Node Criticality Assessment in a Blockchain Network

Aditya Shyam Bazari
Department of Computer Engineering, Delhi Technological University
New Delhi, India
adityabazari_bt2k16@dtu.ac.in

Akash Aggarwal
Department of Mathematical Sciences, Indian Institute of Technology (BHU)
Varanasi, India
akash.aggarwal.mat15@itbhu.ac.in

Waqar Asif
School of Engineering and Mathematical Sciences, City, University of London
London, UK
waqar.asif@city.ac.uk

Marios Lestas
Department of Electrical Engineering, Frederick University
Cyprus
eng.lm@frederick.ac.cy

Muttukrishnan Rajarajan
School of Engineering and Mathematical Sciences, City, University of London
London, UK
r.muttukrishnan@city.ac.uk

ABSTRACT

Blockchain systems are being rapidly integrated in various technologies, with limited work on the effect of the underlying network topology on the blockchain performance. In this work, we investigate the significance of each network node on the overall blockchain performance. This is assessed by selecting critical nodes according to different criticality metrics, and investigating, using simulations, the degradation in performance incurred upon removing these nodes. The most critical nodes are the ones that incur the greatest degradation in performance. The considered performance metrics are the blockchain size and the packet drop rate. Criticality metrics such as Betweennes Centrality, Closeness Centrality and Degree Centrality are compared. It is found that the Sign Change Spectral Partitioning approach, enhanced with Blockchain Specific traffic flow information, is able to identify critical nodes better in the sense that higher degradation in performance is reported upon their removal.

KEYWORDS

blockchain, node criticality, blockchain security, blockchain attacks, network analysis

ACM Reference Format:
Aditya Shyam Bazari, Akash Aggarwal, Waqar Asif, Marios Lestas, and Muttukrishnan Rajarajan. 2020. Node Criticality Assessment in a Blockchain Network. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

With the invent of a decentralized digital currency, bitcoin, technology and network communications today stand at a crossroads[22]. A possible substitute for the traditional communication methodologies is the technology which underpins the bitcoin, blockchain[10]. With the potential to get as big as the Web, blockchain has emerged as one of the key technologies that can be used to form a distributed solution. This has thus motivated researchers to opt for a blockchain based architecture in fields such as the Internet of Things (IoTs) [8][12], smart grid infrastructure [18][23] and the health-care systems [19][28].

Blockchain is a cryptographically verified, distributed ledger that maintains a set of transaction records for the users in a network. These transactions reflect the exchange of information among users and are relayed across the network with the help of data flooding. A transaction, once generated, is flooded across the network which, upon reaching a few highly capable nodes, is verified for authenticity. These highly capable nodes are referred to as miners in the network. Miners maintain the data exchange protocol by flooding the verified transaction blocks onto the network so that every other node can update their ledger. Each network has multiple miners and the one that completes the verification process at the earliest, triggers the ledger update procedure. All this mechanism that defines the blockchain architecture, is built on the underlying assumption that the network is completely connected. This means that each transaction is received by each miner in the same time slot and each data flood initiated by a miner reaches all possible nodes in the network. In reality, blockchain can have topological limitations. For instance, a botnet attack[17] or a DDoS attack[21], on a network has the potential to render the targeted node useless for network communication. Moreover, if this attack succeeds in targeting the most critical nodes in the network, thus partitioning or breaking a chunk of it, then any transaction generated on one side to the other might be verified by the miner due to legitimate user credentials, but would not be able to reach the other part, hence making the solution unviable.

In the past, a lot of research has been done for the identification of these critical nodes that have a higher influence on the network compared to the others. These nodes gain their importance due to the topological structure of the network. A few of the well known approaches are Betweenness Centrality [25] which evaluates the criticality of a node based on the number of shortest path routes

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

a node participates in, Closeness Centrality [24] which takes into account how close a node is to all other nodes in the network and Degree Centrality [15] which uses node degree to highlight the criticality of a node. All these approaches are known to work well in their considered scenarios, but in a blockchain based setup, where the accessibility of a node is of the highest priority, they tend to misjudge the criticality of a node. In this work, we use a sign change based spectral partitioning approach, enhanced with Blockchain specific traffic flow information, to highlight the criticality of a node. The proposed approach deems a node as critical if it lies in a cutset and experiences the highest traffic flow. A cutset is defined as a set of nodes, that observe a change in sign for the Fiedler vector among their neighbours [2]. The proposed approach is evaluated for change in blockchain size and packet drop ratios and it is observed that the proposed approach outperforms existing approaches by showing a greater reduction in blockchain size and a larger packet drop rate upon removal of the most critical nodes from the network.

The remainder of the paper is organized as follows: Section II explains the background, Section III presents the proposed approach and Section IV reports the results. Conclusions are presented in Section V.

2 BACKGROUND

Blockchain has gained immense attention in the recent times with key focus towards intrinsic operational traits of blockchain such as information propagation[6][7], key management[5][16], ledger architectures[1][14][3], blockchain contracts[27][13] and consensus protocols[4][20]. A blockchain network has numerous features that constitute it's overall functioning. Some of the most cardinal of them are briefly explained below:

Transaction. A transaction is a message generated by a sender for a particular receiver in the network. It contains the public key of the sender, public key of the receiver, transmitted data and a signature. This signature is generated by the sender with the sender's private key for improved security.

Block. A blockchain is a chain of data, arranged in blocks. A block has a header which contains several components such as timestamp, hash of the previous block, root hash of merkel tree and nonce value. The body of a block contains all the transactions. These transactions are verified by a miner and then appended onto the chain. Each block is identified with a hashcode, generated using the content inside a block. This includes the actual transactions between nodes, the time stamp, the nonce and the hash of the previous block, thus making it impervious to random double spending attacks as a slight change in even a single parameter would render the block invalid.

Merkel Tree. A merkel tree is a hash based data structure, where each leaf node contains the hash of a transactional block and each non-leaf node contains the hash of its children nodes. A merkel tree summarises all the verified transactions by repeatedly hashing the data and producing a final digital fingerprint.

Timestamp. Every block of transaction has an associated timestamp [26]. This enhances the security of the network and ensures that a newly created block is in line with all existing blocks, where a new block cannot have a timestamp of a time earlier than the ones already added.

Mining. In order to add a transaction into a blockchain, a few nodes that have the required computational capabilities perform a cryptographic operation, referred to as the Proof of Work, onto the incoming transactions. This helps in validating the authenticity of a transaction. A miner receives a small reward for this work.

Proof of Work (PoW). Proof of Work is an approach in which miners are up against each other to add a block and receive the compensation reward. The goal of a PoW approach is to solve the mathematical hash, which gets more complex as the blockchain increases in size[11].

In a blockchain based setup, a sender generates a transaction, signs it with its private key for identification purposes and broadcasts it across the network. Each miner waits for a predefined time to get all the transactions that were generated in that time slot and upon receiving them, combines them to form a block. Each block is verified with its own hash function which also constitutes of the hash of the previous block. A block is then mined by the miner and upon completion, signed using the miner's private key. A signed block is then broadcasted into the network where each node receiving this block adds it to its existing chain.

There are mainly two situations in which a transaction can get rejected by the blockchain model, despite of it being genuine :

- When a transaction is triggered, it is assigned a timestamp. During the validation of that transaction, the process checks if the timestamp aligns with the requirement criteria of the network, as explained above[26]. If the received timestamp is not greater than the median of previous 11 timestamps, the transaction is rejected as the transaction is too old to be validated in the current time.
- The public key of all the nodes of the network are floated in the blockchain network initially. When a node is removed from a network possibly due to malicious attacks, it's public key still remains. During the validation process, the network checks if either the sender or the receiver is null, that is, any of the two is a dead node. If so, the transaction is refused and not validated.

3 PROPOSED APPROACH

In a blockchain environment, the underlying network topology plays a vital part in defining node characteristics. In an ideal scenario, a successful miner would be the one that is capable of receiving all possible transactions in a particular time window with the smallest delay factor. This means that if two miners are on the same network, the one receiving all transactions at the earliest would have an edge over the other. The second miner might not receive all transactions due to network losses or might receive one after the defined time window expires. A time window is referred to as a slot of time in which a miner expects to receive all transactions. Once received, they are mined to form a block. These network losses could be caused due to a compromised/faulty node in the network whereas, the delay in packet reception could be caused due to a bottleneck in the network. In order to eliminate these issues, it is vital to identify these critical nodes in time, which can have drastic affects on the performance of a blockchain network.

Considering the architecture of a blockchain setup, this work exploits the spectral partitioning approach proposed in our previous work [2] and compliments it with new features pertinent to a blockchain setup. It helps in identifying critical nodes in the network with the help of a sign change approach. The proposed approach works by calculating the Fiedler vectors for all nodes in the network and then forming a cutset based on the change in sign that is observed [9]. Fiedler vector values hold both positive and negative signs, these signs help identify two partions of the network. The proposed approach, identifies these nodes which have neighboring nodes with different Fiedler vector sign and labels them as part of a cutset. These nodes represent the bottleneck of the network and removal of all such nodes would lead the network partitioned [2]. Nodes belonging to this cutset have precedence over each other based on the traffic flow patterns that they are observing. In a blockchain setup, network traffic flows to and from a miner so a node belonging to the cutset, that also experiences the maximum traffic flow, is deemed as the most critical node in the network. We formally define the spectral partitioning approach as follow:

Let a simple un-directed graph $G = (V, E)$ consist of a vertex-set V and an edge-set E where |V| = n and |E| = m. The adjacency matrix A is a $n \times n$ matrix in which each row and column corresponds to a vertex of G. Any element, a_{ij}, of this matrix represents the number of edges between vertex i and vertex j. For a graph G, the matrix A would be symmetric about the main diagonal and a_{ij} would have value of either 1 or 0. The diagonal matrix $D = diag(d_1, \ldots, d_n)$ is the degree matrix, elements of which are the degree of all the vertices of the G. For any element d_{ii} of matrix D, i represents the vertex in the graph and the absolute value of d_{ii} is the degree of that particular vertex. Any other element of matrix D, d_{ij} would have a value of 0, where $i \neq j$.

For aforementioned graph G the Laplacian matrix L, is define as:

$$L = D - A \tag{1}$$

The diagonal elements l_{ij} of L are therefore equal to the degree of vertex v_i and off-diagonal elements l_{ij} are -1 if vertex v_i is adjacent to v_j and 0 otherwise.

Eigenvalues and eigenvectors provide an insight into the connectivity of the graph. Let for any matrix A if there is vector $X \in \mathbb{R}^n \neq 0$ such that

$$AX = \lambda X$$

for some scalar λ, then λ is called the eigenvalue of A with corresponding eigenvector X. So the eigenvalues of the Laplacian matrix L are arranged in ascending order such that $0 = \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$. The multiplicity of zero in the eigen values represent the number of disconnected components of a network [9]. The second smallest eigen value μ(G) = λ_2 represents the algebraic connectivity of the network. Smaller the algebraic connectivity the closer the network is in becoming disconnected. After getting a second smallest eigen value λ_2 the corresponding eigen vector $\vec{v} = (v_1, \ldots, v_n)$ is the Fiedler vector of matrix L. The Fiedler vector has both positive and negative entities. Elements with different signs represent connected subgraphs which are poorly connected with each other. So the Fieldler values help in identifying those sections of the graph, removal of which, can potentially split the network into two, also known as the cutset. In a network, the nodes of this cutset S are hence crucial in the overall functioning. For any vertex v, if the sign of the Fiedler value of at least one of its neighbours is different than its own sign then it is included in the cutset S. Nodes inside the cutset S are evaluated for criticality based on the traffic flow $T(x)$ that they observe $\forall x \in S$. The optimization problem originating from this can be defined as:

$$P : CN = \arg\max_{x \in S} T(x) \tag{2}$$
$$\text{Such that } S = \arg\min_{\alpha \in V} \mu(G(V - \alpha))$$

Fig. 1 illustrates the working principles of the proposed approach. Each node calculates its corresponding Fiedler vector value and shares it with the neighbouring nodes. These nodes evaluate these values and identify themselves as critical nodes if they see a change in sign among their neighboring nodes. These values are represented as node labels in Fig. 1. Nodes in the cut set then observe traffic flow through them for the identification of the most critical nodes in the network. Removal of these nodes would have severe affects on the network. They would result in both, packet drops due to node removal and packet drop due to timestamp expiration. Lets consider the removal of node B in the considered scenario. In the former case, if node A generates a transaction for node B it reaches the miner X who evaluates the validity of the transaction and drops it due to non existing node B. In the later case, the removal of node B would redirect all traffic through node C which would create a bottleneck and thus result in timestamp expiration. Transactions will reach miners across the cutset with a delay thus reporting a time earlier than the acceptable time window. This results in packet drops due to time stamp expiration. Furthermore, removal of all nodes from the cutset would render the network partitioned into clusters thus increasing packet drops. This will have adverse affects on the size the blockchain, where a miner would not be able to see any transactions being made on the other side of the network.

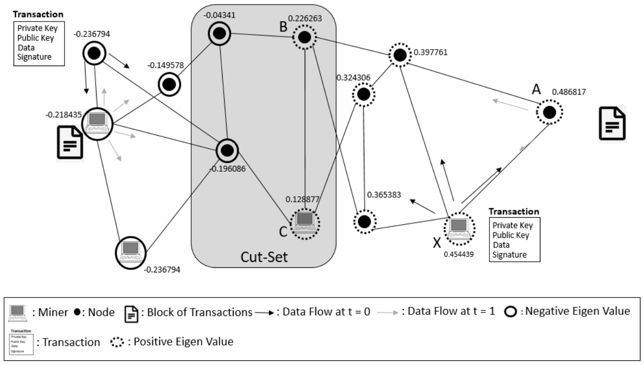

Figure 1: Representation of the Blockchain Model

4 SIMULATION AND RESULTS

In this section, we evaluate the performance of the proposed approach against existing approaches namely, betweenness centrality, closeness centrality, degree centrality and random node removal in a blockchain based environment. We report the change in blockchain size, increase in packet drop due to node removal and increase in packet drop due to hop limit. In this simulation setup, due to simulation environment limitations, a hop count was used instead of timestamp, where the upper limit for acceptable time window is represented as hop limit. The proposed approach was also evaluated for variable network density, variable hop limit and varying number of miner.

We consider a network of a 1000 nodes interconnected using the uniform random distribution generating a 1000 transactions between randomly selected senders and receivers. An evaluation is made after each iteration where an iteration covers a random set of transaction. Each iteration is upper bounded by 100 transactions. The results reported in this section are an average of 50 random network topologies.

In the first set of simulations, we evaluate blockchain outcomes for varying hop window with fixed edge probability p. We assign $p = 10\%$ and evaluate for hop window h of 2, 3, 4, 5, 10, 15 & 20. We assume having a single miner in this simulation for better illustration of the effect of hop window. We remove this assumption in later simulation setups for illustrating the scalability of the approach. We remove 10% nodes belonging to the cutset upon each iteration and evaluate results. This is repeated until the entire cutset has been removed.

Fig. 5 reports that as hop window is increased from 2 to 20 the packet drop due to hop count decreases as shown in Fig. 4. This is merely due to increase in acceptable hop count window, where a transaction is valid even after covering a longer path, thus rendering most transactions valid. Results also indicate a linear increase up till approximately 20 iterations after which packet drops reports no significant change. This is mainly due to the limitation on total number of transactions which is kept to a 1000. The increase in packet drop due to hop count reduces the size of the blockchain as reported in Fig. 2. Packet drop due to null transactions show a similar result for all hop windows due to similarity in the number of nodes that are removed.

In the second set of simulations, we evaluate the outcome for varying edge probabilities with fixed hop window. We assign $h = 4$ and evaluate for edge probabilities, p of 2%, 6%, 10%, 15% & 20%. These simulations are conducted on a single miner framework for better illustration of the effects of network density. We remove 10% nodes belonging to the cutset upon each iteration and evaluate results. This is repeated until the entire cutset has been removed.

Fig. 9 reports that as the edge probability of the network is increased from 2% to 20%, the packet drop due to hop count decreases as shown in Fig. 8. This is due to the increase in network density, consequently resulting in more number of connections amongst the nodes. The increase in inter-node connectivity reduces the number of hops required to reach from sender to receiver. Fig. 7 highlights the packet drop due to null transactions increases. This can be accounted for by considering the cutset size. In every simulation, the nodes are being removed until the cutset is completely empty. For a denser network, the cutset size is larger, given the increased connectivity amongst the nodes. Hence, more nodes are being removed for denser network. As a result, the null nodes in the network increases, thereby increasing the packet drops due to null transactions. These drops collectively affect the blockchain size, but the drops due to

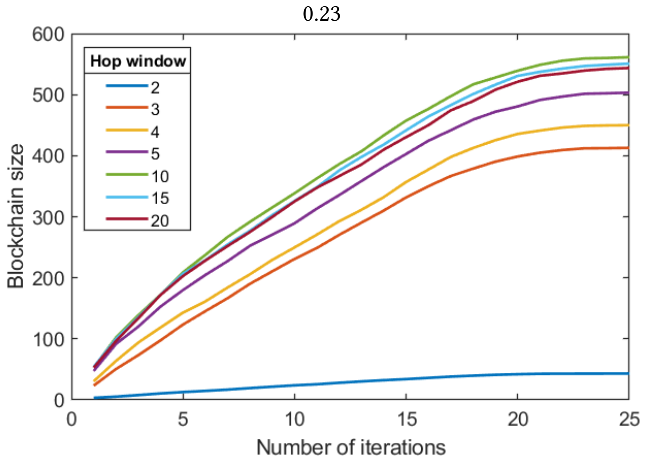

Figure 2

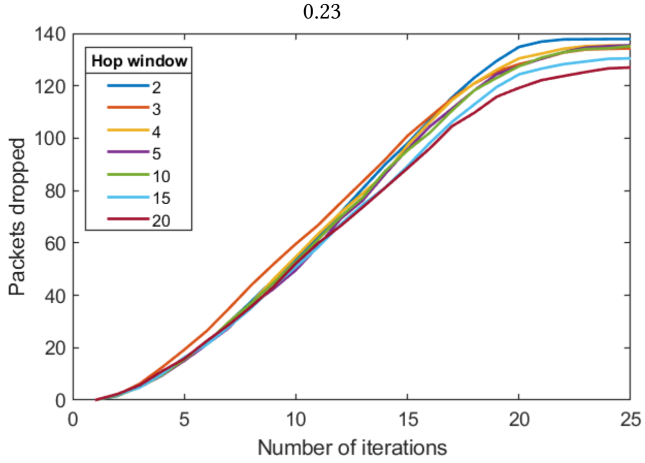

Figure 3

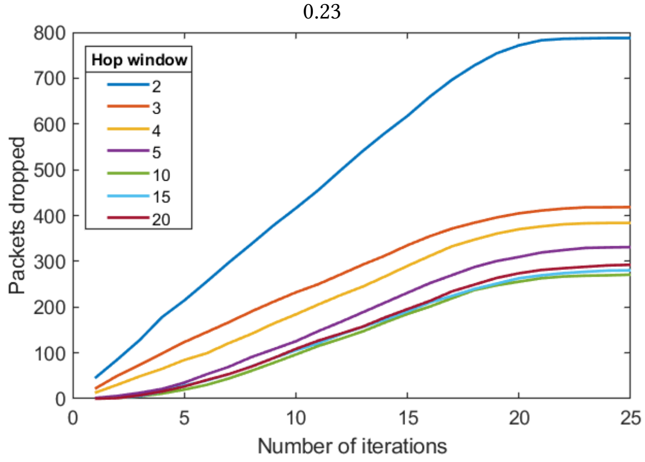

Figure 4

Figure 5: Variations in hop window (2: Blockchain Size, 3: Packet Drops due to null transactions, 4: Packet Drops due to hop expiry)

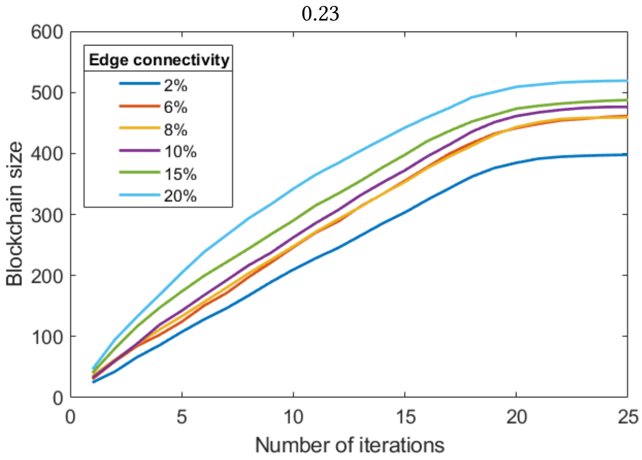

Figure 6

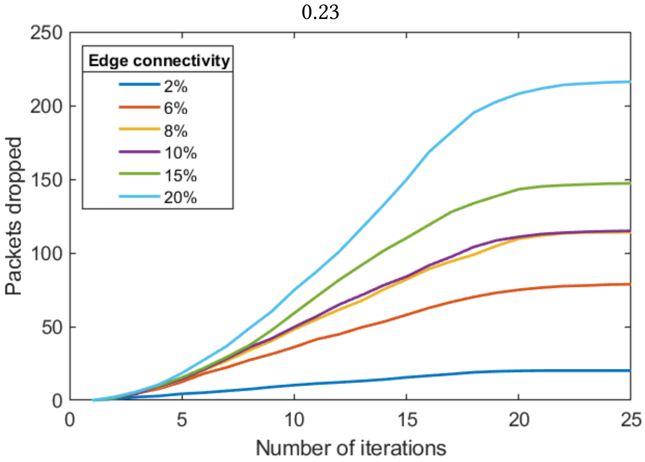

Figure 7

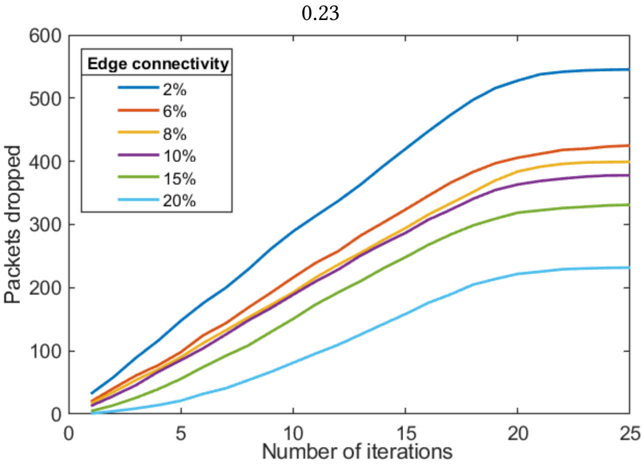

Figure 8

Figure 9: Variations in Graph Density (6: Blockchain Size, 7: Packet Drops due to null transactions, 8: Packet Drops due to hop expiry)

hop limit outweighs the drops due to null transactions, resulting in the outcomes as reported in Fig. 6.

In the third set of simulation we compare our proposed method with commonly used network analysis methods namely Random Node Selection, Degree Centrality, Closeness Centrality and Betweenness centrality. We assign $h = 5$ and $p = 10\%$. These simulations are conducted on a single miner framework for better illustration of the effects of network density and hop window.

We remove 10% of the most critical nodes of the network upon each iteration where the number of nodes removed is upper bounded by 2.5%, 5% and 10% of the whole network as reported in Fig. 12, Fig.16 and Fig.20 respectively. We see that the packet drops due to hop window is maximum for our proposed approach, followed by Betweenness Centrality, Closeness Centrality, Degree Centrality and Random Node Selection respectively, as reported in Fig. 12, Fig.16 and Fig.20. The packet drops due to null transactions remain similar for all the methods as the number of nodes removed are same as reported in Fig. 11, Fig.15 and Fig.19. These two factors, subsequently, affect the blockchain size, with the blockchain size being the least for our proposed approach, followed by Betweenness Centrality, Closeness Centrality, Degree Centrality and Random Node Selection respectively, as reported in Fig.10, Fig.14 and Fig.18. The proposed approach outperforms the existing approaches due to a better selection of better nodes, which in turn creates a bottleneck in the network, thus causing packet drops and delays. A similar trend is observed when the percentage of network being removed is increased from 2.5% to 10%.

In the final set of simulations, we justify the scalability of our proposed approach with multiple miners and compare it with various network analysis approaches. We assign $h = 4$ and $p = 10\%$ and average out the outcomes of 25 random network topologies. We remove the entire cutset in the beginning of the iterations for each network analysis method.

Fig. 22 and Fig. 23 highlight the results for 3 and 5 miners respectively with the similar patterns. Our proposed method significantly outperforms the other methods as reported in Fig. 24. In our approach, after the removal of the cutset, the entire network gets partitioned into two parts. This results in each part separately maintaining it's own copy of the blockchain, inconsistent with the other. The two blockchains run simultaneously in our proposed method due to partition of the network but for other methods, no such partition takes place.

5 CONCLUSION

This work proposes a node criticality analysis approach for blockchain scenarios. The proposed approach leverages our previous work on Spectral Partitioning and induces certain blockchain traffic flow metrics to it. Attacks are simulated on the network based upon the criticality of nodes obtained from the proposed approach and nodes are thus removed. The effects on blockchain are compared for various network densities and hop windows. The method is then compared to the majorly used existing approaches, like Betweenness Centrality, Closeness Centrality and Degree Centrality based upon the extent to which communications in the network

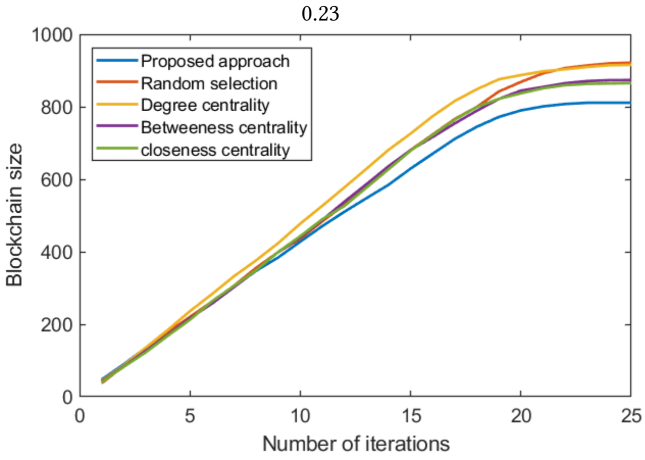

Figure 10

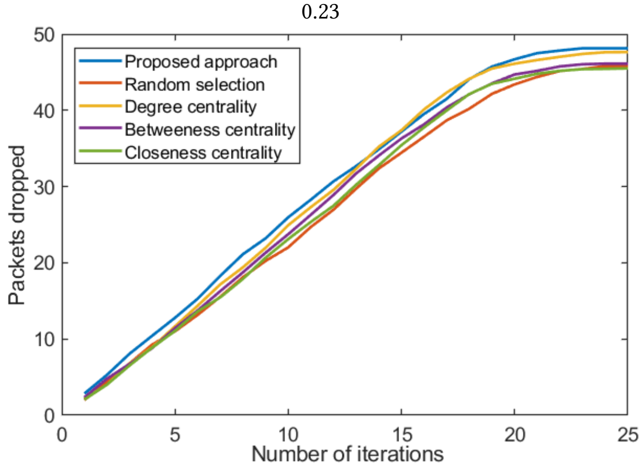

Figure 11

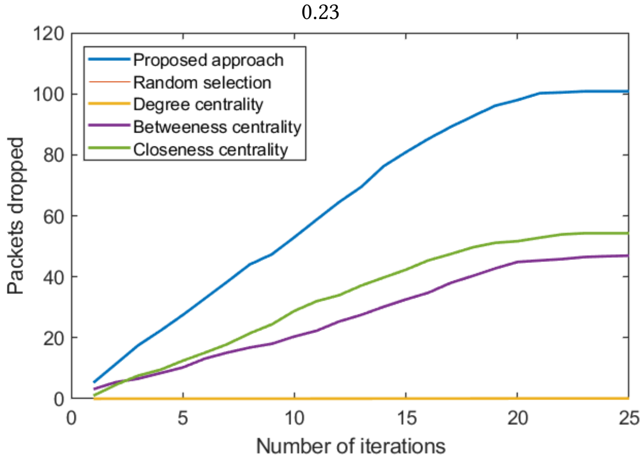

Figure 12

Figure 13: 2.5% of the network attacked (10: Blockchain Size, 11: Packet Drops due to null transactions, 12: Packet Drops due to hop expiry)

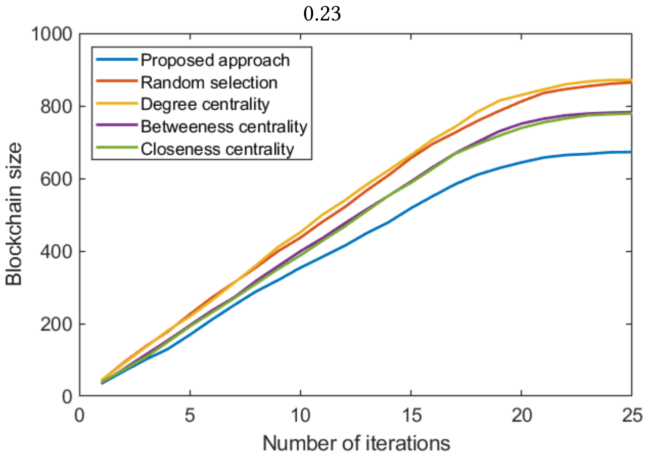

Figure 14

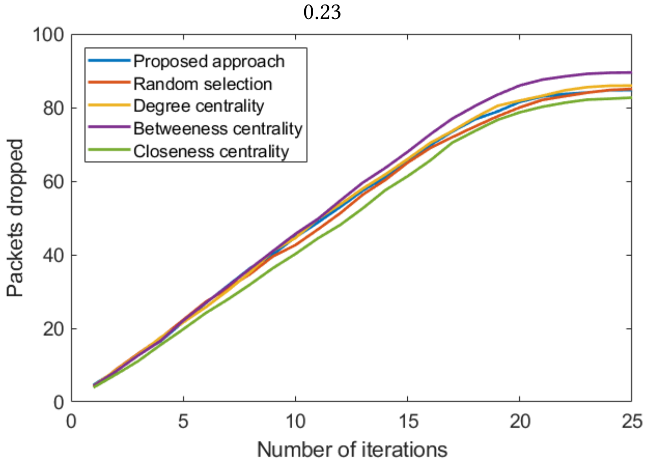

Figure 15

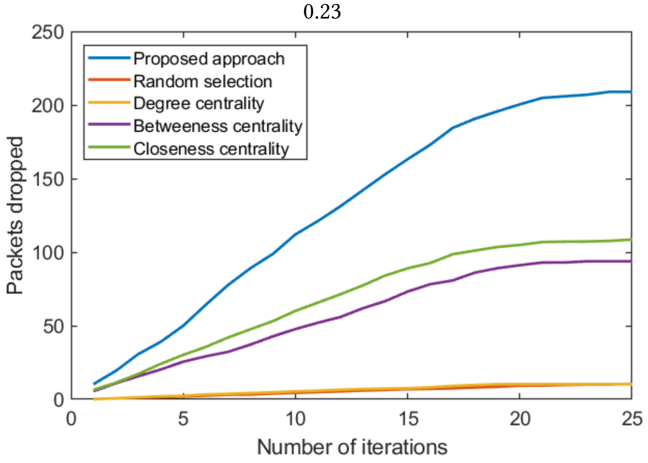

Figure 16

Figure 17: 5% of the network attacked (14: Blockchain Size, 15: Packet Drops due to null transactions, 16: Packet Drops due to hop expiry)

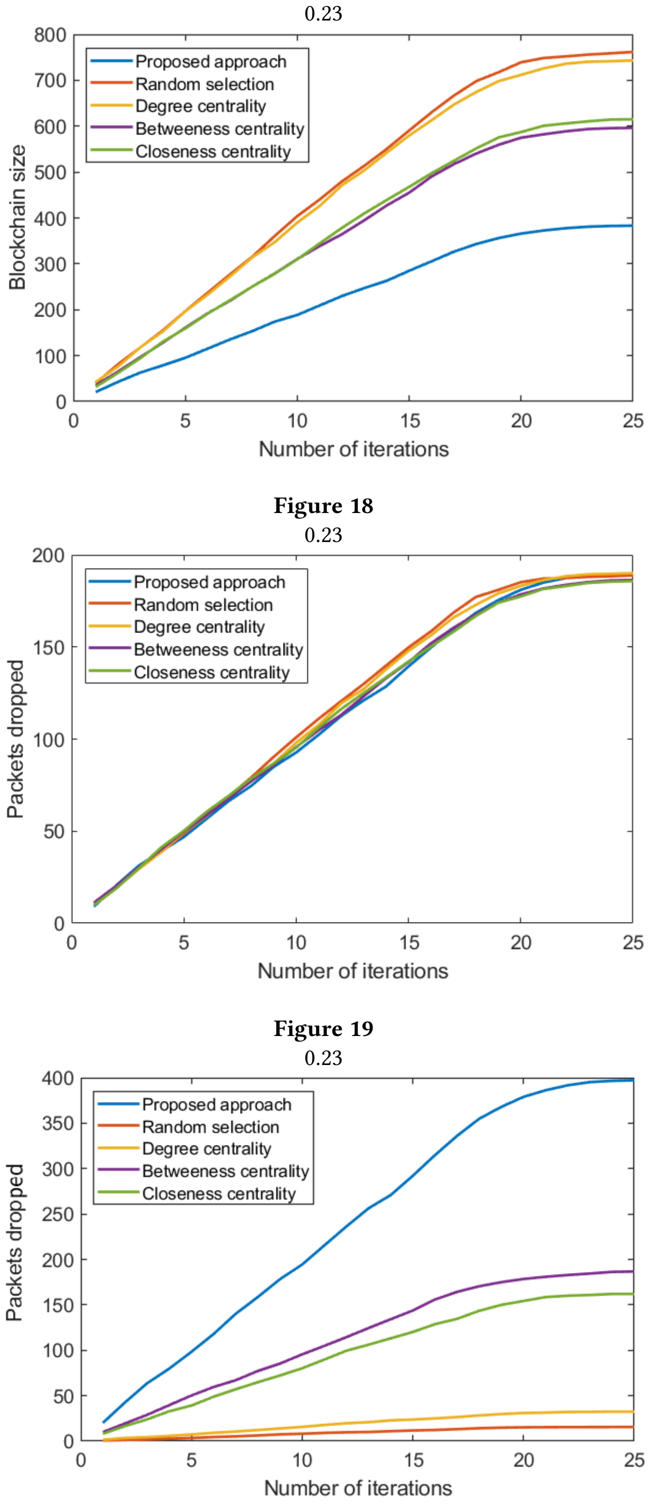

Figure 18

Figure 19

Figure 20

Figure 21: 10% of the network attacked (18: Blockchain Size, 19: Packet Drops due to null transactions, 20: Packet Drops due to hop expiry)

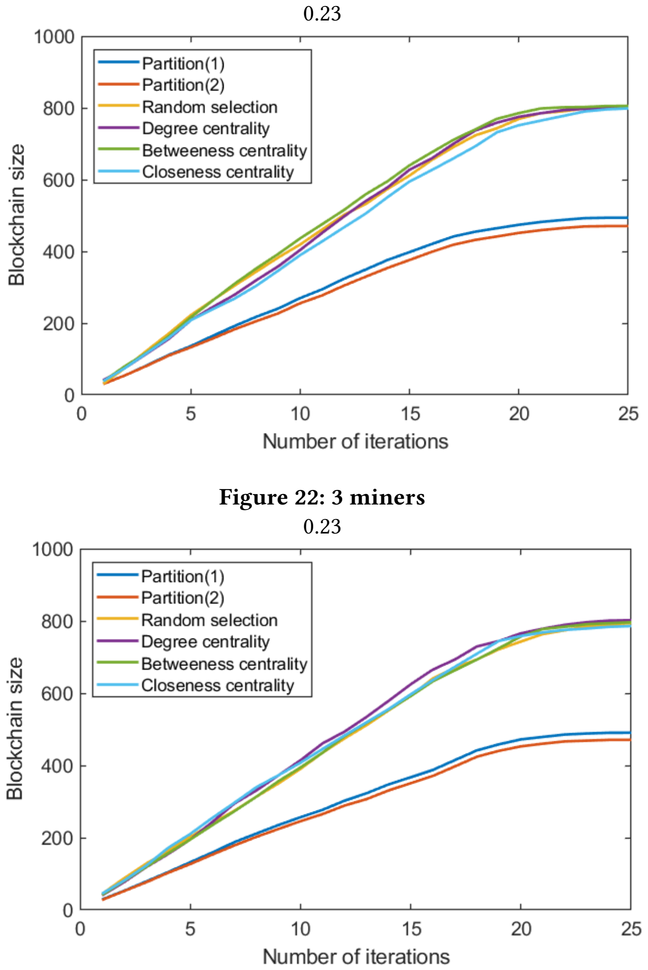

Figure 22: 3 miners

Figure 23: 5 miners

Figure 24: Multiple Mining Scenario

are affected. The proposed approach outperforms the existing approaches and is thus found to be better in identifying the critical nodes.

REFERENCES

[1] Saveen A Abeyratne and Radmehr P Monfared. 2016. Blockchain ready manufacturing supply chain using distributed ledger. (2016).

[2] Waqar Asif, Marios Lestas, Hassaan Khaliq Qureshi, and Muttukrishnan Rajarajan. 2015. Spectral partitioning for node criticality. In *IEEE Symposium on Computers and Communication (ISCC)*. IEEE, 877–882.

[3] Christian Cachin. 2016. Architecture of the hyperledger blockchain fabric. In *Workshop on distributed cryptocurrencies and consensus ledgers*, Vol. 310.

[4] Christian Cachin and Marko Vukolic. 2017. Blockchain consensus protocols in the wild. *arxiv preprint arxiv:1707.01873* (2017).

[5] Nicolas T Courtois and Rebekah Mercer. 2017. Stealth Address and Key Management Techniques in Blockchain Systems.. In *ICISSP*. 559–566.

[6] Christian Decker and Roger Wattenhofer. 2013. Information propagation in the bitcoin network. In *IEEE P2P Proceedings*. IEEE, 1–10.

[7] Joan Antoni Donet Donet, Cristina Perez-Sola, and Herrera-Joancomart. 2014. The bitcoin P2P network. In *International Conference on Financial Cryptography and Data Security*. Springer, 87–102.

[8] Ali Dorri, Salil S Kanhere, Raja Jurdak, and Praveen Gauravaram. 2017. Blockchain for IoT security and privacy: The case study of a smart home. In *IEEE international conference on pervasive computing and communications workshops (PerCom workshops)*. IEEE, 618–623.

[9] Miroslav Fiedler. 1973. Algebraic connectivity of graphs. *Czechoslovak mathematical journal* 23, 2 (1973), 298–305.

[10] Suman Ghimire and Henry Selvaraj. 2018. A Survey on Bitcoin Cryptocurrency and its Mining. In *26th International Conference on Systems Engineering (ICSEng)*. IEEE, 1–6.

[11] Shihab Shahriar Hazari and Qusay H Mahmoud. 2019. A Parallel Proof of Work to Improve Transaction Speed and Scalability in Blockchain Systems. In *IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC)*. IEEE, 0916–0921.

[12] Minhaj Ahmad Khan and Khaled Salah. 2018. IoT security: Review, blockchain solutions, and open challenges. *Future Generation Computer Systems* 82 (2018), 395–411.

[13] Ahmed Kosba, Andrew Miller, Elaine Shi, Zikai Wen, and Charalampos Papamanthou. 2016. Hawk: The blockchain model of cryptography and privacy-preserving smart contracts. In *IEEE symposium on security and privacy (SP)*. IEEE, 839–858.

[14] Tsung Ting Kuo, Hyeon-Eui Kim, and Lucila Ohno-Machado. 2017. Blockchain distributed ledger technologies for biomedical and health care applications. *journal of the American Medical Informatics Association* 24, 6 (2017), 1211–1220.

[15] Vito Latora and Massimo Marchiori. 2007. A measure of centrality based on network efficiency. *New journal of Physics* 9, 6 (2007), 188.

[16] Ao Lei, Haitham Cruickshank, Yue Cao, Philip Asuquo, Chibueze P Anyigor Ogah, and Zhili Sun. 2017. Blockchain-based dynamic key management for heterogeneous intelligent transportation systems. *IEEE Internet of Things journal* 4, 6 (2017), 1832–1843.

[17] Chao Li, Wei Jiang, and Xin Zou. 2009. Botnet: Survey and case study. In *Fourth International Conference on Innovative Computing, Information and Control (ICICIC)*. IEEE, 1184–1187.

[18] Esther Mengelkamp, Benedikt Notheisen, Carolin Beer, David Dauer, and Christof Weinhardt. 2018. A blockchain-based smart grid: towards sustainable local energy markets. *Computer Science-Research and Development* 33, 1-2 (2018), 207–214.

[19] Matthias Mettler. 2016. Blockchain technology in healthcare: The revolution starts here. In *IEEE 18th International Conference on e-Health Networking, Applications and Services (Healthcom)*. IEEE, 1–3.

[20] Mitar Milutinovic, Warren He, Howard Wu, and Maxinder Kanwal. 2016. Proof of luck: an efficient blockchain consensus protocol. In *proceedings of the 1st Workshop on System Software for Trusted Execution*. ACM, 2.

[21] Jose Nazario. 2008. DDoS attack evolution. *Network Security* 2008, 7 (2008), 7–10.

[22] Sardar Vallabhbhai Patel. 2018. Currency at a Cryptic Crossroads: Decrypted. *SVP National Police Academy* (2018), 74.

[23] Claudia Pop, Tudor Cioara, Marcel Antal, Ionut Anghel, Ioan Salomie, and Massimo Bertoncini. 2018. Blockchain based decentralized management of demand response programs in smart energy grids. *Sensors* 18, 1 (2018), 162.

[24] Ahmet Erdem Sariyuce, Kamer Kaya, Erik Saule, and Umit V Catalyiirek. 2013. Incremental algorithms for closeness centrality. In *IEEE International Conference on Big Data*. IEEE, 487–492.

[25] HB Mihiri Shashikala, Roy George, and Khalil A Shujaee. 2015. Outlier detection in network data using the Betweenness Centrality. In *SoutheastCon 2015*. IEEE, 1–5.

[26] Pawel Szalachowski. 2018. (Short Paper) Towards More Reliable Bitcoin Timestamps. In *Crypto Valley Conference on Blockchain Technology (CVCBT)*. IEEE, 101–104.

[27] Hiroki Watanabe, Shigeru Fujimura, Atsushi Nakadaira, Yasuhiko Miyazaki, Akihito Akutsu, and Jay Junichi Kishigami. 2015. Blockchain contract: A complete consensus using blockchain. In *IEEE 4th global conference on consumer electronics (GCCE)*. IEEE, 577–578.

[28] Xiao Yue, Huiju Wang, Dawei Jin, Mingqiang Li, and Wei Jiang. 2016. Healthcare data gateways: found healthcare intelligence on blockchain with novel privacy risk control. *journal of medical systems* 40, 10 (2016), 218.