IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received November 30, 2021, accepted January 2, 2022, date of publication January 5, 2022, date of current version January 13, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3140401

A Secure and Efficient Trust Model for Wireless Sensor IoTs Using Blockchain

NADEEM JAVAID, (Senior Member, IEEE)
Department of Computer Science, COMSATS University Islamabad (CUI), Islamabad 44000, Pakistan
School of Computer Science, University of Technology Sydney (UTS), Ultimo, NSW 2007, Australia

e-mail: nadeemjavaid@comsats.edu.pk

ABSTRACT Wireless Sensor Internet of Things (WSIoTs) face various challenges such as unreliable data communication, less cost efficiency, security issues and high energy consumption due to their deployment in hostile and unattended environments. Moreover, the node's rapid energy dissipation due to the void holes and imbalanced network deployment has a bad impact on the network performance. To overcome the aforementioned issues, a blockchain based trust model for WSIoTs is proposed in this paper. Moreover, the Dijkstra algorithm is used to propose a routing protocol for performing efficient communication between network nodes while simultaneously avoiding void holes between ordinary sensor nodes and a sink node. Furthermore, to provide transparency in the network, all the transactions performed by the nodes are recorded in the blockchain in an immutable manner. Moreover, the Proof of Authority (PoA) consensus algorithm is used to validate and add the transactions in the blocks. Besides, a distributed platform, known as interplanetary file system, is used in WSIoTs for reliable and cost-effective storage. The simulation results show that PoA performs 13% better than proof of work consensus algorithm. The proposed routing protocol and trust model are validated in terms of gas consumption, throughput, nodes' status and energy consumption.

INDEX TERMS Blockchain, Dijkstra algorithm, trust model, void hole avoidance, Wireless Sensor Internet of Things.

I. INTRODUCTION

In Wireless Sensor Internet of Things (WSIoTs) [1], [2], the Ordinary Sensor Nodes (OSNs) are deployed in the environment for its monitoring. The OSNs send data to main servers, which are responsible to aggregate the data sensed by ONS. While the Sink Nodes (SNs) act as relay nodes that establish communication between OSNs and the main server. The applications of WSIoTs cover the areas of smart city, medicine, military, etc. The sensing nodes are either statically or dynamically deployed in a specific area to monitor and detect different events and to collect the respective data. WSIoTs may face several challenges: small data storage capacity, void holes, routing and security issues, and low throughput. Besides, for the reliable and efficient delivery of data across the network in WSIoTs, the designing of routing protocols has gained much importance. The routing path from the source to the destination is decided by these protocols. However, the protocols have some limitations due to the use of terrestrial medium such as low Packet Acceptance Ratio (PAR), short battery life, noise, void holes, etc. These parameters are important to check the reliability of the network.

Different types of protocols are proposed for optimal route finding including the geographic routing [3], fuzzy routing [4], transmission adjustment routing [5], etc. The geographic routing is also referred as position based routing that provides services, e.g., content-centric networking and location-aware services. Besides this, in WSIoTs, greedy algorithms are implemented for finding the shortest path. Some other techniques also collaborate with these greedy algorithms to find the optimal solution, i.e., the shortest path with minimum energy consumption and large network lifetime. In routing, the WSIoTs face the issues of rapid energy dissipation and the presence of void holes, which have a direct impact on the stability of the network. A large amount of energy dissipation and the creation of void holes degrade the performance of the network. A void hole is

The associate editor coordinating the review of this manuscript and approving it for publication was Barbara Masini.

a field inside a network where a node cannot detect its neighbors to transfer the received packet. The origin of a void hole is based upon the following two grounds: there is no forwarder node ahead and a node is dead due to excessive communication. Moreover, the IoSTs face many issues like a single point of failure, extra cost, network congestion, etc., due to the involvement of a third party [6]–[8]. To solve these issues, a decentralized protocol, referred as blockchain, is proposed, which operates without the involvement of any third party [9]–[11].

The rapid development in blockchain technology in the field of industry and academia has grabbed the attention of researchers [12]. The idea of blockchain was presented by a cryptography mailing list in 2008 [13]. Initially, it was implemented to perform secure and reliable financial transactions. Afterward, its applications were increased manifolds. Blockchain is considered as one of the influential technologies that bring many benefits. Some of them include security, transparency, immutability and trustworthiness. Its main characteristic is decentralization. Moreover, the blockchain maintains a peer to peer connection where all nodes are associated with each other and can equally participate in the network operations. With the integration of different systems, the blockchain also provides data encryption, distributed consensus and monetary incentives that are helpful to achieve trust in the environment. In literature, blockchain technology is used to overcome the issues related to security, data storage, untrustworthiness, single point of failure and extra cost due to third party involvement, etc.

Besides, the WSIoTs face several security issues, which are either caused by internal network nodes or external nodes. Sometimes, the network nodes become selfish and perform malicious operations to harm the network. The attacks performed by these nodes are known as internal attacks. On the other hand, some external nodes also try to hinder the network performance and they attack either the network nodes or the data. These types of attacks are known as external attacks. Both of these attacks are necessary to be avoided to maintain the reliability and efficiency of the network. In literature, many trust models and security protocols are presented to overcome the aforementioned issues. In this paper, a system comprising a trust model and a routing protocol is proposed for WSIoTs to overcome the aforementioned challenges. A part of this work is published in the conference [14]. The main contributions of the paper are as follows:

- a routing protocol using Dijkstra algorithm is proposed to find the shortest path from OSNs to the SN,
- the void holes are avoided during route finding procedure,
- the malicious activities in the network are avoided by a blockchain based trust model,
- PoA consensus mechanism is used to minimize the computational overhead, caused due to Proof of Work (PoW) and
- a distributed storage platform, known as Interplanetary File System (IPFS), is used to provide a cost effective data storage solution for WSIoTs.

The organization of the remaining paper is as follows. The problem statement is presented in Section II. An overview of the related work is presented in Section III. Section IV presents the proposed model and its detailed discussion. Moreover, the simulation results and their discussion are given in Section V. Furthermore, the effectiveness of smart contract and proposed model is discussed in Sections VI and VII, respectively. Finally, this work is concluded in Section VIII.

II. PROBLEM STATEMENT

The conventional routing protocols require a central trusted authority to ensure the identification and authentication of the network nodes. This trusted authority causes different issues like extra monetary cost and single point of failure. To solve these issues, blockchain based routing protocol is presented in [15], in which malicious attacks in the networks are avoided by the blockchain. However, this protocol uses the PoW mechanism, which reduces the network performance and increases the use of resources. Furthermore, the authors in [16] present a trust model based on blockchain technology for WSIoTs. However, the trust model uses the PoW consensus mechanism, which requires high energy consumption and powerful computational resources. So, the proposed system is not suitable for resource constrained sensor network environments. Moreover, a blockchain based privacy protection mechanism is proposed in [17] for providing privacy to the users in the crowdsensing networks. This mechanism uses the confusion mechanism encode algorithm to overcome the privacy issue of the users. However, it is concluded that using the confusion mechanism encode algorithm along with double-SHA256 algorithm in resource constrained crowdsensing algorithm involves unnecessary computations. Besides, an opportunistic routing algorithm is presented in [18] for void hole avoidance. The multi-hoping concept is utilized in this proposed algorithm to enhance the network's energy efficiency. However, the algorithm's security and reliability are not discussed. Moreover, in [16], [17], the entire data of WSIoTs is stored on the blockchain. However, it is very costly to store data on the blockchain.

III. RELATED WORK

Blockchain is used in different fields of life due to its several features. Therefore, in this section, existing work of blockchain in various fields is studied.

In [19], a consortium blockchain is used to provide secure services to the Internet of Things (IoT) devices. The services are provided to IoT devices from trusted edge service providers with low latency and high throughput. Moreover, the PoA consensus mechanism is used and compared with the PoW mechanism in terms of latency, throughput and packaging time. The authors in [20] highlight an issue in the conventional blockchain technology that it is difficult to be

implemented in mobile devices as the technology uses a PoW consensus mechanism, which requires high computational power for mining. This issue is tackled in [21] by presenting a blockchain based framework for mobile edge computing. In this framework, the multipliers based algorithm's alternating direction method and stochastic geometry theory are presented. The efficiency of the proposed algorithm is determined by its comparison with the benchmark algorithm.

A blockchain based data sharing system with integration of Long Range Wide Area Network (LoRaWAN) is presented in [22]. Trustfulness of network operators and the network coverage are maintained by a LoRaWAN server. Moreover, the network data is verified, however, it is not maintained for a long time. Furthermore, an incentive mechanism using blockchain is presented in [23], which motivates the users to store the data of WSNs. For data integrity, hashes are computed and compared with the preserved hashes of original data. The efficiency in the consensus mechanism is obtained through the Provable Data Possession (PDP) technique. Although, PDP is efficient to recognize damaged data, however, it cannot recover the data.

Furthermore, a blockchain based data transmission scheme is presented in [24]. It uses a multi-link communication tree to handle concurrent data transmission. Moreover, the proposed scheme is capable to handle nodes' failure in the network. It is observed from the experiments that the proposed scheme efficiently handles 15% node failure. However, in the presence of 30% failed nodes, both delay and communication time are increased. The authors in [25] address the issue of access control in data intensive applications by proposing a blockchain based scheme. In this scheme, deep learning technique and consensus mechanism are used to authenticate the channel state information. It is also analyzed that the spectral efficiency is increased by this scheme. In [26], a blockchain based branching scheme is presented to handle data of Intelligent Vehicles (IVs). Also, the data is tracked and verified by the blockchain. The trustworthiness of IVs is analyzed by presenting trust points. However, the duplication of data and state change are major issues with branching. The duplication increases with an increase in load. The authors in [27] analyze the need to revise the current storage mechanisms to save the storage space. A network coding based distributed storage framework is proposed and embedded into the existing blockchain to store the encoded packets of the block. Two deterministic techniques are implemented to save the storage space. The analysis shows that the storage space is largely saved using the above mentioned mechanism. However, due to different encoded packets on different nodes, a consensus mechanism cannot be applied to defend the system from pollution attacks.

IV. PROPOSED SYSTEM MODEL

With the invention of sensor networks, routing protocols have attained the attention of researchers. The protocols are used in sensor networks to send the data from one node to the other. However, they suffer from different challenges: finding the shortest path, congestion, low packet delivery ratio, void hole problem, etc. So, my concern is to find the shortest path by avoiding the void holes to efficiently deliver the data from OSNs to SN. In my model, the blockchain is deployed on SNs, and the credentials of all participating nodes and the environmental data are stored in IPFS. Besides, the data is transmitted through the shortest path. The procedure for the selection of this path is given below.

In order to send the sensed data, the OSN follows the shortest path. The OSN finds the distance from the nearby SNs using the x and y coordinates. The distance is calculated using the Euclidean distance formula [2]. After finding all possible routes, the shortest route is found towards the SN. For this purpose, only those nodes are selected that lie near SN because the selection of nodes lying far from that SN leads to longer routes and void holes. This path exploration is performed by the Dijkstra algorithm. Two sets of nodes are passed to the aforementioned algorithm. The first is a set with no void nodes and the second is the set with void nodes. Moreover, the Dijkstra algorithm finds the shortest paths from the source node to the SN by building a path tree using the set of nodes having minimum distance between them. After finding the paths, an optimal path is chosen from the set of minimum distance paths (shortest path tree). The path finding by Dijkstra algorithm compromises the privacy of network. However, my main objective is to achieve transparency in the network so that no node can perform a gray hole or a black hole attack. If any node performs a gray hole or a black hole attack, it will be easily detected and removed from the network. In this way, my proposed model ensures secure and real time data transmission.

Algorithm 1 Shortest Path Calculation by Dijkstra Algorithm

```
Function Dijkstra
Input Parameters: (S, s, d)
Output: shortest distance matrix distance[U]
distance[s] = 0
For (each V in S)
  If (V != s)
    distance[V] = infinity
  End If
  V → Q
  While (Q != ∅)
    V = V ∈ Q having minimum distance[V]
    remove V from Q
  End While
end For
For (each U of V)
  New_dis = distance[V] + length(V, U)
  If (V != s)
    distance[U] = New_dis
  End If
end For
return distance[U]
```

The network deployment is of two types: sparse and dense. In the sparse deployment, the void holes are created when a large number of sensor nodes die in the dense deployment. Moreover, the reasons behind the creation of void holes are as follows. Either the nodes in a particular region die because of excessive energy usage or a node does not have any forwarder node within its communication range. In FIGURE 1, a node

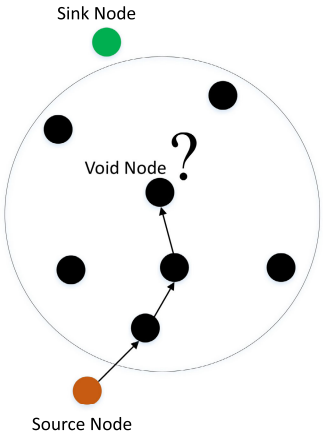


FIGURE 1. Occurrence of a void hole.

with question mark (?) sign shows the occurrence of a void hole. It means that the question marked node cannot send data to the next nodes, as it has no node in its transmission range.

Initially, the distance of each OSN to the nearest SN is evaluated using the Euclidean distance formula. Afterward, the void hole nodes are found, which are simply ignored, as they lead to the wastage of the computational resources. As the battery of a network has a direct relation with the operational life of a sensor network ($Network_{life} \propto Network_{residual\ energy}$), so, battery wastage means less network lifetime. Moreover, a set is maintained, which has nodes with no void holes.

In the current scenario, the Dijkstra algorithm takes the set with no void holes along with the starting and finishing Identities (IDs) of source nodes and SNs, respectively. The other inputs like details of storage and computation capabilities are provided to the algorithm in segments. These segments are $M \times 3$ matrices with the format of [ID, n1, n2]. Here, ID, n1 and n2 represent integer values, starting ID and finishing ID, respectively.

Algorithm 1 takes three parameters as input: S, s and d. Here, S, s and d represent segments, source ID and destination ID, respectively. Initially, the distance is set as ''0'' from the source to the destination. For each vertex, the distance from the source to the other exploring nodes is set to infinity (lines 6-8). The vertex (V) is added to the queue (Q) (line 9). Where, V is explored with minimum distance between source and destination (lines 10-13). Afterward, the shortest distance among the neighbors that are not yet explored is calculated (lines 16-20). Here, New_{dis} is the new shortest distance and distance[U] is the final shortest distance, which is returned by the function.

A. PROPOSED TRUST MODEL

In this paper, the procedure for routing from OSNs to the SNs is defined. The proposed routing protocol follows the shortest path using the Dijkstra algorithm from the source nodes to the SNs. Void hole avoidance is also performed to improve the performance of the network. Moreover, the security of the network is maintained and malicious activities by the nodes are prevented by a blockchain based trust model. Additionally, PoA is used instead of PoW to avoid high cost computational operations. Firstly, the key components of the trust model are discussed. Secondly, the brief discussion of the trust model is provided.

1) KEY COMPONENTS

The permissionless blockchain is used in my network because the network is deployed for environmental monitoring where the users should be allowed to request data from SN without permission. Any end user can request data from SN. The permissionless blockchain is deployed on SNs and main servers. The SNs collect and aggregate the data and the main server is responsible for validating transactions while monitoring of the network. The key components of the system model are OSNs, SNs and the main server.

- **Ordinary Sensor Nodes:** They are the most important nodes in the network that perform the monitoring of surroundings and collection of environmental data. Afterward, these nodes send the gathered data to the nearest SNs.
- **Sink Nodes:** They are the relay nodes that establish communication with OSNs, SNs and the main server. Three main operations that are performed by the SNs include adding new nodes, collecting data and executing the smart contract. SNs are responsible to add new nodes in the network after consensus development by PoA. Moreover, the OSNs sense the data from environment and send it to SNs where it is stored temporarily. The data of each OSN is distinguished by the SN using its location and ID. This data is stored to remove redundancy from the data. Then, this data is sent to IPFS for distributed storage and saving the data storage cost of blockchain. The blockchain is deployed on SNs and the main server that contains a database to permanently store the transactions of nodes to provide transparency in the network. Furthermore, there are two types of data that are stored on SNs: transaction data and sensed data. The transaction data consists of all the details of transactions performed between two nodes. While the sensed data is the environmental reading that is sensed by sensor nodes in the network. Moreover, the SNs can access the main server's data. Although the main server publishes the smart contract, it is executed by the SN to validate and store data.
- **Main Server:** It is a trustworthy entity in the network, which is responsible for processing the data sensed by OSNs, publishing smart contracts and assigning activity to nodes. All the transactions' records of ONs are stored on the main server and SNs because blockchain is deployed on them. Moreover, the transactions are distinguished by their locations and IDs. The transactions are stored in an immutable database, which can only be accessed by the main server and the authorized SNs.

In FIGURE 2, by keeping in mind the ease of readers, the overall working of the proposed system model is summarized

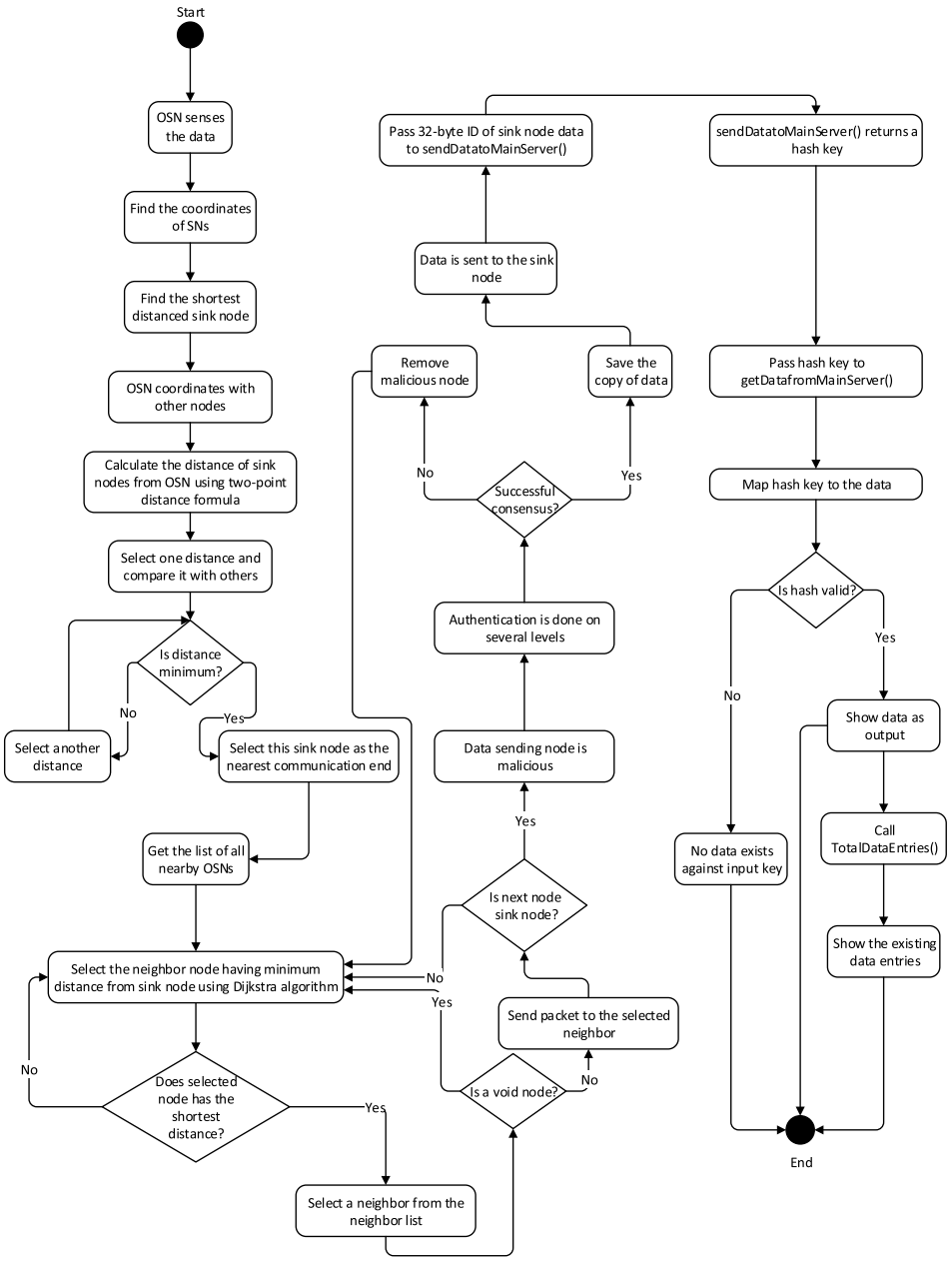


FIGURE 2. **Workflow of the proposed system model.**

and presented in the form of a flowchart. Moreover, FIGURE 3 presents the components involved in the proposed blockchain based trust model and their connections with each other. The OSNs establish connections with SNs to send the sensed data of their surroundings to them. The SNs are connected with the main server. They receive the data, and further transfer it to other SNs and main server for processing and storage. These nodes validate the data of other nodes. Moreover, SNs maintain a distributed ledger to record communication operations of all network nodes. The main server examines the network and its nodes. If any node stops working or performs malicious operations, it is discarded by the main server.

The data sent by the SNs is secured using the private key. This encryption key is unique and the data is only accessible using this key. The process of data requesting and accessing is presented in FIGURE 4. From the figure, it is clear that an SN requests for data using a hash value. The main server first checks the authenticity of SN. In case 1, if an SN is authentic, then its input hash is mapped with the data. If the main server finds any data regarding the input hash, then the requested data is sent to the SN. In case 2, if SN is not authenticated by the main server, it means that the SN is malicious and the main server will not approve its request for data. In case 3, where SN is validated but data is not found for the entered hash, then the request of SN is turned down with a response

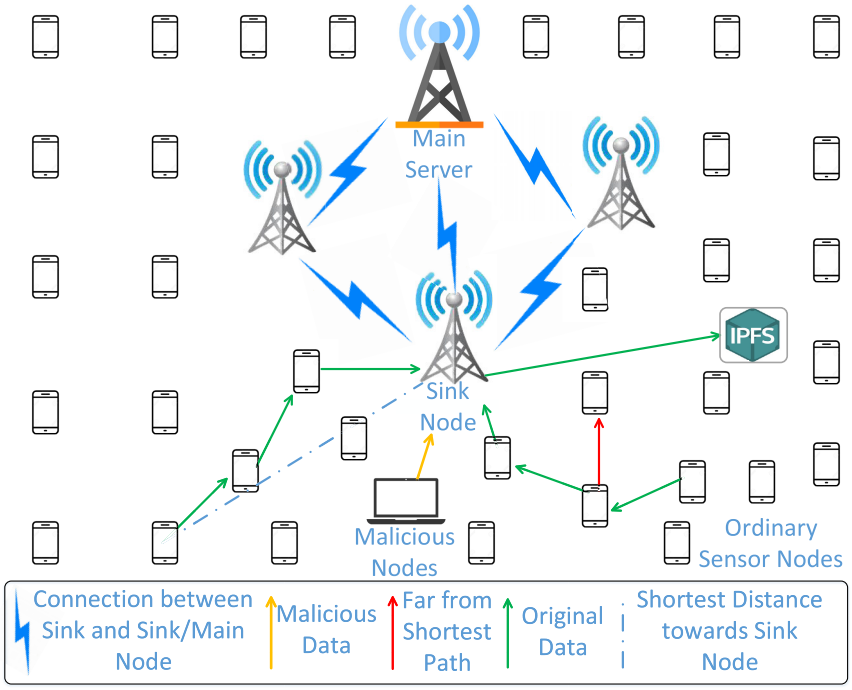


FIGURE 3. The proposed blockchain based system model.

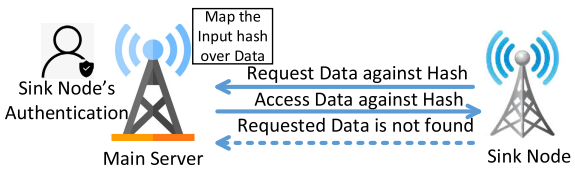


FIGURE 4. Data requesting and accessing process.

message. The response message is like "Requested data is not found". This approach further increases the security of the network along with the privacy of data.

2) TRUST MODEL

In my blockchain based trust model, the ordinary nodes initially sense data from the environment. Then, these nodes send the data to sink nodes for further processing. The sink nodes collect the data from ordinary nodes and aggregate this data. In the aggregation of data, the redundancy is removed by sink nodes. Moreover, the sink nodes are also responsible for adding new nodes into the network. These nodes are added after validating them by the validator. In my proposed model, the PoA consensus mechanism is used. PoA mechanism is responsible for validating the transactions in the network and adding the blocks into the blockchain. In PoA, each block is validated and integrated into the blockchain by an authorized group of nodes.

PoA is not computationally intensive like PoW. The reason is that the miners in PoW, who are responsible for adding

Algorithm 2 Blockchain Based Trust Model

1. **Function** Trust Model
2. /*ON are ordinary sensors*/
3. /*SN is the sink node that aggregates the data of ON*/
4. /*D_i is the data that is sensed by ON*/
5. **Input Parameters**: (ON, SN, D_i)
6. /*V is verdict about node*/
7. **Output**: V
8. $D_i \rightarrow SN$
9. **For** (i=0; i $\leq ON$; i++)
10. Aggregated Data $= D_i$
11. **End For**
12. **PoA Consensus Algorithm**
13. **For** (all performed transactions)
14. **If** (transaction is validated by the validator)
15. Transaction is valid
16. Block is added into the blockchain
17. V = The node is legitimate
18. **Else**
19. The transaction is malicious
20. Node is revoked and removed from the network
21. V = The node is malicious
22. **End If**
23. **End For**

the block into the blockchain, are selected after solving a mathematical puzzle. All the miners are provided with a complex puzzle and they have to solve it as soon as possible. The miner that solves the puzzle first becomes the miner node. This node is then responsible for validating the transactions and adding blocks into the blockchain. Therefore, a large amount of computational overhead is incurred in PoW during the selection of miner nodes. On the other hand, in PoA, there is pre-selected node (validator) is responsible for validating the transactions and adding the blocks. The nodes are selected on the basis of their reputation

in the network. However, an issue with PoA is that it makes the network centralized to some extent and compromises the distributed nature of blockchain. Moreover, the transactions are monitored by validator and the nodes that perform malicious transactions are detected. The detected malicious nodes are restricted from performing further transactions and immediately removed from the network. Moreover, As a pre-selected node is responsible instead of 51% nodes for validating the transaction, therefore, my proposed network is a byzantine fault tolerant network. my blockchain based trust model is explained in Algorithm 2.

Furthermore, in PoA, a hash value is calculated for each transaction using a hash function. The generated hash value is also known as transaction hash. In the hash function of PoA, an input value of random size is selected and converted into a fixed-size hash value. Equation 1 shows the hashing operation performed by a hash function in PoA consensus algorithm.

$$f(a) = b. \tag{1}$$

where the input value is represented by a while b shows the generated hash value for the input a. For instance, the fixed-size hash value generated by the Keccak-256 for the input value ''hello'' is ''1c8aff950685c2ed4bc3174f3472287b56d9517b9c948127319a09a7a36deac8''. Once the hash of any data is generated, it cannot be changed into original data (de-hashing). So, it does not reveal the original data.

3) INTERPLANETARY FILE SYSTEM

The nodes in WSIoTs are resource constrained and do not have much storage and computational resources, so, the data of all these nodes is stored on SN after its processing. Moreover, I have implemented blockchain on the sink nodes. Therefore, when the data of the whole network is stored on the blockchain, then the monetary cost of data storage is too high. The reason is that when 1 MB of data is stored on the blockchain, it costs 14151.68 US dollars [28]. To solve this issue, I use IPFS in my WSIoTs network. IPFS is a storage platform in which the data is stored distributedly on different devices. The workflow of IPFS is given below and explained in Algorithm 3.

1) SN requests IPFS for storing its processed data by sending this data to IPFS, encrypted by its private key.
2) IPFS receives this data, decrypts it and calculates the hash of data. Then this hash is sent to SN as the ID of data.
3) In this step, IPFS divides the data into small chunks of 256 kBs and stores them on different distributed devices.
4) When the SN wants its data, it requests the data using the hash provided by IPFS in step 2.
5) After the request of data is received, IPFS collects the data from distributed devices and aggregates it.
6) In the last, the aggregated data is encrypted with the private key of SN and is sent to this SN.

Algorithm 3 Interplanetary File System for Distributed Storage

Function IPFS
/*Sink node requests the *IPFS* for data storage*/
Input Parameters: ($IPFS$, $Hash_{ipfs}$)
/*D is the data that sink node requests to store*/
Output: D
Step 1 - Data Storage Request
Sink node sends request to *IPFS*
If Ethereum address of sink node exists in blockchain
  Hash of data $Hash_{ipfs}$ is calculated
  IPFS divides the data in small chunks
Step 2 - Data Acquirement Request
  SN requests the data from *IPFS*
  If $Hash_{ipfs}$ is present on *IPFS* server **then**
    IPFS aggregates the data
    IPFS sends data to the *SN*
  Else
    message(The data is not present)
  EndIf
Else
  SN is a malicious node
EndIf

V. SIMULATION RESULTS AND DISCUSSION

In this section, the smart contract is evaluated in terms of transaction and execution costs, and an in-depth discussion of the simulation results is presented. The proposed model is simulated using Remix Integrated Development Environment (IDE), MetaMask, Ganache and MATLAB. The smart contract is developed in the Remix IDE. All the rules of data transfer are written in Solidity language and the nodes in the network follow these rules to request and send the data. On the other hand, all nodes are deployed in MATLAB and the simulation of network in sending and receiving the data packets is performed in MATLAB. I iterate the network for 3000 rounds. Simulation parameters are shown in TABLE 1. Throughput is the measure of the number of packets a system can process within a unit time. FIGURE 5 depicts that the throughput of the deployed network gradually increases with the increase in the number of rounds. Initially, the network has low throughput because the network is going through the deployment phase and other procedures like shortest path finding and void hole avoiding phases. When all the shortest routes from OSN are found and void holes are avoided, then the network becomes stable. The network throughput is only used to send and receive the data by different nodes. It is the reason that performance of the network is increased after route finding and void hole avoidance phases.

Moreover, PAR is another important parameter to check the reliability of the network. According to FIGURE 6, PAR is maximum during initial rounds of the network, however, it decreases continuously after 1100^{th} round [14]. At the start, the network shows maximum output because at this time all nodes are active and working properly. As the rounds pass, the energy of nodes depletes in sending and receiving the packets. Therefore, the nodes begin to die one by one, which directly affects the performance of the network. The figure shows that after 2500 rounds, all the nodes die due to their energy depletion, which results in zero PAR value.

TABLE 1. Simulation parameters for the proposed routing protocol.

Parameter	Value
Network dimensions	1500m × 1500m
Location of SN 1	325 × 325
Location of SN 2	1050 × 325
Location of SN 3	325 × 1050
Location of SN 4	1050 × 1050
Location of SN 5	750 × 1230
Location of main server	750m × 750m
Number of OSNs	150
Number of SN	5
Number of main servers	1
Transmission range of each OSN	100m
Data transmitting energy	$5.0\ e^{-04}$ J
Data receiving energy	$5.0\ e^{-08}$ J
Initial energy of each OSN	7 J
Maximum number of rounds	3000
Alive node's status	1
Dead node's status	0

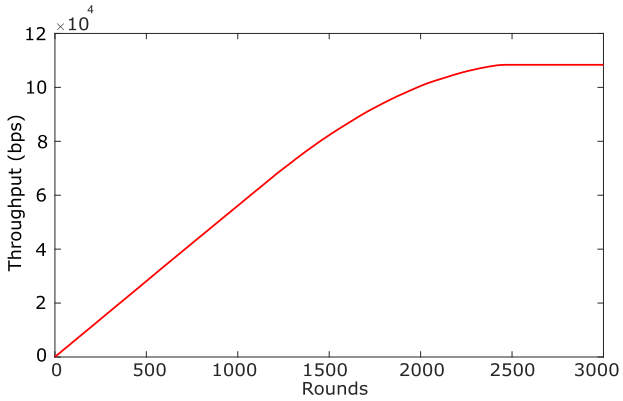


FIGURE 5. Throughput of the network.

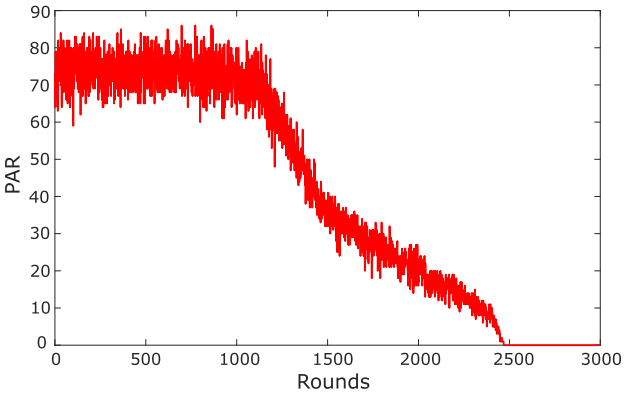


FIGURE 6. PAR of the network.

This zero PAR value means that the network has stopped working.

Besides, the residual energy has a direct relation with the lifetime of the network. The residual energy of the whole network is shown in FIGURE 7. Initially, the total residual energy of the network is 1050 Joules (J). With the increase

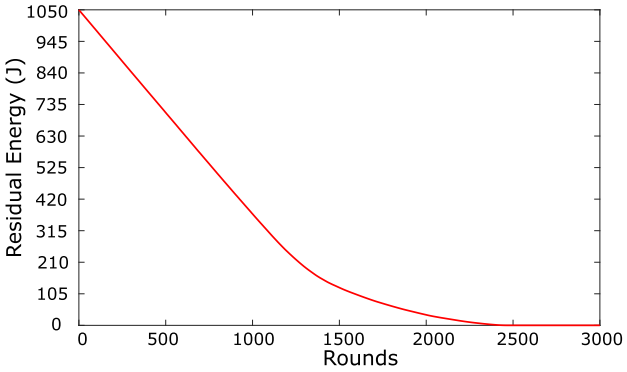


FIGURE 7. Residual energy of OSNs.

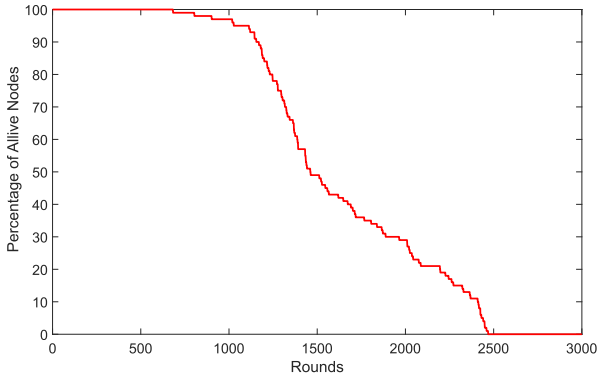


FIGURE 8. Percentage of alive nodes.

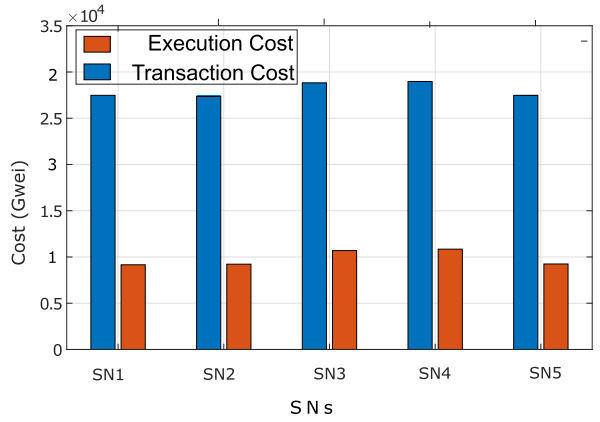


FIGURE 9. State checking costs of SN.

in the number of rounds, every node uses energy in different operations like finding the shortest path, avoiding void holes, sending and receiving the data packets, etc. It causes a gradual decrease in the residual energy of all participating nodes. Hence, the residual energy of the whole network depletes after 2500 rounds and the network collapses, as shown in FIGURE 7. In the same way, FIGURE 8 shows the percentage of alive nodes in the network. The nodes in the working state are known as alive nodes. The figure depicts the liveliness of sensor nodes with the increasing number of rounds. All sensor

TABLE 2. Costs for states of the sink nodes.

Parameters	Values
transaction hash	0xfbaef, ..., 840f96
from	0xca35b7, ..., fa733c
to	Clustering.StateOfSNs() 0xbbf28, ..., 732db
transaction cost	27886 gas (Cost only applies when called by a contract)
execution cost	6614 gas (Cost only applies when called by a contract)
hash	0xfbaef, ..., 840f96
input	0x38e, ..., fccad
decoded input	{}
decoded output	{"0": "string: Current state of Sink Node 1 is: 1", "1": "string: Current state of Sink Node 2 is: 1", "2": "string: Current state of Sink Node 3 is: 0", "3": "string: Current state of Sink Node 4 is: 0", "4": "string: Current state of Sink Node 5 is: 1", "5": "string: State of selected Ordinary Node is: 1"}
logs	[]

nodes are alive till approximately 2500 rounds after which no further transmission of data is observed.

A. SMART CONTRACT EVALUATION

The smart contract is a piece of code that has business rules and operates without the involvement of any third party. Whenever transactions are performed in the Ethereum blockchain, a cost is paid in terms of gas. There are two types of costs: transaction and execution. The former is the cost calculated during the deployment of smart contract code in the Ethereum blockchain. Whereas, the latter is the cost calculated during the execution of smart contract functions.

In the blockchain, for each computational operation, the amount of gas is fixed. Even with the changes in the value of ethers, the gas cost does not change. Whereas, the price or value of ether changes according to the market operations. The simulations are performed by taking transaction and execution costs from the Remix IDE. From the results, it is seen that the execution cost is less as compared to the transaction cost. The main reason is that the transaction cost is paid when deploying a smart contract in the blockchain network while execution cost is paid when executing a function of the smart contract. The costs are calculated until all the nodes in the network are dead. Moreover, the state (active or inactive) of an SN is evaluated through other SNs and the main server. It is also assumed that when the node is in active state, it is considered as a trusted SN. It is seen from FIGURE 9 that the transaction and execution costs incurred when checking the states of SNs are approximately the same. The reason is that the same function of the smart contract is used by all SNs to check the state of any particular SN. The costs for the states of SNs are given in TABLE. 2.

Moreover, in FIGURE 10, the transaction performed by five individual SNs is evaluated in terms of transaction and execution costs. From the figure, it is observed that both costs for each SN are the same. All SNs in the network perform almost the same tasks like getting data from OSNs, adding new nodes and finding the shortest route. Furthermore,

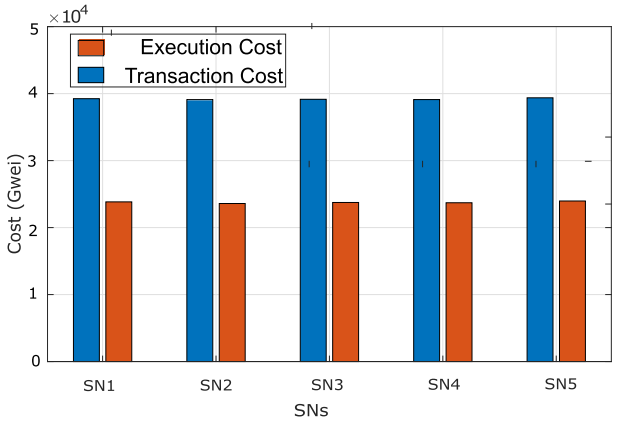


FIGURE 10. Gas consumption of SNs.

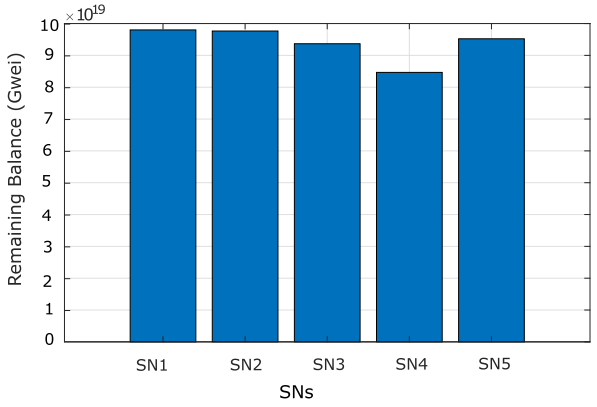


FIGURE 11. Remaining balance of SNs.

TABLE 3. Costs for different operations.

Function	Execution Cost (Gwei)	Transaction Cost (Gwei)
Network Deployment Cost	4272505	5674465
Costs for Checking Individual SN's State		
SN 1	9155	30427
SN 2	9221	30493
SN 3	10692	31964
SN 4	10846	32118
SN 5	9243	30515
SNs 1 Present State	6639	27911
SNs 2 Present State	6614	27886
Costs for Data Packet Processing		
SN 1	23837	46325
SN 2	23595	46019
SN 3	23749	46237
SN 4	23705	46193
SN 5	23969	46457

TABLE 3 shows the comparison of both costs for different functions.

FIGURE 11 shows the remaining balance of all SNs after transferring different data packets. These balances are written in "Gwei", which is the unit of gas in the Ethereum network. Moreover, this balance is converted

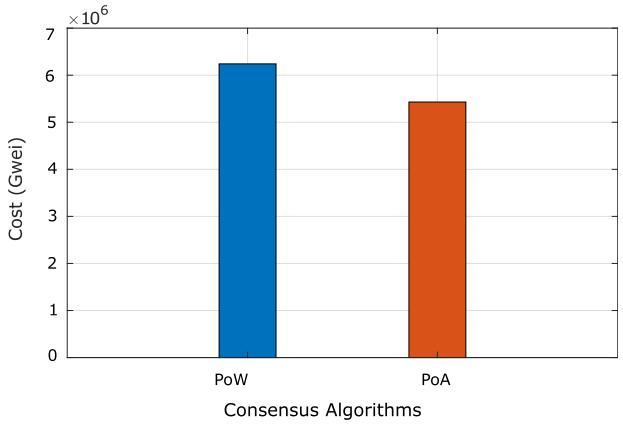


FIGURE 12. Gas consumption of PoW and PoA.

into ethers, which are used to pay the transaction fees. The conversion of balance into ethers is done by dividing the current balances with $1 * 10^{17}$. Besides, the remaining balances of SN 1, SN 2, SN 3, SN 4 and SN 5 are "97999999999994051982", "97655551010110101020", "93655551010115551040", "84656210176217804110" and "95213527631076317630" Gwei, respectively. From the figure, it is observed that SN 4 has the lowest balance among all other SNs because it receives and processes more data packets as compared to other SNs. The transaction order of SNs is given as $trans_{maxtomin} = SN\ 4 > SN\ 3 > SN\ 5 > SN\ 2 > SN1$. Where, $trans_{maxtomin}$ is the maximum to minimum transaction order in which SNs perform the transactions.

Moreover, the transaction cost of PoW and PoA is also compared, as shown in FIGURE 12. It is seen from the figure that the transaction cost for PoW and PoA is 6241214 Gwei and 5429857 Gwei, respectively. These results show that PoA performs almost 13% better than PoW. The reason is that in PoW, a complex mathematical puzzle needs to be solved for the selection of the miner node. On the other hand, in PoA, a pre-selected validator is responsible for validating the transactions. The validator is selected from all interested SNs and there is no need of solving the puzzle. It is the reason that the gas consumption in PoA is less than PoW. However, PoA compromises the decentralized nature of blockchain because the validator is a pre-selected node that validates the transaction and adds the block into the blockchain. The validator is selected on the basis of its reputation score, it is the reason that reputation is needed in my network. Due to the selection of validator on the basis of reputation score in PoA, the blockchain network becomes centralized to some extent.

VI. SMART CONTRACT ANALYSIS

I have considered the following security parameters to show the effectiveness of my smart contract. Figure 13 shows the security analysis of my smart contract using Oyente.

```
(venv)root@12718c3fb077:/home/oyente/oyente# python oyente.py -b Consensus.sol
Running, please wait...
           ============ Results ============
           Time Dependency      False
           Concurrency Bug      False
           Reentrancy Bug       False
```

FIGURE 13. Security analysis by Oyente.

A. TIME DEPENDENCY

Time dependency is one of the major attacks that are performed by the malicious miner nodes. In the blockchain, when a block is created, a timestamp is given to it. This time shows the generation time of the block. In a time dependency attack, the malicious miner nodes try to change the timestamp of a block for their interests. In my model, the time dependency attack is not possible because the PoA consensus algorithm is used. In this consensus algorithm, there is a pre-selected trusted validator, which is responsible for validating the transactions and adding the blocks into the blockchain. When any malicious miner node tries to change the timestamp of a block, it should be validated by the validator in the network. Therefore, the time dependency attack cannot be performed in my smart contract.

B. CONCURRENCY BUG

This attack occurs when more than one function of the smart contract are triggered simultaneously. When two or more independent functions are triggered at the same time, then undesired outcomes are occurred, which affect the overall performance of the network. In my model, all the functions of the smart contract are triggered one after the other and this attack can not be performed, as shown in Figure 13.

C. REENTRANCY BUG

In a reentrancy attack, the malicious node calls a faulty function again and again to affect the performance of the network. When any faulty function is called in a recursive manner, then other functions can not be executed and the overall operability of the network is affected. In my model, this attack can not be performed because when an attacker triggers a faulty function again and again, it would be identified by SNs. The SNs can easily identify the malicious node because the smart contract is deployed on them and pre-selected node is responsible for validating the transaction. It is the reason that this attack can not be performed in my smart contract.

VII. FORMAL SECURITY ANALYSIS

The formal security analysis of the proposed system model is carried out by considering the following attacks.

A. MAN IN THE MIDDLE ATTACK

As the nodes in the WSIoTs are distributed, therefore, it is very important to establish trust in the WSIoTs. Sometimes, the quality and credibility of data are compromised due to the man in the middle attack. In this attack, a malicious node

secretly listens to the communication happening between two parties and tries to alter the data that they exchange. However, this attack is not possible in my proposed model because the data is sent and received by the nodes in an encrypted form. Therefore, any malicious node can not get the actual data and change it. In this way, the credibility of data is achieved in my proposed model.

B. SYBIL ATTACK

Privacy leakage is one of the critical issues in the WSIoTs. Some malicious nodes forge the IDs of legitimate nodes in the network and pose to be legitimate nodes with a large number of forged identities. To solve this issue, the public key infrastructure is used in my proposed model. In this way, all the data (either from OSNs to SNs or from SNs to IPFS) and nodes' credentials (to SNs for registration) is sent in an encrypted form. The sender encrypts the data or credentials with the private key of the receiver, which is only known to the entity itself. So, the data or credentials are readable only by the receiver and no malicious node in the network can read them because it does not have the decryption key (private key of the receiver).

C. DATA TAMPERING ATTACK

Data integrity is another security parameter that is to be ensured in a trustworthy network. In my model, the data integrity is ensured using the Keccak-256 hashing technique. When the data is requested by any node in the network, its hash is calculated and stored on the blockchain. After this, the data in the encrypted form is sent to the requester. The requester after receiving the data calculates its hash on its own and compares it with the hash stored on the blockchain. If these two hashes match, then the data is considered to be valid and data integrity is ensured.

D. BLACKHOLE AND GRAYHOLE ATTACKS

In grayhole and blackhole attacks, the malicious relay nodes either randomly or constantly drop packets [29]. In this way, the packet does not reach the destination, which reduces the efficiency of a network. However, these attacks are not possible in my network because when the path is found by Dijkstra algorithm, the routing table is made and stored at the source node. The malicious nods will be easily detected when it drops the packets frequently. Therefore, my proposed model ensures secure data transmission.

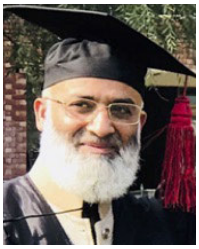
VIII. CONCLUSION AND FUTURE WORK

In WSIoTs, sensor nodes are deployed that are responsible for monitoring the nearby area, gathering data and transferring it to other nodes via a wireless medium. However, certain challenges limit the performance of these networks, which are no shortest path, void holes, rapid energy consumption, etc. To solve these problems, a routing protocol is introduced, which finds the shortest path using the Euclidean distance formula and the Dijkstra algorithm. Additionally, the proposed algorithm avoids the void holes, which results in less battery usage of the network nodes and ultimately helps the network in the long run. Moreover, the security of the WSIoTs is maintained by the proposed blockchain based trust model. Therefore, the nodes in the network can communicate securely. Furthermore, the PoW consensus mechanism is replaced with the PoA consensus mechanism because PoW includes extra computations, which result in network performance deficiency. Furthermore, IPFS is used for reliable and cost-effective storage in the WSIoTs. I have performed extensive simulations to testify the routing protocol and trust model. The simulation results show that the PoA consensus mechanism consumes almost 13% less gas as compared to the PoW consensus mechanism. In the future, I will integrate blockchain technology with other existing routing protocols and will make a comparison to evaluate the performance of each protocol.

REFERENCES

[1] N. Javaid, A. Sher, H. Nasir, and N. Guizani, "Intelligence in IoT-based 5G networks: Opportunities and challenges," *IEEE Commun. Mag.*, vol. 56, no. 10, pp. 94–100, Oct. 2018.
[2] N. Javaid, "Integration of context awareness in internet of agricultural things," *ICT Exp.*, to be published, doi: 10.1016/j.icte.2021.09.004.
[3] K.-V. Nguyen, C.-H. Nguyen, P. Le Nguyen, T. Van Do, and I. Chlamtac, "Energy-efficient routing in the proximity of a complicated hole in wireless sensor networks," *Wireless Netw.*, vol. 27, no. 4, pp. 3073–3089, May 2021.
[4] M. Selvi, S. V. N. Santhosh Kumar, S. Ganapathy, A. Ayyanar, H. Khanna Nehemiah, and A. Kannan, "An energy efficient clustered gravitational and fuzzy based routing algorithm in WSNs," *Wireless Pers. Commun.*, vol. 116, no. 1, pp. 61–90, Jan. 2021.
[5] N. Javaid, "NADEEM: Neighbor node approaching distinct energy-efficient mates for reliable data delivery in underwater WSNs," *Trans. Emerg. Telecommun. Technol.*, p. e3805, Dec. 2019.
[6] W. Ali, I. U. Din, A. Almogren, M. Guizani, and M. Zuair, "A lightweight privacy-aware IoT-based metering scheme for smart industrial ecosystems," *IEEE Trans. Ind. Informat.*, vol. 17, no. 9, pp. 6134–6143, Sep. 2021.
[7] I. Ullah, N. U. Amin, A. Almogren, M. A. Khan, M. I. Uddin, and Q. Hua, "A lightweight and secured certificate-based proxy signcryption (CB-PS) scheme for E-prescription systems," *IEEE Access*, vol. 8, pp. 199197–199212, 2020.
[8] S. D. Muruganathan, D. C. F. Ma, R. I. Bhasin, and A. O. Fapojuwo, "A centralized energy-efficient routing protocol for wireless sensor networks," *IEEE Commun. Mag.*, vol. 43, no. 3, pp. S8–13, Mar. 2005.
[9] P. P. Ray, B. Chowhan, N. Kumar, and A. Almogren, "BIoTHR: Electronic health record servicing scheme in IoT-blockchain ecosystem," *IEEE Internet Things J.*, vol. 8, no. 13, pp. 10857–10872, Jul. 2021.
[10] S. Namasudra, G. C. Deka, P. Johri, M. Hosseinpour, and A. H. Gandomi, "The revolution of blockchain: State-of-the-art and research challenges," *Arch. Comput. Methods Eng.*, vol. 28, no. 3, pp. 1497–1515, May 2021.
[11] A. Bhardwaj, S. B. H. Shah, A. Shankar, M. Alazab, M. Kumar, and T. R. Gadekallu, "Penetration testing framework for smart contract blockchain," *Peer–Peer Netw. Appl.*, vol. 14, no. 5, pp. 2635–2650, Sep. 2021.
[12] M. Allouche, M. Mitrea, A. Moreaux, and S.-K. Kim, "Automatic smart contract generation for internet of media things," *ICT Exp.*, vol. 7, no. 3, pp. 274–277, Sep. 2021.
[13] D. Unal, M. Hammoudeh, and M. S. Kiraz, "Policy specification and verification for blockchain and smart contracts in 5G networks," *ICT Exp.*, vol. 6, no. 1, pp. 43–47, Mar. 2020.
[14] A. Mateen, J. Tanveer, N. A. Khan, M. Rehman, and N. Javaid, "One step forward: Towards a blockchain based trust model for WSNs," in *Proc. 14th Int. Conf. P2P, Parallel, Grid, Cloud Internet Comput. (3PGCIC)*, 2019, pp. 57–69.
[15] G. Ramezan and C. Leung, "A blockchain-based contractual routing protocol for the Internet of Things using smart contracts," *Wireless Commun. Mobile Comput.*, vol. 2018, pp. 1–14, Nov. 2018.

[16] W. She, Q. Liu, Z. Tian, J.-S. Chen, B. Wang, and W. Liu, "Blockchain trust model for malicious node detection in wireless sensor networks," *IEEE Access*, vol. 7, pp. 38947–38956, 2019.
[17] B. Jia, T. Zhou, W. Li, Z. Liu, and J. Zhang, "A blockchain-based location privacy protection incentive mechanism in crowd sensing networks," *Sensors*, vol. 18, no. 11, p. 3894, 2018.
[18] Z. Rahman, F. Hashim, M. F. A. Rasid, and M. Othman, "Totally opportunistic routing algorithm (TORA) for underwater wireless sensor network," *PLoS ONE*, vol. 13, no. 6, Jun. 2018, Art. no. e0197087.
[19] M. Liu, F. R. Yu, Y. Teng, V. C. M. Leung, and M. Song, "Computation offloading and content caching in wireless blockchain networks with mobile edge computing," *IEEE Trans. Veh. Technol.*, vol. 67, no. 11, pp. 11008–11021, Nov. 2018.
[20] Y. Xu, G. Wang, J. Yang, J. Ren, Y. Zhang, and C. Zhang, "Towards secure network computing services for lightweight clients using blockchain," *Wireless Commun. Mobile Comput.*, vol. 2018, pp. 1–12, Nov. 2018.
[21] J. Lin, Z. Shen, C. Miao, and S. Liu, "Using blockchain to build trusted LoRaWAN sharing server," *Int. J. Crowd Sci.*, vol. 1, no. 3, pp. 270–280, Sep. 2017.
[22] Y. Ren, Y. Liu, S. Ji, A. K. Sangaiah, and J. Wang, "Incentive mechanism of data storage based on blockchain for wireless sensor networks," *Mobile Inf. Syst.*, vol. 2018, pp. 1–10, Aug. 2018, doi: 10.1155/2018/6874158.
[23] J. Li, "Data transmission scheme considering node failure for blockchain," *Wireless Pers. Commun.*, vol. 103, no. 1, pp. 179–194, Nov. 2018.
[24] D. Lin and Y. Tang, "Blockchain consensus based user access strategies in D2D networks for data-intensive applications," *IEEE Access*, vol. 6, pp. 72683–72690, 2018.
[25] M. Singh and S. Kim, "Branch based blockchain technology in intelligent vehicle," *Comput. Netw.*, vol. 145, pp. 219–231, Nov. 2018.
[26] M. Dai, S. Zhang, H. Wang, and S. Jin, "A low storage room requirement framework for distributed ledger in blockchain," *IEEE Access*, vol. 6, pp. 22970–22975, 2018.
[27] R. V. Rosa and C. E. Rothenberg, "Blockchain-based decentralized applications for multiple administrative domain networking," *IEEE Commun. Standards Mag.*, vol. 2, no. 3, pp. 29–37, Sep. 2018.
[28] *Ethereum*. Accessed: Oct. 7, 2021. [Online]. Available: https://ethereum.stackexchange.com/questions/872/what-is-the-cost-to-store-1kb-10kb-100kb-worth-of-data-into-the-ethereum-block
[29] Y. T. Hsieh and C. Y. Ku, "Detection of gray hole attack in software defined networks," in *Proc. Int. Conf. Electron. Bus. (ICEB)*, 2018, pp. 231–239.



NADEEM JAVAID (Senior Member, IEEE) received the bachelor's degree in computer science from Gomal University, Dera Ismail Khan, Pakistan, in 1995, the master's degree in electronics from Quaid-i-Azam University, Islamabad, Pakistan, in 1999, and the Ph.D. degree from the University of Paris-Est, France, in 2010. He is currently an Associate Professor and the Founding Director of the Communications Over Sensors (ComSens) Research Laboratory, Department of Computer Science, COMSATS University Islamabad, Islamabad. He is also working as a Visiting Professor with the School of Computer Science, University of Technology Sydney, Sydney, Australia. He has supervised 137 master and 24 Ph.D. theses. He has authored over 900 articles in technical journals and international conferences. His research interests include energy optimization in smart grids and in wireless sensor networks using data analytics and blockchain. He was recipient of the Best University Teacher Award from the Higher Education Commission of Pakistan, in 2016, and the Research Productivity Award from the Pakistan Council for Science and Technology, in 2017. He is also Associate Editor of IEEE Access and an Editor of *Sustainable Cities and Society* journals.