Benchmarking Performance of Ethereum Blockchain on Resource Constrained Devices

Suhail Al Marzouqi, Michael Baddeley, Martin Andreoni Lopez
Secure Systems Research Center (SSRC), Technology Innovation Institute (TII)
Abu Dhabi, United Arab Emirates
{suhail, michael, martin}@ssrc.tii.ae

***Abstract*—Blockchain has challenged many of the conventions around digital security. In essence, blockchain supports a decentralized platform maintained by peers instead of a single entity. Furthermore, the data in the blockchain is immutable and is being held in a secure and encrypted way. However, running the blockchain on resource-limited devices, such as a consumer PC or a low-power Raspberry Pi as opposed to dedicated servers, is demanding due to the resource limitation in energy, memory, and time taken to validate the transaction on the blockchain. This paper explores these limitations by evaluating and benchmarking the blockchain framework Geth: a terminal interface for the Ethereum blockchain which makes it possible to create a private blockchain in addition to joining the actual blockchain. This article employs a private blockchain within Geth and benchmarks the blockchain to highlight the differences (and limitations) between the devices. Specifically, we make the following observations: (i) the time it takes to validate and add the transaction to the blockchain on a Raspberry Pi 4 is markedly slower compared to an Intel i9-10885H CPU @ 2.40GHz, (ii) the transaction between PCs compared to transactions between Pis is around two orders of magnitude higher. These valuable insights can be used to help researchers in the design and implementation of blockchain-driven security architectures for distributed systems, such as industrial IoT deployments and UAV swarms.**

***Index Terms*—Blockchain, IoT, Geth**

I. Introduction

Over the last decade, blockchain, and the Distributed Ledger Technology (DLT) underpinning it, has emerged as a revolutionary data management framework to establish consensus and agreements in a trust-less and distributed environment. Blockchain offers an immutable, transparent, secure, and auditable ledger to verify the integrity and traceability of information/assets during their life cycle by eliminating a central authority's involvement. For example, a swarm of Unmanned Aerial Vehicles (UAV) on a mission to collect certain data about a specific area using wireless sensing nodes [1] can potentially be accessed by a bad actor, and the data tampered with. However, blockchain solves this issue by making it harder for the bad actor to alter the data [2]. Essentially, if the actor changes the data of one block, they would have to change the data of all the blocks leading to the first block in the blockchain. Nevertheless, this raises two main concerns. Firstly, IoT devices lack the resources to handle the blockchain size as the chain scales. Secondly, validating and adding the transaction to the blockchain takes some time – which can overwhelm infrastructure in terms of computational power. For example, when considering available computing resources, IoT devices have a small memory capacity in MBs, while an average blockchain can often take up GBs [3]. Since all devices would have to keep a local copy of the ledger, this is prohibitive to many devices considered in blockchain-driven security use-cases. To examine these issues and present a benchmark of blockchain performance on commodity devices, the paper presents a benchmark design and an evaluation of the resource-limited devices with the Geth framework.

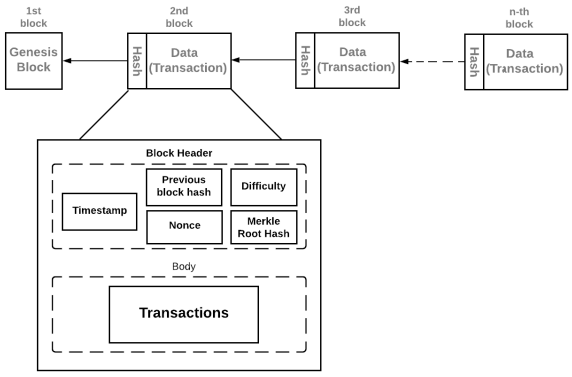


Fig. 1: Overview of the blockchain structure.

Specifically, this article makes the following contributions:

- We evaluate the performance of the (Ethereum driven) Geth blockchain framework [4] for two different devices.
- We benchmark the resource efficiency of the blockchain on these devices: in terms of memory, energy consumption, and transaction execution time.

II. Background

A. Blockchain and Consensus Mechanisms

A blockchain (as demonstrated in Figure 1) is a distributed system of transactions organized in blocks. Distributed peers maintain these blocks. Each block comprises a block header and a body. A block header includes the previous block's hash, timestamp, nonce, difficulty level, and the Merkle root hash. The body includes the transactions. In a blockchain, the users that execute and confirm transactions are called miners. Before digitally signing and appending the transaction into the blockchain network, these miners verify the transactions by following a mining mechanism. Since the blockchain is a distributed public ledger, it holds immutable data in a secure

978-1-6654-7038-4/22/$31.00 ©2022 IEEE
DOI 10.1109/CPS-IoTBench56135.2022.00009

and encrypted way. It also ensures that the transactions cannot be altered, thus preventing data and execution tampering. Blockchain's design achieves decentralization, traceability as well as execution, and data transparency [5]. A blockchain can be either public or private; public implies anyone can validate blocks, while private means only trusted parties could validate blocks. Blockchain also provides services such as authentication, confidentiality, and more. While in traditional security approaches, those services are provided by centralized platforms, this centralization poses a single point of failure and can often represent a considerable security risk. However, by distributing security transactions across all nodes in the ledger, blockchains address the single point of failure threat. Typical blockchain architecture is composed of the following elements: *node*, a participating device in the blockchain. *Transaction*, holds information; *block*, holds a set of transactions visible to all nodes in the blockchain; *chain*, a sequence of blocks; *miners*, perform validation process to add a block to the blockchain; *validation mechanism*, a set of rules to carry out the operations in the blockchain. Moving on into the consensus mechanisms. Proof of Work (PoW) is the mechanism for validating a blockchain block. In this approach, miners perform some work, usually complex mathematical problems that are easy to validate. Each block in the blockchain has a PoW; it depends on the time it takes to validate a block from the miner, the more computational power the miner has, the faster the validation is. On the other hand, it can suffer from some detrimental shortcomings. Over time rewards will decrease; hence the miners will too [6]. Proof of Stake (PoS) is an extra step to PoW, which solves the shortcoming problem. Instead of miners, there are forgers. More money the forgers have means more mining power they have for validating blocks [6]. Proof of Concept (PoC) [7] is to prove if a concept has a real value to the world. It is developing a blockchain solution and seeing if it can become a reality or not. Unlike the Proof-of-Work (PoW), which was used in this experiment highly dependent on the resources of the IoT device, higher computational power results in less execution time for the transaction.

III. Benchmark Design and Evaluation

Our benchmark evaluates the Geth blockchain platform [4], an open-source implementation of the Ethereum blockchain, written in Go Lang, and utilizes the PoW mechanism present in ethereum. This paper employs Geth for sending and receiving transactions between the devices (a Raspberry Pi 4 and an Intel i9-10885H CPU @ 2.40GHz) by creating a private blockchain with multiple peers in it. Our goal is to evaluate trade-offs in terms of a) memory consumption, b) energy consumption, c) time taken to validate and add the transaction to the blockchain.

A. Benchmark Process

The benchmark process is as follows; we connect all devices (a.k.a peers) to the private blockchain. Then, we initialize the Genesis block, which is the root block of the blockchain; the difficulty and other factors are determined in the Genesis block. The next step is sending the transactions from one device to another, as seen in Figure 2. We analyze three different scenarios:

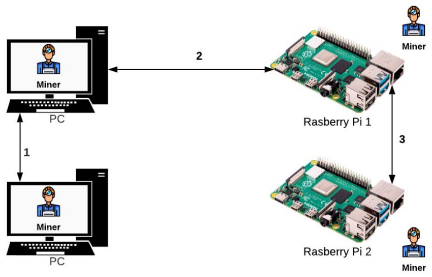


Fig. 2: Benchmark's development process. Connecting the devices to the blockchain for sending and receiving transactions between them.

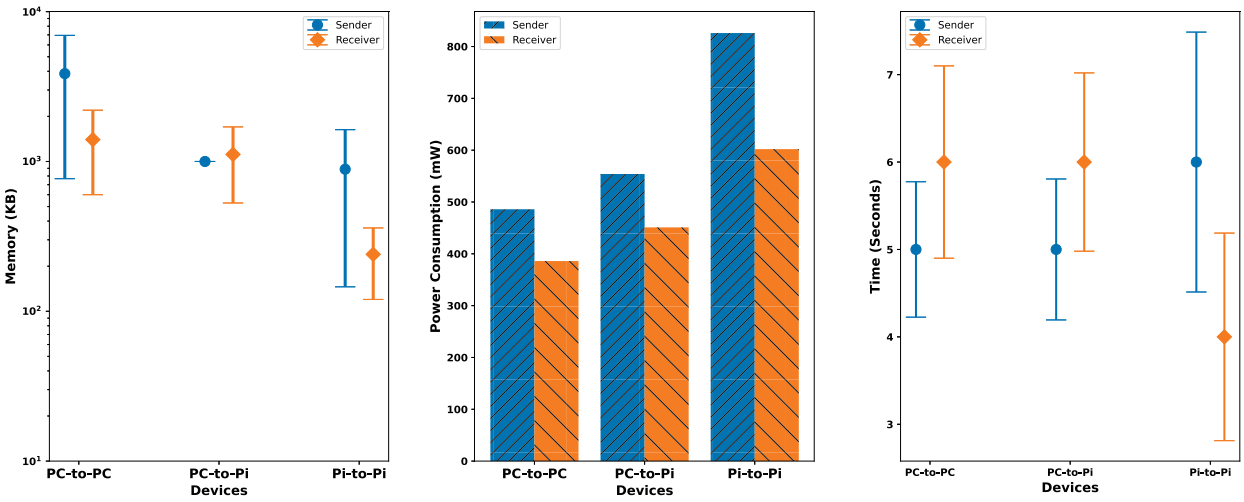
- Transactions between PCs (i.e., *PC-PC*).
- Transactions between a PC and Pi (i.e., *PC-Pi*).
- Transactions between Pis (i.e., *Pi-Pi*)

The evaluation will try to answer the following questions: (i) to what extent is inference technically feasible on limited resource devices, and (ii) what is the overhead of blockchain on IoT devices in terms of memory consumption, energy consumption, and time to validate and add the transaction?

B. Experimental Setup

Software setup. We deploy Geth on each device and initialize all of the devices with the genesis block, set the IP address of each device as the peer address, and fix each device to have its own port. For interacting with the blockchain, we use Geth attach [4]. Geth attach is the terminal page to send the commands and interact with the blockchain. We can also see the connected peers. We initialize each device with the Genesis block for our experiment and peer them by their IP addresses. The Genesis block [4] is the first block in the blockchain; it can be set up in a private blockchain to have some specific parameters, such as the gas limit, which dictates the cost of sending/receiving a transaction. Also, we can set the difficulty level of mining a block and allocate some ether to certain miners participating in the blockchain. We start each miner in each device to add a block to the blockchain to get incentive (private ethereum). We say private ethereum, meaning we let the miners mine the blocks and add them to the private blockchain, and get rewarded for it. The real ethereum blockchain will take around eight days to get one ether; as more blocks get mined in the real blockchain, the more the difficulty becomes. The peering of the devices can be checked from the platform with a specific command denoting the IP addresses of the peered devices. All the blocks in the blockchain are mined by a random device which also can be known from the framework. Finally, a transaction from one device to another can be initiated to send and receive some ether. In all tests, ten Ethers were sent from one IoT device to another within the span of 15 seconds.

Hardware setup. We use the Raspberry Pi 4 Model B that features a `Broadcom BCM2711`, quad-core Cortex-A72



(a) Memory. (b) Average (mean) power. (c) Time.

Fig. 3: Resources consumed during the processes of executing a transaction and adding it to the blockchain.

(ARM v8) 64-bit SoC at 1.5GHz, with 4GB LPDDR4-3200 SDRAM. We employ an 8 core `Intel i9-10885H` at 2.40GHz Laptop with 32.0 GB, of which only 16 GB's of RAM and 4 Cores are being used for this experiment. We use the Linux utility `powerTOP` to measure the power consumption.

C. Benchmarking

We repeat each experiment 10 times. We report the average values of memory consumed by the process of validating and adding the transaction to the blockchain. We also report the energy consumption based on the process and how much power was consumed throughout the transaction. Finally, We also report the average inference execution time based on how long the process has run and stopped. For all benchmarks, the standard deviation is conveyed as $\pm$ within Table I.

Memory consumption. In Figure 3a, we present the memory consumption by reporting the average memory consumption for each case being PC-to-PC, PC-to-Pi, and Pi-to-Pi. It can be seen that the memory consumption of the sender in both the PC-to-PC and Pi-to-Pi case are higher than the receiver. We notice that the highest memory consumption of all the cases is the PC-to-PC case, being at 10^4 KBs. While it can be noticed that the Pi-to-Pi case memory consumption is two orders-of-magnitude lower by 10^2 KBs, this is because the PC has more computational power compared to the Pis. It is able to mine blocks more than the Pis throughout the transaction. In the case of PC-to-Pi, the PC had no deviation in all the tests. It always came out consuming 10^3 KBs. The PC and Pi are the only working peers in this test, and the Pi manages to due the mining for a timeless than the PC since it has lower computational power. As for the receiver, the Pi, it can be noticed that there is a deviation of 586.58 KB's as shown in Table I. Finally, we notice that the sender, in all cases, consumes much more memory than the receiver.

Power consumption. In this part, we report the energy consumption based on the process and how much did it consume throughout the transaction using powerTOP. Figure 3b presents a comparison of the power consumed during the inference of a transaction being validated and added to the blockchain. We notice a similar trend in the receiver in all the cases. Pi-to-Pi has the highest power consumption at an average of 825 $\pm$ 1363.1 mW for the sender and approximately 600 $\pm$ 1027.7 mW for the receiver. In Table I, we report the average power consumption and the standard deviation of all the cases. PC-to-PC consumes 458 mW and 385 mW for the sender and receiver, respectively. In contrast to the Pi-to-Pi case, the PC has more computational power than the Pis. However, due to their low computational power, the Pis try their best to validate and add the transaction to the blockchain, hence, the high power consumption. In all cases, the sender has consumed higher power than the receiver due to the initiation of the transaction.

Inference execution time. We continue with the average inference execution time for each case shown in Figure 3c and Table I. We notice that both the PC-to-PC and PC-to-Pi cases in the sender take around 5 $\pm$ 0.8 seconds on average to validate and add the transaction to the blockchain, showing similar behavior. However, the Pi-to-Pi sender case takes a bit longer, with around 6 $\pm$ 1.5 seconds for validating and adding the transaction to the blockchain. In all cases, the receiver showed a similar deviation and took approximately 5 seconds to receive the transaction and have it in its local blockchain.

However, while many devices participate in a real-world blockchain, in these experiments, there are only two active devices. Sender miners will send the transaction and then

TABLE I: Average values of Memory Consumption (KB)

Transaction	Memory (KB)		Power (mW)		Time (s)	
	Sender	Receiver	Sender	Receiver	Sender	Receiver
PC-to-PC	3858 ± 3090.5	1400 ± 800	486	386	5 ± 0.77	6 ± 1.1
PC-to-Pi	1000	1115 ± 586.6	554	451	5 ± 0.81	6 ± 1.01
Pi-to-Pi	888 ± 742.4	240 ± 120	826	602	6 ± 1.48	4 ± 1.19

stop mining. Receivers then start mining until they receive the transaction. While one would expect the Pi-to-Pi time to be greater than that of PC-to-PC, we hypothesise that (counter-intuitively) the limited resources of the Pi sender means it does not have the computational power to mine quickly. More time is therefore given for the transaction to come through at the receiver and allows it to focus it is resources firstly on the mining, and then on the transaction. The PC, with greater resources, will quickly finish the transaction meaning the receiving node must process mining and the transaction at the same time.

D. Discussion and Limitations

Generally, we observe that the results of each benchmark solely depend on the capabilities of the device. The more computational power the device has, it will be able to validate and add the transaction to the blockchain quicker. At the same time, since it can validate at a higher rate, it will mine the blocks most of the time and consume more memory. This is essentially a trade-off between performance and memory consumption.

In contrast, the power consumption for IoT devices is far greater as devices are pushed to their computational limits, and we observed that power on low-powered IoT devices is consumed more than the PCs. The time it takes to validate the transactions and add them to the blockchain in the PCs are faster than the Pis due to the resources. In terms of memory, The consumption is at its lowest for the Pi-to-Pi case compared to the PC-to-PC case due to the mining of the block. Also, the addition of peers to the network could play a significant role in the time for validation and addition of the transaction into the blockchain.

Specifically, we find that the trade-off for these benchmarks can be higher or lower depending on the settings of the Genesis block in Geth, but security is at risk too. We also observe limitations as the number of transactions increases. This, naturally, causes an increase in the size of the blockchain, which may lead to memory starvation where low-powered IoT devices may not have enough resources to handle such scalability.

IV. Related Work

This paper uses the Geth framework for all the devices under test. Interestingly, however, an alternative DLT framework, IOTA [8], positions itself as a cryptocurrency for the IoT. In IOTA, the main distinction from other ledger technologies is the concept of a *tangle*. The tangle is a Directed Acyclic Graph (DAC) for storing transactions. Nodes issue these transactions in the tangle. If a new transaction is to be added to the chain, in Bitcoin [9] it would be added to the block after a node solves a very complex puzzle or mathematical problem. On the contrary, the tangle would be the approval of two previous transactions. Thus, transactions are issued without any fees.

Hang and Kim [10] present an implementation of blockchain using IoT devices for sensing data in real-time and is usable in any user's device like phones, for example. First, the authors present a smart contract for the sensor and actuator events in case they pass the threshold to collect some data. After that, they analyzed the registration of devices into the blockchain with the use of permissioned blockchain [11]. They tested with a different number of devices: 50,150,250, and 500 and showed that 50 devices registration (50 transactions) with four IoT devices being PCs running docker and Pis took around 2.3 Seconds. In our case, the Geth framework for ten transactions took around 4.5 seconds for all cases as the sender. This is mainly the case because of the consensus mechanism, the mechanism used for validating the transaction and adding it to the blockchain. Additionally, the Hang and Kim [10] approach used the PoC as mentioned in the background section; they proposed an architecture for having devices getting authenticated to collect some data, and before being added to the blockchain, they get verified, and another entity validates them to be added into the blockchain.

Ramachandran [12] is the official IOTA energy benchmark for the chrysalis edition of IOTA. The author mentions that the consensus mechanism is PoW, like the one in the Geth framework. However, it operates differently since the PoW in IOTA is not necessarily done by the node itself, meaning the node can only relay the message/transaction instead of calculating the hashes to find the nonce that validates the transaction. Looking at results on Table 12 in [12] it can be seen that they used the Raspberry Pi 3 and 4 for the tests with Easy PoW meaning it was done on a client, not the node itself. The author normalized the data from having 0 Transactions Per Seconds (TPS), 50 TPS with Easy PoW consumed approximately 214 mW. 100 TPS consumed around 300 mW. Finally, the mainnet, the actual IOTA blockchain, consumes 1045 mW. All of these power evaluations were just for relaying the message. We used the PoW on our nodes to validate and add the transaction to the blockchain; for example, in the Pi-to-Pi case, it is the highest of all cases in power consumption terms because of the computational power it has.

Özyılmaz and Yurdakul [13] have also done some initial work on the computational and storage factors stating they are

not necessary to run the blockchain on IoT. Therefore, it is possible to run the blockchain on IoT devices but not run as efficiently as possible.

Elsts et al. [14] have done some work on the creation of a transaction bundle in iota and be able to send data and some Iota's with different devices; in particular, we focus on the CPU and Pi. The transaction bundle is basically a bunch of transactions object, and these objects contain an address of the sender/receiver, some value of iota to send/receive, tag, timestamp, current and last indices. Looking at Table 4 in [14] they show the average time measured and energy consumption estimated for the IOTA operations. The time it takes to sign the transaction differs from our presented work in this paper since another entity, the IoT proxy, is doing the validation. Nevertheless, it is worth noting that the time it takes to do so is very little, around 370 ms for a Raspberry Pi 3B, and a Core i7 CPU takes 3.5 ms. Our results show that the PC is faster than the Pi in the execution time. In addition, the power consumption of the PoW is calculated in Joules and can be converted to Watts by dividing the time it took to complete the PoW. For example, the Raspberry Pi took around 82 seconds to complete the PoW and consumed approximately 55 Joules. Converting it to mW, this is around 670 mW. In our PC-to-Pi, the Pi consumed 554 mW as shown in Table I. While as a sender and the receiver in the Pi-to-Pi case, the power consumption values are 826 mW and 602 mW, respectively, which is quite similar to the authors' results.

V. Conclusions

Blockchain has been one of the most prominent and discussed technologies within the last decade. Particularly for distributed use-cases, such as scenarios common in IoT systems, blockchain technology promises secure, validated transactions without reliance on a central authority. However, with all the claimed advantages that blockchain provides, it is essential to analyze and understand the performance and capabilities of this technology on different devices – particularly the resource-limited devices often targeted by proposed blockchain use-cases. This paper shows the trade-offs between using blockchain on a PC (such as could be found on commercial drones) and on a Raspberry Pi (as used in many IoT deployments). We then presented a benchmark to evaluate the Geth framework for both a PCs and Pis, revealing significant differences and trade-offs between the different cases presented, We firstly find that the PC-to-PC case consumes much more memory than the other cases but consumes the least energy, and therefore has implications for power-limited applications such as UAVs Secondly, our evaluation shows differences in execution time mainly between the PC-to-PC case and PC-to-Pi cases due to the differences in computing power. Comparing our work with the related work, we saw differences in power consumption due to the different consensus algorithms used. For future work implementing Proof of Authority (PoA), similar to Remote Proof-of-Work in IOTA, would better compare results against current SOTA literature. Finally, it would be valuable to evaluate and compare how resource-limited devices perform on other frameworks like Solana [15] and test different consensus algorithms.

References

[1] Z. Cui, X. Fei, S. Zhang, X. Cai, Y. Cao, W. Zhang, and J. Chen, "A hybrid blockchain-based identity authentication scheme for multi-wsn," *IEEE Transactions on Services Computing*, 2020.

[2] M. A. Lopez, M. Baddeley, W. T. Lunardi, A. Pandey, and J.-P. Giacalone, "Towards secure wireless mesh networks for uav swarm connectivity: Current threats, research, and opportunities," in *2021 17th International Conference on Distributed Computing in Sensor Systems (DCOSS)*, 2021, pp. 319–326.

[3] . Blockchains, "Blockchain Size: Everything You Need To Know," 2021. [Online]. Available: https://101blockchains.com/blockchain-size/

[4] "Go Ethereum: Official Go implementation of the Ethereum protocol," https://geth.ethereum.org/, 2021.

[5] M. Kadadha, H. Otrok, R. Mizouni, S. Singh, and A. Ouali, "Sensechain: A Blockchain-based Crowdsensing Framework for Multiple Requesters and Multiple Workers," *Future Generation Computer Systems*, 2020.

[6] M. T. Hammi, B. Hammi, P. Bellot, and A. Serhrouchni, "Bubbles of Trust: A Decentralized Blockchain-based Authentication System for IoT," *Comp. & Sec.*, 2018.

[7] H. Anwar, "Blockchain Proof of Concept: Enterprise POC Guide," 2021. [Online]. Available: tinyurl.com/3smhftrj

[8] S. Popov, "The tangle," 2018. [Online]. Available: tinyurl.com/56ye8b9p

[9] N. Satoshi, "Bitcoin: A Peer-to-Peer Electronic Cash System," *Consulted*, 2008.

[10] L. Hang and D.-H. Kim, "Design and Implementation of an Integrated IoT Blockchain Platform for Sensing Data Integrity," *Sensors*, 2019.

[11] G. Iredale, "Introduction to Permissioned Blockchains," 2019. [Online]. Available: https://101blockchains.com/permissioned-blockchain/

[12] N. Ramachandran, "Energy Benchmarks for the IOTA Network (Chrysalis Edition)," 2021. [Online]. Available: https://blog.iota.org/internal-energy-benchmarks-for-iota/

[13] K. Özyılmaz and A. Yurdakul, "Work-in-Progress: Integrating Low-Power IoT Devices to a Blockchain-based Infrastructure," 2022. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8094378/

[14] A. Elsts, E. Mitskas, and G. Oikonomou, "Distributed Ledger Technology and the Internet of Things: A Feasibility Study," in *Proc. of the 1st Workshop on Blockchain-enabled Networked Sensor Systems*, 2018.

[15] A. Yakovenko, "Solana: A New Architecture for a High Performance Blockchain," 2021. [Online]. Available: https://solana.com/solana-whitepaper.pdf