Blockchain-based BATMAN protocol using Mobile ad-hoc Network (MANET) with an Ensemble Algorithm

Upendra Singh (upendrasingh49@gmail.com)
Shri Govindram Seksaria Institute of Technology and Science https://orcid.org/0000-0002-8215-1552
Sumit Kumar Sharma
Generali Insurance Group: Assicurazioni Generali SpA
Mukul Shukla
Shri Govindram Seksaria Institute of Technology and Science
Preeti Jha
IIT Indore: Indian Institute of Technology Indore

Research Article

Keywords: mobile ad-hoc network (MANET), Byzantine Fault Tolerance, Ensemble Algorithm, Blockchain

Posted Date: July 13th, 2021

DOI: https://doi.org/10.21203/rs.3.rs-673489/v1

Noname manuscript No.
(will be inserted by the editor)

Blockchain-based BATMAN protocol using Mobile ad-hoc Network (MANET) with an Ensemble Algorithm

Upendra Singh* · **Sumit Kumar Sharma** · **Mukul Shukla** · **Preeti Jha**

Received: date / Accepted: date

Abstract A MANET is a decentralized type of wireless network of mobile devices, it can also be defined as an autonomous system of nodes. All the nodes in the network are connected by wireless links and are mobile. They can come together and form a network without any support from any existing network infrastructure. MANET is a new field of study based on blockchain in a wireless ad-hoc environment. However, the main challenge for blockchain applications in ad-hoc networks is how to adapt to the extreme computational complexity of block validation while preserving the characteristics of blockchain and include nodes in the validation process. This article proposes a blockchain-based mobile network (MANET) with an ensemble algorithm. The proposed scheme provides a distributed environment for MANETS routing using a blockchain based on the Byzantine Fault Tolerance (BFT) protocol. Taking advantage of the better approach of mobile ad-hoc networking (BATMAN) to incorporate the concept of blockchain into the MANET as a representative protocol. The proposed method named Extended-BATMAN (E-BATMAN) incorporates the concept of blockchain into BATMAN protocol using MANET. As a secure, distributed and reliable platform, Blockchain solves most BFT security issues, with each node performing repeated security operations individually. The experimental analysis of the proposed ensemble algorithm is based on four parameters such as packet delivery rate, average end-to-end latency, network throughput, and energy. All of these parameters show better results with the proposed ensemble protocol than with existing state-of-the-art protocols.

Keywords mobile ad-hoc network (MANET) · Byzantine Fault Tolerance · Ensemble Algorithm · Blockchain

U. Singh
Shri Govindram Seksaria Institute of Technology and Science, Indore, MP, India
E-mail: Upendrasingh49@gmail.com

SK. Sharma
Iffco Tokyo General Insurance Ltd., MP, India

M. Shukla
Shri Govindram Seksaria Institute of Technology and Science, Indore, MP, India

P. Jha
Indian Institute of Technology Indore, MP, India

1 Introduction

Satoshi Nakamoto introduced a blockchain technology in 2008, as part of our peer-to-peer system called Bitcoin (Nakamoto, 2019). It has also proven itself as a technology that can improve data quality, flexibility, and reliability. However, the blockchain environment does not make the technology more robust and network-bound. Network sharing does not occur frequently on social networks such as the Internet backbone and 4G/5G networks, but network segregation can branch block and create two different chains. Since the two chains cannot coexist, one (shorter) is usually removed to maintain the integrity of the world. Losing data may or may not be the problem. Although a new type of blockchain with strong network delivery capabilities may be useful for infrastructure networks, MANET can have significant benefits (Cordova et al., 2020). As a result, many new MANET protocols have emerged, including optimized link state routing (OLSR) (Clausen et al., 2003), ad-hoc on-demand distance vector routing (AODV) (Perkins et al., 2003), Better Approach to Mobile Ad-hoc Networking (BATMAN) (Sanchez-Iborra et al., 2014), and dynamic source routing (DSR) (Varaprasad et al., 2013). The nodes themselves use this protocol to select the forward path from the source to the future and push packets in this way.

Researchers working on MANET-based routing protocols to maintain trust using blockchain are as follows:

Laube et al. (2019) showed that a DAG-based framework can be used to solve partitioning problem with MANET network mobility. They defined the partition problem as a set of problems that must be addressed when the network topology changes. In this work, we use the BATMAN routing protocol for a multi-hop ad hoc mobile network under development by the German" Freifunk" community. It is intended to replace the Optimized Link State Routing Protocol (OLSR) (Kulla et al., 2012a). Recently, blockchain trustless properties began to be researched to plan collaboration requirement components in numerous frameworks. (Machado and Westphall, 2021) presents a thorough and itemized survey of deals with blockchain-empowered information sending motivations for multi-hop MANETs. In this, Machado and Westphall (2021) contextualized selfish trouble making in explicit kinds of MANETs and why it influences information conveyance unwavering quality. We likewise summed up pre-blockchain motivating force components that animate helpful conduct and introduced an outline of blockchain highlights that could uphold impetus systems.

Many researchers have worked on developing a secure mechanism to communicate over the network. (Omar et al., 2012) have proposed an authentication mechanism, which ensures the links are secure before any sort of communication over the network. However, since MANETs are constantly changing (Eschenauer et al., 2002), the private key can be owned by a malicious entity, even if there are no unauthorized outsiders (Yang et al., 2019). The Byzantine Fault Tolerance Protocol (BFT) (Kotla and Dahlin, 2004) was chosen to implement blockchain operations, BFT is a property of a system that is capable of withstanding the class of faults derived from the Byzantine Generals problem. This means that the BFT system can continue to function even if some of the nodes fail or act maliciously (Aublin et al., 2013). Furthermore, we use denial contradictions with a fictitious node mechanism (DCFM) (Lwin et al., 2020a; Schweitzer et al., 2015) as a representative detection mechanism to identify malicious intruders and trusted nodes, as this is one of the efficient schemes that was recently introduced, Lwin et al. (2020a). The proposed system evaluation system that can fulfill the objectives of MANETs based on blockchain technology. We have identified the challenges and design of simplified blockchain-based trust management in mobile ad-hoc networks, proposed by Lwin et al. (2020a).

In our work, we structure our framework in four different stages: First stage calculates Trust value, the second stage delegates BFT using bully election to elect speaker, transaction claims/node validation and block generation using delegated BFT based on Extended-BATMAN protocol is applied on third stage, and finally maintenance is achieved on stage four.

This paper is broadly standardized as follows: In section 2, we have given some preliminary information. The proposed algorithms are explained in section 3. Results of the experiments performed various parameters are reported in section 4. Finally, conclusions are given in section 5.

2 PRELIMINARIES

This section describes the different ways in which the proposed algorithm can be supported.

2.1 Overview of BATMAN

Batman protocol decentralizes route knowledge; in other words, the routing tables are not available for the entire (Sliwa et al., 2019) network. To have the best gateway to communicate with the destination node, each node does an assignment of a single-hop neighbor in the mesh. As a result, an efficient and very fast routing scheme has been developed, which builds a collective intelligence network, and allows a low CPU and, consequently, a low battery consumption for each node (Johnson et al., 2008). This protocol works as follows: an OriGinator Message (OGM) is broadcasted regularly by each node, because of that link-local neighbors learn about the existence of the node. Each Link-local neighbor receiving the OGM message relies on it and rebroadcasts, this is done according to certain BATMAN forwarding rules. Due to the aforementioned broadcasting mechanism, the Batman mesh network gets flooded because of OGM messages (each node receives them at least once), or due to packet loss in communication links or their TTL value. The route quality is estimated by the number of OGM messages received from a particular node through each local link neighborhood. To find the best route to a particular end node, Batman finds the best path by counting OGM messages received from each node in the network and logs forwarded by the link-local neighbor. Batman uses this information and maintains a table that has an entry for a good link-local path for each node in the network. Every OGM has a serial number, Batman uses this number to distinguish between new OGM packets and their copies. Note that OGMs only functions as hello packets and contain no information about routing tables, connection status, etc., and they are not routing information exchange packets.

This protocol has received a great deal of attention in the research community. As a result, a significant amount of work is analyzed in assessing routing efficiency in different scenarios. For example, Kulla et al. (2012b) extensively studies performance under different environments and different node conditions (Kulla et al., 2011, 2010). However, the BATMAN protocol has little coverage for QoS/QoE

support for multimedia and VoIP communications, and even less for low energy consumption of nodes. These services have some strict requirements, including delay, sensitive constraints that make traffic management very complex. Therefore, to support these time-sensitive communications, the capabilities of common MANET routing protocols, especially the BATMAN protocol, need to be evaluated.

2.2 DCFM: Representative Attack Mitigation Scheme in BATMAN

DCFM is a representative project, we have incorporated it into the proposed project. Before discussing DCFM, we must discuss the NIAs (Node Isolation Attacks); these attacks are explicitly addressed by DCFM. Kannhavong et al. (2006) have first described NIAs, a type of DoS attack against OLSR. DoS stands for Denial of service; this attack aims to isolate the intended victim from the network. In such attacks, Batman routing nodes are used to find MPR nodes (multipoint relay selector set) with maximum range over their neighbors. Specifically, the victim gets blocked by the attacker node and doesn't receive the control packet. In the beginning, the attacker finds the place in the victim's transmission range and identifies its two-hop neighbors by exchanging hello messages with the victim. After that it yields a fake hello message, stating that the victim's two-hop neighbor is a one-hop neighbor. Due to this the victim gets the false information and selects the attacker as their only MPR. This MPR selection by routing protocol is based on selecting minimum MPRs. Therefore, the attacker node is the only node that transmits contact information from the victim and abandons the victim's messages. As a consequence, the other nodes, which are not receiving messages from the victim node, are removed from the network topologies.

2.3 Trust Managements using Blockchain in MANET

A blockchain consists of a list of records. All the records are stored in blocks, and each block is consisting of three things: a hash pointer to the previous block, timestamp, and listof transactions. Using the previous block's hash for link- ing, each block links itself to the previous block and the blockchain is formed. A blockchain design is resistant to modification of its data. Once data is recorded in any given block, it cannot be altered retroactively without the alter- ation of all subsequent blocks. A blockchain can be used as a distributed ledger that records transactions between two parties. Blockchain technology has created the backbone ofa new dimension of the internet as it allows digital informa- tion to be distributed but not copied. Originally, blockchain

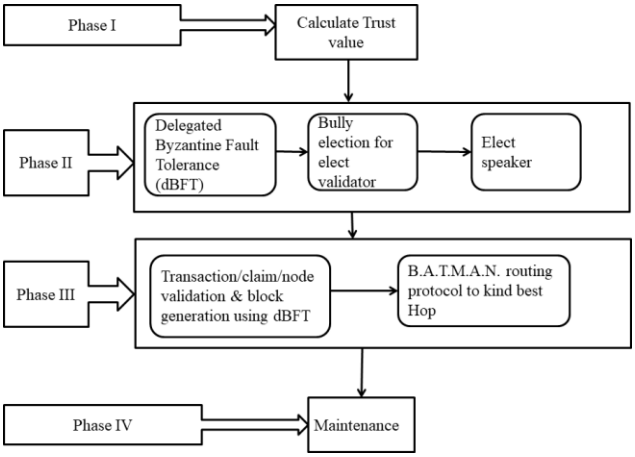


Fig. 1: Workflow of the proposed architecture.

was devised for digital currency i.e Bitcoin, but now the tech community has found other potential uses of technology.

A large number of blockchain-based applications are emerging in recent times, covering numerous fields includ- ing Real-time IoT operating systems, financial services, reputation systems, and so on. After the occurrence of any transaction, the information is broadcasted to all the peers in the network. A special group of participating nodes, which is called miners, attempts to lock transactions from the transaction pool that satisfies a cryptographic hash function. The mining process produces a block, which requires considerable computing power and is also prob- abilistic. Whilst block mining is hard, verifying a correct block is not (Dennis and Owen, 2015). Peiris et al. (2020) introduced a Blockchain-based distributed reputation model for ensuring trust in mobile ad-hoc networks. Recently, Liu et al. (2020) developed a B4SDC, a blockchain system for security-related data collection in MANETs.

3 PROPOSED WORK

In this paper, we have proposed block-chain-based dis- tributed trust installation system, and for this, we have adopted a blockchain-based architecture to manage trust performance and maintenance in MANET. In particular, we have implemented a public blockchain architecture to meet the extreme resource consumption and long validity time of existing block-chain technologies in dynamic and latency-sensitive environments. Fig 1 shows the components of the proposed system.

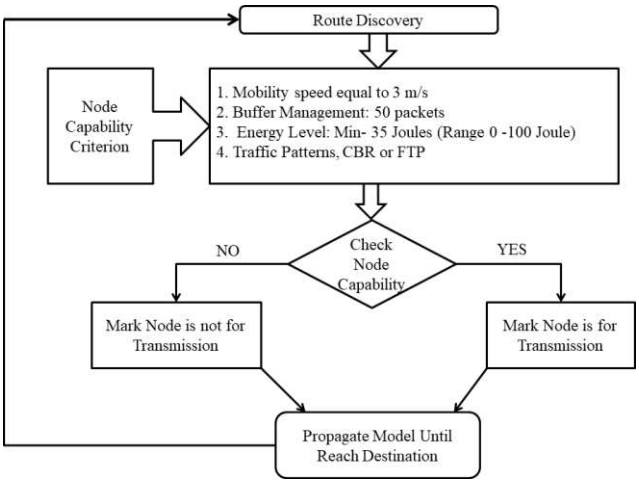


Fig. 2: Workflow of Trust Value computation.

3.1 Stage I: Trust Value Computation

In this work, a distributed trust system was developed, which improves the reliability and scalability of the network. Our focus is on how the trusted network is built and not on how trust value will be calculated. Also, for reducing the same attacker attacking again, information about the attacking node is spread throughout the network whenever a node searches for an enemy nearby. Our proposed scheme uses various discovery and trust models, but we adopted DCFM (Schweitzer et al., 2015) as a representative scheme and is applicable in the solution presented in this article. DCFM rules are appli- cable on all the nodes for detecting malicious neighbors.

As discussed earlier, the DCFM algorithm analyzes the network topology information from neighboring nodes to detect adversaries. If any discrepancy is found in the received information, such sending nodes are identified as a malicious node and a heavy fine are levied on the verification nodes. When such an adversary node is found in the network, using blockchain technology, that node's information is shared across the network so that it can be removed from the network. Figure 2 represents how to do Trust Value computation.

The Additive Increase/ Multiply Decrease (AIMD) (Marti et al., 2000) is a scheme that controls the Trust value (TV) of each node and specifies explicit and fair rewards and penalties for residential nodes, and which also includes adversaries of MANET. As the name implies, the node's TVs are added and multiplied, where the addition factor (α) and the multiplication factor (β) are used respectively in equation 1. In the DCFM detection strategy TV of the attacking node is multiplied by the β value -1, which is also the worst network penalty. Such nodes' information (negative TV values) is distributed throughout the network. By having this information each resident removes those nodes from the connection validation of this information (block transaction). A normal trust value lies between 0 to 1, a negative trust value simply means the node is forbidden from the network. However, some nodes are selfish and do not intentionally attack neighboring nodes, the value of β of such node is different, as explained in equation 2. In contrast, a working node will increase the TV by adding an α value. A high TV of MPR nodes makes the trust level determination fair, and hence it should be high. Therefore, α is adjusted according to equation 3. In the case of the MPR node, will be set to 0.7 and its TV will be for honest nodes that are not MPRs, and it will accumulate high TVs over time. Every node value is zero initially, however, every node can have the trust value, "1" as the highest. The initial TV value for every node in the network is zero. The numerator part of α value in Equation 3 i.e., $\sum_{k=1}^{n-1} m_i j^k m_i j^k + 1$ depicts number at which node i chooses node j as its MPR node to forward packets for k iterations (starting from when j begins to have a connection with i to when i calculates $j's$ TV.

$$TV = \begin{cases} TV * \beta & if\ a\ node\ misbehaves \\ TV * \alpha & otherwise \end{cases} \quad (1)$$

$$\beta = \begin{cases} -1 & if\ a\ node\ is\ an\ attacker\ according\ to\ DCFM \\ 0.7 & otherwise \end{cases} \quad (2)$$

$$\alpha = \begin{cases} \max\left[\frac{\sum_{k=1}^{n=1} m_i j^k}{m_i j^{k+1}}, 0.5\right] & if\ a\ node\ is\ an\ MPR\ node \\ 0.5, & otherwise \end{cases} \quad (3)$$

Algorithm 1: Trust Value Computation

Input: Mobility, Buffer management, Energy level
Output: destination

```
Initialize:
    Mobility = 0 < mobility < 500m/s
    Buffer management = 15 < Buffer < 100
    Energy level = 0 < energy < 100
    Traffic pattern = CBR or FTP
if (Node capability == yes) then
    Mark node for the transmission
else
    Leave the node
while (current position == destination) do
    repeat
        step 1 to 5
    until (current position == destination);
```

Additionally, a collaborative approach is introduced to our security solutions, intending to make the proposed

system more efficient. Although MANETs (Hernandez-Orallo et al., 2014) previously considered cooperative networking, earlier nodes perform individual detection processes for most of the security modes available for the proactive routing protocol. In DCFM, the search takes place at every interval of Hello. However, our solution reduces the investigation interval based on the number of neighbors near the node, synergistic effects of nearby neighboring nodes can help in extending this interval. Nodes that meet the following principles can be examined collaboratively rather than individually (Lwin et al., 2020a).

The Algorithm 1 discusses the steps for computation of trust value (TV). In line 1, initialization of route discovery and current position has been done. Node capability check is performed on Line 2. Then mark node for the transmission, else leave node, as shown in Line 5. Thereafter, the current position is matched, and if it is the destination, then repeat step 1 to 5.

3.2 Stage II: Ensemble Algorithm (Delegated ByzantineFault Tolerance (dBFT))

The proposed model begins by calculating the trust values of the nodes. After calculating trust values of nodes, model elects a validator node using election algorithm. Then delegated Byzantine Fault Tolerance (dBFT) selects a speaker node and remaining acts as a delegate. After that speaker verifies the claims and create hashes, and then sends a proposal to delegates. The delegates also verify and compare the results of a speaker with results of delegates. If results are matched, then the block is generated else request is discarded. After block generation, the model verifies all the transactions. Furthermore, the model uses a BATMAN routing protocol to find the next best hop and packet is sent to the specific hop. The model continuously maintains all properties of blockchain as well. Algorithm 2 show steps of dBFT.

Algorithm 2: Delegated Byzantine Fault Tolerance (dBFT)

Input: TV; Array of trust values
Output: validator
1 V = max(TV);
  V is the array of nodes eligible to become validator
2 Bully Election(V)
3 **Return** coordinator validator

Election of validator The highest TVs nodes are the eligible ones to become validators in the network. To determine the block creator node out of such nodes, the bully election (Hernandez-Orallo et al., 2014) strategy is adopted. It is also a commonly used election algorithm in distributed environments, as the name implies, the bully election algorithm makes a node with the highest identification number accepted as coordinator for other nodes. The node intending to be the leader communicates with another node with a higher priority. If a response is received from any of these nodes, it refuses to become the coordinator. Otherwise, it becomes the coordinator in the network. When a node with the highest priority directly claims the coordinator role, in such case communication overhead becomes lowest, and which is the best scenario. Therefore, this algorithm is well suited for a MANET environment. Likewise, the node with the highest TV can become the validator in a MANET blockchain. θ is the threshold value that determines whether a node is acceptably reliable to become a validator node. Unlike in a bully election, a node cannot self declare itself as a trustworthy node, which means that it required a neighboring node. If node i and j are neighbors and j have a TV above the threshold, it sends a claim message (i, j, TV-Claim, one-hop-count) $prKey_i$, where i is the follower node of j, j is the validator node claimed by i and trust value and one-hop neighbor count of j is put in TV-Claim and one-hop-count, respectively. $prKey_i$ is the private key of i, which signs the claim message. Similarly, using piggybacking on a TC message every node with neighbors with TVs above the threshold can broadcast a claim message to the entire network. j becomes the validator if these two conditions are met: TV of j node is the highest and no malicious claims on nodes i and j from other nodes in the network. In a claim message, to avoid a situation where two or more nodes have the same trust value and cannot be concluded which node should be selected, a one-hop count is added. In such a case the node with the highest number of one-hop count has more chances to become a validator. Here, the energy of node i is reduced in broadcasting a claim message for node j. As MANET is a resource-hungry environment, a reward should be given for the voting node i.

Delegation Process This step takes a list of nodes with their corresponding Trust value (TV) and then applies a choice algorithm to select certain nodes as validation nodes. In the delegation process, this node array selects one of the nodes as the speaker node and selects all the remaining functions as representatives. The speaker verifies and calculates the hash values of the pending claims and sends them to the representatives for questioning. Thus, if the results of the comparison of the speaker and the representative are greater than or equal to 66.6%, the hashes calculation is performed, then a new block pending claims or transactions is added or discarded. Algorithm 3 explains the delegation process.

Algorithm 3: Delegation Process

Input: Speaker *S*, CN; array of validators
Output: block
Initialize: CN = Validators
Select speaker *S* from CN, and consider all others as delegates *D*.
S is responsible for constructing new block from waiting claims.
Verify *S* and calculate hash
D validates the results of *S*
D share and compare the results of *S*
if *(skP ≥ 66.6%)* **then**
    Block added ;
else
    Discard Request ;

3.3 Stage III: Transaction Validation and Block Generation

In a network, block transactions update the trust value irrespective of the type of node, it can update the trust values of malicious nodes. Firstly, the validator node or delegate node generates the block whenever a transaction is received. The Better Approach to Mobile Ad-hoc Networking (BATMAN) protocol approach is used for this implementation in which transactions are propagated through MPR nodes. Each node *n*n will send an encrypted transaction (*n*, transaction) *prKeyn*, where transactions are encrypted by theprivate key of *n*n.

Extended-BATMAN (E-BATMAN) Extended-BATMAN (E-BATMAN) The BATMAN protocol can be described (simplified) as follows: Each node sends a broadcast message (called the original message or OGM) to notify its neighbors of its existence. According to certain rules, these neighbors will rebroadcast the OGM to notify them, such as the first initiator of this message. Therefore, messages from all the senders swamp the network. The size of OGM is small, a typical raw packet size is 52 bytes including IP and UDP overhead. The OGM contains the following fields at least: the sender's address, the address of the node sending the packet, the TTL, and the sequence number.

OGMs suffer from loss or delay of packets as they move through the mesh, this can be due to poor or saturated wireless links. Therefore, OGMs will flow quicker and more reliably on good routes. Suppose that an OGM has been received one or more times, which includes the sequence number specified by the sender of a particular OGM. Each relay node receives OGM at most once. Only those received from the neighbor were identified as the best next-hop currently (the neighbor with the highest ranking) to the original OGM initiator.

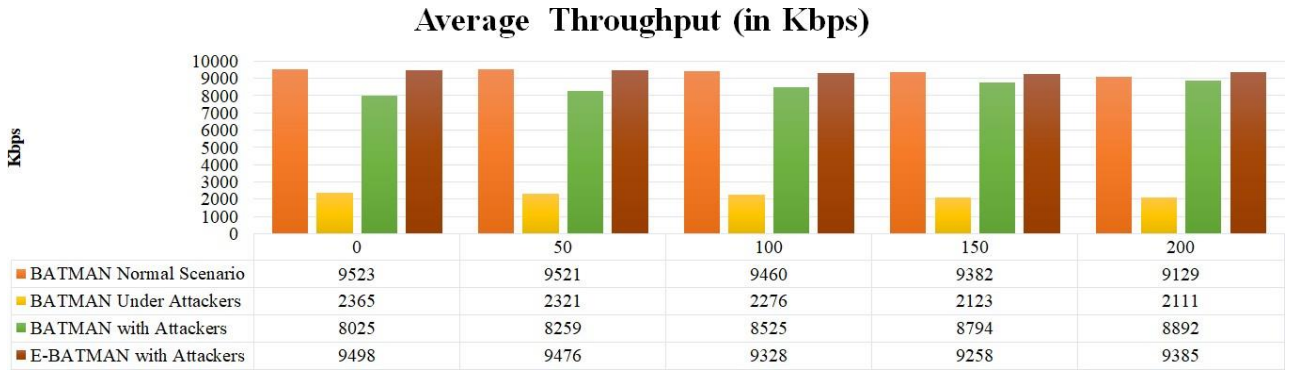
In this way, the OGMs are selectively inundated by the network, informing the receiving nodes of the existence of other nodes. By receiving its OGM, a *X* node learns that there is a *Y* node in the distance. When its one-hop neighbors resend OGM from node *Y*. If the node *X* has more than one neighbor, the sender receives messages faster and more reliably via one of its single hop neighbors. The neighbor needs to send data to the remote node. The protocol then selects this neighbor as the current best next hop to the message sender and configures its routing table. Algorithm 4 shows the steps of BATMAN protocol.

Algorithm 4: E-BATMAN Algorithm

Input: OGMs
Output: best hop node
Each node Broadcast OGMs to its neighbours
Neighbours rebroadcast OGM'S to prove their existence
OGM's are oringator of messages of size 52 byte.
Including IP and UDP overhead
if *(node neighbour > 1)* **then**
    Best hop node= current node
else
    repeat
        step 1 to 4
    until *(Best hop node= current node)*;

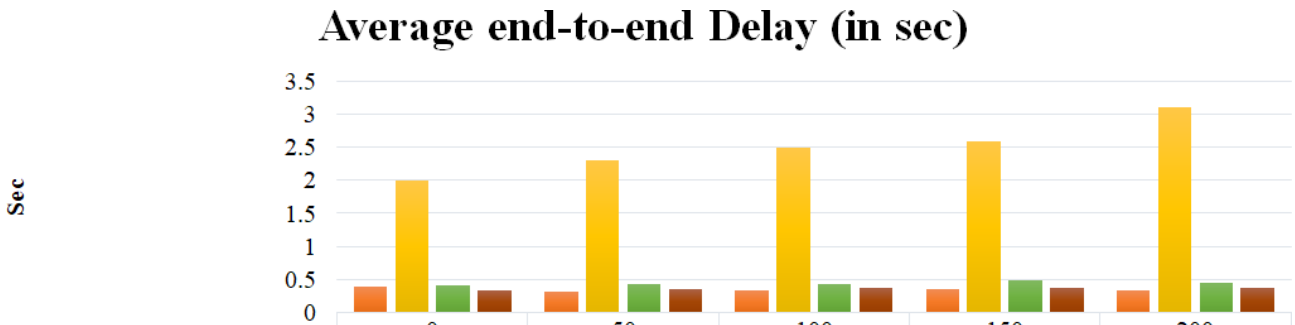
Block Configuration The structure of the block must contain information that is included in the block and how the representative node configures it. In a blockchain system, transactions are stored as a block and the network is chained by a block. A hash value (SHA-256 algorithm) is added to the block, which is obtained directly from the transaction data. This provides instability in the blockchain. So, a little variation in the transaction data alters the hash value. For chaining of the blocks, the previous block's hash is added as data in the current block for chaining. It means that any data change in a block of chain disrupts all the blocks in the blockchain. The only format accepted by Block is as follows: a hash signature starting with 10 consecutive zeros. By this rule, there is a data element called nonce. The value of this data element is changed repeatedly, and this is for obtaining a legitimate hash value.

A trusted MANET blockchain has blocks containing transaction data and metadata described above (timestamp, transaction hash, delegate id, and nonce). When a transaction hash, transaction generator ID and TVs issued by the transaction generator, as well as the representative id, are added to provide a denial of block transactions by the nodes. When the network is formed, the first block of the blockchain is called" Genesis Block", which is defined with a blank list of transactions.



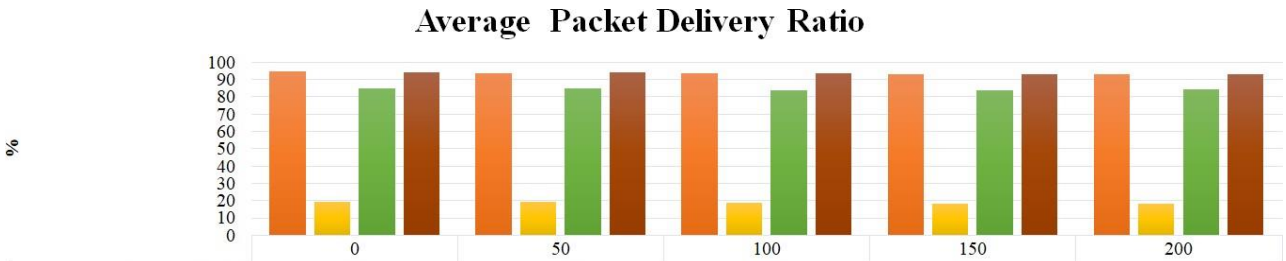
	0	50	100	150	200
BATMAN Normal Scenario	9523	9521	9460	9382	9129
BATMAN Under Attackers	2365	2321	2276	2123	2111
BATMAN with Attackers	8025	8259	8525	8794	8892
E-BATMAN with Attackers	9498	9476	9328	9258	9385

Fig. 3: Results of Average Thoughput using proposed E-BATMAN and existing BATMAN protocol (Lwin et al., 2020b).



	0	50	100	150	200
BATMAN Normal Scenario	0.39	0.31	0.34	0.35	0.34
BATMAN Under Attackers	2	2.3	2.5	2.6	3.11
BATMAN with Attackers	0.42	0.44	0.43	0.48	0.46
E-BATMAN with Attackers	0.34	0.36	0.38	0.37	0.38

Fig. 4: Results of Average e2e-delay using proposed E-BATMAN and existing BATMAN protocol (Lwin et al., 2020b).



	0	50	100	150	200
BATMAN Normal Scenario	94.82	93.84	93.85	93.18	92.89
BATMAN Under Attackers	19.35	19.21	18.96	18.39	18.28
BATMAN with Attackers	84.9	85.1	83.6	83.7	84.2
E-BATMAN with Attackers	94.29	94.01	93.56	93.21	93.09

Fig. 5: Results of PDA using proposed E-BATMAN and existing BATMAN protocol (Lwin et al., 2020b).

3.4 Block Maintenance

There are two types of nodes in a blockchain environment, which are as follows: full nodes and light nodes, former handles blockchain and the latter does not handle the entire blockchain and relies heavily on the information of entire nodes. Due to such nature of MANETs, our environment has also adopted this concept. When a new node tries to connect to the network, it obtains access to the blockchain data. Initially, a node is included in the network as a lightweight node, which means that it can only download the blocker's header. Although soon after entering the network, a new node can act as a light node, it is also capable of generating transactions (attacker detection/TV calculation) on the network. Initially when there is no full node available in the network, until then the network's host node acts as a temporary full node for the relay block headers.

4 EXPERIMENTAL RESULTS

The experimental results obtained from the proposed approach are presented in this section. Evaluation parameters are used; these considerations are Packet Delivery Ratio, Average End-to-End Latency, Network Throughput, and Energy. This result section includes 105 mobile nodes in-network.

4.1 NS3 Simulation Parameters

This comparison of the proposed algorithm and existing (Lwin et al., 2020b) is shown in subsection. Table 1 represents simulation parameters.

4.2 Performance Evaluation

4.2.1 Average Throughput (AT)

The data retrieval at the destination node in any unit of the time interval is termed as throughput (Taha et al., 2017).

$$AT = (number\ of\ bytes\ received * 8/ simulation\ time) * 1000kbps \quad (4)$$

4.2.2 Average end-to-end Delay (e2e delay)

The time utilized by a packet to reach source to destination is called the end-to-end delay (Taha et al., 2017).

$$e2e\ delay = \frac{\sum_{i=1}^{n} (Ri - Si)}{n} \quad (5)$$

4.2.3 Packet Delivery Ratio (PDR)

The data packets' ratio sent to the data packets received is termed as the PDR (Taha et al., 2017). Mathematically, it can be defined as follows:

$$PDR = \frac{number\ of\ packets\ recieved}{number\ of\ packets\ sent} * 100 \quad (6)$$

4.3 Results and Discussion

This section compares the performance of the proposed Extended-BATMAN algorithm with Existing BATMAN protocol. Figure 3 shows the average throughput of the current work and the proposed E-BATMAN protocol. Here, BATMAN with Attackers, BATMAN Under Attackers, and another BATMAN Normal Scenario are compared with the proposed E-BATMAN method. In existing Lwin et al. (2020b), BATMAN with Attackers protocol was getting minimum Throughput (Kbps) is 8025, and Maximum Throughput (Kbps) is 8892. Another protocol BATMAN Under Attackers was getting minimum Throughput (Kbps) is 2111, and Maximum Throughput (Kbps) is 2365. Additionally, we compared another existing proposed protocol BATMAN Normal Scenario. BATMAN Normal Scenario, getting minimum Throughput (Kbps) is 9129, and Maximum Throughput (Kbps) is 9523. Figure 3 shows the throughput of our proposed work, and we designed protocols E-BATMAN with Attackers. In the proposed Blockchain-based Extended BATMAN (E-BATMAN) with Attackers, the protocol was getting minimum Throughput (Kbps) is 9258, and Maximum Throughput (Kbps) is 9498. Figure 4 shows the results obtained from the proposed E-BATMAN's performance with the existing BATMAN protocol, BATMAN with Attackers, BATMAN Under Attackers, and another BATMAN Normal Scenario. The BATMAN with Attackers protocol provides a minimum delay of 0.42 sec and a maximum of 0.48 sec. Additionally, BATMAN Under Attackers produces a minimum delay of 2 sec. Maximum delay of 3.11 sec. We compared another existing proposed protocol BATMAN Normal Scenario. BATMAN Normal Scenario, getting minimum delay is 0.31 sec, and Maximum delay is 0.39 sec.

Figure 4 shows the throughput of our proposed work; we designed protocols E-BATMAN with Attackers. The proposed Blockchain-based Extended BATMAN (E-BATMAN) with Attackers produces a minimum delay of 0.34 sec and a maximum of 0.38 sec.

Figure 5 shows PDR (%) of proposed E-BATMAN and existing works like BATMAN with Attackers, BATMAN Under Attackers, and another BATMAN Normal Scenario. In existing (Lwin et al., 2020b) BATMAN with Attackers protocol was getting minimum packet delivery ratio (%)

Table 1: Simulation Parameters

Parameters	Specification	Parameters	Specification
Network Simulator	NS-3, Version 3.33	PHY /MAC Protocol	IEEE 802.11
Network Size	1 km x 1 km	Propagation Model	Two-ray ground
Connection Protocol	UDP/TCP	Mobility Model	Random Direction 2 d Mobility Model
Data Type	Constant Bit Rate (CBR)/FTP	Channel Type	WifiPhyStandard: For example, 802.11b, 802.11n, etc.
Source/Destination	Random	Antenna Model	test-parabolic-antenna
Data Packet Size	256 bytes	Simulation time (Second)	200
Simulation Protocol	BATMAN, E-BATMAN	Language	C++ and python
Simulation Scenario	105	No of Malicious Nodes	5% out of the scenario

is 84.2, and Maximum packet delivery ratio (%) is 85.1. Another protocol BATMAN Under Attackers was getting a minimum packet delivery ratio (%) is 18.28, and the Maximum packet delivery ratio (%) is 19.35. WE compared another existing proposed protocol BATMAN Normal Scenario. BATMAN Normal Scenario, getting minimum packet delivery ratio (%) is 92.89, and Maximum packet delivery ratio (%) is 94.82. Figure 3 shows the PDR (%) of our proposed work, and we designed protocols E-BATMAN with Attackers. In the proposed Blockchain-based Extended BATMAN (E-BATMAN) with Attackers, the protocol was getting minimum packet delivery ratio (%) is 93.09, and Maximum packet delivery ratio (%) is 94.29.

Figure 6 shows the comparative analysis for examined parameters like packet delivery ratio (%), Throughput (Kbps), and Delay(s). the proposed result compared with existing work (Lwin et al., 2020b), and we get improvement result in term of all parameters explained above.

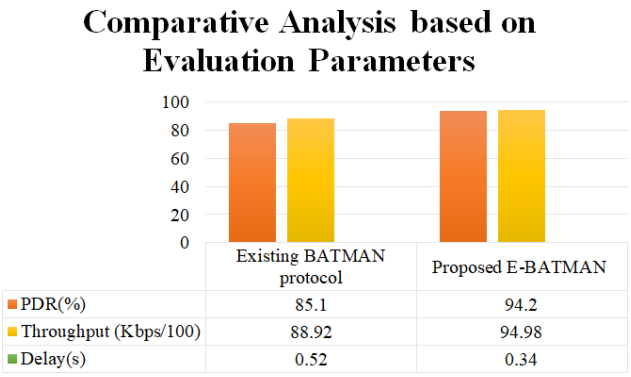


Fig. 6: Comparative analysis based on Evaluation parame- ters for proposed E-BATMAN and existing BATMAN pro-tocol (Lwin et al., 2020b).

5 CONCLUSION

In this paper, we proposed a novel approach which generates distributed trust value in MANETs. We implemented blockchain concept in BATMAN protocol termed Extended BATMAN (E-BATMAN). Simulation results demonstrated that distributed trust value provides strong network security. The overall complexity is reduced because there is no information lost using the proposed E-BATMAN protocol even though the attacker changes their location and attacks different network nodes. The network is safe. Besides this, each node's responsibility is reduced. Additionally, our blockchain-based E-BATMAN protocol using MANETs is reliable, scalable, and available. In future, we wish to test our proposed scheme's feasibility with various routing protocols in MANETs.

Compliance with Ethical Standards

Ethical approval: This manuscript does not contain any studies with human participants or animals performed by any of the authors.

Funding: No funding was received from any organization for conducting the study of the submitted work and preparation of this manuscript.

Conflict of interest: The authors of this manuscript declare that they have no conflict of interest.

Informed Consent: The research papers which are used for the study of the submitted work has been cited in the manuscript and the details of the same has been included in the reference section.

References

Aublin PL, Mokhtar SB, Que´ma V (2013) Rbft: Redun- dant byzantine fault tolerance. In: 2013 IEEE 33rd International Conference on Distributed Computing Systems, IEEE, pp 297–306

Clausen T, Jacquet P, Adjih C, Laouiti A, Minet P, Muhlethaler P, Qayyum A, Viennot L (2003) Optimized link state routing protocol (olsr)

Cordova D, Laube A, Pujolle G, et al. (2020) Blockgraph: A blockchain for mobile ad hoc networks. In: 2020 4th Cyber Security in Networking Conference (CSNet), IEEE, pp 1–8

Dennis R, Owen G (2015) Rep on the block: A next generation reputation system based on the blockchain. In: 2015 10th International Conference for Internet Technology and Secured Transactions (ICITST), IEEE, pp 131–138

Eschenauer L, Gligor VD, Baras J (2002) On trust establishment in mobile ad-hoc networks. In: International workshop on security protocols, Springer, pp 47–66

Hernandez-Orallo E, Olmos MDS, Cano JC, Calafate CT, Manzoni P (2014) Cocowa: A collaborative contact-based

watchdog for detecting selfish nodes. IEEE transactions on mobile computing 14(6):1162–1175

Johnson D, Ntlatlapa NS, Aichele C (2008) Simple pragmatic approach to mesh routing using batman

Kannhavong B, Nakayama H, Kato N, Nemoto Y, Jamalipour A (2006) Analysis of the node isolation attack against olsr-based mobile ad hoc networks. In: 2006 International Symposium on Computer Networks, IEEE, pp 30–35

Kotla R, Dahlin M (2004) High throughput byzantine fault tolerance. In: International Conference on Dependable Systems and Networks, 2004, IEEE, pp 575–584

Kulla E, Ikeda M, Barolli L, Miho R (2010) Impact of source and destination movement on manet performance considering batman and aodv protocols. In: 2010 International Conference on Broadband, Wireless Computing, Communication and Applications, IEEE, pp 94–101

Kulla E, Ikeda M, Hiyama M, Barolli L, Miho R (2011) Performance evaluation of olsr and batman protocols for vertical topology using indoor stairs testbed. In: 2011 International Conference on Broadband and Wireless Computing, Communication and Applications, IEEE, pp 159–166

Kulla E, Hiyama M, Ikeda M, Barolli L (2012a) Performance comparison of olsr and batman routing protocols by a manet testbed in stairs environment. Computers & Mathematics with Applications 63(2):339–349

Kulla E, Ikeda M, Oda T, Barolli L, Xhafa F, Takizawa M (2012b) Multimedia transmissions over a manet testbed: problems and issues. In: 2012 Sixth International Conference on Complex, Intelligent, and Software Intensive Systems, IEEE, pp 141–147

Laube A, Martin S, Al Agha K (2019) A solution to the split & merge problem for blockchain-based applications in ad hoc networks. In: 2019 8th International Conference on Performance Evaluation and Modeling in Wired and Wireless Networks (PEMWN), IEEE, pp 1–6

Liu G, Dong H, Yan Z, Zhou X, Shimizu S (2020) B4sdc: A blockchain system for security data collection in manets. IEEE Transactions on Big Data

Lwin MT, Yim J, Ko YB (2020a) Blockchain-based lightweight trust management in mobile ad-hoc networks. Sensors 20(3):698

Lwin MT, Yim J, Ko YB (2020b) Blockchain-based lightweight trust management in mobile ad-hoc networks. Sensors 20(3):698

Machado C, Westphall CM (2021) Blockchain incentivized data forwarding in manets: Strategies and challenges. Ad Hoc Networks 110:102321

Marti S, Giuli TJ, Lai K, Baker M (2000) Mitigating routing misbehavior in mobile ad hoc networks. In: Proceedings of the 6th annual international conference on Mobile computing and networking, pp 255–265

Nakamoto S (2019) Bitcoin: A peer-to-peer electronic cash system. Tech. rep., Manubot

Omar M, Challal Y, Bouabdallah A (2012) Certification-based trust models in mobile ad hoc networks: A survey and taxonomy. Journal of Network and Computer Applications 35(1):268–286

Peiris P, Rajapakse C, Jayawardena B (2020) Blockchain-based distributed reputation model for ensuring trust in mobile adhoc networks. In: 2020 International Research Conference on Smart Computing and Systems Engineering (SCSE), IEEE, pp 51–56

Perkins C, Royer EM, Das S (2003) Ad-hoc on-demand distance vector routing (aodv). Tech. rep., Internet-Draft, November 1997. draft-ietf-manet-aodv-00. txt

Sanchez-Iborra R, Cano MD, Garcia-Haro J (2014) Performance evaluation of batman routing protocol for voip services: a qoe perspective. IEEE Transactions on Wireless Communications 13(9):4947–4958

Schweitzer N, Stulman A, Shabtai A, Margalit RD (2015) Mitigating denial of service attacks in olsr protocol using fictitious nodes. IEEE Transactions on Mobile Computing 15(1):163–172

Sliwa B, Falten S, Wietfeld C (2019) Performance evaluation and optimization of batman v routing for aerial and ground-based mobile ad-hoc networks. In: 2019 IEEE 89th Vehicular Technology Conference (VTC2019-Spring), IEEE, pp 1–7

Taha A, Alsaqour R, Uddin M, Abdelhaq M, Saba T (2017) Energy efficient multipath routing protocol for mobile ad-hoc network using the fitness function. IEEE access 5:10369–10381

Varaprasad G, Narayanagowda SH, et al. (2013) Implementing a new power aware routing algorithm based on existing dynamic source routing protocol for mobile ad hoc networks. IET networks 3(2):137–142

Yang J, He S, Xu Y, Chen L, Ren J (2019) A trusted routing scheme using blockchain and reinforcement learning for wireless sensor networks. Sensors 19(4):970