Communication Protocol Impact on Energy Efficiency of Blockchain Application

Analysis of Energy Efficiency of Android Blockchain Application Using UDP or QUIC

Tomasz Puczel[1]

Supervisors: Johan Pouwelse[1]**, Bulat Nasrulin**[1]

[1]**EEMCS, Delft University of Technology, The Netherlands**

A Thesis Submitted to EEMCS Faculty Delft University of Technology,
In Partial Fulfilment of the Requirements
For the Bachelor of Computer Science and Engineering
June 22, 2025

Name of the student: Tomasz Puczel
Final project course: CSE3000 Research Project
Thesis committee: Johan Pouwelse, Bulat Nasrulin, Examiner: Koen Langendoen[1]

Abstract

Blockchain is already a widely adopted solution which can achieve decentralized storage in trustless settings. However, it is infamous for its high energy demands, making it difficult to operate on mobile phones with limited battery life. Among many design decisions in a blockchain implementation, is the choice of the communication protocol used for messaging between the participants. This work systematically evaluates the energy efficiency of custom blockchain implementations: over UDP and over QUIC with Iroh Rust crate for Android devices. The results demonstrate a higher energy demand by the QUIC-based implementation, with up to 34% faster battery discharge on average compared to the UDP approach. Investigations show different energy usage per CPU and Wi-Fi components, and network traffic profile during the application's operation. These findings highlight the trade-offs related to energy efficiency in designing a blockchain-based system.

1 Introduction

Blockchain is a technology that enables storing data without requiring any central authority or pre-established trust between parties [1, 2]. In such systems, the participants (also referred to as nodes or users) store all or a part of a ledger of immutable blocks and collectively agree on a current state of truth through a certain consensus mechanism. Blockchain technology has seen increasing prevalence and has already been researched and adopted in many areas, including healthcare [3], energy systems [4], and most prominently in cryptocurrencies, such as Bitcoin [1] or Ethereum [5]. Despite the shared foundation, there exist differences between the implementations, for example, in the areas of cryptography, reaching a consensus, the users' interactions, and the underlying communication between the nodes.

One of the implementation decisions when designing a blockchain-based system is the choice of the communication protocol used for the pairwise interactions between the nodes. Blockchain uses peer-to-peer networking, where various protocols offer different characteristics with certain trade-offs. The selection between a lightweight but unreliable protocol such as UDP (User Datagram Protocol) or a more robust one like QUIC, can have further implications on various aspects, such as the system's complexity or energy consumption.

A functional communication protocol, together with other components, allows for running the system on a live network. Currently, these networks are mainly composed of desktop or server-grade devices [6]. However, smaller devices, such as smartphones and other mobile or embedded ones, are gaining prevalence these days. Smartphones are commonly used in a range of daily tasks such as banking, e-commerce, or even electronic identity, while embedded devices are used in industrial systems. All these applications could benefit from blockchain adoption on such target devices, but a common characteristic of these is the constrained resources. Aside from storage or computation capabilities, a crucial factor is the battery life, making the energy efficiency of blockchain solutions an important concern.

At the same time, blockchain technology faces a few serious limitations related to its scalability [7], sustainability of the whole system, and resource management on an individual node. Energy efficiency is a critical aspect of blockchain's operation on a smartphone, and is one of the factors often mentioned when considering lightweight blockchains [8]. Among the many factors that influence it are computation, storage, and network communication. However, the impact of the specific communication protocol choice and other related factors in a blockchain application for Android smartphones remains underexplored.

This research investigates the energy efficiency of a simplified blockchain implementation for Android devices, inspired by a certain blockchain design (TrustChain) and built collaboratively from scratch as a part of this work. The primary focus is on the underlying communication protocols, but other factors are investigated as well. In other words, this research paper addresses the following questions:

- How energy efficient is a TrustChain-inspired blockchain implementation on Android mobile devices?
- How do UDP and QUIC protocols affect the energy efficiency of such a blockchain application?
- What factors contribute to the energy efficiency of such implementations the most?

The structure of this paper is as follows. Section 2 presents the relevant background, Section 3 reviews related work in this area. Then Section 4 explains the methodology of the research. The findings are presented and discussed in Section 5 and Section 6 respectively. Section 7 reflects on the integrity and ethics of the research. Section 8 suggests further research directions and lastly, Section 9 concludes the paper.

2 Background

This section outlines the foundational concepts and system components relevant to the research conducted and the approach taken in this study. We begin by examining blockchain technology, with particular emphasis on its core components and its evolving architectural variations. Subsequently, we analyze the networking layer, which is selected as the main experimental variable in this study, given its role in peer-to-peer messaging. Finally, we explore the energy efficiency aspect with a particular focus on its relevance to mobile platforms, including a discussion of possible limiting factors, and blockchain-induced overheads.

2.1 Blockchain Technologies

Blockchain Fundamentals

Blockchain technology makes anonymized, trustless, and decentralized data storage possible. A few concepts interact together in order to achieve these goals.

In a general case of a blockchain, participants of the network are the devices identified by their public key, that are cryptographically linked to their private key. Data is stored in structures called blocks, which form a linear sequence, chained together in an immutable order. Each block contains a few fields to make the whole system functional.

The payload is a number of bytes (specific and subject to interpretation per every particular application). Next, each block, except for the first *genesis block*, includes the hash of the previous block. This is computed using a secure cryptographic hash function, which in turn, places the block in a specific, immutable position in the whole chain. Moreover, each block is digitally signed by its creator using their private key, while all other nodes that receive a block can verify its signature's correctness by using the sender's public key. Note that public keys serve as a way to address participants, so this system provides a degree of anonymity [9].

Yet another crucial component of the blockchain is the consensus mechanism, which determines how the nodes agree on a single version of the ledger. Bitcoin utilizes the Proof-of-Work (PoW) process where the nodes need to perform intensive computations (*mining*) to propose a new block. To this date, many other consensus algorithms have been devised [10], including Proof-of-Stake (PoS), which another widespread cryptocurrency - Ethereum - switched to in 2022 [11].

Similarly to how consensus mechanisms can vary significantly between blockchain systems, any other part of the whole architecture can be different as well. This can include the cryptographic algorithms used or the transport protocol used to transfer the blocks between nodes which is the main focus of this work.

TrustChain Architecture

TrustChain, as developed in the study by Otte et al. [12], is another blockchain-based idea capable of performing trusted transactions. In this design, instead of preventing fraud using a typical consensus algorithm, the objective is to make the fraud detectable. A model to determine the trustworthiness of the participants was introduced as part of the specification. Moreover, in this architecture, nodes only maintain their own chain of blocks, but each block references both the sender's and receiver's previous block. More on the guarantees and characteristics of TrustChain can be found in the original work.

This model served as an inspiration for the implementation developed for the purpose of this research. More details on the implementation can be found in Section 4.

2.2 Networking and Communication Protocols

Communication between the nodes is the backbone of any distributed system. In centralized applications, the communication between the clients relies on central servers which relay messages. In contrast, peer-to-peer (P2P) networks, such as those used in blockchain systems, offer a decentralized approach, where messages in principle are transferred directly between equivalent nodes. In order to facilitate transfers between specific applications a different communication protocol can be used, which can have significant implications on the performance of the application.

In this research two protocols have been selected for the main comparison: UDP and QUIC. Note that other protocols could be analyzed as well, however, they were left out of scope and are suggested in Section 8 as further research direction.

UDP

UDP (User Datagram Protocol) is a lightweight, connectionless protocol used to deliver packets to other destinations on an Internet Protocol network [13]. While it offers checksums for data integrity, it provides no guarantees on ordering or reliability of the transfer itself. Its simplicity results in low overhead, however, can lead to inefficiencies or not delivering data over an unreliable network.

QUIC

QUIC [14] is a modern, relatively recent transport protocol developed by Google. It builds on top of UDP, which allows for easy adoption, and is extended to support reliable and multiplexed communication, and includes TLS 1.3 for encryption. Unlike traditional TCP (Transport Control Protocol), QUIC removes some of the overhead introduced, for example, the round-trips during connection setup [15]. It also supports flow and congestion control, which makes it an attractive alternative for certain applications. In 2017 it was estimated that 7% of the whole internet traffic was transported over QUIC [15].

2.3 Energy Efficiency

Energy efficiency is an important aspect of all software solutions. On the one hand, it affects the overall sustainability of the whole system, which is highly relevant in the contemporary world. Moreover, energy consumption is one of the factors that determines whether a certain application is suitable for a particular type of devices.

With the growing use of mobile and other Internet of Things devices, many applications can benefit from the support of these platforms. However, such devices usually have limited resources available. This includes their computation power, storage available, and, most relevant to this research, battery life. Smartphones in particular are an important device in people's daily lives, but a trade-off of their convenience is their limited energy budget. For this reason, highly energy-draining applications are not feasible for operation, as they can lead to too fast discharging and cause overheating.

Key factors to the energy efficiency of mobile applications include: screen time, CPU utilization, and network traffic [16]. Since a blockchain application must receive, send and process data packed in the blocks, these activities are inevitably a part of the application's lifecycle.

3 Related Work

Recent years have seen a number of research studies focusing on the energy consumption of blockchain solutions, with the goal of evaluating their implementations or assessing their suitability for constrained devices.

Ometov et al. [17] stated that "the use of constrained devices is generally underestimated in the context of blockchain". Their work surveyed existing blockchain designs suitable for smartphones and introduced a novel consensus mechanism. In their work, they compared it against existing Proof-of-Work implementations and they found that mechanisms based on Proof-of-Work algorithms significantly deplete smartphone battery capacity, whereas consensus based on Proof-of-Authority algorithm has little impact on smartphone's battery life. They performed and provided a detailed description of measurements of voltage, temperature, and battery level on a number of Android devices.

Another research by Sedlmeir et al. [18] pointed out that the power consumption of Bitcoin was at particularly high levels. It identified that work performed by every single node, particularly to mine the new blocks, as the main contribution to the overall power consumption, which was also validated by Bada et al. [19]. Other research also mentioned that verification of the incoming block proposals [20] is among the most energy-draining tasks of operating a blockchain node. These studies build an understanding of the overall energy consumption but are not specific to the case of communication protocol impact on a smartphone's energy profile.

Similarly, Escobar et al. [21] focused on measuring the energy consumption of the specific hashing and Proof-of-Work computations using Running Average Power Limit technology. They proposed certain modifications to the algorithms used in Merkle Tree hashing and mining computations. In their work, they obtained up to 20% improvements for these cases. This work targeted power consumption of specific aspects of the whole implementation, but did not consider the communication aspect nor did it evaluate the case of smartphones.

Some research has been done into a comparative analysis of communication protocols with evaluations of their potential energy efficiency impact. The study by Kumar and Dezfouli [22] indicated that using QUIC in their use-case of MQTT reduced processor usage by up to 83% as compared to TCP. On the other hand, the research by Sun et al. [23] indicated that receiving packets with UDP has a very low impact on devices' energy consumption, while TCP imposes significant cost.

Research Gap

To the best of the author's knowledge, existing literature lacks research into the energy efficiency of a blockchain implementation for smartphones with the main focus on the underlying communication protocols. This work addresses this identified research gap, by performing systematic measurements and comparing the energy consumption across different protocols. The details of the approach are outlined in the next section.

4 Methodology

This section outlines the methodology employed to conduct this research on the energy efficiency of blockchain implementation for smartphones with a particular focus on the impact of communication protocols. It first describes the custom blockchain implementation used for this study. Next, it details the measurement setup, including the devices, tools, and energy-related metrics collected. Finally, the experiment design is presented, covering the preparations, scenarios, and automation procedure.

4.1 Implementation

Although working blockchain platforms exist, they are often too complex and not easily customizable, which makes it infeasible to analyze certain isolated aspects under controlled, custom-defined conditions. Hence, in order to allow for precise control over protocol-level behavior and for energy profiling required for this research, a custom, simplified blockchain implementation was developed for Android smartphones (Section 8 proposes research into other platforms that were left out of scope). A similar approach was taken in prior studies [24].

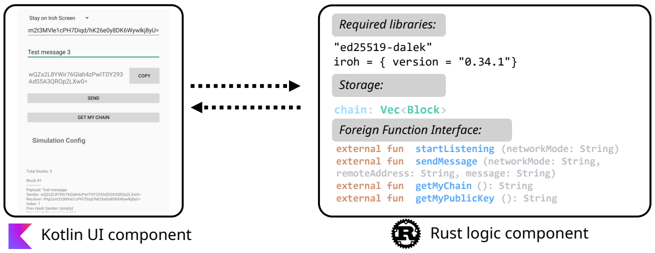
Another alternative considered was the analysis of publicly available datasets about the energy efficiency of blockchain solutions for smartphones. However, to the best of the author's knowledge, no such datasets exist which would allow for this research's questions investigation.

Design Overview

On a high level, the Android smartphone application operates through two interconnected components, as illustrated in Figure 1.

The first is a lightweight layer of Kotlin code, that provides the user with an interface to interact with the app and captures user's input, which mimics a proper application. All actions requiring some blockchain logic are delegated to the Rust component. Its facade is exposed via the Java Native Interface, serving as the Foreign Function Interface. The Rust code in turn provides the blockchain functionalities, which include: blocks storage, hash computation, cryptographic signature generation and verification, and peer-to-peer communication over the network.

This modular setup, with a clearly defined contract between the components, increases portability and allows for further reuse of the same logic.



Stay on Iroh Screen

m2t3MVle1cPH7Diqd/hK26e0y8DK6Wywlkj8yU=

Test message 3

wQZs2L8YWir76Glah4zPwITDY293AdS5A3QROp2LXw0=

Simulation Config

Total blocks: 3

Block #1

Payload: Test message
Sender: wQZs2L8YWir76Glah4zPwITDY293AdS5A3QROp2LXw0=
Receiver: PAp2um2t3MVle1cPH7Diqd/hK26e0y8DK6Wywlkj8yU=
Index: 1
Prev Hash Sender: (empty)

Required libraries:

```
"ed25519-dalek"
iroh = { version = "0.34.1"}
```

Storage:

```
chain: Vec<Block>
```

Foreign Function Interface:

```
external fun  startListening (networkMode: String)
external fun  sendMessage (networkMode: String,
remoteAddress: String, message: String)
external fun  getMyChain (): String
external fun  getMyPublicKey (): String
```

Figure 1: Implementation overview

TrustChain Logic Implementation

The block structure in this research is inspired by the TrustChain design. Each block encapsulates the desired payload and other fields necessary for correct operation of the whole blockchain (also shown in Figure 2):

- sender_pubkey, receiver_pubkey - public keys of the sender and receiver respectively
- sender_sign, receiver_sign - cryptographic signatures of the block by the sender and receiver (using their respective secret keys)

– `prev_hash_sender`, `prev_hash_receiver` - hashes pointing to the previous block in the respective chains

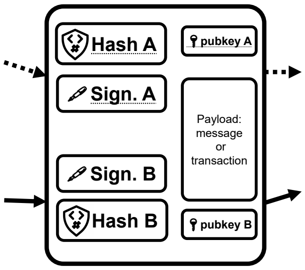


Figure 2: Block structure visualized

Again inspired by the TrustChain design, in the developed implementation the messages are exchanged directly in sender-receiver pairs, rather than being broadcast to all the nodes.

A message exchange is initiated by one of the nodes on the network. The sender wraps the payload in a block, containing their signature and hash of their previous block, which is next sent to a desired destination, identified by a public key. Upon receiving a block, the receiver validates the signature provided by the sender and sends back a completed block with their previous hash and their signature included. Lastly, the sender verifies the receiver's signature. The nodes proceed to store the completed blocks in their own chains. The entire process is illustrated in Figure 3.

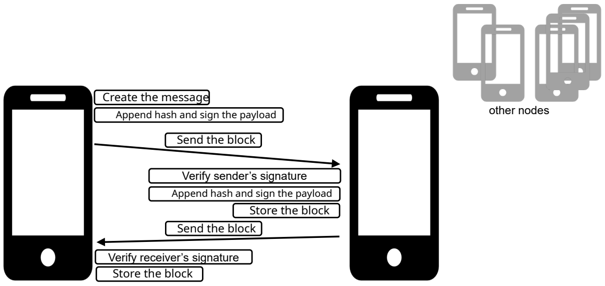

other nodes
Create the message
Append hash and sign the payload
Send the block
Verify sender's signature
Append hash and sign the payload
Store the block
Send the block
Verify receiver's signature
Store the block

Figure 3: Communication and logic overview

Resilience against malicious actors and other attacks was not in the scope of this project; addressing these concerns is left for future work (see Section 8). The measurements were performed using this simplified implementation, assuming only trustworthy nodes.

Communication Protocols Overview

The main research objective is to evaluate the impact of the underlying protocol used for peer-to-peer communication. As argued in section 2.2, in this research two communication protocols: QUIC and UDP, were selected for comparison. The developed application supports two separate, independent modes, each corresponding to one of the protocols.

In order to facilitate the communication over UDP, upon app creation a UDP socket is opened by the Rust component. In an indefinite loop, incoming datagrams are received into a buffer using Rust's `recv_from` method. Similarly, outgoing messages are sent to a remote address via the `send_to` method which wraps the block in a UDP datagram.

Support for QUIC protocol is achieved using the Iroh[1] Rust crate. Iroh abstracts away node discovery and the NAT traversal needed for peer-to-peer messaging. Upon application startup, an Iroh Endpoint is instantiated along with an associated public-private-key pair. This Endpoint is used to read messages via a receiving stream after opening a connection. Similarly, to send a message, a connection to a remote node ID is opened and the bytes representing the block are written to the sending stream.

4.2 Measurements Setup

The measurement setup was crucial for conducting reproducible and insightful research on the energy efficiency of blockchain implementations for Android smartphones.

Devices Used

Physical smartphones were chosen for measurements as opposed to emulators. This more accurately mirrors [25] the behavior of a desired blockchain application deployed on real-world target devices with the energy sourced from an actual smartphone's battery.

The specifications of the two selected devices that were available and used for the experiments are summarized in Table 1.

Device Model	Role	Android version	Battery capacity (total)
Samsung A50	Sender	11	4000 mAh
Xiaomi 11S	Receiver	11	5000 mAh

Table 1: Device specifications

Tools and Metrics

Multiple tools exist for collecting energy efficiency-related metrics for an Android smartphone application. This section presents the tools and available metrics ultimately selected for measurements and briefly mentions other tools that were considered but not used.

First of all, Battery Manager from Android[2] and an Android broadcast with battery status changes provide access to several instantaneous or moving average battery parameters about battery and charging properties. The following were identified as the most insightful ones for energy profiling:

- charge counter, battery level - instantaneous remaining battery capacity and percentage left
- average current - (moving) average of battery current; the sign indicates charging or discharging
- voltage - 5-second median of the battery voltage level
- temperature - 5-second median of the battery temperature

[1] https://www.iroh.computer/

[2] https://developer.android.com/reference/android/os/BatteryManager

These metrics were gathered every 60 seconds throughout the simulations and saved to a file.

Other available metrics, such as instantaneous battery current or health statuses, were not chosen due to their high correlation with the selected parameters, instability in reading or limited relevance to the analysis.

Another native Android tool that provides information about system services is `dumpsys`. By specifying the `batterystats` service through the Android Debug Bridge (`adb`) utility, recent statistical data about the battery usage is written to a file. On top of the global statistics, this also provides a breakdown of per-process (per UID) energy and network insights, including the capacity drains attributed to certain categories (e.g., CPU, Wi-Fi), or the total number of packets sent and received.

System traces recorded using a native system tracing utility were also evaluated as a source of fine-grained device activity insights over a short period of time. They offer a comprehensive overview of the smartphone's behavior, down to threads' activity level. This method was only used for investigation of short periods of activity, due to a large size of generated reports.

A few alternative measurement approaches and tools were recognized, but not integrated in the research due to the limitations; analysis using these is suggested in Section 8.

Power Profiler[3] and macrobenchmarks[4] with related PowerMetric metrics are both capable of reporting energy-related data across specific categories (e.g., CPU, display, memory, network) but were not available on the devices used in the experiment; the documentation specifies these are supported on Pixel 6 and subsequent models.

Battery Historian was a tool developed by Google for visualization of Android `bugreports`, however, it is no longer supported.

Hardware measurement tools, such as the Monsoon Power Monitor, provide highly precise energy consumption data by physical measurements drawn from the battery, however, they were not available to employ in this study.

4.3 Experiment Design

In order to ensure reproducibility and reduce the risk of human errors, the experiments were triggered and configured using a script. A dedicated simulation configuration component was developed in the user interface of the app which allowed for running simulations of exchanging blockchain messages between two nodes with customizable input parameters. Adjustments were possible for:

- block payload size - number of bytes in the payload
- transmission rate - number of blocks sent per second
- termination criterion - governs whether the simulation would stop upon reaching: a predefined duration of the test, number of messages, battery level threshold, or continue running indefinitely

The automated simulation supported both underlying implementations. This allowed for systematic testing with both communication protocols and in turn, a comparison of these.

Every experiment followed a standardized procedure:

1. All other applications and non-essential services (e.g., Bluetooth) were terminated or switched off.
2. The desired destination public key (and possibly remote address) was configured.
3. The desired configuration values were set, with the following default values: 256 bytes of payload, target of 10 messages per second.
4. The screen brightness was set to maximum.
5. The previous battery data gathering by `dumpsys batterystats` was reset.
6. The simulation was initiated.

The default parameters were chosen arbitrarily to reflect a plausible operational blockchain implementation and remained consistent across all the experiments conducted during this research. Experiments with different parameters are among the suggested future work directions.

During the simulation, the metrics were collected from the device initiating the block exchanges. This node creates and sends the blocks to a predefined destination and actively listens for the blocks, verifies the incoming ones, and stores them if they are correct, as described in section 4.1.

The experiments were repeatedly performed with the default parameters for both underlying communication protocols. During each of them, at 60-second intervals, the relevant metrics were queried from the Battery Manager and persisted in a file. At the end of each of the simulations, a summary of battery data was fetched from `dumpsys batterystats`. Independently, when a simulation with the same configurations was running, a more detailed system trace over a short time was recorded for further analysis.

5 Findings

This section reports the results from multiple experiments, conducted using the methodology described in Section 4. They quantitatively compare the energy efficiency of the blockchain application using QUIC with Iroh and UDP as the underlying communication protocol.

Section 5.1 presents the metrics collected over multiple long-duration runs. Section 5.2 analyzes the energy consumption breakdown per category.

During the research, eight long-duration simulation runs were executed on a Samsung A50 smartphone. Each simulation lasted at least 6000 seconds (100 minutes) and used the same default parameters. Table 2 details all the experiment runs conducted to collect the data.

Although some runs exceeded 6000 seconds, they exhibited the same behavior and patterns throughout. Therefore, to allow for comparison of all eight runs within a common time frame, the results were truncated at 6000 seconds in this analysis, without any loss of relevant insights. The complete datasets are available alongside the codebase[5].

[3] https://developer.android.com/studio/profile/power-profiler

[4] https://developer.android.com/topic/performance/benchmarking/macrobenchmark-metrics#power

[5] https://github.com/tomDelftRP/blockchain (access: June 2025)

No.	Specification	Protocol	Total duration (s)
1	Device: Samsung A50 Target rate: 10 blocks / s Payload size: 256 bytes	QUIC with Iroh	6200
2			14000
3			12900
4			11300
5		UDP	7600
6			12700
7			11000
8			21000

Table 2: Details of the experiment runs

5.1 Long-term Battery Metrics

Firstly, Figure 4 presents the battery capacity level over time. The graph indicates that with either of the communication protocols, the battery drain was rather steady and linear. Over 6000 seconds, when the implementation with UDP was running on average about 881 mAh were drained, while in the case of QUIC with Iroh, this was 1183 mAh of the battery's initial 3746 mAh capacity. These correspond to 24% and 32% of the battery capacity respectively. The capacity drop is about 34% higher for the implementation using QUIC with Iroh.

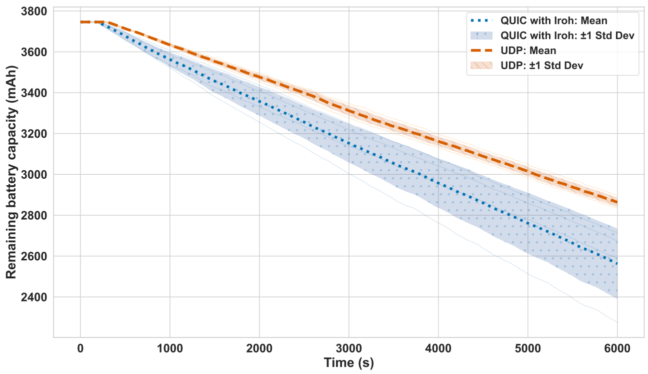


Figure 4: Average battery capacity remaining (mAh)
In this and the next charts, the thick line represents the arithmetic mean of the data across all runs. The shaded and hatched areas denote the range of one standard deviation. Thin, semi-transparent lines represent individual experiment runs

This discharge is caused by the current flowing out of the battery, which is illustrated in Figure 5. Initially, in the first 60 seconds, a substantial drop down to approximately -850 mA for UDP and -1250 mA for QUIC with Iroh is observed. In the subsequent measurements, in the case of UDP, the current stabilizes at the level of about -450 mA with a standard deviation of about 20 mA. The QUIC with Iroh implementation is less stable, with the mean fluctuating between -550 mA and -725 mA and a standard deviation of about 100 mA. Two of the runs using QUIC with Iroh had much higher variability over time, while the other ones were more stable, but still at a higher level of outflowing current.

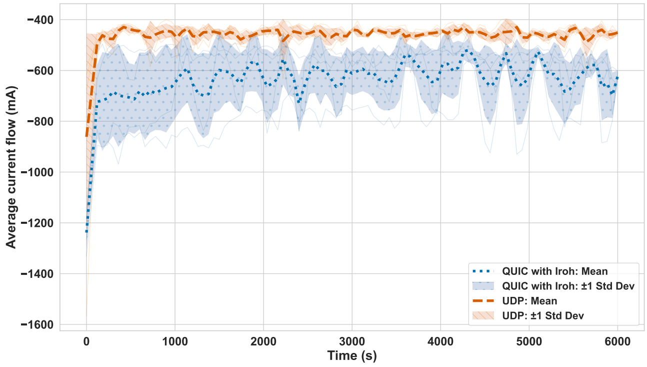


Figure 5: Average current from the battery (mA)

The output voltage of the battery steadily decreases over time, with a significant initial drop, as shown in Figure 6. The initial drop is larger in the case of QUIC with Iroh and is followed by a slightly faster decline. After 6000 seconds voltage decreased from initial 4.25 V to 3.89 V for the UDP implementation, and to 3.82 V for the QUIC with Iroh one.

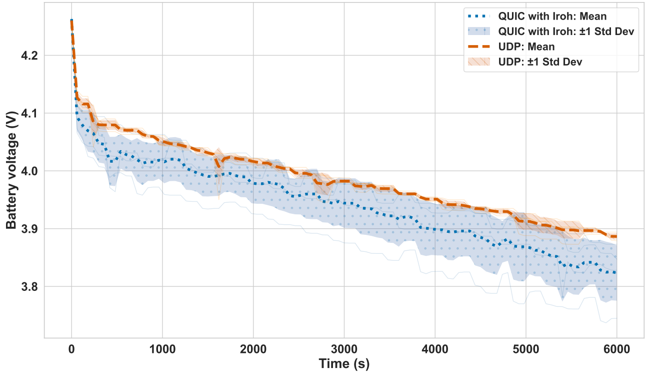


Figure 6: Average battery voltage (V)

Figure 7 depicts that in the initial 6000 seconds, the battery temperature rises rapidly at the start, then typically slows down or stops, or even slightly fluctuates in the longer runs. The total temperature increase for QUIC with Iroh ranged between +7 °C and +10.5 °C with an average of +9.3 °C, while the runs with communication over UDP resulted in only a +4.5 °C to +8 °C increase with an average of +6.7 °C.

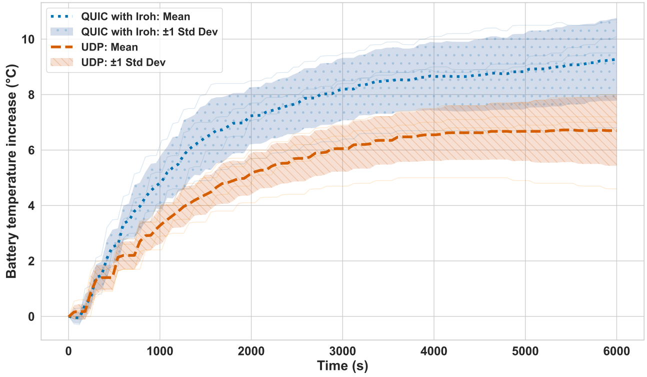


Figure 7: Average battery temperature increase (°C)

5.2 Components Analysis

The breakdown of the application's estimated power use per CPU and Wi-Fi system component (the only two provided by `dumpsys batterystats`) over the simulations for both implementations is shown in Figure 8.

Consistently across all the runs, the implementation using QUIC with Iroh drained more battery capacity than the UDP alternative, confirming the findings from Section 5.1.

The reported breakdown also shows that the proportion consumed by the CPU is significantly higher for the implementation using QUIC with Iroh, ranging from 82% to 85% with an outlier of 62% in the third run. On the other hand, for UDP trials the proportion of energy drained by the CPU is consistently between 7.9% and 9.9%, with the complimentary Wi-Fi component accounting for over 90%.

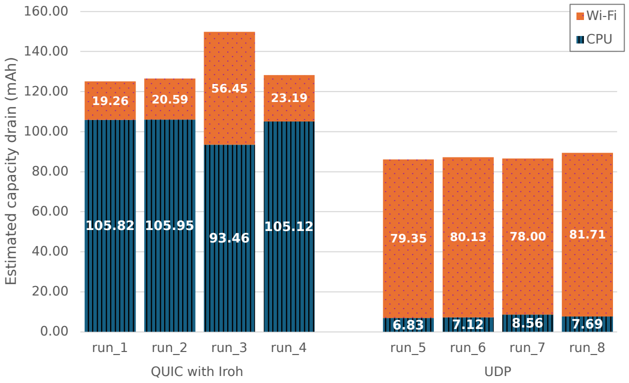


Figure 8: Estimated capacity drain (mAh) per CPU and Wi-Fi

To take a step towards understanding the underlying factors causing these differences, the network traffic summary provided by `dumpsys batterstats` was firstly analyzed and, as expected given the properties of the protocols, exhibited notable differences between the two implementations. UDP sends and receives precisely 1 packet per block, while the implementation using QUIC with Iroh averaged between 3.7 and 4.4 of sent and received packets per block when reusing the same connection, and even up to an average of 14 packets when creating a new connection per every block. This network traffic contrasts with both the proportions and the nominal values of the capacity drops attributed to the Wi-Fi category by the `dumpsys batterstats` utility. Lastly, analysis of a 20-megabyte system trace recorded over 20-second periods for both of the implementations, shows a higher number of threads spawned by the QUIC with Iroh implementation: 25 compared to 21. This difference, partially explained by the developed code, can be a factor contributing to the higher energy consumption, particularly the portion attributed to the CPU.

Section 8 encourages further research into low-level causes and properties specific to the underlying communication protocol which explain the impact on the energy profile of a blockchain application.

6 Discussion

The main objectives of this work were to evaluate the energy efficiency of the proposed blockchain application for Android smartphones, to compare the impact of different communication protocols (specifically UDP and QUIC with Iroh) used for peer-to-peer messaging, and to understand the factors affecting the energy efficiency. The findings outlined in Section 5 provide a basis for addressing these aspects.

The experiment results provide evidence that continuous operation of the application, which involves exchanging messages, performing cryptographic operations, and persisting the data, increases the device's power consumption. Both implementations caused additional battery drain, resulting in a faster discharge of the battery, confirming that even such a simplified implementation imposes a considerable energy burden on a mobile device.

A comprehensive comparison between the proposed implementations using UDP and QUIC with Iroh as the underlying communication protocols demonstrates significant differences in energy usage and related metrics. Across all the evaluated metrics - in particular, battery capacity drain, current flow, and voltage level - the implementation using UDP was found to be less energy-demanding. Specifically, when utilizing QUIC with Iroh the average discharge was 34% higher than when using UDP. Further investigation shows a significantly higher proportion of energy consumed by CPU (usually between 82% - 85%) when using QUIC with Iroh while for UDP implementation more power use was attributed to the Wi-Fi component, despite much lower network traffic.

These findings confirm that an operating blockchain contributes to the smartphone's energy consumption, which is in line with the previous research by Ometov et al. [17]. The analysis also indicates that there exists a significant difference even when varying just one implementation aspect, namely the communication setup. Moreover, further inspection showed that even a single detail such as reusing the connection significantly affects the network traffic which might affect the power consumption.

The results highlight the need to consider the trade-offs between energy efficiency and reliability and other features offered by some protocols. While many studies point out that QUIC is comparably or more energy efficient than TCP [26, 27], this study shows that lightweight UDP has even less overhead. As could be expected, the increased network traffic and CPU utilization in the case of QUIC with Iroh are likely causing the more intensive use of the smartphone's battery. Further analysis of the properties and steps of the protocols involved in the application's operation and the related trade-offs are suggested as a future research direction.

The continuous operation of such a blockchain application was shown to considerably affect the usability of the mobile device with a communication protocol appearing to be a factor contributing to the energy consumption. Shorter battery discharge time and quick temperature rise are of primary importance for smartphones and other devices with limited resources and hence need to be taken care of when making design choices, such as about the communication protocol.

7 Responsible Research

This section reflects on the academic integrity, and ethical and environmental concerns associated with this research. It emphasizes the reproducibility and replicability of the process carried out and considers the ethical and environmental implications of the experiments and findings.

LLM usage LLMs were used in this research to improve the language in the report, explain certain concepts, and provide examples of code snippets. Every output was critically assessed and reviewed by the authors.

7.1 Reproducibility and Replicability

All the data for this research was exclusively collected as part of the work, as opposed to using publicly available datasets. To ensure replicability, the Methodology section explains all the design decisions and simplifications, as well as provides a detailed account of the steps of the experimental process. This includes specifications of the devices used, the required preparations and configurations for an experiment, and the tools and metrics used for assessment. The experiments were conducted within the described conditions, and the recorded results were transparently reported. Moreover, care has been taken to describe the code used for performing the evaluation, and the codebase itself, including versioned dependencies and the collected data, is publicly available online[6].

These measures stay consistent with core scientific values - transparency and honesty, which are outlined in the TU Delft Code of Conduct[7].

Such information is sufficient to replicate the setup and evaluations, as long as the utilized tools stay in use and are available. Devices in a different condition or with different hardware specifications might introduce variability.

7.2 Ethical Aspects

The research did not involve any sensitive data or human participation. The data collection process only involved physical devices. Nonetheless, this work does intersect with a broader ethical context.

The results of this research can contribute to further development of blockchain technology, in particular their adoption on smartphones. While innovations towards decentralization might be desirable, such technologies are currently under smaller control of the authorities, which may lead to an increase in unregulated transactions that can be used for illegal purposes [28, 29].

7.3 Environmental Aspects

As it was noted before, currently the blockchain technology has significant energy demand. This research aims to contribute toward more energy efficient solutions which can have a positive impact on sustainability. However, the multiple experiments carried out throughout this research have a direct environmental cost.

The experiments involved discharging the battery and consequently using up the energy to charge it back again. Although on a global scale, the energy consumption of the experiments of this research was minimal, it was nonetheless energy consumed solely for this purpose. Lastly, the repeated experiments might have affected the battery health of the devices used.

Future studies should consider this aspect when performing experiments, especially when on a larger scale.

8 Future Work

This research was conducted using a specific approach and several simplifications (described in Section 4) due to feasibility constraints. Based on the limitations and the findings, several suggested future work directions are listed below:

- Expand blockchain implementation, integrating consensus mechanism or trustworthiness calculations, full chain verification, and malicious parties tolerance - to evaluate their impact on energy consumption.
  Moreover, optimize the developed blockchain implementation using energy-aware techniques.
- Improve measurement setup: consider leveraging a warm-up phase, and other types of profiling tools and repeat the tests under varied conditions (e.g., parameters, network).
- Investigate direct causes of the energy consumption, narrowing it down to concrete components of the application or the phone's activity and isolating the network communication aspect further.
- Explore other alternatives for the communication protocol (e.g., TFTP, TCP, uTP, DCCP) and for the other aspects of the whole blockchain architecture.
- Evaluate energy efficiency on various mobile platforms (iOS), smartphone models, and other resource-constrained devices.
- Analyze more extensively the impact of concrete properties of the examined protocols and the trade-offs related to employing different ones as the underlying communication implementations.

9 Conclusion

Blockchain technology changes the way the data is stored, and is capable of achieving this without a central authority and pre-established trust. It is already deployed in applications such as cryptocurrencies, which are widely used in society. However, most of the current designs are highly energy intensive, making the technology unsuitable for the whole class of resource-limited devices, in most cases also smartphones. An energy efficient implementation suitable for such devices would enable them to participate in these networks and support the decentralization movement.

This study explores and evaluates the impact of two alternative choices for the underlying communication protocols used in peer-to-peer messaging implementation, namely UDP and QUIC with Iroh Rust crate.

[6] https://github.com/tomDelftRP/blockchain (access: June 2025)

[7] https://www.tudelft.nl/en/about-tu-delft/strategy/integrity-policy/tu-delft-code-of-conduct

A standardized, replicable experiment demonstrated a higher energy demand of the implementation over QUIC with Iroh in comparison to the one over UDP. Across conducted experiments, the discharge was happening at a 34% faster rate and the battery temperature rose on average 2.6 °C more.

The measurement tools also indicated that the implementations differed in the CPU and Wi-Fi related energy consumption. QUIC over Iroh exhibited a significantly higher proportion of energy drain of the app consumed by the CPU and smaller Wi-Fi related consumption than the UDP implementation, despite larger network traffic. The values, however, changed after introducing a connection reuse mechanism, which proved that certain implementation decisions can affect the energy profile of the app significantly. Overall, it is evident that the protocol choice itself influences the overall app's energy consumption and has implications on the device's usability as well as trade-offs with the application's functionality.

This work contributes to the body of knowledge on the energy efficiency of a blockchain application for Android smartphone devices with a focus on underlying communication protocols. It provides a methodology to both develop and evaluate a simplified blockchain design. Due to time constraints, the scope of this research was limited, but it is highly encouraged to pursue this direction further, with a more robust architecture and with extended measurement setup.

Maybe in the near future, a decentralized world with smartphone devices participating in blockchain networks and not excessively draining their batteries will be the reality.

References

[1] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system. *Cryptography Mailing list at https://metzdowd.com*, 03 2009.

[2] Aaron Wright and Primavera De Filippi. Decentralized blockchain technology and the rise of lex cryptographia. *Available at SSRN 2580664*, 2015.

[3] Cornelius C Agbo, Qusay H Mahmoud, and J Mikael Eklund. Blockchain technology in healthcare: a systematic review. In *Healthcare*, volume 7, page 56. MDPI, 2019.

[4] Moein Choobineh, Ali Arabnya, Behrouz Sohrabi, Amin Khodaei, and Aleksi Paaso. Blockchain technology in energy systems: A state-of-the-art review. *IET Blockchain*, 3(1):35–59, 2023.

[5] Vitalik Buterin et al. Ethereum white paper. *GitHub repository*, 1(22-23):5–7, 2013.

[6] Sehyun Park, Seongwon Im, Youhwan Seol, and Jeongyeup Paek. Nodes in the bitcoin network: Comparative measurement study and survey. *IEEE Access*, 7:57009–57022, 2019. doi: 10.1109/ACCESS.2019.2914098.

[7] Kyle Croman, Christian Decker, Ittay Eyal, Adem Efe Gencer, Ari Juels, Ahmed Kosba, Andrew Miller, Prateek Saxena, Elaine Shi, Emin Gün Sirer, et al. On scaling decentralized blockchains: (a position paper). In *International conference on financial cryptography and data security*, pages 106–125. Springer, 2016.

[8] Denis Stefanescu, Leticia Montalvillo, Patxi Galán-García, Juanjo Unzilla, and Aitor Urbieta. A systematic literature review of lightweight blockchain for iot. *IEEE Access*, 10:123138–123159, 2022.

[9] Qi Feng, Debiao He, Sherali Zeadally, Muhammad Khurram Khan, and Neeraj Kumar. A survey on privacy protection in blockchain system. *Journal of network and computer applications*, 126:45–58, 2019.

[10] Leo Maxim Bach, Branko Mihaljevic, and Mario Zagar. Comparative analysis of blockchain consensus algorithms. In *2018 41st international convention on information and communication technology, electronics and microelectronics (MIPRO)*, pages 1545–1550. Ieee, 2018.

[11] Dominic Grandjean, Lioba Heimbach, and Roger Wattenhofer. Ethereum proof-of-stake consensus layer: Participation and decentralization. In *International Conference on Financial Cryptography and Data Security*, pages 253–280. Springer, 2024.

[12] Pim Otte, Martijn de Vos, and Johan Pouwelse. Trustchain: A sybil-resistant scalable blockchain. *Future Generation Computer Systems*, 107:770–780, 2020.

[13] Jon Postel. User datagram protocol. Technical report, 1980.

[14] Jana Iyengar and Martin Thomson. Rfc 9000: Quic: A udp-based multiplexed and secure transport. *Omtermet Emgomeeromg Task Force*, 2021.

[15] Adam Langley, Alistair Riddoch, Alyssa Wilk, Antonio Vicente, Charles Krasic, Dan Zhang, Fan Yang, Fedor Kouranov, Ian Swett, Janardhan Iyengar, et al. The quic transport protocol: Design and internet-scale deployment. In *Proceedings of the conference of the ACM special interest group on data communication*, pages 183–196, 2017.

[16] Pijush Kanti Dutta Pramanik, Nilanjan Sinhababu, Bulbul Mukherjee, Sanjeevikumar Padmanaban, Aranyak Maity, Bijoy Kumar Upadhyaya, Jens Bo Holm-Nielsen, and Prasenjit Choudhury. Power consumption analysis, measurement, management, and issues: A state-of-the-art review of smartphone battery and energy usage. *ieee Access*, 7:182113–182172, 2019.

[17] Aleksandr Ometov, Yulia Bardinova, Alexandra Afanasyeva, Pavel Masek, Konstantin Zhidanov, Sergey Vanurin, Mikhail Sayfullin, Viktoriia Shubina, Mikhail Komarov, and Sergey Bezzateev. An overview on blockchain for smartphones: State-of-the-art, consensus, implementation, challenges and future trends. *IEEE Access*, 8:103994–104015, 2020. doi: 10.1109/ACCESS.2020.2998951.

[18] Johannes Sedlmeir, Hans Ulrich Buhl, Gilbert Fridgen, and Robert Keller. The energy consumption of blockchain technology: Beyond myth. *Business & Information Systems Engineering*, 62(6):599–608, 2020.

[19] Abigael Okikijesu Bada, Amalia Damianou, Constantinos Marios Angelopoulos, and Vasilios Katos. Towards a green blockchain: A review of consensus mechanisms and their energy consumption. In *2021 17th International Conference on Distributed Computing in Sensor Systems (DCOSS)*, pages 503–511, 2021. doi: 10.1109/DCOSS52077.2021.00083.

[20] Shubhani Aggarwal, Neeraj Kumar, and Prosanta Gope. An efficient blockchain-based authentication scheme for energy-trading in v2g networks. *IEEE Transactions on Industrial Informatics*, 17(10):6971–6980, 2021. doi: 10.1109/TII.2020.3030949.

[21] Cesar Castellon Escobar, Swapnoneel Roy, O Patrick Kreidl, Ayan Dutta, and Ladislau Bölöni. Toward a green blockchain: Engineering merkle tree and proof of work for energy optimization. *IEEE Transactions on Network and Service Management*, 19(4):3847–3857, 2022.

[22] Puneet Kumar and Behnam Dezfouli. Implementation and analysis of quic for mqtt. *Computer Networks*, 150:28–45, 2019. ISSN 1389-1286. doi: https://doi.org/10.1016/j.comnet.2018.12.012. URL https://www.sciencedirect.com/science/article/pii/S1389128618310776.

[23] Yuxia Sun, Junxian Chen, Yong Tang, and Yanjia Chen. Energy modeling of iot mobile terminals on wifi environmental impacts. *Sensors*, 18(6):1728, 2018.

[24] Fabian Knirsch, Andreas Unterweger, and Dominik Engel. Implementing a blockchain from scratch: why, how, and what we learned. *EURASIP Journal on Information Security*, 2019:1–14, 2019.

[25] Mohammed K Alzaylaee, Suleiman Y Yerima, and Sakir Sezer. Emulator vs real phone: Android malware detection using machine learning. In *Proceedings of the 3rd ACM on International Workshop on Security and Privacy Analytics*, pages 65–72, 2017.

[26] Faheem Iqbal, Moneeb Gohar, Hanen Karamti, Walid Karamti, Seok-Joo Koh, and Jin-Ghoo Choi. Use of quic for amqp in iot networks. *Computer Networks*, 225:109640, 2023. ISSN 1389-1286. doi: https://doi.org/10.1016/j.comnet.2023.109640. URL https://www.sciencedirect.com/science/article/pii/S1389128623000853.

[27] Sidna Jeddou, Fátima Fernández, Luis Diez, Amine Baina, Najid Abdallah, and Ramón Agüero. Delay and energy consumption of mqtt over quic: An empirical characterization using commercial-off-the-shelf devices. *Sensors*, 22(10):3694, 2022.

[28] Sean Foley, Jonathan R Karlsen, and Tālis J Putniņš. Sex, drugs, and bitcoin: How much illegal activity is financed through cryptocurrencies? *The Review of Financial Studies*, 32(5):1798–1853, 2019.

[29] Steven David Brown. Cryptocurrency and criminality: The bitcoin opportunity. *The Police Journal*, 89(4):327–339, 2016.