Enhancing the Raft Consensus Algorithm

TALAKOTI SNIGDHA[1], BARMA VISHWANTH[2], N. RAMA DEVI[3]
[1, 2, 3] Department of Computer Engineering and Technology, Chaitanya Bharathi Institute of Technology, Hyderabad, India

Abstract— Use of IOT devices increased rapidly and most of the data is collected, that data is being developed into smart data to understand the activities of users in public spaces. Huge data sets are difficult to analyse using traditional processing techniques, so organisations turned to conversion of smart data by using advanced AI and Machine Learning techniques. To ensure data transparency and privacy for personal and sensitive information, private blockchains are applied with Raft Consensus algorithm. Huge data results in more number of nodes which increase transactions and number of messages. To reduce overall system degradation, the collected transactions are divided into certain amount of transactions into cells, where these cells are optimised in the blockchain system using Raft algorithm. Therefore, we developed an algorithm which is cell based and reduces massive transactions in the smart data market.

Index Terms—Raft consensus algorithm, machine learning, federated learning, privacy.

I. INTRODUCTION

Today we live in a world of data overflow, where massive information is collected from diverse sources like IOT devices causing increasing digital traffic. Smart data, a key part of smart city development is rapidly expanding and shedding light on human activity from individual to citywide levels. Human mobility is vital for understanding human behaviours and activities, especially in health concerns like respiratory diseases including covid-19. Around 4 billion mobile phones worldwide have sensors for the location that track human movement patterns and adherence to restriction, using such data raises privacy issues as information is collected and shared without explicit permission. Hence large companies are turning to blockchain due to its transparency and security in managing data, especially personal data, for efficiency, smaller networks are use of mechanisms like pacts algorithms for agreement and security. By using the cell based Raft consensus algorithm to handle transactions efficiently, we have grouped them into manageable segments to maintain stability.

It is clear that we are now immersed in a flood of data, as evidenced by the many events produced on various platforms. The growth of IoT devices and internet traffic is huge, and projections show an increase in the number of users and data volumes. The amount of connected devices is to reach 124 billion in the next decade. This increase in smart data, in particular, plays an important role in the development of smart cities.

These new big data sets provide valuable insights into human activity, both at the individual and municipal levels. They contribute to the complex relationship between human mobility and urban resource distribution. Human mobility, in particular, is important for behavioral understanding and is an important factor for the spread of respiratory diseases such as COVID-19. Location sensors in 4 billion mobile phones worldwide allow for movement tracking and detection of restrictions

However, withholding such sensitive information raises privacy concerns. While these data sources can be made public, they also require private permission before companies can share them. While data transparency is important for individual privacy, many businesses ’terms and policies are unclear about sharing personal information.

With the introduction of the European Union General Data Protection Regulation empowers EU citizens to manage the collected data and use of their own data Although GDPR is for the EU, it is increasingly expected by multinational companies will be reflected in the management of customer data . Many companies are turning to blockchain and other distributed ledger technologies to improve data management and protect personal information, including mobility data

With its immutable nature and transparent transactions, blockchain technology offers solutions to data security and privacy concerns. Consensus algorithms play an important role in ensuring consistency of messages in blockchain networks, directly affecting performance and efficiency. To overcome the limitations of traditional PoW and PoS algorithms, federation/private blockchains use effective consensus mechanisms such as Paxos and Raft algorithms.

For example, the Raft algorithm is implemented in the Hyperledger Fabric, making consortium/private blockchains suitable for a variety of business applications. However, there are challenges such as leader node load balancing and message volume. To overcome these, the proposed cell-based raft (CBR) consensus algorithm divides tasks into cells, ensuring system stability and continuity in smart data markets

Conclusion Continuity presents both opportunities and challenges. The proposed CBR consensus algorithm offers a promising solution to enhance blockchain performance in handling large transactions.

II. LITERATURE REVIEW

The paper *"On Using Raft Over Networks: Improving Leader Election"* describes the Raft consensus algorithm and a key part of the algorithm focuses on improving leadership elections Raft ensures that all state updates with systematic a the split is sent to a leader, which then It repeats these recent updates of all other nodes. The response time of the system, which is determined by the delay between the leader and other nodes, is influenced by the time required for various nodes to assume leadership following a node failure or leadership change. Each node, in this scenario, carries a probability of becoming a leader, impacting the anticipated response time. These probabilities are shaped by random intervals during which nodes await leader failure detection and engage in competition for leadership. The paper contends that these intervals should be uniform across all nodes but puts forth a mathematical model to assess the likelihood of generating favoured leaders. Through test bed experiments using an open-source raft implementation and it validates the observed parameters.

The Paper *"Cell Based Raft Algorithm for Optimised Consensus Process on Blockchain in Smart Data Market"* addresses the challenges posed by IoT devices and highly developed traffic, leading to smart data f relationship development, helps in crime prediction and understanding a about COVID-related -19 infections but ensuring data privacy while keeping sensitive personal information is a challenge, as many systems lack clarity on data tracking.

To overcome these issues, the paper proposes consortium/private blockchains using the Raft algorithm. However, the Raft algorithm requires processing multiple messages for groups of nodes for a single transaction, which leads to system degradation as the number of nodes increases. To mitigate this, the paper introduces the Cell-based Raft (CBR) consensus algorithm. The algorithm at hand partitions tasks into cells, thereby minimising the volume of exchanged messages while preserving the fundamental principles of the Raft algorithm. Additionally, the paper suggests employing a combined learning algorithm to ascertain the optimal cell size within a blockchain system, ensuring both high accuracy and data privacy. This methodology is poised for further refinement, particularly in the context of a smart data market featuring numerous services, where its complexity can be advanced.

The paper *"Performance Analysis of Raft Consensus Algorithms for Private Blockchains*" addresses the important issue of consensus in blockchains, focusing on network stability, which can significantly impact blockchain performance but often receives little attention if given compared to other aspects such as threat modeling.

The paper focuses on evaluating the performance of the Raft consensus algorithm in networks characterised by extremely low packet loss rates. It introduces a straightforward yet precise analytical model designed to assess the probability of distributed network segmentation. This model offers estimations of the likelihood of network partitioning based on variables such as network size, packet loss rate, and election timeout at each time point.

To validate the accuracy of the model, the paper employs the Raft simulator and observes a consistent agreement between the experimental and analytical results. This proposed model, in essence, has the potential to predict both the time and probability of network segmentation.

The paper *"FRCR: Raft Consensus Scheme Based on Semi Asynchronous Federal Reconstruction"* addresses the increasing size of distributed rafts and the corresponding decrease in cluster throughput The raft consensus algorithm requires constant optimization to achieve changes in a complex and evolving application environment. To address these challenges, the paper proposes a Federal Reconstruction Committee Raft (FRCR) consensus framework.

The FRCR algorithm leverages Federation Reconstruction technology for the training, updating, and testing of feature data sets within Raft nodes. Subsequently, the algorithm executes the program to discern proficient nodes, establish the committee structure, and enhance voting quality and speed. Notably, the algorithm integrates a semi-asynchronous buffer mechanism and a countermeasure against malicious node attacks, effectively mitigating security inconsistencies in federation aggregation.

The efficacy of FRCR is predominantly validated through experimental analysis of the seven-stage consensus algorithm. The results demonstrate its capacity to enhance the performance and flexibility of the Raft consensus algorithm within a distributed system.

The paper *"Study of Response Time and Availability of RAFT Consensus in a Distributed SDN Control Plane"* investigates the effect of controller clustering on software-defined networking (SDN) control plane response time although SDN provides flexibility and simplicity in network performance but it need infrastructure network robustness Need for maintenance and flight durability.

The research focuses on the distributed consensus algorithm implemented on RAFT, Open Daylight and ONOS SDN controller platforms. RAFT ensures data store replication, candidate selection after controller failure, and controller state restoration upon successful repair. The paper introduces a framework for analysing SDN cluster organisations in terms of response time and availability parameters.

Using Stochastic Activity Networks, the authors model RAFT performance, failure injection, and cluster recovery processes. Real-world experiments are conducted to collect rate parameters for a representative group return model. Furthermore, the paper proposes a rapid rejuvenation method to reduce the reaction time to failures caused by software errors, and ensure system scalability in time vertical. The paper focuses on the response time and availability of RAFT consensus algorithm in distributed SDN control planes.

SDN provides flexibility and flexibility of network operations, but critical networks require complexity and longevity and long-term control planes. For robustness, multiple distributed controllers are used, clustering for complexity and rapid failure. However, the effect of controller grouping on total system response time is not well understood in current literature. The study examines the RAFT consensus algorithm implemented on the OpenDaylight and ONOS SDN controller platforms. RAFT is responsible for data-store replication, candidate selection after a controller failure, and recovery of the controller state upon successful repair. In order to evaluate the performance of RAFT, the study introduces a framework for conducting a numerical analysis of different SDN cluster organisations regarding response time and availability parameters.

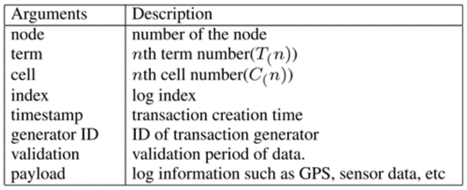
III. EXISTING PROBLEMS

Smart data markets must address the challenge of ensuring data transparency, and to show how their personal information is collected, shared, and transformed in the mechanisms. Integrating both public and private blockchains in a hybrid architecture for smart data markets due to the combination of different consensus models. The choice of consensus algorithms, such as Paxos, and Raft, for private blockchains requires careful consideration to balance security, scalability, and performance according to the specific use case. The leader-centric nature of the Raft algorithm in maintaining log sequence brings challenges related to single points of failure. Leader crashes or absence can trigger frequent leader elections, impacting system stability. Ensuring efficient and timely leader elections is essential in the Raft algorithm. Rapid leader changes due to crashes can lead to interruptions and hinder overall system performance.

Recovering from leader crashes and restoring consensus can be complex, as the process of electing a new leader involves communication and synchronised among follower servers. Maintaining a consistent state machine across distributed servers during the log log replication period is critical but can be difficult due to potential discrepancies in log entries and execution sequences. The correctness of blog replication relies on accurate matching of ten numbers and indexes in messages, posing challenges in cases of data loss or inconsistencies. As the blockchain network grows in terms of participants and transactions, maintaining the efficiency and effectiveness of the consensus algorithm becomes increasingly challenging, requiring scalable techniques and solutions. Unlike traditional Raft where servers prioritise maintaining identical logs for the latest client commands, blockchain systems can optimise for smaller log sizes. This is because, depending on the system's policy, the log might only contain essential data points from the transactions, rather than the entire transaction itself. As a result, message sizes tend to be lower. However, this efficiency comes at a cost: network load remains directly tied to the number of messages exchanged. This can overwhelm leaders if the message volume is high.

IV. METHODOLOGY

Cell size have to be determined in order to process transactions. The optimal cell size is obtained by data on the throughput of existing log replication through federated learning. The log replication format is different from the traditional raft algorithm.



Arguments	Description
node	number of the node
term	nth term number($T_(n)$)
cell	nth cell number($C_(n)$)
index	log index
timestamp	transaction creation time
generator ID	ID of transaction generator
validation	validation period of data.
payload	log information such as GPS, sensor data, etc

Figure 1: Description of the arguments that required for log.

This will leads to include various elements to be included in a transaction or message and presents a method to cope with various situations to keep the transaction consistent. Change in process of leader election in raft algorithm.

- Initially the servers are in the follower state and values of log arguments are initially set to zero.
- No leader is present in the blockchain network, so the followers can’t receive the heartbeat messages from the leader, so all the server involves in the leader election mode.
- The very first follower who identifies the absence of leader changes then, it is converted to candidate state.

Candidate sends the request vote to the other followers, the term number is incremented. Follower has to options (true/ false), returns the value based on two conditions

- Term Number (candidate) > Term Number (follower)
- Cell Number (candidate) >= Cell Number (follower)

If both the conditions satisfy, follower sends “True”.

In the event that a candidate secures a higher number of votes, they assume the position of the leader. The initial action of the leader involves comparing the hash values of the last committed cell and the current cell among the followers. If these values do not align, the new leader instructs the followers to overwrite and update the information stored by the newly elected leader. This algorithm triggers the servers to enter a new leader election mode, resulting in an increase in both the term and the cell numbers. Consequently, the cell number is reset whenever a new leader is elected, as re-election could introduce potential errors in the system. After leader election, the actual transactions are processed and CBR algorithm proceeds with the log replication process. With the introduction of cells to group transactions, alleviating leader-related bottlenecks, exchanges efficiency by committing to the state machine in cell-sized units. Pre-determines cell size for all participating servers in the CBR consensus algorithm, that attain optimal cell size via federated learning, using log replication data.

The GDPR empowers EU citizens to control their data, prompting multinational companies, including those using zar technology, to enhance transparency in managing customer information. Blockchain emerges as a solution for increased control and protection of personal data, even in transportation-related information. Its inherent inflexibility and transparent network design make it a formidable solution for ensuring data security. Acting as a distributed data system or shared ledger, blockchain records irreversible transactions without the consent of network participants.

To overcome the challenge, the study suggests dividing the system into cells within an acceptable range, reducing complexity without additional protocols. The next section reviews previous research on blockchain systems, raft consensus algorithms, and federated learning for smart data markets. Finally, the fifth section summarizes the paper and suggests future research directions.

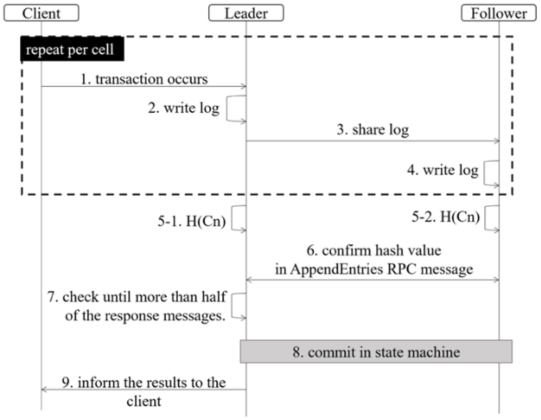


Figure 2: Flowchart of changed leader election protocol in the consensus algorithm.

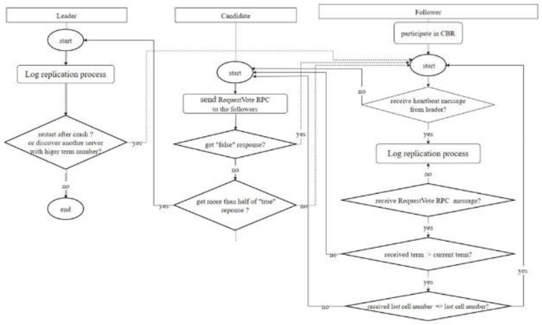


Figure 3: Sequence diagram of the cell-based raft consensus algorithm.

V. IMPLEMENTATION

Define the Components:

- Nodes/Peers: Each participant in the network will be a node or peer.
- Leader Election Module: Raft involves a leader-follower model. Implement the leader election process.
- Log Replication Module: Ensure that the logs are replicated across all nodes.
- Consensus Module: Implement the core consensus algorithm based on Raft.

```
src
|-- main
|   |-- java
|       |-- blockchain
|           |-- Node.java
|           |-- RaftConsensus.java
|           |-- LogEntry.java
|           |-- NetworkLayer.java
|           |-- ElectionTimer.java
|           |-- BlockchainClient.java
|-- test
    |-- java
        |-- blockchain
            |-- RaftConsensusTest.java
            |-- NetworkLayerTest.java
            |-- BlockchainClientTest.java
```

Initially, all servers are in the follower state with log argument values set to zero in the CBR blockchain network. No leader is present in the network, and followers are unable to receive heartbeat messages. Consequently, all followers transition to leader election mode. Upon detecting the absence of a leader, a follower shifts to the candidate state with the intention of becoming the new leader.

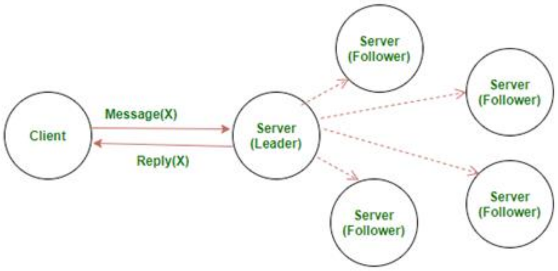


Figure 4: Servers receiving messages from client.

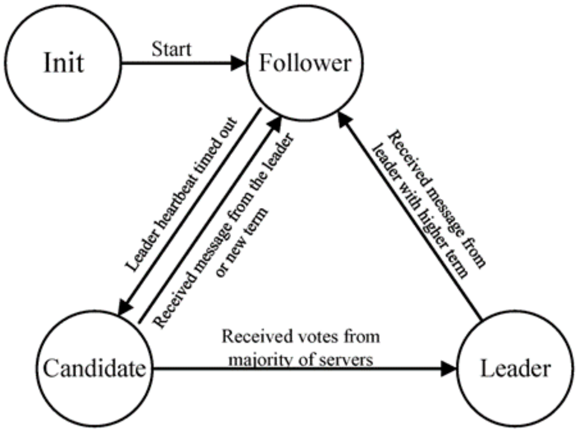


Figure 5: Leader election process of raft consensus algorithm.

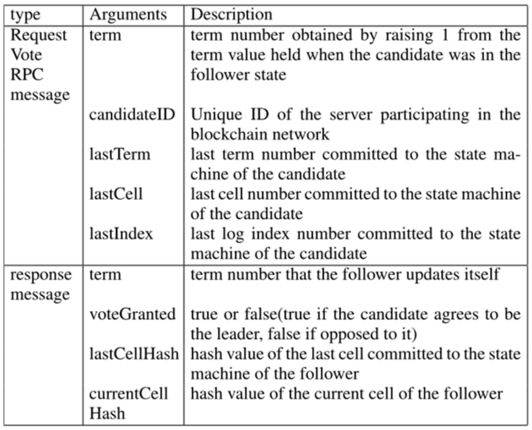
The blockchain network can be broadly categorized into two types: public blockchain and private blockchain, each serving distinct purposes. In a public blockchain, there is no restriction on the number of anonymous nodes that can participate. However, all nodes communicate securely through encryption. Participants in a public blockchain are encouraged to abide by the terms of the agreement to ensure the best possible outcome.

Mining a block in a public blockchain will take a lot of time due to the involvement of unknown users. As a result, public blockchains are not suitable for environments that require agility. In contrast, a federation or private blockchain reaches a much faster consensus of only authenticated nodes.

In this research, a cell-based raft (CBR) mechanism, derived from the existing raft algorithm, is employed for designing a blockchain system. The system's performance is assessed by comparing it with the conventional raft system. Unlike the Raft algorithm in traditional distributed systems, where altered values are stored in logs, the Raft algorithm in blockchain systems retains a log in a transaction line or block, persisting through the consensus process.

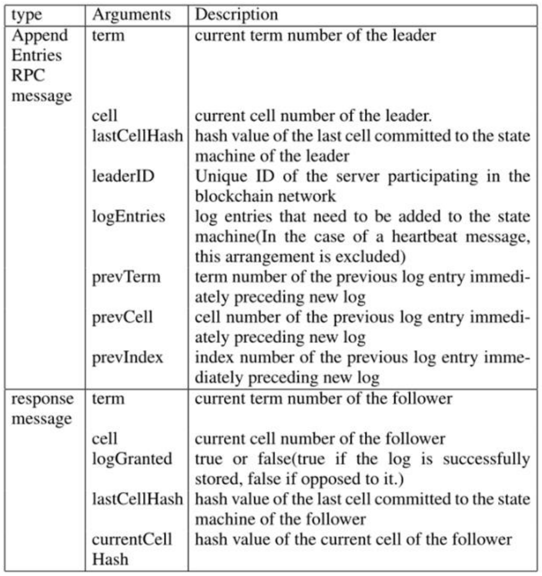
To implement CBR, virtual nodes are developed through an open-source platform, as it was not possible to obtain and agree on more than five physical devices for testing The Raft algorithm is well known and used on projects on GitHub. A Java-based project was chosen for this study because of its stability and advanced features. The project did not require any additional protocols and used pre-defined libraries. The hash logs stored in the cell are generated using the SHA-256 function.

The simulation scenarios used for implementation are described in detail. The original Raft algorithm has been implemented by researchers in various languages and tools. This study used a Java-based project from the official RAFT environments, as it provided a stable environment and necessary functionality without the need for additional libraries.



type	Arguments	Description
Request Vote RPC message	term	term number obtained by raising 1 from the term value held when the candidate was in the follower state
	candidateID	Unique ID of the server participating in the blockchain network
	lastTerm	last term number committed to the state machine of the candidate
	lastCell	last cell number committed to the state machine of the candidate
	lastIndex	last log index number committed to the state machine of the candidate
response message	term	term number that the follower updates itself
	voteGranted	true or false(true if the candidate agrees to be the leader, false if opposed to it)
	lastCellHash	hash value of the last cell committed to the state machine of the follower
	currentCell Hash	hash value of the current cell of the follower

Figure 6: Description of the arguments for request vote in raft consensus algorithm.



type	Arguments	Description
Append Entries RPC message	term	current term number of the leader
	cell	current cell number of the leader.
	lastCellHash	hash value of the last cell committed to the state machine of the leader
	leaderID	Unique ID of the server participating in the blockchain network
	logEntries	log entries that need to be added to the state machine(In the case of a heartbeat message, this arrangement is excluded)
	prevTerm	term number of the previous log entry immediately preceding new log
	prevCell	cell number of the previous log entry immediately preceding new log
	prevIndex	index number of the previous log entry immediately preceding new log
response message	term	current term number of the follower
	cell	current cell number of the follower
	logGranted	true or false(true if the log is successfully stored, false if opposed to it.)
	lastCellHash	hash value of the last cell committed to the state machine of the follower
	currentCell Hash	hash value of the current cell of the follower

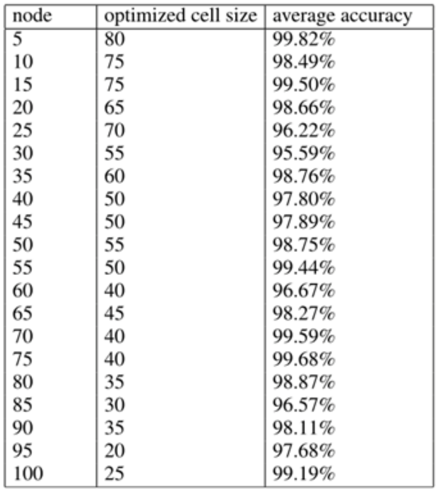
Figure 7: Description of the arguments for append entries in raft consensus algorithm.

VI. RESULT ANALYSIS

This study focuses on achieving a consensus on the cells and evaluates the performance of the cell-based raft (CBR) algorithm through various simulations Compared with the traditional raft algorithm, the study shows that they form a network of strong and efficient.

The report Raft algorithm faces challenges as the number of nodes increases, so leaders have to deal with the increasing number of messages from followers that shows the increase in the number of messages that leaders have to process as the number of nodes changes Raft algorithm-based blockchain process increments To validate behavior, As the number of nodes increases from 5 to 100, the AppendEntries RPC messages are measured. The figure illustrates a gradual increase in AppendEntries RPC messages, reaching a maximum with constant continuity. In contrast, the traditional Raft algorithm demonstrates a decrease in Transaction Per Second (TPS) as the number of nodes increases. For example, in a traditional raft-based blockchain with 75 nodes, the TPS is 45.43 to 43.90. This decline indicates that the load-balancing issue of a candidate in a traditional raft-based blockchain can have a significant impact on the overall performance of the system.

To overcome this problem, the proposed CBR algorithm determines an acceptable cell size for the blockchain consensus process. The study measures the number of messages failed by TPS in a 70-node CBR-based blockchain by increasing the cell size to the maximum number of statically adjustable nodes, starting from cell size 10 and gradually increases by 5. TPS Initially increases with cell size, but when the cell size reaches 45, the TPS decreases rapidly This decrease occurs because as the cell size increases the follower's logs are stored in the same order as the log of the leader. Before a cell is committed, participants send a failure response to the candidate from an AppendEntries RPC message, delaying the consensus process. When the cell size is 45, the number of failed response messages increases rapidly, indicating that the cell size may not be increased unconditionally. Instead, the optimal cell size for both promoter and prophet adaptation was determined.



node	optimized cell size	average accuracy
5	80	99.82%
10	75	98.49%
15	75	99.50%
20	65	98.66%
25	70	96.22%
30	55	95.59%
35	60	98.76%
40	50	97.80%
45	50	97.89%
50	55	98.75%
55	50	99.44%
60	40	96.67%
65	45	98.27%
70	40	99.59%
75	40	99.68%
80	35	98.87%
85	30	96.57%
90	35	98.11%
95	20	97.68%
100	25	99.19%

Figure 8: Optimised cell size and average accuracy of the node derived through federated learning.

VII. OVERALL CONTRIBUTIONS AND ADVANCEMENTS

Traditional Raft algorithm is efficient but not optimised for real-time smart data markets. Struggles to maintain consistent logs for latest client commands due to limited commit speed in the massive communication of the follower nodes. Aims to tackle issues in blockchain systems using Raft algorithm. Currently the Raft struggles with one transaction per log, causing bottlenecks. With the introduction of cells to group transactions, alleviating leader-related bottlenecks, enhances efficiency by committing to the state machine in cell-sized units.

Pre-determines cell size for all participating severs in the CBR consensus algorithm, that attain optimal cell size via federated learning, using log replication data. The CBR algorithm offers handling real-time transactions within smart data markets by implementing cells to groups transaction and allowing commit to state machine.

CONCLUSION

Blockchain adoption for smarter services in sectors like healthcare is growing, but challenges arise while applying the Raft algorithm, the main consensus method, to environments requiring widespread node participation. This study introduces the Cell-Based Raft (CBR) consensus algorithm, aiming to ease the leader node's load by reducing message traffic. CBR divides transactions into cells without additional protocols, maintaining traditional Raft principles while boosting stability in dynamic smart data markets. The proposed algorithm dynamically adjusts cell sizes through federated learning, enhancing real-world processing efficiency by adapting to varying transaction volumes. Simulations demonstrate its efficacy, decreasing the leader's burden and increasing Transaction Per Second (TPS) through message reduction, all while ensuring communication stability and log sequence integrity.

REFERENCES

[1] S. Nižetić, P. Šolić, D. López-De-Ipiña González-De-Artaza, And L. Patrono, ‘‘Internet Of Things (Iot): Opportunities, Issues And Challenges Towards A Smart And Sustainable Future, ’’J. Cleaner Prod., Vol. 274, Nov. 2020, Art. No. 122877.

[2] A. Wang, A. Zhang, E. H. W. Chan, W. Shi, X. Zhou, And Z. Liu, ‘‘A Review Of Human Mobility Research Based On Big Data And Its Implication For Smart City Development, ’’Isprs Int. J. Geo-Inf., Vol. 10, No. 1, P. 13, Dec. 2020.

[3] M. Yechezkel, A. Weiss, I. Rejwan, E. Shahmoon, S. Ben-Gal, And D. Yamin, ‘‘Human Mobility And Poverty As Key Drivers Of Covid-19 Transmission And Control, ’’Bmc Public Health, Vol. 21, No. 1, Pp. 1–13, Dec. 2021.

[4] P. Voigt And A. Von Dem Bussche, ‘‘The Eu General Data Protection Regulation (Gdpr), ’’In A Practical Guide, 1st Ed. Cham, Switzerland: Springer, 2017.

[5] Z. Zheng, S. Xie, H.-N. Dai, X. Chen, And H. Wang, ‘‘Blockchain Chal- Lenges And Opportunities: A Survey, ’’Int. J. Web Grid Services, Vol. 14, No. 4, Pp. 352–375, 2018.

[6] D. Berdik, S. Otoum, N. Schmidt, D. Porter, And Y. Jararweh, ‘‘A Survey On Blockchain For Information Systems Management And Security, ’’Inf. Process. Manage., Vol. 58, No. 1, Jan. 2021, Art. No. 102397.

[7] W. Fu, X. Wei, And S. Tong, ‘‘An Improved Blockchain Consensus Algo- Rithm Based On Raft, ’’Arabian J. Sci. Eng., Vol. 46, Pp. 8137–8149, Feb. 2021.