A Lightweight Proof of Stake Voting Mechanism with Byzantine Agreement and Cryptographic Sortition for Telemedicine Systems

Denis Wapukha Walumbe
Department of Information Technology, Murang'a University of Technology, Kenya

Gabriel Ndung'u Kamau
Department of Information Technology, Murang'a University of Technology Murang'a, Kenya

Jane Wanjiru Njuki
Department of Information Technology, Murang'a University of Technology, Murang'a, Kenya

ABSTRACT
With the rising integration of blockchain in critical domains such as healthcare, designing efficient, lightweight, and privacy-preserving consensus mechanisms remain a significant challenge. Existing Proof-of-Stake (PoS) implementations often incur high computational and communication overhead, making them unsuitable for telemedicine systems. This study proposed LightweightPoS, a novel voting mechanism designed for this environment. The proposed mechanism incorporates a cluster-based voting to minimize message complexity, Byzantine Agreement protocol for robust fault tolerance and cryptographic sortition to ensure fairness and privacy. This implementation slashes global communication, reducing message complexity by over 95% compared to traditional PoS models. The study evaluated the proposed and baseline mechanisms through simulations using real-time telemedicine data sensors. The results demonstrated that the proposed mechanism consistently achieved sub-10ms latency, high transaction throughput (up to 2400 TPS) and low energy consumption (~0.002kWh per round). It significantly outperformed baseline mechanism like Algorand and Ouroboros. Furthermore, the system included an effective Byzantine node detection, ensuring reliability under adversarial conditions. This work contributes a practical consensus voting mechanism that balances privacy and regulatory compliance. It provides a robust foundation for deploying blockchain technology in privacy-sensitive telemedicine applications.

General Terms
Privacy, Algorithms, Performance, Blockchain

Keywords
Blockchain, Consensus, Proof-of-Stake, telemedicine systems, Byzantine agreement, Cryptographic sortition

1. INTRODUCTION
Telemedicine systems depend on distributed networks for real-time sharing of data and decision making. However, they face significant challenges in ensuring data privacy. Those networks require consensus mechanism that are low-latency, energy-efficient, and preserves privacy especially for resource constrained devices. While Proof of Stake (PoS) reduces the need for the high computational power demanded by Proof of Work (PoW), it still does not adequately meet the requirements for small, resource-constrained devices such as those in telemedicine systems [1]. Telemedicine systems require low-latency, energy-efficient, and fault-tolerant consensus protocols that ensure data privacy. The current PoS are not designed or optimized for lightweight devices. It results in high energy consumption and computational overhead.

To address these challenges, this study developed a novel consensus mechanism termed LightweightPoS. This mechanism incorporates a modified BA protocol with cryptographic sortition. BA protocol serves as the core voting technique to achieve consensus among distributed nodes. The modified BA protocol that lowers computational and communication complexity while ensuring data privacy. Cryptographic sortition ensures fairness and enhances validator privacy. It further integrates cryptographic sortition to enhance validator selection and data privacy [2]. Overall, the study presents a voting mechanism that makes PoS lightweight suitable for telemedicine systems. Through empirical evaluation, the study has demonstrated improved performance of PoS model in telemedicine system scenarios.

Blockchain is a revolutionary technology that began with Bitcoin enabling a peer-to-peer digital currency network [3]. Over decades, Bitcoin has grown resulting in a digital reserve estimated to be more than a trillion dollars [4]. Ethereum came into play and extended the concepts of Bitcoin in creating programmable currency with smart contracts. Smart contracts have resulted to many applications [1]. Inspired by the smart contracts, the technology was developed, advanced and spread in many fields, such as industry [5], agriculture, healthcare, administration [6], smart cities [7], and Internet of Things network. In blockchain technology, data storage, cryptography, consensus models and architecture are the core features[3,5]. Consensus models the core of blockchain and affects the efficiency, privacy, security and stability of ecosystem. Therefore, it is necessary to study consensus algorithms if blockchain technology is to be deployed in low powered devices such as telemedicine systems.

The consensus algorithm is responsible for privacy and stability of the operations in the system. It is the core of the blockchain technology where it dictates the procedures of accountability among the nodes, how a new block is generated and validated; and how transaction fee is earned [8]. Proof of Work (PoW) presented rewards in terms of Bitcoins for solving complex mathematical problem. The PoW became popular in numerous applications [9]. However, the increasing popularity of PoW exposed the drawback of the technology, that systems do not scale and it is not fully decentralized [9]. Computational resource demand as the problem became more complex presented another limitation [10]. Overall, the high computational power has resulted to high carbon footprint. Proof of Stake was proposed to overcome the drawbacks of PoW [6], [11] where scalability and computational demands

were addressed.

PoS systems use validators for transaction processing. To become a validator in PoS system, the participants lock stake or token. The state of the ledger is decided by the selected validators through the consensus process [2]. The consensus process is where many nodes that do not trust each other arrive at an agreement on the validity of the block. When a new block is added, i.e., successfully added to the chain of blocks, stakes are forged to reward the validators for processing the transactions. The reward is measured as stake or the wealth that a node can use to transact. Stake is an important element in PoS systems since the stability of the system depends on the stakes.

2. RELATED WORK

Consensus algorithms are the core functionality of blockchain technology and impact its implementation. The study discussed improvements in the algorithms underpinning PoS voting mechanism from literature. They all try to improve different aspects of PoS by giving their solutions to improve the specific aspects. However, some scholars have focused on different aspects of consensus algorithm apart from voting mechanism.

Luo et al. [12] proposed a two election processes where one selects the representatives and second election selects the winner. In the first election, network participants vote to select a group of representative nodes responsible for validating transactions and maintaining the ledger. In the second election, these representatives vote among themselves to select the final leader, responsible for generating new block. The proposed modification to improve on DPoS consensus mechanism with aim of improving decentralization. The two election system enhances decentralization reducing concentration of power as witnessed in traditional DPoS.

Xu et al. [13] explored DPoS as well in efforts to improve its decentralization. The scholars proposed a concept of virtual stake for keeping track of truthful witness votes during voting. The approach provided a motivator for the voters to remain truthful since at the end of the round there was a reward. The virtual stake helps in determining the leader with the aim of discouraging misbehaving witnesses from submitting blocks. However, the voters retain the rights to give portion of their votes to any candidate witness.

Delegated Proof of Stake with downgrade DDPoS was put forward by Yang et al [14]. The main goal of this was to address the issues with malicious witness behavior in DPoS algorithm. DDPoS uses the concept of Proof of Work (PoW) where computational power instead of stake is used for election of nodes. Downgrade mechanism is used to quickly downgrade malicious nodes in the system. One vote per node is also introduced in this approach to ensure fairness of the voting process.

Chen et al [15] proposed improvement on the voting mechanism of PoS by introduction of vague set and node impact factor. The solution aimed at single vote voting mechanism in PoS and fixed total nodes issues. The node's voting rate is determined by computing neighboring nodes voting status and the node's impact factor. The authors proposed the use of fuzzy value computation approach in addition to establishing the sum number of the agent nodes [16]. It is aimed at improving voting and participation of the nodes on a network.

The scholars above addressed various issues in voting mechanism for PoS such as decentralization, fairness, enthusiasm of nodes in the voting process, and malicious node selection. The researchers noted that the discussions centered around decentralization, privacy and enthusiasm of nodes to participate in voting mechanisms are critical. The researchers noted that most of the proposed solutions increases the computation needs of PoS voting mechanism. The concept of low computation, energy constrained and low memory devices was not considered in the design proposals. Therefore, most of the proposed solutions are not suitable for telemedicine systems.

Byzantine agreement (BA) protocols are important in achieving consensus in a distributed system where node's behaviors maybe malicious or unpredictable. BA is based on traditional theories like exponential information gathering (EIG), the King algorithm and Ben-Or's randomized protocols [17]. The theories guarantee termination, consensus and validity but faces drawbacks in high communication and round complexity [18]. EIG requires an exponentially large exchange of messages, where the King algorithm depends on partial synchronous to achieve faster agreement. Randomness has been implemented in Ben-Or's protocol to reduce communication overhead to realized faster agreement.

Although the traditional protocols guarantee termination, they are often impractical in dynamic or large networks because of the $O(N^2)$ communication complexity, lack of scalability and multi-round dependency during which messages can be lost.

Recent studies have used BA protocols in IoT and embedded systems. In such environment, nodes face constraints in memory, energy and computation resources [1]. The application of BA in Proof-of-Stake have emerged as feasible alternatives. There is a tradeoff between consistency or fault tolerance to gain in latency and energy efficiency [13]. The tradeoff raises critical questions on privacy and trustworthiness of consensus when deployed in constrained resource environment. There is need to balance not only energy and latency but throughput and fault tolerance with regards to capabilities of the devices.

Telemedicine systems increasingly depend on distributed systems that facilitate real-time diagnostics, regular monitoring of patient, and remote care [19], [20], [21], [22]. These systems entail network of small, mostly mobile devices like the blood glucose monitors and wearable vital trackers. the environment that they are operating is unreliable with potential failures.

The client-server models have struggled in maintaining fault tolerance, availability, and privacy. Consequently, blockchain technology has been explored in ensuring trustless coordination, enforcing data immutability and reduce dependency on traditional data storage [23]. Nonetheless, there are challenges regarding integration of consensus mechanisms that are lightweight. This is more complicated where there is need to maintain privacy with strict regulatory and privacy constraints.

The related work establishes that while significant research there has focused on improving consensus mechanism, a gap remains. These solutions are not designed for resource constrained environment such as telemedicine systems. This work directly addresses this gap by proposing a lightweight consensus mechanism operating efficiently on low-power devices while maintaining robust privacy.

3. RESEARCH METHODS

3.1 Introduction

The proposed solution was developed through a systematic process that involved designing the core artifacts, refining them

iteratively, and performing functional tests. The tests were to ensure proper operation. After confirming that the mechanism worked as intended, the researchers conducted comparative simulations to evaluate the and validate the Lightweight Proof-of-Stake(LightweightPoS) voting mechanism against the existing approaches. It was measured against two established PoS protocols: Algorand and Ouroboros Praos.

3.2 Goal of the simulation

The experiments were designed to:

1. Quantify performance improvements in a resource-constrained telemedicine environment
2. Validate theoretical claims about communication and computational efficiency
3. Assess privacy preservation capabilities
4. Measure sustainability under varying network conditions.

3.3 Simulation Environment

The following describes the hardware and software simulation environment.

3.3.1 Hardware configuration

The following were hardware specification where the experiment was conducted.

Table 1. Hardware configuration for simulation

Item	Specification
Processors	Intel Core i7-10700k CPU @ 3.80GHz
Memory	32 GB DDR4 RAM
Storage	1 TB NVMe SSD
Graphics	NVIDIA GeForce RTX 2070 Super
Operating System	Ubuntu 22.04 LTS (64-bit)
Windows with virtual environment for Ubuntu	

3.3.2 Software stack

The simulation environment comprised:

1. Bevywise IoT Simulator: emulated 100-100 wearable telemedicine devices with configurable energy profiles, memory constraints and computational capabilities.
2. MQTT protocol: a lightweight messaging protocol for device communication used CrystalMq broker
3. Data pipeline: synthetic medical data generation, such as heart rate, temperature
4. Blockchain implementation- custom lightweight PoS protocol (Python)
5. Baseline implementations: Algorand and Ouroboros are adapted from open-source.

3.3.4 Experimental Variables

Independent Variables

1. Network size: 50-1100 nodes (both fixed and randomized configurations)
2. Node distribution: pre-configured topologies (50,70,90,150 nodes) and randomized topologies (65,165,174 nodes)
3. Byzantine nodes ratio: 5%, 10%, 20% of total nodes
4. This device emulated Block parameters: maximum of 100 transactions per block and 50% local vote threshold for cluster consensus

Dependent variables (performance metrics)

1. Throughput: transactions per second (TPS), Block rate, and Latency
2. Energy efficiency: kWh per block processed and battery drain per consensus round
3. Resource consumptions: network messages per block
4. Privacy metrics: byzantine node detection rate and privacy compliance scores (PCS).

3.4 Testing Scenarios

Objective: to establish performance benchmarks under controlled conditions

Configuration:

1. Fixed node counts (50,70,90,150)
2. 0% Byzantine nodes
3. 5 simulated rounds per configuration

Scenario 2: sustainability and scalability testing

The objective was to evaluate performance degradation with network growth

Configuration:

1. Randomized node counts (65,165,174)
2. Variable Byzantine ratios (65,165,174)
3. 3 simulated rounds per configuration

Scenario 3: Fault Tolerance (malicious attack)

Objective: to measure consensus reliability under adversarial conditions

Configuration:

1. Fixed 150-node network
2. Byzantine ratios from 5% to 33%
3. Measurement of: detection accuracy, false positive rate, consensus success rate

Scenario 4: Privacy compliance

Objective: Quantify HIPAA/GDPR alignment based on weighted measures for data exposure, access control, data compression, and cryptographic analysis of leader selection.

3.5 Data Collection and Analysis

An automated logging system was implanted in Python. Key performance metrics were aggregated using the Pandas library and subsequently visualized with Matplotlib. A custom front end dashboard with real time monitoring of the simulation was developed.

3.6 Validation Approach

The experimental results were validated using baseline comparison against Algorand and Ouroboros Praos. Quantified performance of the proposed solution and the baseline were compared under same conditions where quantified performance improved under controlled conditions.

3.7 Ethical considerations

The following were ethical considerations:

1. Synthetic patient data were generated to avoid real patient exposure
2. Baseline implementation used unmodified open-source code for fair comparison
3. Energy measurements accounted for virtualization overhead.

3.8 Limitation of the study

The following were limitations of the study.

1. Simulation against real-world deployment variance-the experiments were conducted in a controlled simulation instead of actual telemedicine hardware. Factors such as unstable network conditions, device heterogeneity and hardware-specific constraints were not fully captured.
2. The study used a fixed cluster size in initial experiments – cluster size rigidity may not be optimal for a highly dynamic network. Randomized cases were used to overcome this challenge.

4. PROPOSED MECHANISM

This section presents the core design and operational principles for the proposed lightweight voting mechanism designed for telemedicine environment. It discusses the features and the voting techniques that contributes to the lightweight mechanisms nature. The aim is to show how these choices collectively support privacy preserving, efficient, and reliable consensus in telemedicine.

4.1 Features of proposed mechanism

The proposed lightweight voting mechanism presents a solution to the inefficiencies of the existing PoS-based voting mechanism. It introduces cluster-based pre-consensus, Byzantine agreement BA and cryptographic sortition techniques. The proposed solution has the following key features

4.1.1 Cluster-Based Pre-Consensus

The nodes are grouped into clusters on the basis of network topology or proximity. Clusters vote to get a leader that will participate at global voting and the local consensus is reached within the clusters. This reduces the number of messages exchanged globally on the network. Hence minimizing communication overhead.

4.1.2 Byzantine Agreement with Cluster leaders

The cluster leaders participate in the Byzantine Agreement process to reach a global consensus. BA offers high tolerant where even if some cluster leaders are faulty, the system can still reach a consensus. With cluster leaders' quick finality is achieved hence reducing computational demand on small devices.

4.1.3 Cryptographic sortition

To ensure fairness and privacy, validators are elected using hash-based lottery sortition (HBL) in the voting process. This technique is to prevent malicious nodes from predicting or manipulating the selection process, enhancing privacy of the system.

4.2 Election Process

There are 3 rounds that are Cluster-Based Pre-Consensus voting, BA voting and Final voting in the proposed voting mechanism.

4.2.1 Cluster-Based Pre-Consensus

The election process is designed to cluster nodes into groups based on proximity. The aim is to reduce overall message communication within the network. Reduces message overhead from O (N^2) to O($n+k^2$). Allow each cluster to select a leader that will participate in global voting. To rapidly and effectively complete the election process, the network categorizes the number of validators into clusters. The network can only Partition NN into kk clusters. The total number of the nodes on the network must satisfy the following condition

Condition 1:

The total summation of the partition for clusters is given as:

$$\bigcup_{i=1}^{k} C_i = N, \qquad C_i \bigcap C_j = \emptyset \; \forall \; i \neq j$$

Condition 2:

For each node μ∈ Ci:

Verify T and compute vote v_(u)∈ {0(REJECT),1(ACCEPT)}

Broadcast (μ, v_(u))to Ci

Explain:

Verify(T): lightweight checks for the timestamp freshness and validation signature

The binary votes 1 for accept and 0 for reject minimizes the bandwidth for better performance on lightweight systems. The intra-cluster broadcast scope save energy instead of global flooding with messages.

condition 3:

For cluster Ci

$$\text{Decision}_i = \begin{cases} 1 \; if \; \sum_{u \in C_i} \left(v_u \geq [\frac{C_i}{2}] \right) \\ 0 \qquad otherwise \end{cases}$$

Condition 4:

Leader Election via HBL

The voting decision is encrypted before broadcasted.

For C_i, elect leader $Leader_i$ = arg min$u \in C_i$ HBL (sku, α)

Explanation

The secrete key sku and the shared randomness α serve as inputs to the HBL.

The output is Proof π and random value y: leader = node with last y

For privacy, the Byzantine nodes cannot manipulate leader selection

4.2.2 Byzantine Agreement (BA) Round Voting

Leader Commitment:

Each $Leader_i$ computes:

ZKPi←ZK-SNARK ($Decision_i$, HBL_i)

Broadcast ($Leader_i$, $Decision_i$, ZKP_i) to all leaders.

Explanation:

ZK-SNARK is where the decision of the leaders re computed honestly with revealing full clusters votes using Zero-Knowledge Proof.

The bandwidth efficiency is achieved by only leaders

communicating globally hence less messages.

Vote Validation:

For each received (Leader$_j$, d_j, π_j):

Verify ZKP$_j$ and HBL$_j$

If valid, add d_j to Votes

Byzantine filtering ensures that malicious votes are rejected. Those are votes with invalid proofs or HBL outputs.

Global Consensus:

Compute

$$\text{ConsensusState} = \begin{cases} 1 \; if \; \frac{\sum_{d_j \in Votes} d_i}{k} \geq Q \\ 0 \qquad otherwise \end{cases}$$

The global consensus is achieved with quorum Q value for the vote to be valid.

Agreement:

If $\sum d_j$ =ConsensusState$\geq \lceil Q \cdot k \rceil$, finalize ConsensusState.

Quorum Q is achieved at 76% to ensure that the system tolerates up to f< n/3 faults.

Termination if Q is greater or equal to k leaders agree, voting protocol finalizes

4.2.3 Final Decision

The final decision is based on the consensus state either accepted or reject.

$$\text{Finalize}(T) = \begin{cases} \text{ACCEPTREJECT if ConsensusState} = 1 \\ REJECT \qquad otherwise \end{cases}$$

4.3 Byzantine Agreement Voting Algorithm

This solution is suitable particularly to telemedicine where computational power, and storage are limited. To reduce the message overhead further, the proposed solution incorporated clustered voting into the BA voting mechanism. The solution is particularly designed for telemedicine systems. The algorithm for the proposed solution is shown:

Algorithm 1: Voting for Block Validator with Byzantine Agreement

Output: Consensus Decision (ACCEPT or REJECT)
START: Consensus process begins
Initial Setup:
1: Input:

- ✓ Network Nodes: N={N_1,N_2,…,Nm}
- ✓ Clusters: C={C_1,C_2,…,C_k}
- ✓ Stakeholders: S={S_1,S_2,…,S_k} (nodes with staked tokens)
- ✓ Transaction: T (proposed block data)
- ✓ Quorum: Q (e.g., 67%)
- ✓ Rounds: R (voting rounds)
- ✓ Cryptographic Sortition Function: HBL()

Step 1: Clustered Voting (Local Consensus)
2: For each cluster Ci ∈C do
3: Each node Nj∈Ci verifies T and votes ACCEPT or REJECT.
4: ≥50% of nodes in Ci vote ACCEPT then
5: ClusterDecisioni←ACCEPT
6: Else
7: ClusterDecisioni←REJECT
8: End If
9: End For
10: For each cluster Ci ∈ C do
11: Select leader Li via HBL:
- Each node computes HBL(stake,seed).
- Node with lowest HBL score becomes LiLi.
12: Li commits to ClusterDecisioni using ZKP.
13: End For
Step 2: Byzantine Agreement (Global Consensus)
14: For each leader Li do
15: Broadcast ClusterDecisioni to other leaders.
16: End For
17: If ≥Q (e.g., 67%) of leaders propose ACCEPTACCEPT then
18: GlobalConsensus←ACCEPT
19: Else
20: GlobalConsensus←REJECT
21: End If
Step 3: Final Decision
22: If GlobalConsensus==ACCEPT then
23: Append T to blockchain.
24: Else
25: Discard T.
26: If R>0 then
27: Decrement RR and restart consensus.
28: End If
29: End If
END

---

5. EMPIRICAL EVALUATION

To evaluate the proposed lightweight voting PoS mechanism, a telemedicine environment was set up using Bevywise IoT simulator. The simulation mimicked real-world environment by modeling telemedicine systems both wearable and embedded devices. The devices communicated over the MQTT protocol. There were two sets of node configuration, randomized and non-randomized [27]. Randomized is where the network simulator picks varying number of nodes while non-randomized is where the nodes were pre-configured. The goal was to assess performance of the network under different operational loads with varying node and structured topologies.

The key performance metrics such as latency, transaction throughput (TPS), processing time, block generation rate, and energy consumptions were captured and analyzed[28]. To ensure stability and repeatability, each experiment consisted of multiple rounds. The baseline algorithms used for comparison were Algorand and Ouroboros Praos, implemented using publicly available source code for PoS and simulated under same simulator.

5.1 Performance Evaluation

This section presents the results of comprehensive evaluation comparing: (1) the proposed lightweight PoS voting mechanism, (2) Algorand and (3) Ouroboros Pras. Algorand and Ouroboros Praos used the legacy code available on GitHub account. The algorithms were written in Python where they mined blocks from telemedicine system. The message from the simulator was received through the MQTT broker. The performance was measured using different parameters such as battery drain, throughput, latency, block creation rate and Transaction per Second. The data below shows the simulation parameters configurations for the study simulation.

Table 2. Averaged metrics (5 rounds)

Nodes	Algorithm	TPS	Latency (s)	Block Rate
50	LightweightPoS	2324.90	0.003	0.461
	AlgorandLight	298.92	6.855	0.375
	OuroborosLight	453.77	8.324	0.162
70	LightweightPoS	2426.27	0.007	0.119
	AlgorandLight	80.78	11.335	0.077
	OuroborosLight	74.20	13.213	0.114
90	LightweightPoS	1909.80	0.001	0.044
	AlgorandLight	257.08	8.790	0.030
	OuroborosLight	678.26	10.425	0.044
150	LightweightPoS	1864.57	0.001	0.097
	AlgorandLight	377.44	8.286	0.064
	OuroborosLight	298.99	11.636	0.097

Table 2 shows the summary of experiment 1 with 5 rounds of each experiment configured as per the number of nodes. LightweightPoS maintains low latency below 10ms and high 1800 TPS up to 150 nodes, this demonstrating strong scalability.

The lightweight Proof-of-Stake (LightweightPoS) developed in this study was designed with focus on maximizing performance for blockchain telemedicine systems. The core features of blockchain; voting mechanism, cryptographic framework, architectural design, and storage are impacted by the key metrics like latency, energy efficiency, transaction per second (TPS) and block creation rate.

Randomized node results

To test the scalability and stability, experiments were based on number of nodes to assess how the proposed solution performed.

Table 3 Randomized experiment results

Nodes	Algorithm	TPS	Latency (s)	Processing Time (s)	Block Rate
165	Lightweight PoS	1210.20	0.0041	0.0052	0.0781
165	AlgorandLight	178.34	7.81	9.74	0.0438
165	OuroborosLight	272.72	10.08	12.60	0.0780
65	Lightweight PoS	2019.48	0.0033	0.0041	0.1643
65	AlgorandLight	222.62	5.03	6.27	0.0917
65	OuroborosLight	154.97	6.43	8.03	0.1641
174	Lightweight PoS	1121.23	0.0044	0.0055	0.1594
174	AlgorandLight	95.89	3.92	4.88	0.0892
174	OuroborosLight	230.13	5.05	6.31	0.1592

Table 3 provides a snapshot of average results across the three randomized node experiments (165,65 and 174 nodes). LightweightPoS steadily outperformed the baseline algorithms in all measured metrics. It is observed that LightweightPoS consistently attains the lowest latency and highest TPS. They are critical metrics for real-time decision making in telemedicine systems such as emergency situations.

5.1.1 Metrics-Based Evaluation

The metrics were computed and presented in visual form, as well as data exported from JSON format to CSV using the Pandas library in Python.

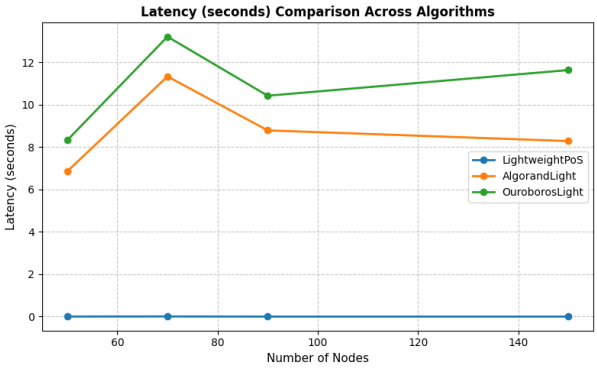

Figure 1: Latency performance for algorithm comparisons

5.1.1.1 Throughput

The performance of the proposed algorithm was evaluated against the established PoS algorithm Algorand [26] and Ouroboros Praos [27]. The proposed algorithm has been evaluated on performance for throughput. In blockchain systems, throughput is measured by how work is done in a given time. It was measured by TPS, latency, and block rate. Transactions per second (TPS) is the total number of successful transactions per second. Figure 1 shows the throughput of the proposed consensus algorithm, which shows that the number nodes results to a linear change in TPS in all three algorithms. TPS linearly increases with network size. LightweightPoS reaches over 2400 TPS in small networks.

5.1.1.2 TPS

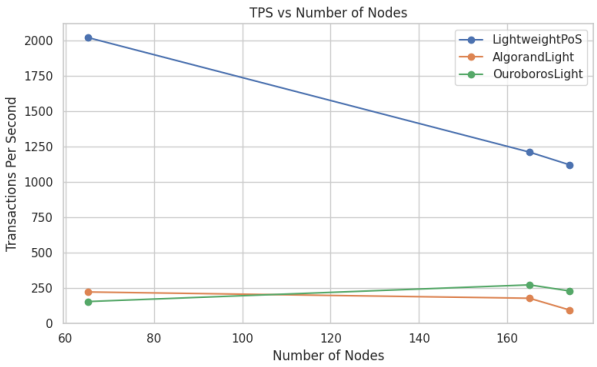

Figure 2: Average TPS for randomized nodes

The results in figure 2 show high TPS for proposed LightweightPoS compared to Algorand and Ouroboros. Although there is a decline from 65 to 165, the decline

stabilizes over time. Showing the stability of the proposed solution to handle network dynamic changes. The network moved from 165 nodes to 65 then 174 nodes but the results are consistent.

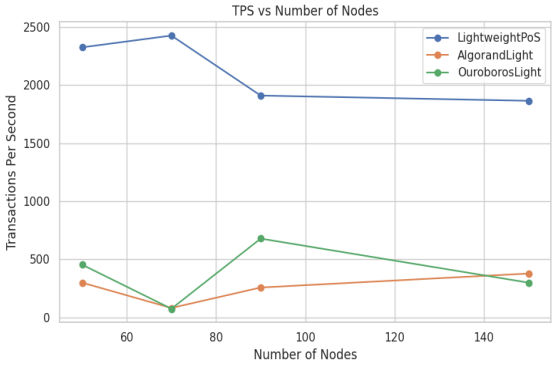

Figure 3: Pre-selected node experiment results for TPS

The TPS performance shown in Figure 3 shows that the proposed algorithm is faster than Algorand and Ouroboros Praos, reaching up to 2400 transactions per second.

As per [25] TPS is a good metric to measure the blockchain. The cluster-based voting mechanism ensures that there is a local-first design for faster latency, hence maintaining high TPS, outperforming traditional PoS models that depend on global messages.

5.1.1.3 Block Rate

This metric refers to how frequently new blocks are added to the blockchain in a given time. It is a critical performance metrics since it reflects the network responsiveness, efficiency of consensus and system scalability.

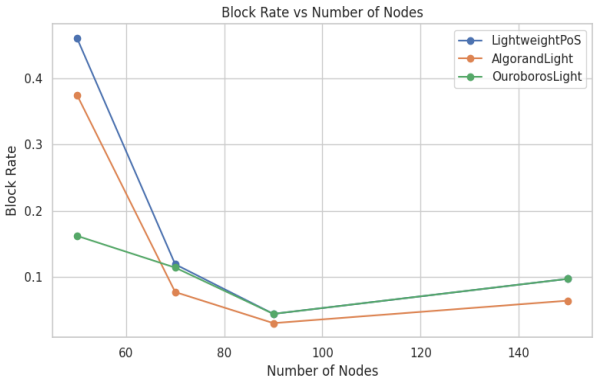

Figure 4: Comparison of rate of block generation

Figure 4 shows the generation of blocks by the three algorithms. The results shows consistency of the proposed algorithm with the existing ones. The performance on block generation does not vary significantly.

5.1.2 Communication overhead (messages/ network load)

This is a measure of resource consumption. It is in terms of the messages or data packets exchanged during consensus. Low communication messages imply less energy, CPU usage and bandwidth required. The messages exchanged per block mined were counted. The simulations align with the complexity simplification $0(n + k^2)$.

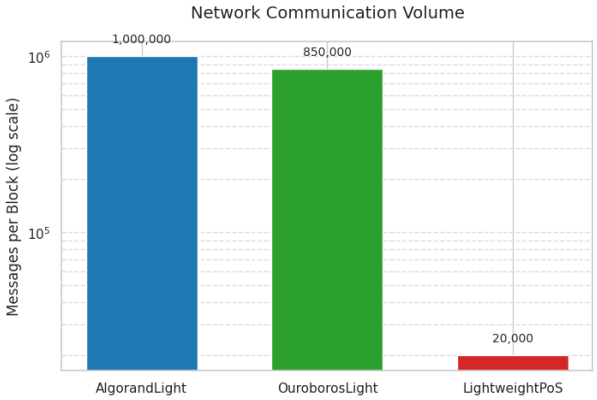

Figure 5: Message communication per block generated

LightweightPoS reduced message volume from 1,000,000 in traditional PoS to about 20,000 messages by implementing cluster-based communication.

Table 4. Summary of Results for Resource Consumption across node sizes

Metric	AlgorandLight	OuroborosLight	Lightweight PoS
Energy (kWh)	0.0044	0.0041	0.0020
Battery Drain (%)	0.22	0.21	0.17
Communication Volume	1,000,000 msgs	850,000 msgs	20,000 msgs

The resource consumption analysis shows that LightweightPoS is significantly more efficient in all node configurations. It consumes 50% less energy, drains less battery life per block mined, and reduces message complexity by about 90%. The results affirm the suitability of the proposed solution in low powered telemedicine systems where minimal resource utilization is critical for longevity of the devices and patient safety.

6. CONCLUSION AND RECOMMENDATIONS

The primary goal of the study was to develop a lightweight voting mechanism suitable for resource-constrained telemedicine systems. The aim was to ensure a secure, efficient, and privacy-preserving blockchain solution. Through the design and implementation of the Lightweight Proof-of-Stake (LightweightPoS) protocol, this goal was successfully achieved.

The mechanism introduced a cluster-based consensus technique leveraging on Byzantine Agreement (BA). Additionally, cryptographic sortition and zero-knowledge proofs were implemented for enhanced privacy and fairness. The goals were to reduce communication and computational overhead associated with traditional PoS protocols. The solution is feasible for devices with limited battery life, memory, and processing power.

In conclusion, the LightweightPoS mechanism addresses the critical need for a lightweight, secure, and privacy-aware consensus solution in telemedicine systems. Through a combination of empirical robustness with a sound theoretical

foundation, the protocol is a viable foundation for blockchain integration in telemedicine systems. It supports privacy-sensitive data exchange, validator efficiency and operates efficiently on constrained telemedicine systems.

Recommendations: Based on consistent performance in all critical metrics, LightweightPoS is a significantly effective consensus mechanism. It is suitable for privacy-preserving, scalable blockchain deployments in telemedicine systems.

Further study: While the current implementation of the LightweightPoS consensus mechanism has provisions for identifying Byzantine nodes, it lacks a punishment and positive reinforcement mechanism. In real-world distributed systems, especially in a sensitive environment such as telemedicine, strategic and sustained cooperation(reward) as well as deterrence (punishment) are important. This is referred to as a reward system. Punishment alone discourages bad behaviors but fails to reward good behaviors.

In the future, the researchers intend to enhance it to incentivize nodes in compatible behavior among network participants, achieving a quick and reliable consensus process.

7. ACKNOWLEDGEMENT

The authors sincerely acknowledge the Department of Information Technology, Murang'a University of technology, for providing research facilities, technical resources and institutional support that made the study possible. The authors appreciate the valuable input and feedback from colleagues and peers during development and review of this work.

8. REFERENCES

[1] I. Abraham and D. Dolev, "Byzantine agreement with optimal early stopping, optimal resilience and polynomial complexity," in *Proceedings of the Annual ACM Symposium on Theory of Computing*, 2015. doi: 10.1145/2746539.2746581.

[2] S. Y. Lin, L. Zhang, J. Li, L. li Ji, and Y. Sun, "A survey of application research based on blockchain smart contract," *Wireless Networks*, vol. 28, no. 2, 2022, doi: 10.1007/s11276-021-02874-x.

[3] S. N. Khan, F. Loukil, C. Ghedira-Guegan, E. Benkhelifa, and A. Bani-Hani, "Blockchain smart contracts: Applications, challenges, and future trends," *Peer Peer Netw Appl*, vol. 14, no. 5, 2021, doi: 10.1007/s12083-021-01127-0.

[4] D. R. Kowalski and A. Mostéfaoui, "Synchronous byzantine agreement with nearly a cubic number of communication bits," in *Proceedings of the Annual ACM Symposium on Principles of Distributed Computing*, 2013. doi: 10.1145/2484239.2484271.

[5] C. Rupa, D. Midhunchakkaravarthy, M. K. Hasan, H. Alhumyani, and R. A. Saeed, "Industry 5.0: Ethereum blockchain technology based DApp smart contract," *Mathematical Biosciences and Engineering*, vol. 18, no. 5, 2021, doi: 10.3934/MBE.2021349.

[6] Y. Wei, M. Xiao, N. Yang, and S. Leng, "Block Mining or Service Providing: A Profit Optimizing Game of the PoW-Based Miners," *IEEE Access*, vol. 8, 2020, doi: 10.1109/ACCESS.2020.3010980.

[7] A. Norta, P. Dai, N. Mahi, and J. Earls, "A public, blockchain-based distributed smart-contract platform enabling mobile lite wallets using a proof-of-stake consensus algorithm," in *Lecture Notes in Business Information Processing*, 2019. doi: 10.1007/978-3-030-04849-5_33.

[8] V. Deval *et al.*, "Mobile Smart Contracts: Exploring Scalability Challenges and Consensus Mechanisms," *IEEE Access*, vol. 12, 2024, doi: 10.1109/ACCESS.2024.3371901.

[9] H. H. M. Mahmoud, W. Wu, and Y. Wang, "Proof of learning: Two Novel Consensus mechanisms for data validation using Blockchain Technology in Water Distribution System," in *2022 27th International Conference on Automation and Computing: Smart Systems and Manufacturing, ICAC 2022*, 2022. doi: 10.1109/ICAC55051.2022.9911156.

[10] A. Angelucci and A. Aliverti, "Telemonitoring systems for respiratory patients: technological aspects," *Pulmonology*, vol. 26, no. 4, 2020, doi: 10.1016/j.pulmoe.2019.11.006.

[11] N. Lasla, L. Al-Sahan, M. Abdallah, and M. Younis, "Green-PoW: An energy-efficient blockchain Proof-of-Work consensus algorithm," *Computer Networks*, vol. 214, 2022, doi: 10.1016/j.comnet.2022.109118.

[12] Y. Luo, X. Deng, Y. Wu, and J. Wang, "MPC-DPOS: An efficient consensus algorithm based on secure multi-party computation," in *ACM International Conference Proceeding Series*, 2019. doi: 10.1145/3376044.3376061.

[13] Y. Xu, X. Yang, J. Zhang, J. Zhu, M. Sun, and B. Chen, "Proof of engagement: A flexible blockchain consensus mechanism," *Wirel Commun Mob Comput*, vol. 2021, 2021, doi: 10.1155/2021/6185910.

[14] F. Yang, W. Zhou, Q. Wu, R. Long, N. N. Xiong, and M. Zhou, "Delegated proof of stake with downgrade: A secure and efficient blockchain consensus algorithm with downgrade mechanism," *IEEE Access*, vol. 7, 2019, doi: 10.1109/ACCESS.2019.2935149.

[15] R. Chen, L. Wang, and R. Zhu, "Improvement of Delegated Proof of Stake Consensus Mechanism Based on Vague Set and Node Impact Factor," *Entropy*, vol. 24, no. 8, 2022, doi: 10.3390/e24081013.

[16] A. H. Alamoodi, O. S. Albahri, A. A. Zaidan, H. A. Alsattar, B. B. Zaidan, and A. S. Albahri, "Hospital selection framework for remote MCD patients based on fuzzy q-rung orthopair environment," *Neural Comput Appl*, vol. 35, no. 8, 2023, doi: 10.1007/s00521-022-07998-5.

[17] H. Qin, Y. Cheng, X. Ma, F. Li, and J. Abawajy, "Weighted Byzantine Fault Tolerance consensus algorithm for enhancing consortium blockchain efficiency and security," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 10, 2022, doi: 10.1016/j.jksuci.2022.08.017.

[18] M. K. Aguilera and S. Toueg, "The correctness proof of Ben-Or's randomized consensus algorithm," *Distrib Comput*, vol. 25, no. 5, 2012, doi: 10.1007/s00446-012-0162-z.

[19] F. Koehler *et al.*, "Impact of remote telemedical management on mortality and hospitalizations in ambulatory patients with chronic heart failure: The telemedical interventional monitoring in heart failure study," *Circulation*, vol. 123, no. 17, 2011, doi: 10.1161/CIRCULATIONAHA.111.018473.

[20] L. E. Nohr, "Telemedicine and patients' rights.," *J Telemed Telecare*, vol. 6 Suppl 1, 2000, doi: 10.1258/1357633001934573.

[21] C. M. Fausett, M. P. Christovich, J. M. Parker, J. M. Baker, and J. R. Keebler, "Telemedicine Security: Challenges and Solutions," *Proceedings of the International Symposium on Human Factors and Ergonomics in Health Care*, vol. 10, no. 1, pp. 340–344, Jun. 2021, doi: 10.1177/2327857921101241.

[22] F. Rezaeibagha and Y. Mu, "Practical and secure telemedicine systems for user mobility," *J Biomed Inform*, vol. 78, 2018, doi: 10.1016/j.jbi.2017.12.011.

[23] T. L. Zhou, B. W. Xu, L. Shi, and Y. M. Zhou, "Measuring package cohesion based on client usages," *Ruan Jian Xue Bao/Journal of Software*, vol. 20, no. 2, 2009, doi: 10.3724/SP.J.1001.2009.00256.

[24] R. Hao, X. Dai, and X. Xie, "Doppel: A BFT consensus algorithm for cyber-physical systems with low latency," *Journal of Systems Architecture*, vol. 148, 2024, doi: 10.1016/j.sysarc.2024.103087.

[25] M. J. Fischer, N. A. Lynch, and M. S. Paterson, "Impossibility of distributed consensus with one fault process," 1982.

[26] C. Dwork, N. Lynch, and L. Stockmeyer, "Consensus in the presence of partial synchrony," *Journal of the ACM (JACM)*, vol. 35, no. 2, 1988, doi: 10.1145/42282.42283.

[27] H. Kim, J. Park, M. Bennis, and S. L. Kim, "Blockchained on-device federated learning," *IEEE Communications Letters*, vol. 24, no. 6, 2020, doi: 10.1109/LCOMM.2019.2921755.

[28] H. J. Ko and S. S. Han, "TPS Analysis, Performance Indicator of Public Blockchain Scalability," *Journal of Information Processing Systems*, vol. 20, no. 1, 2024, doi: 10.3745/JIPS.04.0304.

[29] H. U. Kumar and R. Prasad, "Algorand: A Better Distributed Ledger," in *1st IEEE International Conference on Advances in Information Technology, ICAIT 2019 - Proceedings*, 2019. doi: 10.1109/ICAIT47043.2019.8987305.

[30] B. David, P. Gazi, A. Kiayias, and A. Russell, "Ouroboros Praos: An Adaptively-Secure, Semi-synchronous Proof-of-Stake Blockchain," 2018, doi: 10.1007/978-3-319-78375-8_3.

[31] N. Ettaloui, S. Arezki, and T. Gadi, "An Overview of Blockchain-Based Electronic Health Record and Compliance with GDPR and HIPAA.," 2024.

[32] A. , Arabsorkhi and E. Khazaei, "Blockchain Technology and GDPR Compliance: A Comprehensive Applicability Model.," 2024.

[33] A. Sengupta and H. Subramanian, "User Control of Personal mHealth Data Using a Mobile Blockchain App: Design Science Perspective," *JMIR Mhealth Uhealth*, vol. 10, no. 1, 2022, doi: 10.2196/32104.

[34] A. J, D. P. Isravel, K. M. Sagayam, B. Bhushan, Y. Sei, and J. Eunice, "Blockchain for healthcare systems: Architecture, security challenges, trends and future directions," 2023. doi: 10.1016/j.jnca.2023.103633.

[35] A. Liu *et al.*, "A GRA-Based Method for Evaluating Medical Service Quality," *IEEE Access*, vol. 7, 2019, doi: 10.1109/ACCESS.2019.2903684.

[36] M. Ayenew *et al.*, "Enhancing the performance of permissionless blockchain networks through randomized message-based consensus algorithm," *Peer Peer Netw Appl*, vol. 16, no. 2, 2023, doi: 10.1007/s12083-022-01407-3.