”How Small Can a Blockchain Be?” Optimizing Storage of Lightweight IoT MANET DLTs

***Abstract*—Internet of Things (IoT) networks are growing in prominence. Blockchains and their broader distributed ledger technologies (DLTs) can connect them in distributed drop-in and drop-out networks. This is the first paper to analyze Mobile Ad-Hoc Networks (MANETS) and how the storage capacity of limited range peer-to-peer gossip networks are reflected by the different architectures, consensus algorithms, block parameters, and current generation lightweight cryptography. We will benchmark blockchains, Directed Acyclic Graphs (DAGs), and Tree-Chain across multiple consensus algorithms and compare the storage requirements to previously studied throughput metrics. Ultimately we'll show that Tree-Chain is the current best storage solution because of it's redundant transaction prohibition, how DAGs/blockchains can implement this feature for improved performance, and design a lightweight DLT block and transaction header that is customizable for desired security levels. The underlying simulator can be a springboard for all IoT-MANET and smart-city applications to optimize parameters and decide: ”How small can a blockhain be?”**

***Index Terms*—Blockchain, IoT, MANET, DLT, Smart-City**

I. Introduction

Distributed Ledger Technology (DLT) has matured and more diverse/customizable alternatives to blockchains have emerged with different properties [1]. Our focus is on Internet of Things (IoT) Mobile Ad-Hoc NETworks (MANETS), the perfect intersection of distributed lightweight nodes and decentralized ledgers. Ad-hoc public and permissioned ledgers are applicable towards; smart cars, toll-systems, crowd-sourcing, contact tracing, delivery, drones, and taxiing. The parameters of our simulation can be tuned to represent pedestrians, automotive vehicles, bikes, sensors, or sidewalk automated delivery robots [2]. We envision smart-cities/homes/factories using drop-in and drop-out networks to relay information, and this study fills the following gap: ”What DLTs, consensus algorithms, and parameters should I use for an optimized IoT DLTS?”

We set out to benchmarks and optimize 4 unique architectures in Internet of Iot-MANET DLTs; blockchains, a Directed Acyclic Graph (DAG) (archetypically represented by IOTA Tangle [3]), Hedera HashGraph [4], and Tree-Chain [5]. In a previous paper [6] we benchmarked throughput and Expected Confirmation Transaction (ECT) of transactions [6]. We concluded that HashGraph's asynchronous ordering mechanism scaled extremely poorly with network density, and was no longer relevant, DAG Proof of Location (PoL) was the optimum for throughput, Proof of Work (PoW) was the worst consensus algorithm in all scenarios, and all DLTs' throughput scaled positively with network density. This second chapter reports the storage requirements of the unique architectures under different storage strategies, minimum/maximum transactions per block, and lightweight crpytographic candidates.

II. Simulator

Our novel python simulator, discretizes urban movement of range-limited agents running full-nodes of a DLT with a gossip network of ”transactions” (defined in II). By varying parameters such as; number of agents, wireless network connectivity (Bluetooth, Wi-Fi, LPWAN), minting difficulty, map, consensus algorithm, and block size, we can report the optimum of each scenario.

The following simulations were run on two maps, a spoke-and wheel highway system with Low Power Wide Area Network (LPWAN) peer-to-peer gossip, and a dense downtown Wi-Fi peer-to-peer network. The differences are insignificant and only results from the highway system are shown here. All simulations use the optimum confirmation time parameters discussed in [6], and are run for 10,000 transactions and 167 ”minutes.”

Network density has no effect on the total storage requirements in our simulations, because the rate of new transactions is the same. The database of agent registration, including agent IDs and public keys would have to increase with network density, but that is not covered in this short paper for brevity.

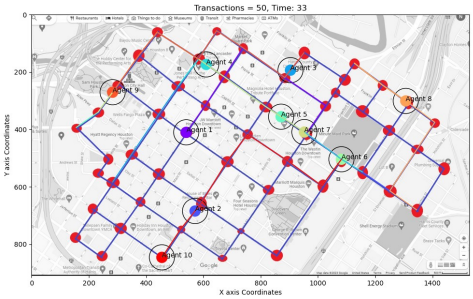

Fig. 1. Simulation visualization, each labeled agent's P2P communication range is denoted by the black circle. Each agent's previous location is visualized as a color trail.

III. Storage Strategies

A. Accumulation

The anti-storage strategy, every agent accumulates all transactions and blocks. Every agent is a full node in the DLT, and can bootstrap newcomers by themselves and check all previous transactions and blocks. Inevitably any DLT will grow larger with time and outstrip the storage capabilities of lightweight agents.

Our implementation allows discarding of orphaned blocks that will not affect the permanent DLT.

B. Pruning Blocks

The following are pruning techniques to discard historical data and limit storage requirements. We use the classification established in Abdelhamid et al. [7]

- **Simple Block Pruning/Sharding:** Palm et al. [8] describe an efficient pruning and distributed bootstrapping system limited to permissioned DLTs. We implement a version where all blocks are hashed and modularized by $\frac{\#\text{Agents}}{2}$ and assigned to two agents for historical storage.
- **Balance-based Synchronization:** The last N blocks are considered the most important, and the running total "balance" of each agent is all that is saved. Historical data can be compressed to a change in agent balances.
- **State-based Synchronization:** distinct from balance-based, because there is an implied smart contract or other statefulness of the DLT. Ethereum Full Sync and Fast Sync utilized this strategy, and Kim et al [9] created a novel alternative for Ethereum. Our IoT-MANET DLT does not utilize smart contracts so we will not implement and test state-based strategies.

C. Road Side Units (RSUs) or Base Stations

RSUs are static, more powerful infrastructure that can bridge the cyber-physical gap between lightweight peer-to-peer IoT networks and the broader internet. [10] We model RSUs as interconnected toll-stations, that collect all historical data for bootstrapping and more complex DLT operations beyond a lightweight IoT agent.

IV. DLT Block & Transaction Structure

A. Data Structures

Tables I and II show our proposed DLT block header, transaction header, transaction input, and transaction output. A digital signature is necessary to provide authentication, and thus transactions will have to have a digital signature appended to all transactions.

B. Compression

We tested a battery of Ubuntu based hash algorithms and achieved a reliable 1.695 compression rate across all algorithms, and all data sizes. Agents can compress historical blocks, and still quickly identify each block by either not compressing the block ID, or placing them in a predictable pattern in external memory.

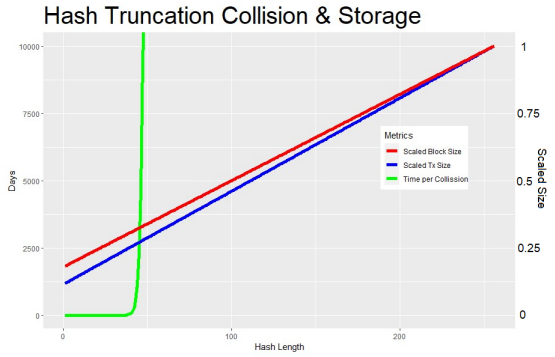

Fig. 2. Hash Truncation effect on data structure size and collisison resistance.

Kim, Noh, and Cho [11] developed a storage compression consensus to optizime blockchains. Melding compression and consensus algorithms in the future to create a Proof of Compressions/Size would be a valuable addition to the lightweight DLT community.

C. Hash Function

Table III presents all of the NIST lightweight hash finalists [13]. Based off of Abed et al. [15] storage, security, power, and throughput metrics. Alfrhan et al. [16] compare SHA-2, SHA-3/Keccak, and PHOTON. Based on both comparisons and the amount of attention PHOTON has received, we tentatively selected it for our lightweight DTLs.

Future work will run independent execution and library size of the 4 finalists compared to TurbkSHAKE, the 12-round instance of Keccak, the underlying primitive under KangarooTwelve.

We benchmark PHOTON-Beetle, on a AMD Ryzen 5 3600 6-Core Processor with a NVIDIA GeForce GTX 1660 Ti GPU, and achieve 1.3 million million bytes hashed per second, which we use to derive figure 2. This collision model assumes that an adversary is trying to replace a specific previous transaction or block with a false transaction. We do not account for birthday attacks [17], that analysis would be relevant to future work on IoT lightweight security constraints.

With this analysis we selected 48 bits of truncated hashing, which provides 13244 days or 36.29 years of time for a midrange desktop to succeed in a specific collision attack. This parameter is adjustable depending on the security constraints of the network, which will naturally be lower than traditional ledgers because of the lightweight IoT requirements of the network.

D. Digital Signature

We selected SEMECS [18], for our digital signature system, because it has the smallest tag size of 32 bytes. SEMECS is a K-time signature scheme; public keys are $(2 * K + 1) * 32$ bytes long, and private keys are 32 bytes. This paper does not deal with the key-management system to use a K-time signature, but we suggest gossip using new keys will also include the new public key signature signed by the last private

Proposed IoT DLT Block Structure (208 Bits)			
Bits	Name	Data Type	Description
8	Version	uint8	Architecture/block version
32	Timestamp	uint32	Time of minting
32	Minter/Minters	uint32[maxMinters]	List of minting agent's addresses
48	PreviousBlockHash	PHOTON hash	Hash of previous block header
32	Nonce	uint32	Random data to overcome difficulty
8/40/35	Consensus Specific	uint8/base32[8]	Necessary Difficulty for PoW, Geocode location for PoL Wi-Fi/LP-WAN
48	Hash Root of Txs	PHOTON hash	Root of Merkle Tree containing Transactions
*	Transaction List	Txs[NumTxs]	List of Transactions

TABLE I

DATA STRUCTURE OF LIGHTWEIGHT IOT-MANET BLOCK HEADER, PHOTON HAS A 256 BIT HASH, BUT SIZE CAN BE EXCHANGED FOR DECREASED SECURITY BY TRUNCATING IT TO 48.

Proposed Transaction Structure (≈244 Bits)			
Bits	Name	Data Type	Description
32	Time	uint32	Time of creation
4	NumInputs	uint4	Number of Inputs for this transaction
52*Inputs	Input List	Input[NumInputs]	List of all UTXOs to be consumed by the transaction.
4	NumOutputs	uint4	Number of Outputs for this transaction
16*Outputs	Output List	Output[NumOutputs]	List of UTXOs to be generated by the transaction.
Proposed Input Structure (52 Bits)			
48	Outpoint Hash	PHOTON hash	The previous transaction that contains the spendable output
4	Outpoint Index	Tx Index	The index within the previous transaction's output array to identify the spendable output
Proposed UTXO Structure (16 Bits)			
16	Value	uint16	Monetary value of this UTXO, max value is 65536

TABLE II

DATA STRUCTURE OF LIGHTWEIGHT IOT-MANET TRANSACTION HEADER AND INPUTS/OUTPUTS. WE SIMULATE THAT EACH TRANSACTION HAS 3 INPUTS AND OUTPUTS, BASED ON THE BITCOIN BLOCKCHAIN'S INPUT/OUTPUT FREQUENCY.

NIST Finalist	**Building Block**	**Mode**	**Digest Size**
PHOTON-Beetle	PHOTON_{256} Permutation	Sponge	256
ASCON	ASCON Permutation	Sponge	256
Romulus	Skinny-128-384+	MDPH [12]	256
SPARKLE	SPARKLE_{384}	Sponge	256
Xoodyak	Xoodoo Permutation	Sponge	256

TABLE III

NIST LIGHTWEIGHT IOT HASH FINALISTS [13], PHOTON WITH THE AUTHENTICATED ENCRYPTION MODE BEETLE [14] IS OUR IMPLEMENTED CHOICE.

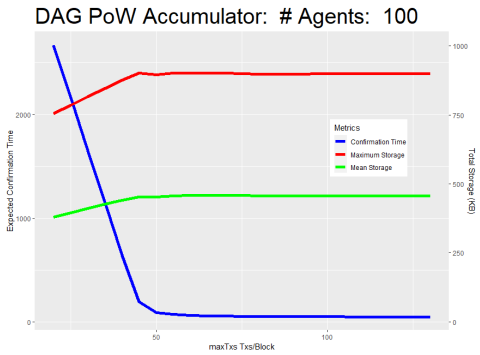

Fig. 3. Block ceiling is a bottleneck with no benefits for lower values in all scenarios.

key signature. A tag will be appended to all transactions to provide authentication, making the practical size of each transaction 500 bytes exactly instead of 244.

This is the current largest storage requirement and will be a major topic of interest for future IoT development minimizing digital signatures while maintaining security.

V. BLOCK SIZE

We vary two parameters, the minimum block size, and maximum block size and study the resulting storage requirements in MANET environments with different storage strategies.

Maximum block size is always left as large as possible, our simulations showed that there is no improvement in storage spikes or any performance benefit from lowering maximum block size.

Minimum block size is specific to each DLT/consensus mechanism scenario. Figure 4 shows that asynchronous peer-to-peer networks are prone to redundant transaction minting, wasting space on both DAGs and blockchains, regdardless of the consensus mechanism.

DLT & Consensus	**Mean Storage**	**Maximum Storage**	**Confirmation Time**
PoW Blockchain	438	869	108
PoL Blockchain	310	650	51
PoW DAG	452	887	19
PoL DAG	288	582	**1.2**
Tree-Chain*	**149**	**390**	2953

TABLE IV

STORAGE RESULTS OF OPTIMUM SPACE REQUIREMENTS. TREE-CHAIN MONOTONICALLY IMPROVES STORAGE REQUIREMENTS WITH "MINIMUM TX/BLOCK", SO IT CAN IMPROVE FURTHER, THE EXPRESSED TREE-CHAIN STATISTICS REQUIRES 55 TRANSACTIONS PER BLOCK.

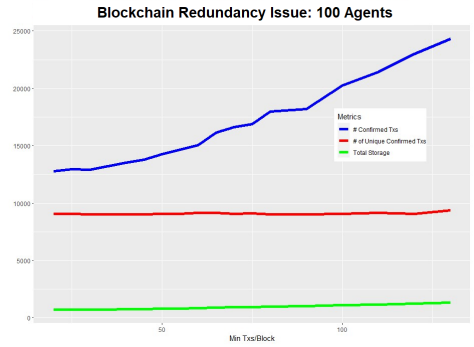

Fig. 4. In blockchains and DAGs, when block minting is delayed, redundancy increases, as multiple agents asynchronously submit the same transaction.

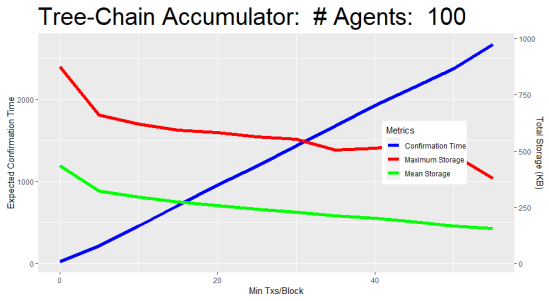

Fig. 5. Tree-Chain Accumulator

Fig. 6. Tree-Chain trades confirmation time for overall storage efficiency.

VI. Conclusion

Cryptography and security guarantees are the largest factor for storage size.

Reducing redunant transaction minting is the best architectural decision. Blockchains and DAGs have no natural safeguards to stop distributed networks from creating redundant transactions. Tree-Chain's bin system solves this problem and can potentially be implemented by PoW/PoL blockchains and DAGs; each agent can only mint transactions that are somehow assigned to them. Either by random hash, geo-spatial location, or some other method.

For lightweight nodes to operate in perpetuity, pruning isn't enough, either heterogeneous RSUs or balance-based pruning is necessary to keep the spatial requirements below a ceiling. Tree-Chain with RSUs can comfortably be run on 20 kB of memory for blocks and the transaction pool, small enough for a $46 Arduino MKR WAN 1310 to run a full node.

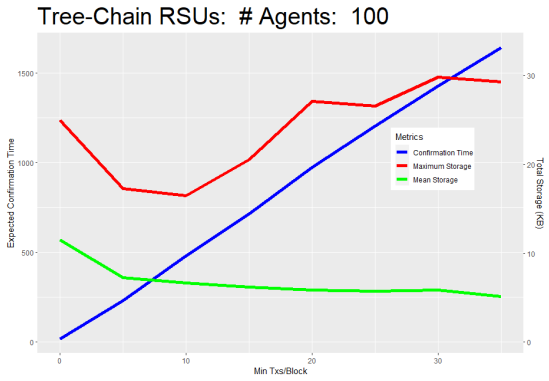

Fig. 7. Tree-Chain with RSUs (6 in this simulation), show the best overall storage requirements. This system can be used in perpetuity with 20 kB of storage, fitting comfortably on an Arduino.

References

[1] N. Troutman and W. Shi, "Survey of spatially aware blockchains for iot," *In-Review*, 2023.

[2] D. Jennings and M. Figliozzi, "Study of sidewalk autonomous delivery robots and their potential impacts on freight efficiency and travel," *Transportation Research Record*, vol. 2673, no. 6, pp. 317–326, 2019.

[3] W. F. Silvano and R. Marcelino, "Iota tangle: A cryptocurrency to communicate internet-of-things data," *Future generation computer systems*, vol. 112, pp. 307–319, 2020.

[4] L. Baird, M. Harmon, and P. Madsen, "Hedera: A governing council & public hashgraph network," *The trust layer of the internet, whitepaper*, vol. 1, pp. 1–97, 2018.

[5] A. Dorri and R. Jurdak, "Tree-chain: A fast lightweight consensus algorithm for iot applications," in *2020 IEEE 45th Conference on Local Computer Networks (LCN)*. IEEE, 2020, pp. 369–372.

[6] N. Troutman and W. Shi, "'which blockchain is best?', simulating iot-manet dlts: Network scalability confirmation time in smart cities," *Submitted*, 2024.

[7] M. M. Abdelhamid, L. Sliman, R. Ben Djemaa, and B. Ait Salem, "Abischain: Towards a secure and scalable blockchain using swarm-based pruning," in *Proceedings of the 2023 Australasian Computer Science Week*, 2023, pp. 28–35.

[8] E. Palm, O. Schelén, and U. Bodin, "Selective blockchain transaction pruning and state derivability," in *2018 Crypto Valley Conference on Blockchain Technology (CVCBT)*. IEEE, 2018, pp. 31–40.

[9] J.-Y. Kim, J.-M. Lee, Y.-J. Koo, S.-H. Park, and S.-M. Moon, "Ethanos: Lightweight bootstrapping for ethereum," *arXiv preprint arXiv:1911.05953*, 2019.

[10] Z. Hu, Z. Zheng, T. Wang, L. Song, and X. Li, "Roadside unit caching: Auction-based storage allocation for multiple content providers," *IEEE Transactions on Wireless Communications*, vol. 16, no. 10, pp. 6321–6334, 2017.

[11] T. Kim, J. Noh, and S. Cho, "Scc: Storage compression consensus for blockchain in lightweight iot network," in *2019 IEEE International Conference on Consumer Electronics (ICCE)*. IEEE, 2019, pp. 1–4.

[12] S. Hirose, "Some plausible constructions of double-block-length hash functions," in *International Workshop on Fast Software Encryption*. Springer, 2006, pp. 210–225.

[13] M. S. Turan, M. S. Turan, K. McKay, D. Chang, L. E. Bassham, J. Kang, N. D. Waller, J. M. Kelsey, and D. Hong, *Status report on the final round of the NIST lightweight cryptography standardization process*. US Department of Commerce, National Institute of Standards and Technology, 2023.

[14] A. Chakraborti, N. Datta, M. Nandi, and K. Yasuda, "Beetle family of lightweight and secure authenticated encryption ciphers," Cryptology ePrint Archive, Paper 2018/805, 2018, https://eprint.iacr.org/2018/805. [Online]. Available: https://eprint.iacr.org/2018/805

[15] S. Abed, R. Jaffal, B. J. Mohd, and M. Al-Shayeji, "An analysis and evaluation of lightweight hash functions for blockchain-based iot devices," *Cluster Computing*, vol. 24, pp. 3065–3084, 2021.

[16] A. Alfrhan, T. Moulahi, and A. Alabdulatif, "Comparative study on hash functions for lightweight blockchain in internet of things (iot)," *Blockchain: Research and Applications*, vol. 2, no. 4, p. 100036, 2021.

[17] M. Bellare and T. Kohno, "Hash function balance and its impact on birthday attacks," in *Advances in Cryptology-EUROCRYPT 2004: International Conference on the Theory and Applications of Cryptographic Techniques, Interlaken, Switzerland, May 2-6, 2004. Proceedings 23*. Springer, 2004, pp. 401–418.

[18] A. A. Yavuz and M. O. Ozmen, "Ultra lightweight multiple-time digital signature for the internet of things devices," *IEEE Transactions on Services Computing*, vol. 15, no. 1, pp. 215–227, 2019.