in the header of transaction, RN validates the PK of senders (n) and then identify its region. Following this step, it finds the corresponding LRN of sender via its PK. Finally, locate the block of street in LRN and then update it, provided that the signature of transaction is valid.

Clearly, in addition to the overhead of transaction header, the cost of this algorithm is $O(n^2)$ and used for N neighboured regions, thereby posing an overhead of $O(n^3)$ on MSM of RN.

Algorithm 2: Updating street conditions of LRN without SM

```
If the header of transaction includes update then
//Validate the PK of RN, which sends transaction, by MSM of RN
For (i = 0; i < n; i++)
  \\ Identify the region of RN;
   If Neighboured-Region-PK[i] == PK then
      For (j=0; j< m; j++)
       // Identify the LRN of that region;
         Locate LRN associated to PK of
         Neighboured-Region-PK[i];
Locates the block of corresponding street in
its LRN using street-ID in transaction;
If verifying the signature in the transaction is successful then
   Update transaction and send to the peer LRNs
Else
   Ignore appending transaction;
```

Fig. 4. Updating street conditions of LRN without SM

However, in proposed method, algorithm 3, every RN has an independent SM for updating every LRN associated with one of neighboured

regions. Because of unique address of SM, the nearest RN in neighboured region sends transaction directly to corresponding SM even without header overhead. For instance, RN of region 10 has three SMs responsible for updating three light blockchain (LRNs) associated with regions 1, 8, 4. Each SM updates its own LRN in RN independent from the MSM as well as other SMs of RN.

Algorithm 3: Updating street conditions of LRN with SM

```
\\Validate the PK of RN which sends transaction
If PK is valid then
    Locates the block of corresponding street using ID in transaction;
    If verifying the signature in the transaction is successful then
        Update street data and sent to the peer LRNs
    Else
        Ignore appending transaction;
```

Fig. 5: Updating street conditions of LRN with SM

Regarding routing, in partnership with the related SM, MSM explore the condition of streets towards destination. For example, a vehicle in region 10 sends a request for going to a street in region 4. MSM sends destination to its own SM associated to LRN of region 4. Afterwards, MSM explore the data of its own region and data provided by SM towards destination. Precisely, two smart managers, MSM and corresponding SM, perform a routing process in parallel and very fast asynchronously.

C. InformationChain

Proposed method in the previous sections covers around 85% of vehicle journeys in a big city, such as Tabriz, with minimum response time, computation and storage resources. To support the remaining journeys (approximately 10%) which are long distance and pass from several regions, Routing Nodes need the data of streets in all regions which demands high network bandwidth to transmit data and considerable storage as well as computation resources to store and process massive amount of data, imposing huge cost on municipalities. To tackle this issue, this paper proposes a novel fast decentralized method, Information Chain, whereby the optimal route of every region in the way of destination is processed via related RN in parallel and asynchronous and subsequently sent to related node in a secure way.

In this proposed method, while receiving a request whose destination is out of both own region and neighboured regions, RN sends a transaction including its own PK as well as signature, the PK of car, the number of destination region and street name to neighboured RN forwarding it to other RNs towards destination. Intermediate RNs located in the way of destination finds at least 2 optimal routes related to their own regions and put them in a block. Mentioned processes are carried out in parallel and asynchronous through the RN of regions involved, reducing routing time considerably. Following that, the RN of destination region sends back block to previous RN which chains it to its own block and in turn transmit to prior node. Finally, origin RN receive a chain of blocks. Every block includes optimal routes associated to an intermediate region towards destination.

3.2.3. Vehicle to RN

A vehicle, which requests a route, sends a message including its own PK and destination to nearest RN via Fog layer. When receiving a message, RN finds route depends on mentioned approaches in previous sections and in turn sends back a message contains PK of FN, PK of vehicle and data (route) to fog layer. Regarding the PK of FN, Related FN gets message and then deliver to corresponding vehicle. A significant point in relation to cars requesting routes is to verify whether the car is genuine without spending much time as the fast response is one of main goals of this architecture. In addition, sending updated route to vehicles without revealing the privacy of car owner, such as the location

of car, as well as imposing heavy traffic on fog network layer to deliver message to cars is described in 3.2.5.

3.2.4. Genuine Vehicle

To verify that whether vehicle is genuine, in [11] the RSI sends a request to other vehicles and ask them whether such a vehicle exists in their vicinity. Although these confirmations can help to prevent the possibility of Sybil attacks, this method is very costly (network overhead) and time consuming and it might be failed in some scenarios. Today's smart environment demands a fast method without waiting time and high network traffic. In proposed approach, after receiving request, RN sends the route to vehicle real-time. However, when arriving to the next street of sending route, vehicle should send its own PK to the FN associated to that street as a confirmation of being in the corresponding street. In otherwise, RN ignores that vehicle as well as updating its route. Therefore, proposed method responses to requests in real-time and utilize a fast light way to verify cars.

3.2.5. Updating the route

During a vehicle journey, some events such as incident might be happened that gives rise to changing the route. In this architecture, the simplest way for sending updated route to car is that RN broadcasts the new route along with the PK of car to Fog layer. However, regarding many vehicles in a big city, this approach imposes heavy traffic on network. To address this problem, this paper proposes a method that deliver route without increasing network overhead as well as revealing the privacy, such as the location of a vehicle. In this method, by considering spent time form origin, RN estimates the location of vehicle and in turn sends the route with the PK of vehicle to only the FNs associated to that area.

3.2. 6. Handling incident

In [11] incident message is sent to police or emergency station. After that, the authority of the station makes call with police or ambulance vehicles and finds a free one and then give the position of incident. Clearly, this is done by human interval, which might be together with mistake and spending much time.

In our proposed architecture, firstly, RN process received message and checks the kind of accident, such as simple car incident or with injured persons. Secondly, decides to notify only police or both police and ambulance. Thirdly, it sends the best route to the nearest free police or ambulance vehicle. Fourthly, it predicts the time that police or ambulance arrive to traffic lights in its own route towards destination. Afterwards, it sends the PK of police or ambulance along with predicted arrived time to traffic lights in that route. Finally, RN sends the PK of these lights along with their positions to police (or ambulance) respectively. When being close to light, it sends a message to the PK of traffic light for confirming that it arrives to light. Following this step, light checks its own list for the PK of allowed police or ambulance as well as time with tolerance 2 or 3 minutes (sooner or later). Having verified the police or ambulance vehicles, light change to green. In addition, another precaution is taken to ensure the safety of proposed method. If lights do not receive a message in predicted time, it sends alarm to the PK of police or ambulance periodically (per 10 s) and requests its position.

3.3. CityBC

In addition to Fast BlockChain (FBC) including multiple sub blockchain networks corresponding to each partition region of city for storing data locally and responding with low-latency, another level of blockchain called CityBC is proposed, which is responsible to store the data of sub-blockchains of all regions of city.

The data stored in this layer can be analysed (future work) to propose optimal partitioning

of city based on several conditions such as the history of the traffic of streets[24], network-to-network communication between regions, the rate of data exchange and requested routes via vehicles. Precisely, CityBC proposes dynamic partitioning in order to reducing interoperable sub-blockchain, result in fast services as well as low network overhead.

4. Simulation Results

To evaluate and simulate the proposed approach, we used Common Open Research Emulator (CORE) network[25] and an open source IoT-Blockchain prototype[11] programmed in python. CORE daemon managing emulation sessions uses Python modules. They can be imported by Python scripts directly. Prototype uses the Pyro[26] technology of python that can integrate different parts of a heterogeneous system and allows to build applications in which nodes can communicate to each other over the network.

Fig.6 demonstrates a blockchain scenario in CORE network Emulator and executing a Blockchain-IoT based prototype on it including 10 options for simulation. In this figure, N2 is blockchain node, N3 is its peer node, N4 is a supportive node to act as the name server for the pyro4 technology and N5 act as the number of transaction sender to blockchain node, such as vehicles in intelligent transportation system.

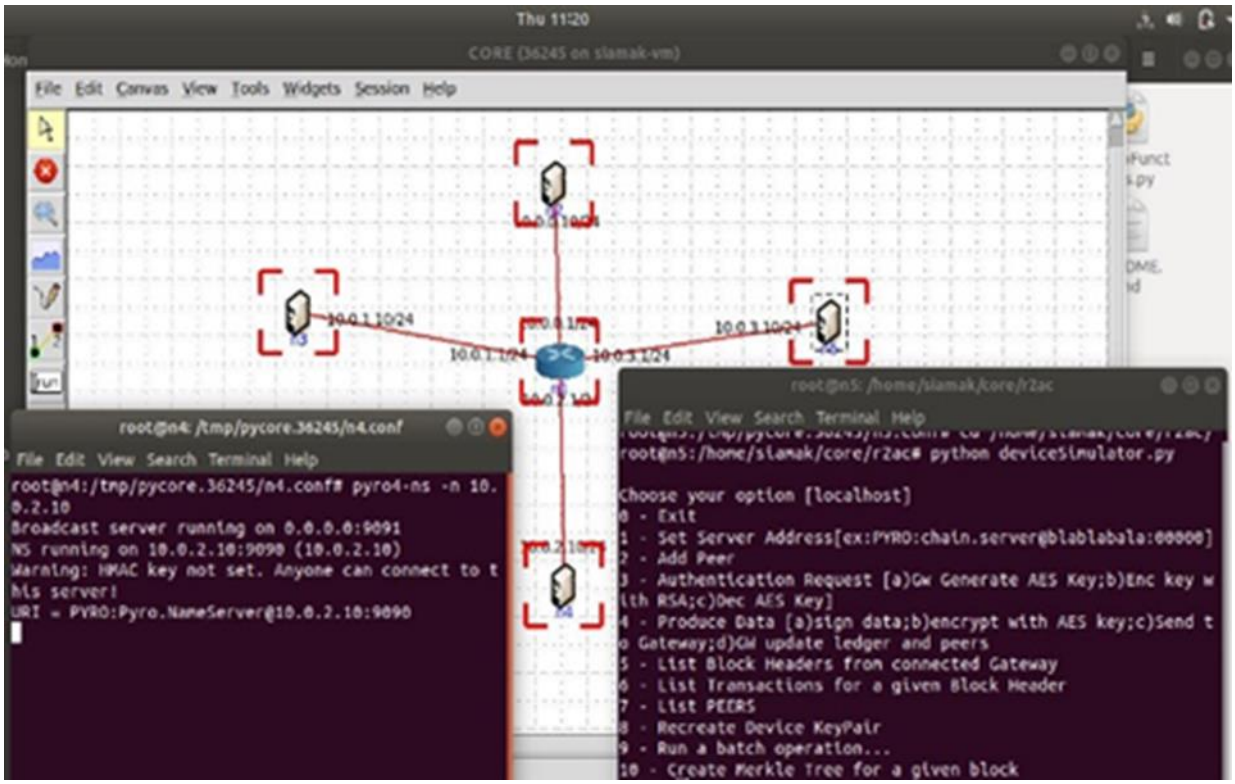

Fig. 6.CORE network Emulator and running imported python prototype

To evaluate the performance of last scenario, different number of transaction sender or participants sending transaction to a blockchain node like roadside infrastructure units (RSIs) are considered. Fig.7 shows the processing time of blockchain node for adding a new transaction to block for 20, 100, 500 and 1000 transaction senders (with 10 transaction) that is 5.2, 5.4, 6.1 and 7.9 millisecond, respectively

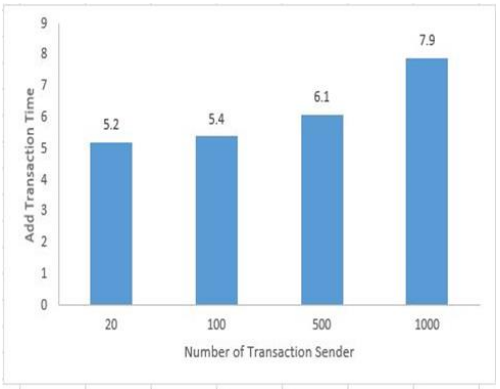

Fig. 7. The processing time of blockchain node for adding a new transaction

Fig.8 demonstrates the required time of updating peer blockchain node for received transaction form another node for various number of participants (20, 100, 500, and 1000) for each blockchain node.

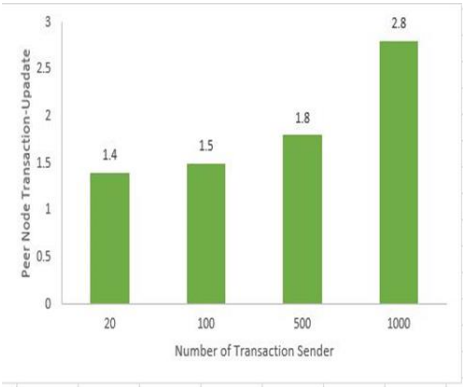

Fig. 8: The required time of updating peer blockchain node

To evaluate required time for adding a block, various number of transactions (10, 100, 500, and 1000) with fixed number of participants (100) per node are taken into account. Fig.9 shows 10 transactions takes 13.9 s and it rises to 16.61 for 1000 transactions, indicating that increasing the number of transactions demands more time to add a new block.

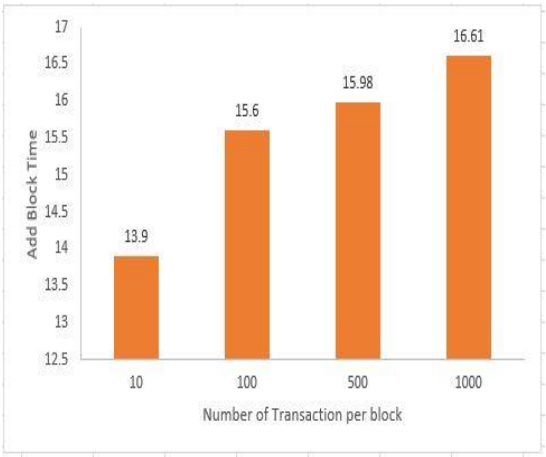

Fig. 9. Required time for adding a block with various transaction

Conclusion

In this paper, a scalable low-latency trusted data exchange architecture for large environment such as smart city is presented. To address security, privacy and centralization issues of such pervasive computing systems, decentralized approach of blockchain is utilized. In the integration of blockchain with such highly distributed systems, we propose a novel partitioning approach to divide a big city to several regions in which there are interconnected blockchain based nodes that share the preprocessed data of streets in corresponding sub blockchain network, result in reducing the overhead of networks as well

as storage resource. Another new approach used in this architecture is light blockchain and its associated smart manager in region-to-region network communication paving the way to low-latency response.

In addition to partitioning, two level blockchain network is proposed where one level, Fast BlockChain (FBC), is responsible for fast service and another, CityBC, stores data in long term and is responsible for analyzing the stored data to provide dynamic optimal partitions. The simulation results show that the results obtained by proposed approach is appropriate for providing fast and reliable blockchain-based platform to exchange trusted data in highly large distributed systems.

References

[1] R. Huo et al., "A comprehensive survey on blockchain in industrial internet of things: Motivations, research progresses, and future challenges," IEEE Communications Surveys & Tutorials, 2022.

[2] M. Kamruzzaman, B. Yan, M. N. I. Sarker, O. Alruwaili, M. Wu, and I. Alrashdi, "Blockchain and Fog Computing in IoT-Driven Healthcare Services for Smart Cities," Journal of Healthcare Engineering, vol. 2022, 2022.

[3] R. Jabbar et al., "Blockchain Technology for Intelligent Transportation Systems: A Systematic Literature Review," IEEE Access, 2022.

[4] B. Negash, A. M. Rahmani, P. Liljeberg, and A. Jantsch, "Fog Computing Fundamentals in the Internet-of-Things," in Fog Computing in the Internet of Things: Springer, 2018, pp. 3-13.

[5] A. Alkhateeb, C. Catal, G. Kar, and A. Mishra, "Hybrid blockchain platforms for the internet of things (IoT): A systematic literature review," Sensors, vol. 22, no. 4, p. 1304, 2022.

[6] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of Things (IoT): A vision, architectural elements, and future directions," Future generation computer systems, vol. 29, no. 7, pp. 1645-1660, 2013.

[7] R. Pepper, "Cisco Visual Networking Index (VNI) Global Mobile Data Traffic Forecast Update," 2013.

[8] P. K. Sharma and J. H. Park, "Blockchain based hybrid network architecture for the smart city," Future Generation Computer Systems, 2018.

[9] S. K. Kotamraju, P. G. Arepalli, L. N. Vejendla, and S. S. Kanumalli, "Implementation patterns of secured internet of things environment using advanced blockchain technologies," Materials Today: Proceedings, 2021.

[10] J. Bambara, P. Allen, K. Iyer, S. Lederer, R. Madsen, and M. Wuehler, "Blockchain: A practical guide to developing business, law, and technology solutions," ed: McGraw-Hill Education, 2018.

[11] R. A. Michelin et al., "SpeedyChain: A framework for decoupling data from blockchain for smart cities," arXiv preprint arXiv:1807.01980, 2018.

[12] A. Reyna, C. Martín, J. Chen, E. Soler, and M. Díaz, "On blockchain and its integration with IoT. Challenges and opportunities," Future Generation Computer Systems, vol. 88, pp. 173-190, 2018/11/01/ 2018, doi: https://doi.org/10.1016/j.future.2018.05.046.

[13] H. Sukhwani, J. M. Martínez, X. Chang, K. S. Trivedi, and A. Rindos, "Performance modeling of pbft consensus process for permissioned blockchain network (hyperledger fabric)," in Reliable Distributed Systems (SRDS), 2017 IEEE 36th Symposium on, 2017: IEEE, pp. 253-255.

[14] M. Vukolić, "The quest for scalable blockchain fabric: Proof-of-work vs. BFT replication," in International Workshop on Open Problems in Network Security, 2015: Springer, pp. 112-125.

[15] T. M. Fernández-Caramés and P. Fraga-Lamas, "A Review on the Use of Blockchain for the Internet of Things," IEEE Access, vol. 6, pp. 32979-33001, 2018, doi: 10.1109/ACCESS.2018.2842685.

[16] G. Fortino, H. Ghasemzadeh, R. Gravina, P. X. Liu, C. C. Poon, and Z. Wang, "Advances in Multi-Sensor Fusion for Body Sensor Networks:

Algorithms, Architectures, and Applications: Guest Editorial," ed: Elsevier, 2018.

[17] K. Bilal, O. Khalid, A. Erbad, and S. U. Khan, "Potentials, trends, and prospects in edge technologies: Fog, cloudlet, mobile edge, and micro data centers," Computer Networks, vol. 130, pp. 94-120, 2018.

[18] Y. Ai, M. Peng, and K. Zhang, "Edge computing technologies for Internet of Things: a primer," Digital Communications and Networks, vol. 4, no. 2, pp. 77-86, 2018.

[19] IOTA. "IOTA " https://www.iota.org/ (accessed 07/10/2018.

[20] K. Yeow, A. Gani, R. W. Ahmad, J. J. Rodrigues, and K. Ko, "Decentralized consensus for edge-centric internet of things: A review, taxonomy, and research issues," IEEE Access, vol. 6, pp. 1513-1524, 2018.

[21] S. Baru, "Blockchain: The next innovation to make our cities smarter," 2018 2018. Accessed: 27/09/2018. [Online]. Available: https://www.pwc.in/publications/2018/blockchain-the-next-innovation-to-make-our-cities-smarter.html

[22] M. S. Alnahari and S. T. Ariaratnam, "The Application of Blockchain Technology to Smart City Infrastructure," Smart Cities, vol. 5, no. 3, pp. 979-993, 2022.

[23] H. Kim and E. A. Lee, "Authentication and Authorization for the Internet of Things," IT Professional, vol. 19, no. 5, pp. 27-33, 2017.

[24] G. Chen and J. Zhang, "Applying Artificial Intelligence and Deep Belief Network to predict traffic congestion evacuation performance in smart cities," Applied Soft Computing, vol. 121, p. 108692, 2022.

[25] CORE. "CORE." https://www.nrl.navy.mil/itd/ncs/products/core (accessed 11/01/2019.

[26] Pyro4, "Pyro4," 2015. [Online]. Available: https://pythonhosted.org/Pyro4/