JOURNAL OF INFORMATION SCIENCE AND ENGINEERING 40, 943-955 (2024)
DOI: 10.6688/JISE.202409_40(5).0001

Evaluating Network Structures in Byzantine-based Consensus Algorithms for Sarawak's Digitalized Pepper Value Chain

RENG-YI KUEH[1], FU-EE TANG[2], HUO-CHONG LING[1,3],
YEE-YONG TAN[2] AND CARRIE LEE-ING HO[2]
[1]*Department of Electrical and Computer Engineering*
[2]*Department of Civil and Construction Engineering*
Curtin University, 98009 Malaysia
[3]*School of Science, Engineering and Technology*
RMIT University, 70000 Vietnam
E-mail: augustinekueh@postgrad.curtin.edu.my[+]; tang.fu.ee@curtin.edu.my; huochong.ling@rmit.edu.vn; {tan.yee.yong; carrie.ho}@curtin.edu.my

Known as the "King of Spice", pepper (Piper nigrum) is the most widely used and traded agricultural product in the world. In Sarawak, pepper is listed as the few industrial crops in producing and contributing the most to the gross domestic product (GDP) of the country, Malaysia. However, the recent price fluctuation in the global pepper market has caused some farmers to abandon pepper farms due to the lower pepper prices against the high cost of input materials and farm maintenance, in addition to limited marketing choices and bargaining power when dealing with buyers due to their geographic remoteness, small production quantity, and variable product. To improve the circumstances, the use of blockchain technology with Byzantine-based consensus algorithm is proposed in this paper to minimize some of the constraints faced by the smallholder farmers. With the PBFT consensus algorithm on-board, the blockchain network (BCN) operates without miners as it relies on the message-transfer mechanism to achieve the total consensus. Besides, the PBFT consensus algorithm with varying network structures were evaluated. Upon comparison between network structures, the group network structure dominated the entirety of the experiment and presented as the proposed network structure, with the addition of specific nodes such as relay, storage, administer and brackets: bench and penalty to facilitate and maintain the longevity of the BCN. The implementation of blockchain technology in Sarawak's pepper industry has the potential to improve the linkages and cohesion between pepper stakeholders but its overall integration would have to be further explored.

Keywords: blockchain technology, consensus algorithm, linkages, network structures, pepper, Sarawak, stakeholders

1. INTRODUCTION

The pepper industry is considered as one of the main contributors to Malaysia's economy, as it is one of the most important and resilient crops of the Malaysian agriculture sector. As demonstrated from the GDP 2017 report, the agriculture sector had contributed 8.2% or RM 96.0 billion to the nation's gross domestic product (GDP) and pepper is considered as a major contributor for this economic activity [2]. Oil palm and rubber contributed 46.6% and 7.3% respectively to the agriculture sector while other commodities such as pepper, cocoa, and paddy contributed 18.6% [2]. According to the Malaysian Pepper

Received January 5, 2023; revised May 3, 2023; accepted August 4, 2023.
Communicated by Wei Kitt Wong.

Board (MPB), the total production of pepper in Malaysia was 30,433 Mt (metric tons), with 11,640 Mt (metric tons) exported to international markets, and the export earning was RM 308.87mil in 2017 [3]. As a result, Malaysia is ranked as the fifth largest pepper-producer in the world. The Malaysian state of Sarawak contributes a total of 95% of the pepper production in Malaysia, and the remaining 5% are from Sabah and Johor [4].

In 2019, it was reported that the country had produced 34,294 Mt (metric tons) of pepper and contributed RM1.95 billion to the nation's GDP, which is around 0.1 percent of Malaysia's commodities GDP [5]. While the pepper crop can provide a major contribution to the nation's and state's agricultural economy, the recent drop in the price of pepper has affected the livelihood of pepper farmers drastically, resulting in many of them having to abandon pepper farming due to the high cost of maintenance, fertilizers and pesticide [5]. Furthermore, pepper farmers have little bargaining power in selling their commodity, as they have limited access to marketing channels [6]. An inherent issue in the Sarawak's pepper industry is the lack of clear linkages between the pepper value chain stakeholders. This prevents smallholder farmers from accessing or participating the pepper value chain. As the proposed solution, a technological medium such as a blockchain platform with an underlying consensus algorithm could provide an intervention and address some of the concerns that are limiting the participation of smallholder farmers in the pepper value chain. Platforms that utilize blockchain technology can provide total provenance and report the status of harvested crops laid within the agricultural value chain. The underlying function can create clear linkages between the pepper value chain stakeholders and track the flow of the pepper products. In this paper, the underlying consensus algorithm is placed as the focal point as it enables the integrity of the commercialization network and helps reducing vulnerability against malicious users, which is important because of the utilization of distributed ledger technology (DLT), whereby online transactions are shared between system nodes for data immutability and the high-degree of security that can be provided by its underlying consensus algorithms [7].

2. LITERATURE REVIEW

Byzantine Fault Tolerance (BFT) is described as a distributed system that is capable of tolerating a class of failures originated from the Byzantine Generals' Problem. Thus far, the Byzantine fault is considered the onerous failure mode to deal with as compared to other class of failure modes. For instance, a node can still pose as having honest data while providing arbitrary data or faulty decision to the consensus network. Without the Byzantine Fault Tolerance in existence, a peer can simply post and transmit erroneous transactions that will render reliability issues to a distributed network. Furthermore, there are no authorities to resolve the occurrence and suspend the malicious node for compromising the consensus network [8].

To solidify a trustless system for a blockchain platform, Byzantine-based consensus algorithms are deployed and govern the continuous operation of the BCN despite the arbitrary interruption by the Byzantine nodes. The BFT-enabled system can allow 1/3 of the replica nodes to be faulty, then refreshed their states with proactive recovery schemes. However, such schemes may not be able to eradicate certain specific issues such as the prolonged process of repairing the compromised replica and the re-synchronization of ac-

tive replicas to serve the network while waiting for the recovery of the faulty nodes. Therefore, the solution provided by current BFT consensus algorithm may not work well in a large-scale BCN of the future, if the issues are still remained and without further optimization to the consensus mechanism [9].

A practical state machine replication algorithm was derived and proposed by Castro and Liskov [10] in 1999 at Massachusetts Institute of Technology (MIT), entitled the Practical Byzantine Fault Tolerance (PBFT) [11]. As the paper presented, the algorithm can tolerate Byzantine faults and offer 2 core attributes, liveness and safety in the asynchronous environment such as the Internet, providing at most $[(n-1)/3]$ out of a total of n replicas that are concurrently faulty or exhibit arbitrary behavior [10]. The liveness attribute ensures the retrieval of replies to the requested clients while the safety attribute will make sure only non-faulty nodes can execute the requests simultaneously [12].

A distributed system can achieve consensus via its message transfer mechanism and underpin the quorum theory in the transmission protocol, which requires the distributed transaction to procure a minimum number of votes so as to perform the requested operation by a client [13]. As a result, the PBFT algorithm is widely known as the main consensus algorithm for permissioned blockchain due to its reliance on voting-based mechanism that helps to generate higher throughput transactions when comparing to other consensus methods, including PoW and PoS, which ubiquitously implemented in public blockchains and consumed a considerably high amount of computational power to solve cryptographic puzzles based on the mining-based mechanism of PoW and stake requirements from PoS [14].

However, its scalability may pose a downside to the gradual expansion of the network as the waiting time increase due to the quadratic time complexity of the algorithm; $T(n) \in O(n^2)$ to the latency of data communication [13]. In addition, nodes are not allowed to simply participate or exit without restarting the whole system, which obstructs the dynamic synchronization of the total number and state of nodes in the distributed network [15].

3. METHODOLOGY

In essence, the PBFT consensus algorithm was separated into three different network structures as test subjects. The test subjects were evaluated based on the performance, reliability, and scalability (PSR) metrics and compared using line charts to manifest the final results and announce the most robust network structure. Subsequently, a proposed network structure and blockchain system architecture were drawn to represent the framework for the backbone of the software layer, in conjunction with the digitalized pepper value chain that connotes the interactivity between radio-frequency technologies and the BCN.

With the qualitative findings presented in [1], blockchain technology is getting more promising to be the main driver in alleviating pain points such as the following:

(1) the lack of product transparency, which may lose its stringent quality when passing to several stages without an automated supervision, and
(2) the lack of clear visibility of alternative linkages, especially to stakeholders that were not directly connected in the pepper value chain.

Therefore, it is important to study the underlying consensus utilized in the BCN and its metrics, which was experimented based on the suggested PBFT consensus algorithm.

As such, this section describes how the PBFT consensus algorithm and selected network structures were programmed and compared in terms of performance, scalability, and reliability (PSR) metrics to gauge the overall efficiency of the proposed network structures. Subsequently, the combination of the group and layer network structures are introduced with additional mechanisms in countering faulty primary or replica nodes during the transactional process. Additionally, a system architecture was drawn to reveal the inner technologies and concepts involved in the devised BCN.

Based on the discussions of PBFT in the literature review, the following test were conducted with three different network structures. In Fig. 1, the structure was basically the vanilla PBFT without additional customization. In Fig. 2, the structure was the network nodes fragmented into segregated groups, meaning there were more than one primary node and were formed into a constituted primary group, with each primary node having their own replica nodes. In Fig. 3, the structure was the primary nodes branched and chained in an imaginary line. All replica nodes were arranged in each knot of the layer to the assigned primary node, eventually producing linked layers with connected primary nodes as the main linkage of broadcasting transaction requests.

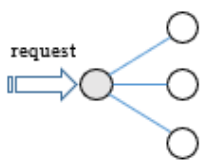


Fig. 1. The basic network structure (grey = primary).

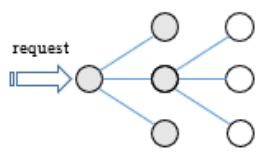


Fig. 2. The group network structure (white = replica nodes).

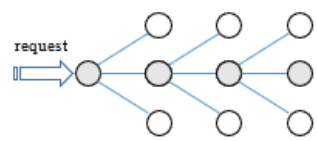


Fig. 3. The layer network structure (only show a branch).

In terms of metrics, the experiment was conducted to observe the three following aspects of the PBFT consensus algorithm with different network structures: namely the performance, scalability, and reliability. Performance was measured by recurring transactions for any boosts or throttles during the process. Scalability is essentially increasing the number of nodes while measuring the completion time per completed transaction. At the same time, reliability will be determined based on how many data packets received with correct digest, message sequence, and key verification. Any anomalies will also be recorded into designated penalty files assigned with node IDs. With the recorded data from each aspect of the consensus algorithm, a direct comparison can be made by charting a line graph and interpret their overall efficiency in completing the simulated transactions.

In depth, consensus simulator was capable in creating and running multiple nodes, which in this experiment would be known as multiple terminal windows. Within each terminal window, automated procedures were executed to undergo these pre-defined consensus phases: request, pre-prepare, prepare, commit, and reply, as shown in Algorithm 1.

Algorithm 1: Message Handler with Goroutines (node.go)

Input: a replica node & node address, nodeTable
Output: acquired mutual consent
begin
  for *//infinite loop*
    Open listening port; Accept incoming packets, data

```
        Generate asynchronous threads with goroutines to handle incoming packets
        Use a declared queue, a channel data type variable to pipe data to handleMsg()
        Generate asynchronous threads with goroutines to handle messages based on packet header
            for //infinite loop inside handleMsg()
                header, payload, sig := splitMsg(data)
                switch(header)
                    case Request: handleRequest(payload, sig)
                    case PrePrepare: handlePrePrepare(payload, sig)
                    case Prepare: handlePrepare(payload, sig)
                    case Commit: handleCommit(payload, sig)
            end for
        end for
end
```

The results collected were the time taken to receive majority votes of acceptance for each transaction in milliseconds, and for each metric was conducted in different methods. For instance, performance metric was determined by solely recording the time taken from a batch of transactions, *e.g.* 20 transactions, then gradually incrementing allocated transactions until it reached a certain saturation or distinction of results between different network structures. The scalability metric was determined by incrementing number of network nodes for every recurrent test, obtaining the results, then observing for any hampering in TPS that was caused by populating nodes to the simulator. The overall objective for this simulator is solely prioritized on allowing a client node to broadcast transaction requests and receive responses from all replica nodes, n or until a minimum consensus has been achieved, using Eq. (1) as the decisive formula.

$$m_c = \frac{n-1}{3} + 1 \tag{1}$$

4. RESULTS AND DISCUSSIONS

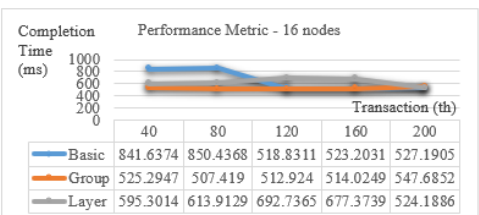
4.1 Performance Metric

The experiment was conducted using a Core i7 (7700HQ) Intel Mobile Processor with 8GB RAM (2400MHZ) to provide a better simulation when opening duplicate applications in a single launch. Generally, the performance metric is to manifest how the network structures were able to process the increasing amount of transactions with the minimum of time required to receive the minimum consensus being received by the client to approve each pending transaction. The time taken to process/consent transactions and broadcast were taken into consideration, along with how well each replica node was interacted in the network without faulty operations such as a lost packet, incorrect message sequence, and failure in digest-key verification for the reliability metric. Actual chaining of blocks and data recording/archiving by the storage nodes were not included in the experiment.

In this performance test, a total of 16 network nodes was evaluated in three different network structures. The network structures were evaluated individually in the literature but never against each other. All network nodes were continuously fed with a total of 200 transaction requests from a designated client node, with a new transaction request being

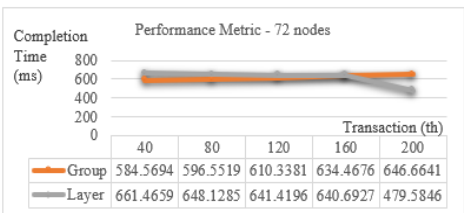
produced after each 10 seconds by the client node since the experiment was conducted on a single machine to prevent bottlenecks and achieve consistent results. The main objective is to evaluate the stability of the consensus mechanism when operates in a prolong period by comparing the total execution of completing between each 40th transaction request to get the best average of results in each round of testing. A total of four iterations were made for each test, *e.g.* $4 * 200$ transactions, with the result for each iteration being added together, then averaged to obtain the final result.

Fig. 4 is a line graph with the completion time as the major factor of the performance metric between the basic, group, and layer network structures. It shows that the group network structure had the least completion time through all the 200 simulated transactions, followed by the layer network structure with approximately 600 milliseconds in completing a simulated transaction. The basic network structure had the most completion time as compared with the other network structures and was able to be on par with the layer network structure in terms of long-term system durability. However, it suffered in the latter experiment by having continuous packet loses and incorrect message sequence due to the bottleneck suffered in the primary node, primarily due to broadcasting and consenting at the same time. Therefore, it is not particularly suitable in having more than 16 replica nodes in the ongoing experiment. All network structures mostly consumed shorter time as transaction progresses due to network stabilization after recurrent testing, which is exhibited in the latter figures, including the figures from the scalability section.



	40	80	120	160	200
Basic	841.6374	850.4368	518.8311	523.2031	527.1905
Group	525.2947	507.419	512.924	514.0249	547.6852
Layer	595.3014	613.9129	692.7365	677.3739	524.1886

Fig. 4. Performance test for basic, group and layer network structures with a total of 16 network nodes.



	40	80	120	160	200
Group	584.5694	596.5519	610.3381	634.4676	646.6641
Layer	661.4659	648.1285	641.4196	640.6927	479.5846

Fig. 5. Performance test for group and layer network structures with a total of 72 network nodes.

Fig. 5 indicates that the group and layer network structures were the only participants to compete against each other, with 72 allocated network nodes per network structure. Based on the pattern, it is ascertaining that both network structures were able to sustain the continuous injection of transaction requests within the PBFT consensus algorithm. Again, the group network structure was able to process the majority of the transaction requests in lesser completion time compared to the layer network structure. Although the layer network structure managed to complete the 200th transaction request within 479 milliseconds, the group network structure was able to run 200 transactions with 84 allocated network nodes simultaneously. However, in the current machine, both network structures were unable to process the maximum total of 128 network nodes due to the similar circumstances faced by the previous basic network structure, which were the data packet loses and incorrect message sequence. To conclude, the group network structure is the overall victor in the performance category as it is capable to process more than 72 replica nodes.

4.2 Scalability Metric

As for the scalability metric, its purpose is to determine the minimum time required to process transactions while increasing the amount of replica nodes for each recurrent test, then observe the pattern drawn on the line graph to justify the capability of the network structure handling the total number of network nodes. The scalability test was progressed with an ascending sequence, starting from a total number of 2^2 to 2^6 network nodes. The results were obtained by averaging the completion time based on four iterations of 100 completed transactions, then compared amongst basic, group, and layer network structures.

Fig. 6 is a line graph with the completion time as the major factor of the scalability metric between the basic, group, and layer network structures. It demonstrates that the group network structure preserved a lower completion time compared to the basic and layer network structure. The layer network structure was gradually increasing its completion time upon adding network nodes to the experiment while the basic and group network structures were shown with a downturn when having 16 network nodes in the simulation, which is caused by the recurrent tests; from the tests of 4 – 12 network nodes that rendered shorter time towards the last test. As mentioned in subchapter 4.1, the basic network structure was unable to participate in the latter experiment due to node latency that caused inherent issues such as data packet loses and incorrect message sequence.

Fig. 7 shows both network structures were having the similar and stable line patterns, with the group network structure requiring the overall lesser completion time. Between 48 and 56 for the layer network structure and between 56 and 64 for the group network structure from the x-axis, the decrease in completion time proved that the slight increase of network nodes will not dampen the efficiency of the PBFT consensus algorithm with these two network structures as the building foundation.

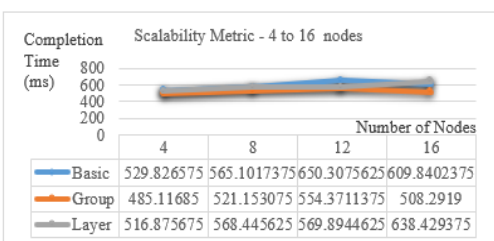


Fig. 6. Scalability test for basic, group, and layer network structures from 4 to 16 network nodes.

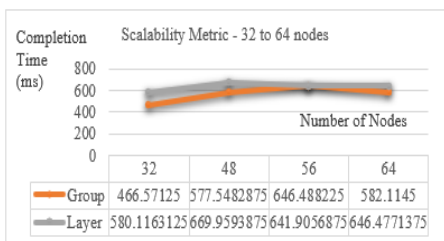


Fig. 7. Scalability test for group and layer network structures from 32 to 64 network nodes.

4.3 Reliability Metric

The reliability test was initiated to discover any faulty operations between node phases. A fault can be a lost packet, an incorrect message sequence, or a failure in digest-key verification. All notable faults were exported to designated node files to determine the severity of a lagged node. As this metric is all about true or false (0 or 1) basis, graphs will not be provided in this particular section. Once a fault is discovered, the network structure is considered ineligible to handle that amount of network nodes.

Throughout the performance and scalability experiments, the basic network structure was the only subject that was unable to progress the experiment with more than 16 network nodes, specifically with 20 network nodes in the experiment, thus making it the least reliable network structure to construct in a consensus mechanism. On the other hand, the group and layer network structure were able to handle up to 72 network nodes in the experiment, with the group network structure having the slight advantage in consensus speed and the additional of 12 consensus nodes allowed in the mechanism. Overall, the group network structure proves to be reliable and therefore, it is chosen to be part of the PBFT consensus algorithm for the blockchain system architecture.

5. PROPOSED NETWORK STRUCTURE AND MECHANISMS

The following is the comprehensive network structure that describes and illustrates the actual flow of a transaction request and the allocation of network nodes in sequence groups. Based on the experimental results, the group network structure has the least completion time in majority of the test. Hence, the proposed network structure will be segmented into groups. Fig. 8 exhibits 3 significant groups: relay, consensus, and pending groups. Each group is tasked with different functionalities and distinct roles. First and foremost, the uppermost group consists of relay nodes, which can be represented by localized servers stationed at MPB branches. Relay machines or localized servers are used to relay transaction requests without participating any ongoing consensus from the client. In addition, MPB branch can register new participants by linking to the localized servers with office computers and update the member listing in the form of Windows GUI. Secondly, the consensus group comprises of primary nodes (grey) and replica nodes (white), with each segment allocated with one primary and three replica nodes. For the purpose of illustration, three replica nodes are allocated, which in reality can be added to a certain capped amount. The actual capped amount is not available as the prototype is not being tested and developed at the time of writing.

Algorithm 2: Group Network Operation (group.go)

Input: a replica node, a nodetable
Output: newNodeTable(grouped), assigned primary node
begin
Initialize an empty map with string to string; newNodeTable
Initialize three arrays of string data type for storing keys based on digits
Initialize a counter: count - when 4 is reached, reset back to 0,
Initialize two flags: match - set to true if the current key-value matches the terminal nodeID,
done - set to true once newNodeTable has filled

for a := range nodeTable
if length equals to 2 //including the word 'N', means a node
append to an array; oneDigitKeys
else if length equals to 3, append to an array; twoDigitKeys
else append to an array; threeDigitKeys
end for

using sort.Strings() to all arrays, then append to key array respectively

```
    for a := range keys // key array
        if a != "C0"
            if count is equal to 0 AND done is false, re-initialize newNodeTable
        Divide the key-value from the sorted nodetable to keyArr and valArr respectively
        if nodeID is equal to a (current loop value), match assigns to true
        Increment the count counter
        if count is equal to 4, count is reset to 0
            if match is true
                for (p := 0; p <= 3; p++)
                    newNodeTable[keyArr[p]] = valArr[p], increment newNodeCount
                    set primary to the keyArr[0], set done to true, set match to false
                end for
        set keyArr and valArr to nil,
        set assignFlag to true
    end for
end
```

As shown in Fig. 8, multiple consensus groups can be formed in this PBFT consensus mechanism. Primary nodes in the group will multicast the transaction request from the designated relay node to its party members (replica nodes) located in the replica group. The furthermost group in the layout is called the pending group, whereby nodes are divided between two brackets: namely the penalty and bench. Network nodes that were caught having multiple faults will be sent to the penalty bracket and waiting to be filtered by the MPB administration panel, mostly by cross-referencing its message logs and the condition of the device. Any irregularities in the message logs will be deleted and be given an updated copy of the blockchain from the administrator. After the filtration, the refreshed node will be sent to the bench bracket, waiting to be formed a new party with other replica nodes with the corresponding district relay node. For instance, Miri district pepper stakeholders will be placed under the management of Miri's relay nodes.

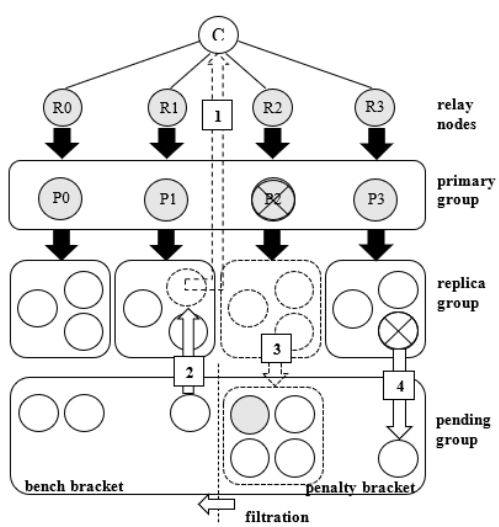


Fig. 8. Group network structure with node penalty and filtration system.

Notice that there are numbered boxes labelled in Fig. 8. The 1st numbered box is when the node has made a transaction request, it will automatically become the client. When that happens, a node from the bench bracket will be immediately placed into the empty slot, as shown by the arrow with the 2nd numbered box. In any case that there are no replacements for the empty slot, the segment will continue the normal-case operation as planned. Subsequently, the 3rd numbered box is when a primary node has attempted multiple faults such as unresponsiveness and malicious acts, the entire segment will be relocated to the penalty bracket and be filtered accordingly. As for the 4th numbered box, the replica node that caused multiple faults will be sent to penalty bracket as opposed to the entire segment. At the same time, a replacement will be made by funneling a bench node to the vacant slot, on the condition of having the similar district. As a clear distinction, the allocation of relay nodes in accordance to Sarawak's districts along with the Node Penalty and Filtration System are differentiated from the existing network structure in the literature to provide additional security and protective mechanisms to the BCN.

Fig. 9 shows the Sarawak's pepper value chain, with each stakeholder participated in the BCN as a consensus node. Additionally, RFID and NFC are utilized as the integrated wireless technology for provenance tracking. Through this layout, individuals are able to be cognizant about the product integrity before purchasing. Alternatively, buyers that are not equipped with a NFC-enabled smartphone can opt for QR scan to view the similar information from the blockchain database. The application is purposely useful for individuals, regardless of the participation in the BCN to identify the originality of the pepper product without having to guess its authenticity amidst purchase decision process. It is also proved to be user-friendly, which can simply scan on the anti-tampered seal imprinted with NFC and QR labels to retrieve product information from external shared databases that was encrypted and provided by the storage nodes in the BCN.

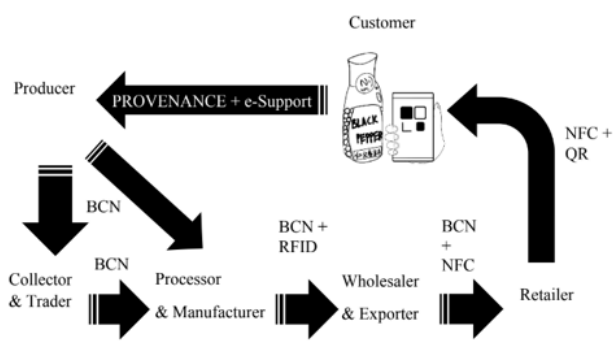


Fig. 9. A digitalized Sarawak's pepper value chain.

6. CONCLUSION

In this paper, network structures in PBFT consensus algorithm were evaluated, namely the basic, group, and layer network structures. All network structures were evaluated based on the PSR metrics, and were compared with time taken as the responding variable of the experiment, except for reliability, which was determined once an anomaly has reached. All results were tabulated and charted into a line graph for better comparison.

Results showed that the group network structure succeeded in surviving the stress test environment, which was equipped with 84 replica nodes in the simulated BCN with less than 1 second per transaction. Therefore, the group network structure was appraised and considered as a robust technique in maintaining the long-term blockchain system for Sarawak's pepper industry. Additionally, proposed network structure and mechanisms were drawn to visualize the actual flow of a transaction request and the allocation of network nodes in sequence groups.

REFERENCES

1. R. Y. Kueh, F. E. Tang, H. C. Ling, Y. Y. Tan, and L. I. Ho, "Digitizing Sarawak's pepper value chain: Uncovering pain points and linkages," *Green Energy, Computing and Sustainable Technology Conference*, 2022, pp. 487-492.
2. Department of Statistics Malaysia, "Selected agricultural indicators, Malaysia, 2018," https://dosm.gov.my/v1/index.php?r=column/cthemeByCat&cat=72&bul_id=UjYxeDNkZ0xOUjhFeHpna20wUUJOUT09&menu_id=Z0VTZGU1UHBUT1VJMFlpaXRRR0xpdz09, 2018.
3. Malaysia's Open Data Portal, "Information on black pepper industry," Department of Statistics, Malaysia, 2019.
4. O. A. Olalere, A. Nour, O. R. Alara, and M. M. Ahmad, "The impact of pepper production on Malaysian economy and the need for nutraceutical diversification," *The National Conference for Postgraduate Research*, Vol. 3, 2016, No. 55368609.
5. Malaysian Pepper Board, "Malaysian pepper board set to spice things up," https://www.mpb.gov.my/mpb/index.php/en/mpb-in-news/352-malaysian-pepper-board-set-to-spice-things-up, 2020.
6. "Farmers struggle with low prices," *The Borneo Post*, https://www.theborneopost.com/2018/11/29/farmers-struggle-with-low-prices/.
7. G. Sylvester, *E-agriculture in Action: Blockchain for Agriculture Opportunities and Challenges*, The Food and Agriculture of the United Nations and the International Telecommunication Union, Bangkok, 2019.
8. G. Konstantopoulos, "Understanding blockchain fundamentals, Part 1: Byzantine fault tolerance," https://medium.com/loom-network/understanding-blockchain-fundamentals-part-1-byzantine-fault-tolerance-245f46fe8419, 2017.
9. H. L. Zhang, "Byzantine fault tolerance for distributed systems," MS Thesis, Department of f Electrical and Computer Engineering, Cleveland State University, 2014.
10. M. Castro and B. Liskov, "Practical Byzantine fault tolerance," in *Proceedings of the 3rd Symposium on Operating Systems Design and Implementation*, 1999, pp. 173-186.
11. L. Seeley, "Introduction to sawtooth PBFT," *Hyperledger*, https://www.hyperledger.org/blog/2019/02/13/introduction-to-sawtooth-pbft. 2019.
12. K. Lei, Q. Zhang, L. Xu, and Z. Qi, "Reputation-based Byzantine fault-tolerance for consortium blockchain," in *Proceedings of IEEE 24th International Conference on Parallel and Distributed Systems*, 2018, pp. 604-611.
13. L. Feng, H. Zhang, Y. Chen, and L. Lou, "Dynamic multi-agent practical Byzantine fault-tolerant consensus in permissioned blockchain," *Applied Sciences*, Vol. 8, 2018, No. 55563884.

14. CrushCrypto, "What is practical Byzantine fault tolerance (PBFT)?" https://crush-crypto.com/what-is-practical-byzantine-fault-tolerance/.
15. Y. Jiang and Z. Lian, "High performance and scalable Byzantine fault tolerance," in *Proceedings of IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference*, 2019, pp. 1195-1202.

Reng-Yi Kueh received the Bachelor of Technology (Computer System and Networking) and Master of Philosophy (Electrical and Computer Engineering) from Curtin University, Malaysia. His research interests include distributed computing and consensus algorithms.

Fu-Ee Tang received the Bachelors and PhD degrees from Victoria University of Manchester. His PhD was focused on pollutant dispersion in coastal flows near a headland. He is the Dean of Learning and Teaching, having served as Head and Coordinator for the Engineering First Year programme from 2006-2011.

Huo-Chong Ling received his B.Eng. (Hons), M.Eng.Sc. and Ph.D. degrees from Multimedia University, Malaysia. He is currently a Senior Lecturer in the School of Science, Engineering and Technology, RMIT University, Vietnam. He serves as an Adjunct Associate Professor in the Department of Electrical and Computer Engineering, Faculty of Engineering and Science, Curtin University, Malaysia. His research interests include steganography, watermarking and access control system.

Yee-Yong Tan received his Bachelor of Engineering (Honours) and Doctorate of Philosophy from Curtin University, Malaysia. His research focuses on the area of subsurface flow constructed wetlands and mathematical modelling of wastewater and faecal sludge treatment.

Carrie Lee-Ing Ho graduated with MSc in Geomatics Engineering (specializing in Engineering for the Environment) from University of Calgary in 2002. Her undergraduate was in Civil Engineering from the University of Manitoba, Canada. Her research focuses on the area of hydrology and water resource management.