Electricity Consumption of a Distributed Consensus Algorithm

WILHELM WANECEK
BACHELOR´S THESIS
DEPARTMENT OF ELECTRICAL AND INFORMATION TECHNOLOGY
FACULTY OF ENGINEERING | LTH | LUND UNIVERSITY

Electricity Consumption of a Distributed Consensus Algorithm

Wilhelm Wanecek
wi2776wa-s@student.lu.se

Department of Electrical and Information Technology
Lund University

Supervisor (LTH): Max Åhman
Supervisor (Stellar): Paul Bellamy

Examiner: Maria Kihl

June 28, 2021

Printed in Sweden
Tryckeriet i E-huset, Lund

Abstract

The colossal electricity consumption of proof-of-work cryptocurrencies such as Bitcoin and Ethereum has caused critical examination of how consensus in blockchain-solutions is designed. Stellar is a decentralized, open-membership payment network built on blockchain technology, with the goal of enabling money to flow between banks, businesses and people across the global financial infrastructure, while minimizing latency and transaction fees. This study seeks to obtain a generalized estimate of the electricity consumption of the Stellar network, leading up with theory on the protocol's consensus algorithm employing federated voting in quorum systems. By dividing the electricity consumption of a single node into four core primitives and applying measurements on a basic validator node, an electricity consumption estimate for running a validator node is constructed. This is then extrapolated to the entire network to obtain a generalized estimate of the electricity required for a single transaction: 0.222 Wh, which turns out to be less than that of Bitcoin by a factor of 10^7 and similar to that of VISA. The results are followed by a discussion on the validity of said estimate, and areas of improvements for the method used, before concluding that by decoupling high electricity consumption from decentralized trust, Stellar provides a blockchain implementation that is not limited by electricity consumption to become an integral part of the global financial infrastructure.

Keywords — Energy efficiency, Payment systems, Stellar, Consensus algorithm, Digital currencies

Popular Science Summary / Populärvetenskaplig sammanfattning

Snabba och tillförlitliga betalningar — utan hög elförbrukning?

I takt med ökad efterfrågan av digitala tjänster har IT-sektorn växt fram som en ny energiintensiv industri. Ett område där digitaliseringen fått fotfäste är inom finansiell teknologi, där digitala valutor och betalningsnätverk erbjuder stora möjligheter. Precis som internet har möjliggjort för datorer över hela världen att kommunicera på en gemensam plattform, vill det amerikanska företaget Stellar med sitt blockkedjebaserade betalningsnätverk utgöra grunden för framtidens globala finansiella infrastruktur. Frågan är bara hur mycket elektricitet det drar?

Digitaliseringen av finanssektorn pågår för fullt. Det höga antalet centralstyrda valutor har på senare tid kompletterats av en uppsjö digitala valutor så som Bitcoin och Ethereum, samtidigt som digitala betalningssystem som Swish, Venmo och AliPay har ökat i popularitet. Men än så länge krävs komplexa avtal med varje betalningssystem och bank, vilket gör det svårt och dyrt för nya tjänster att få fotfäste. Betalningssystemen implementerar även oftast egna protokoll, något som hämmar kommunikation mellan plattformar. Detta gör det långsamt, dyrt och komplicerat att flytta pengar mellan system, banker och nationsgränser.

Stellar har som mål att göra för vår finansiella infrastruktur vad internet gjorde för datorer. Genom deras betalningsnätverk vill Stellar möjliggöra snabba, billiga, och tillförlitliga transaktioner, så att pengar kan flyttas fritt mellan banker, företag och privatpersoner över hela världen. Stellars lösning bygger på blockkedjeteknologi, precis som den numera kända kryptovalutan Bitcoin. Det är tack vare teknologi som Stellar skapar en öppen plattform där en mångfald av aktörer tillsammans kan bygga upp ett tillförlitligt och säkert system utan central styrning.

Men blockkedjeteknologi har på senare tid uppmärksammats av en särskild anledning: deras elförbrukning. I maj 2021 passerade Bitcoin-nätverkets uppskattade årliga energiförbrukning den av Sverige. Kryptovalutor utmärker sig i en IT-sektor som i sin helhet växer så att det knakar, och där elförbrukning hittills precis lyckats hållas i schack av effektiviseringar. En del forskare varnar nu för riskerna att den energiintensiva IT-sektorn kan öka lavinartat i elförbrukning, och kryptovalutor som Bitcoin och Ethereum står för en avsevärd del av det senaste tidens ökning. Om Stellar ska kunna utgöra grunden för framtidens finansiella infrastruktur är det kritiskt att plattformen inte har samma energitörstande beteende som t.ex. Bitcoin.

I denna studie har Stellars energiförbrukning undersökts. Genom att installera en server och ansluta den till betalningsnätverket kunde vi mäta energiförbrukningen från processor samt minnesanvändning, och med hjälp av generella koefficienter kunde vi även uppskatta elförbrukningen från lagring samt nätverkstrafik. Resultaten visar att Stellars nätverk har en låg förbrukning — en faktor av 10 miljoner mindre än Bitcoin per transaktion, samtidigt som det finns gott om utrymme för vidare optimering. Intressant nog visade resultaten på att inte är själva servrarna — minnesanvändning, lagring, och processorkraft — som står för den större delen av energiförbrukningen. Istället är det överföringen av data över vår globala IT-infrastruktur som står för över 94 % av fotavtrycket.

Med en blockkedjelösning som verkar frikoppla extrem energiförbrukning från decentralisering är Stellar inte bara intressant som potentiell finansiell infrastruktur, utan även som inspiration för andra blockkedjelösningar, inom såväl som utanför finansiell teknologi.

Acknowledgements

This Bachelor's thesis would not have been possible without the help of some kind and brilliant people.

At the Stellar Development Foundation, I'd like to thank:

- **Paul Bellamy**, *Software Engineer at SDF*, for suggesting the project to me in the first place, and for invaluable weekly supervision and guidance throughout the entire project. Thank you.
- **Eric Saunders**, *Director of Engineering at SDF*, for welcoming me to Stellar and including me in the weekly stand-ups, for encouraging words and helpful explanations, and for inviting me to speak at the Engineering All-hands.
- **Graydon Hoare**, *Software Engineer at SDF*, for taking the time to explain the intricacies of SCP, and what scaling the protocol may imply in theory as well as in practice.

At Lund University, I'd like to thank:

- **Max Åhman**, *Senior Lecturer, Environmental and Energy Systems Studies, Faculty of Engineering, Lund University*, for supervising the project. Thank you for using your academic experience to guide me in the field of energy systems. The advice on adapting the contents of the text was incredibly valuable when narrowing down the scope and to place the topic in a broader context. Thank you for reading numerous drafts and helping me find a common thread.
- **Maria Kihl**, *Professor, Electrical and Information Technology, Faculty of Engineering, Lund University*, for taking on the work of examining this thesis, despite the short notice.

Contents

List of Figures

List of Tables

Acronyms

API	Application Programming Interface
AWS	Amazon Web Services
BFT	Byzantine Fault Tolerance
BGP	Byzantine Generals' Problem
DLT	Distributed Ledger Technology
CO_2	Carbon Dioxide
CO_2e	Carbon Dioxide equivalents
CPU	Central Processing Unit
DRAM	Dynamic Random Access Memory
EC2	Elastic Compute Cloud
FBAS	Federated Byzantine Agreement System
GCP	Google Cloud Platform
GPU	Graphical Processing Unit
HDD	Hard-disk drive
ICT	Information and communications technology
iff	If and only if
IOPS	Input/Output operations per second
IEA	International Energy Agency
P2P	Peer to Peer
PoW	Proof of Work
PUE	Power Usage Effectiveness
RAM	Random Access Memory
RAPL	Running Average Power Limit
TDP	Thermal Design Power
SCP	Stellar Consensus Protocol
SDF	Stellar Development Foundation
SSD	Solid-state Drive
VM	Virtual Machine
WSC	Warehouse-Scale Computer
XDR	External Data Representation

Chapter 1

Introduction

In recent years, public interest in blockchain technology has skyrocketed, with its advocates promising decentralization, integrity, trust and ownership of data [4]. From having been a niche research area only two decades ago, applications of blockchain (and the slightly broader concept of Distributed Ledger Technology, DLT) are now explored in a diverse set of fields including food sciences [5], education [6], and healthcare [7]. But the most widely known application of blockchain technology exist within financial technology (fintech), as the cryptocurrency named Bitcoin.

Since its launch in 2009, Bitcoin has rapidly grown in popularity, with its peak market capitalization recently having surpassed 1.1 trillion USD [8]. As such, Bitcoin is the dominant cryptocurrency, but far from the only one. In January 2020, over 5 000 different cryptocurrencies were being traded across over 20 000 markets [9], with Ethereum, Tether and Litecoin being other popular cryptocurrencies, and in the spring of 2021, the price of Bitcoin has once again boomed [10, 11].

Cryptocurrency blockchains constitute alternative monetary economic universes, decoupled from the dependency of banks and states to mint or transfer currencies. Blockchain technology, the fundamental concept behind cryptocurrencies, is a method of keeping data synchronized across multiple, independent entities. The technology enables several stakeholders who may be unrelated or have incentives to modify their shared data to agree on and maintain a single dataset, without a central trusted authority like a bank or a government. In the case of cryptocurrencies, the shared state is commonly the ledger listing all transactions in the system, and what we want to avoid is disparate parts of the network disagreeing on what transactions have occurred.

While cryptocurrencies remain the dominant field of application of blockchain technology, especially inside fintech, several other blockchain-based initiatives have come to gain popularity. One of these areas is that of payment networks, where blockchain is integral to achieving a distributed, decentralized model of trust. Examples include Ripple, AlgoRand, Nano, Polkadot and Stellar. As societies around the globe become increasingly digitalized, and several countries including Sweden have started looking at national digital currencies [12], it seems like only a matter of time before the global financial infrastructure becomes digitalized to the core — and blockchain technology may be part of the implementation (as is the case with the initial proposal of Sweden's e-krona [13]). This is especially true for cross-border infrastructure, where actors such as banks or other financial institutions may be unwilling to place trust in one central authority to keep track of transactions. Instead, a blockchain solution can enable several actors to together verify the transactions. In these cases, where blockchain technology can become the digital backbone of financial infrastructure, it is essential that the protocols designed do not reproduce the power guzzling features of certain cryptocurrencies, such as Bitcoin.

Bitcoin and Ethereum apply proof-of-work (PoW) consensus algorithms [14], where solving very difficult, yet arbitrary mathematical problems is key to ensuring the reliability and stability of the decentralized network. Several of the recent payment networks have designed other mechanisms of achieving consensus, for example AlgoRand's Proof-of-Stake algorithm [15], or Stellar's federated voting-based algorithm, the Stellar Consensus Protocol (SCP) [1]. These have the potential to be far less energy intensive than PoW-based blockchains, and may because of this serve an important role of inspiration for future blockchain solutions, within and beyond fintech.

Through a case study on the Stellar payment network, this project serves to investigate the energy consumption of a payment network that does not apply an intrinsically computation-heavy PoW consensus algorithm, with an overarching aim of reasoning about the feasibility of a larger portion of the global financial infrastructure running on such a blockchain based payment network. Section 2 covers background and theory relevant to understanding the problem domain. Section 3 describes and motivates the choice of method. Section 4 presents results, and finally, section 5 places the results in a broader context while discussing the validity.

1.1 Problem Domain & Motivation

In the last decades, ICT infrastructure has grown substantially to meet increased demand for software services [16], to the point that ICT has become a new electricity-intensive industry [17]. Data on the current consumption as well as future predictions remain uncertain [16, 18, 19, 20], with the majority hoping that continued increases in efficiency will keep the energy consumption in check, while others fear a near-future explosion in energy demand. One of the fields where digitalization has expanded is in digital financial systems, with the rise of cryptocurrencies having sparked a race to transform the financial systems. However, researchers fear that the hoped-for efficiency gains in ICT infrastructure may be cancelled out by the increase in electricity demand from cryptocurrency mining [21, 22]. More specifically, Mora et al. highlight the risk of Bitcoin alone pushing global warming above the 2 degree Celsius limit within a few decades [23][1].

[1]Read more about the electricity consumption of data centres in Appendix A

It is clear that blockchain technology is here to stay, with applications expanding beyond the now-dominant cryptocurrency Bitcoin to numerous other fields. Stellar sets out to transform the global financial system through its payment network, supporting any monetary asset. If the Stellar network is to be a sustainable alternative for international transactions at a large scale, the payment network must be sufficiently electricity efficient. While the Stellar Consensus Protocol does not feature the properties that intrinsically demand high energy usage, as with many proof-of-work consensus algorithms such as that of Bitcoin and Ethereum, the electricity consumption of Stellar is still unmapped.

By examining the sources of electricity consumption of the Stellar network, this study hopes to highlight where to focus future optimizations to further reduce the electricity intensity of the Stellar network. Moreover, an overarching aim for this study is to verify if Stellar's claims of low electricity consumption hold true. If they do, it makes SCP a protocol that other blockchain solutions can learn from to avoid an energy-intensive blockchain implementation.

1.2 Research Questions

This study seeks to answer the following two related research questions:

1. How can we estimate the electricity consumption of a single transaction on the Stellar payment network?
2. What is the approximate electricity consumption of a single transaction on the Stellar payment network?

1.3 Limitations

This study is limited to estimating the running electricity consumption of the Stellar network, and does not include estimates associated with the larger lifecycle of hardware, the construction of data centres or ICT infrastructure, nor with possible implicit impacts from e.g. replacing other financial infrastructure.

Neither does it include the electricity consumption of running Horizon, Stellar's optional API server, nor investigate electricity consumption of re-ingestion of history or other fault-handling mechanisms.

Chapter 2

Background and Theory

2.1 Stellar — a payment network

The global financial infrastructure is facing great challenges with an increasingly complex market demanding swift communication beyond borders and currencies. An already large number of traditional currencies has in recent years been complemented by a growing set of cryptocurrencies. Several different payment systems with increasing popularity (e.g. Alipay, Venmo, PayPal and Swish) are to a larger extent substituting cash and traditional bank transactions. But since most payment systems apply their own protocol, moving money across systems, banks and borders is slow, expensive, and complicated.

Much like the internet enabled computers all over the globe to communicate on an open network, Stellar's vision is to unite the world's heterogeneous financial systems on a common, global network, enabling money to flow quickly and at low cost between banks, businesses and people. To realize this vision while ensuring integrity and neutrality, Stellar is a decentralized, blockchain-based open network, supported by the non-profit Stellar Development Foundation (SDF). This is central to Stellar's philosophy. If the global financial system is to act on the same platform, a decentralized system is necessary to enable collaboration beyond geopolitical disagreements and centralized control. Instead of a single authority overseeing the system and its transactions, the trust is distributed across the network.

The Stellar payment network aims to support every currency and speak to every payment system in its native protocol. This means that businesses and people can move money globally in seconds, regardless of the bank used at home or the original currency of their assets. Through their payment network, Stellar also wants to better enable international transactions that are too small to cover the fees charged by today's traditional institutions.

One such case is that of remittances. The World Bank estimates the total sum of received personal remittances during 2019 to more than $656 billion USD [24]. As eloquently explained by Cecchetti and Schoenholtz [25], when migrants send money across borders to their families, it boosts economic activity. Remittances support incomes in some of the poorest countries in the world. Indeed, in Senegal, the Philippines, and Guatemala, remittances exceed 10 % of GDP [25].

Yet, remittances are slow and expensive. Sending $0.50 from the U.S. to Mexico comes with an average transaction fee of as much as $9 [25], and as with most international payments, the latency is measured in days rather than minutes. This makes it impossible to send money abroad quickly in emergencies, and overall adds significant barriers to sending remittances [1].

The Stellar platform facilitates low-cost, universal payments, with its native cryptocurrency — the Stellar Lumen (XLM) — as a universal translator. Money enters and exists the network through anchors, connecting the network with the traditional banking systems. Anchors can be traditional financial institutions or money service businesses. These can issue one-to-one fiat[1]-backed tokens (stablecoins) that can then be traded with interoperability on the network. Anchors can also provide a fiat on/off ramp by connecting the Stellar network to the anchor country's banking system, handling regulatory processes to allow users to effortlessly make deposits and withdrawals [26]. Once fiat money is represented on the network as digital tokens, the network allows money to be traded and converted between different currencies, without having to pay heavy transaction fees or high latency at every step of conversion. As an example, to transfer money from euro (EUR) to Nigerian Naira (NGN), EUR is exchanged with EURT, a 1:1 euro-backed token on the Stellar network. EURT can then be traded for NGNT tokens, directly pegged to the Nigerian Naira, which in turn can exit the network via an anchor and be traded for NGN. The Stellar network charges a minimal transaction fee of about 0.00001 XLM (valued at 0.00000673 USD on the 2021–05–14) to prevent spam.

Since its launch in 2014, Stellar has successfully processed over 675 million transactions, out of which 151 million are in the past six months[2] only [27]. As an open network that keeps track of transactions, Stellar promotes and depends on other organizations to build services on top of the network. For example, SDF has not themselves developed a service that an end-user can use to send remittances. Instead, other organizations have expanded the ecosystem, for example Saldo that facilitates remittances from the US to Mexico, or Tempo and Cowrie that together enable low-fee transactions from EUR to NGN over the Stellar network. Actors in the ecosystem range from plain software services, such as digital wallets (e.g. provided by LOBSTR and StellarX), to financially backed anchor services, such as Bankhaus von der Heydt, who provide a euro stablecoin. Moreover, in January 2021, the Ukrainian Ministry of Digital Transformation announced plans to build Ukraine's virtual asset system using the Stellar network [28]. As illustrated by these examples, Stellar has already become a part of the financial infrastructure with a thriving ecosystem.

[1]Fiat money refers to a currency established as money, commonly by government regulation. Fiat money has a value that is regulated by the parties in exchange or because of government regulation. This, in contrast to commodity money (which has intrinsic value due to its medium, e.g. gold)

[2]As of early June 2021.

The key innovation of Stellar is the Stellar Consensus Protocol, a federated Byzantine agreement protocol enabling secure transactions across untrusted intermediaries. The protocol is designed around a unique voting scheme for reaching consensus on the blockchain, based on a "novel but empirically valid 'Internet hypothesis'" [1]. This is significantly different to Bitcoin's proof-of-work consensus algorithm, where computationally intense cryptographic puzzles are solved to decide who gets to add a block to the ledger. Instead, the two-phase balloting, message passing algorithm is based on trusting selected entities and voting, and hence has the potential to have significantly lowered energy intensity.

In order to understand how and why, the reader must first have an overview of blockchain principles, Byzantine agreement, and Bitcoin's PoW algorithm, before one can compare the design of the Stellar Consensus Protocol.

2.1.1 Blockchain properties of a cryptocurrency

Stellar [1], Bitcoin [14], and other cryptocurrencies are based on blockchain technology. Put simply, a cryptocurrency blockchain is a financial book (or ledger) in which it is permanently listed the balance and transactions of all members, creating a reliable, immutable transaction register. The fundamental property of the blockchain is that no one can practically alter or delete the transactions that make up the history. This absolute permanency of the ledger contents is mainly achieved through the combination of decentralization and the verifiable interdependency of blocks [29].

In a decentralized blockchain, identical copies of the ledger are stored on many servers worldwide (called nodes). This, in contrast to centralized banking systems where each bank keeps a ledger of customer transaction data on its own servers. The nodes are connected to directly to each other, rather than communicating through a centralized server, creating a global peer-to-peer (P2P) network, much like the backbone of the internet. In open-membership blockchain systems, such as Bitcoin or Stellar, anyone can connect a node to the network.

The blockchain is expanded through cryptographically chained blocks of operations (e.g. transactions, withdrawals, or deposits) successively being appended to the blockchain. The links in the chain between the blocks is constructed through each new block including a cryptographic hash depending on the previous block, as well as the set of operations. Cryptographic hashing functions are deterministic, one-way functions, enabling easy verification but making it practically impossible to construct another set of operations with the same hash. This is what prevents the history in the blockchain from being modified. If we have the successive blocks A, B, C, and D, a node could not modify the set of operations in block B without it altering the (easily verifiable) hash that was used to create block C, which in turn would modify the hash of block D. Since every node on the decentralized network holds a copy of the ledger, any node can verify requests and detect attempts to modify the history. See figure 2.1 for a graphical representation of how the operations and the hash of the previous block is used to construct the hash of the next block.

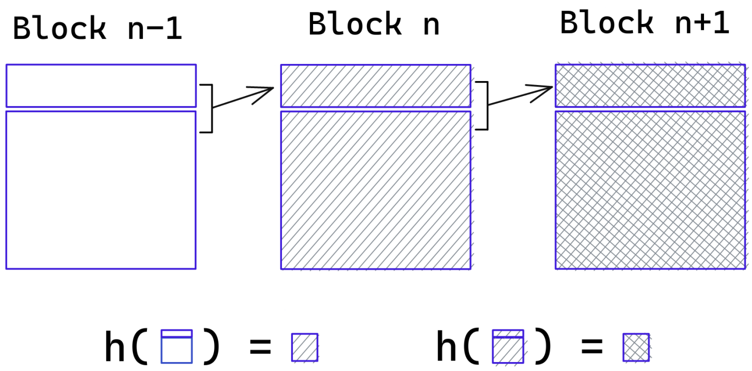

Figure 2.1: A simplified graphical representation of how blocks in a blockchain are a result of the previous block, with the hash function represented by the combining of fill-pattern.

A key aspect where cryptocurrencies differentiate is how and who gets to decide what transactions are included in each new block, and how it is ensured that the network does not split into different parts disagreeing on the history of the ledger, despite failing or ill-behaving nodes, as discussed in the section below on Byzantine Agreement and Byzantine Fault Tolerance.

2.2 Reaching consensus on the blockchain

In a decentralized, distributed blockchain, a critical part of the design lies in how to ensure that a set of independent actors collectively agree on the contents of the ledger, without a central, trusted authority. If different actors would arrive at different conclusions, the network would diverge. Instead, there must be consensus on the blockchain. Moreover, the network must be able to handle a certain number of failing or ill-behaving nodes.

This issue is abstracted to the “Byzantine Generals' Problem” (BGP), a logical problem where a system (particularly distributed systems) should avoid catastrophic failure despite arbitrary failure among some of its actors. It is named after the following allegory used to illustrate the problem, and is a fundamental concept of fault-tolerant computer systems.

Consider a fictional scenario where N generals have surrounded an enemy town, and have to reach consensus on whether to attack or to retreat. If they all attack at the same time, their attempt will be successful. If they all retreat, they can regroup without losses. A half-hearted attack, however would be a catastrophe. As such, it is critical is that they take the same action. The generals communicate with each other via messengers sent to the nearest few generals. The problem is complicated further by the possible presence of treacherous generals, who may cast a vote for a suboptimal strategy or send different messages to different generals. Messages can also fail to be delivered, e.g. if a messenger is caught or gets lost. Since the loyal, "well-behaving" generals do not know which other generals are traitors, they must design a communication protocol such that, in a situation where some generals have received two or more conflicting messages, only the right message is accepted by all, and the false ones are rejected [29].

This problem illustrates the difficulty of reaching consensus in a distributed system, and a translation of concepts to the Stellar network is found in table 2.1. In cryptocurrencies, the problem which the consensus algorithm needs to address is the so-called "double-spend" problem, where the same unit of currency is spent twice in different parts of the network. Imagine an account with a total balance of $10 submitting two transactions at the same time, $10 to person A and $10 to person B. While this is impossible with physical cash, this is a possibility in digital systems. Repeated double-spending would render the entire cryptocurrency worthless. If the entire network always agrees on the set of transactions that have occurred, and have an easy way of verifying that they are in agreement, double-spending is avoidable.

	BGP	Stellar
Objective	Agree on strategy	Agree on transactions to commit to
Spacial Distribution	Separated camps	Distributed nodes in the network
Well-behaving actors	Loyal generals	Truthful nodes
Ill-behaving actors	Treacherous generals	Evil nodes
Ill-behaved action	Suboptimal or inconsistent voting	Add an invalid transaction to the blockchain
Problem	How to know which message is true	How to know which transaction is valid
What to decide on	To attack or retreat	Which transactions to include in block

Table 2.1: Comparison between the Byzantine Generals' Problem (BGP) and the Stellar Payment network for a few different criteria.

A protocol addressing the BGP and in turn the double-spend problem introduces varying extent of Byzantine Fault Tolerance (BFT) — the ability for the system to avoid catastrophic failure despite a certain level of Byzantine fault, or to avoid "Byzantine failure". In the case of cryptocurrencies and blockchains in general, catastrophic failure usually refers to the network splitting in two or more units, with different ledgers, that can no longer agree on the contents of the ledger. The concise definitions below are helpful in understanding BFT.

Definition 2.2.1 (Byzantine fault)**.** Any fault presenting different symptoms to different observers. [30]

Definition 2.2.2 (Byzantine failure)**.** The loss of a system service due to a Byzantine fault in systems that require consensus. [31]

A typical traditional Byzantine agreement concerns a closed system with $N = 3f+1, f \in \mathbb{N} > 0$, and guarantee safety as long as at most f nodes are faulty [1, 29]. A system is considered to be **safe** when no two well-behaving nodes output different decisions, and considered to be **live** (or have liveness) when it does not get stuck in a state from which it cannot recover.

Definition 2.2.3 (Safety)**.** A Byzantine protocol is *safe* when no two well-behaved nodes output different decisions and the unique decision was a valid input (for some definition of valid agreed upon beforehand) [1].

Definition 2.2.4 (Liveness)**.** A Byzantine protocol is *live* when it guarantees that every honest node eventually outputs a decision.

2.3 Bitcoin and proof of work

While Bitcoin as a cryptocurrency serves a different purpose than Stellar's payment network, a brief description of its visionary consensus algorithm may be of value to the reader if it helps in understanding how Bitcoin's high electricity demand is intrinsic to the protocol and why the same does not apply to SCP.

Bitcoin's proof-of-work consensus algorithm relies on the decentralized network of so-called miners solving arbitrary, computationally intensive puzzles. Miners are the servers that perform transaction verification, and it is their activity that contribute to the heavy power consumption of the network. Put simply, the miners repeatedly compete in a cryptographic lottery, where computational power is the deciding factor for winning. The expansion of computational work is central to the entire protocol in avoiding the double-spend problem. In fact, the protocol periodically self-regulates the difficulty to compensate for growth of the network's computing power. As such, the power consumption can be considered intrinsic to the protocol, with a built-in rebound effect for energy consumption necessary to retain safety.

Consider an example transaction of one Bitcoin (BTC) from Alice to Bob on the Bitcoin network. Alice then broadcasts to miners on the network that "Alice wants to transfer 1 BTC to Bob". She authenticates herself to the network by cryptographically signing the request with her private key, which other network members can verify using her public key.

At this point, the transaction is non-validated, and no node has committed to writing the transaction into the ledger. As a first step, the miners verify the solvency of Alice, i.e. if Alice has a balance of at least 1 BTC on the ledger. As the Bitcoin protocol does not implement checkpoints of 'account state', the miners need to process all transactions from the start of history to compute the current balance of Alice. This is one way that Bitcoin intends to make the history more difficult to forge. Someone wanting to change the account balance (e.g. by adding

100 BTC to their balance) cannot only modify the current state, but must instead rewrite history, which (as discussed above) is expensive.

Once the miners have completed the first step, the more computation-heavy second step is initiated. In this phase, miners compete for being the first to embed the new transaction, which is the step where a new block is appended to the chain. A single block in the Bitcoin blockchain contains on average 1500 transactions, having accumulated in the last ten minutes (i.e. Bitcoin's frequency of block creation) [29, p. 2].

The challenge which the miners are competing on, is determining a nonce ("number used once"), that is included in the block along with the rest of the fields. The nonce is a random 8 byte number, such that the cryptographic hash of the entire new block (a 88 byte header, including the nonce, as well as the transaction data) equals or is lesser than a predetermined target. The hashing algorithm used in Bitcoin is *SHA-256* (i.e. it outputs a 256-bit value from an input of arbitrary length), and the target is a small number, e.g. such that the first 72 bits are 0s. Assuming uniform distribution, the probability of in a single attempt finding a number would be 2^{-72}. The exact value of the target, commonly known as the difficulty, is derived based on the average frequency of block verification, which in turn depends on the average computing power of the network. As the computing power of mining machines increase, the difficulty increases (i.e. the target is lowered), which keeps the average frequency of block verification to about once in 10 minutes [29]. As of early May 2021, the network is estimated to perform an all-time high of 177 TH/s (terahashes per second) [32]. This means that a staggering 1.77×10^{14} hashes are computed every second on the network (or 1.06×10^{17} hashes every ten minutes), competing to verify a block. Once such a nonce has been found, the miner broadcasts it to the rest of the network, and the block with transactions is considered verified — the network reaches consensus on the including it in the total set of transactions. The competition for a new block begins.

The reward of verifying a block, i.e. finding a matching nonce, is two-fold. One, the miner receives the sum of transactions fees. Every transaction submitted includes a small transaction fee, and the miner that verifies the block earns the sum of those transaction fees. Two, a fixed value (halving every 210,000 blocks, or about 4 years), currently of 6.25 BTC (in May 2021 valued about \$362,100), is awarded to the miner, which is also how new money is added into the system [29]. The transaction fees amount for about 10 % of the total income, while the fixed reward is the main source of income [33].

The mining for a nonce is the network's method for deterring frivolous or malicious attempts to game the system, and to avoid what is commonly known as the double-spend problem. Since the electricity cost, along with hardware costs, is part of the real-world limiting factor to Bitcoin miners (mining being profitable depends on the balance between the costs of running a mining server, and the value of the possible Bitcoin rewards), and the algorithm self-regulates the difficulty to ensure the mathematical problem remains sufficiently difficult, the electricity consumption can be considered intrinsic to the cryptocurrency. Indeed, in a commentary from 2021, de Vries shows a clear correlation between the price of the currency and the energy consumption [22].

2.4 The Stellar Consensus Protocol

At the core of the Stellar payment network is the Stellar Consensus Protocol, a partially synchronous, federated Byzantine agreement protocol defining the consensus mechanism. It is Stellar's response to Bitcoin's proof-of-work consensus algorithm described above, ensuring that the ledger is safely replicated across all nodes. However, unlike the proof-of-work algorithm of Bitcoin which revolves around computational power to achieve safety, the fundamental mechanism of SCP is that each node declares sets of other nodes they want to stay in agreement with, a relationship transitively connecting the entire network.

SCP is based on voting in quorum[3] systems, a common approach among distributed systems and consensus protocols [34, 35]. While earlier work [36, 37, 38] on quorum systems usually regard the size and members of quorums as static, SCP introduces an open-membership system — a Federated Byzantine Agreement System (FBAS) — where nodes accept different and evolving quorums [1]. This means that nodes can join and leave the network without the need for a centralized membership coordination, whereas in a closed membership system, the entire network needs to be reconfigured when new nodes join.

In order to build up to the protocol in its entirety, a few key concepts need to be covered. To help in constructing an intuitive understanding, consider figure 2.2 as an example of a minimal network.

Nodes Organizations and individuals can take part in the consensus process of the network by running one or more so-called validator nodes. This is commonly achieved through the open source *stellar-core* software, a C++ application maintained by SDF implementing SCP. Unlike with Bitcoin, running a node does not warrant monetary rewards. Instead, the incentive to run validator nodes lies in contributing to the safety and the decentralization of the network.

Quorums To construct an open quorum system, each node v unilaterally declares a subset of nodes on the network to trust and depend on. This is a node's **quorum set**. In figure 2.2, ACD is the quorum set of B, and BC is the quorum set of A. Each node also chooses a **threshold** that defines the minimum fraction of nodes in the given node's quorum set that must agree in order to reach consensus. For example, if B has the threshold $\frac{2}{3}$, any combination of two out of its three nodes would be sufficient for B to proceed — in this case AC, AD, or CD. These sets are known as **quorum slices**. The concept of quorum slices build up to constructing a quorum. A quorum is a (non-empty) set of nodes that contains a slice for each member. In our example, $ABCD$ is a quorum, while ABC is not;

[3] In distributed computing, a quorum is the smallest number of votes required to allow an operation to be performed, e.g. a transaction, often with an added requirement of atomicity and guaranteed replication across the participants in the distributed system.

the reason being that D is in the quorum slices of B and C, and must hence be included in the quorum.

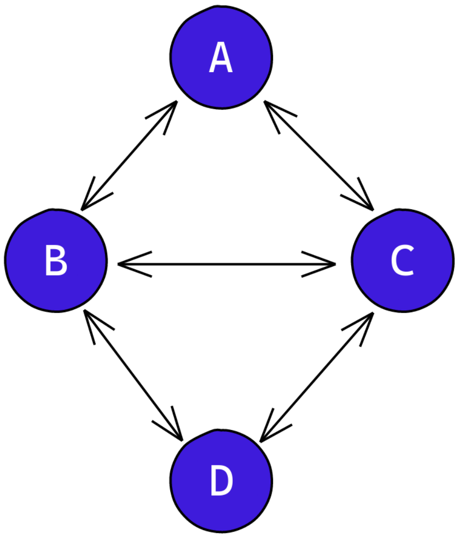

Figure 2.2: A sample network graph with four nodes, $ABCD$, with the bidirectional relationships AB, AC, BC, BD, and CD.

Blocking set Another definition fundamental to SCP is a node's **blocking set**. A v-blocking set is a set of nodes that intersects all of v's quorum slices [1]. Or, put differently, a v-blocking set is any set of nodes in v's quorum set, without which the node v cannot reach consensus. If a node has a threshold such that it requires $\frac{3}{4}$ nodes to reach consensus, any two nodes is a blocking set. Or, returning to our example in figure 2.2, for B, any combination of two nodes (AC, AD, and DC) constitute blocking sets. Or, let's assume C which has the quorum set ABD has a threshold of $\frac{3}{3}$, any node or any combination of nodes is a blocking set (e.g. A, B, and D). It is important to emphasize here that blocking sets are defined per node [39].

Statements Statements are the smallest building block of the Stellar Consensus network, and express opinions which the network wants to agree on regarding operations on the ledger. For example, "I *propose* set T of transactions for this ledger", or "I *am ready to apply* transaction set T to this ledger". Any node's opinion on a statement is based on information from its quorum set, which as we will see, under the application of a distinct set of rules enables reaching consensus on a set of statements.

2.4.1 Rules of federated voting

As a distributed system, SCP applies federated voting to reach consensus on different statements. The protocol must be designed to ensure that nodes do not act on a statement too early, and must know that a statement is safe before it commits to it, much like described above in the Byzantine Generals Problem.

To address this, SCP defines a set of rules for how nodes are to reason about a statement based on information from its quorum set. Any node v can only know what other members of its quorum set has decided with the possibility of messages being delayed or lost, yet, must take decision such that consensus can be reached across all well-behaving nodes of the entire network. To achieve this, every node has to go through three steps of federated voting: **vote, accept**, and **confirm** [39].

Given a statement S, any node v may have one of four opinions on S [39].

- The node does not know anything about S, and hence has **no opinion** on it.
- The node can tell that S is valid and **vote** on it, but cannot determine if it's safe to act on yet.
- The node chooses to **accept** S, as sufficiently many other nodes have supported this statement, but cannot yet determine if it is safe to act upon.
- The node **confirms** S — it is safe to act on. Even if every node in the quorum has not confirmed S, they will not be able to confirm anything else than S.

The federated voting mechanism of SCP declares a set of rules that defines the transitioning between these three states. Given a node v and a statement S, a node can do three things:

- **Vote** for statement S iff[4] S is valid and consistent with the node's previous votes.
- **Accept** the statement S iff *either*
  - Every node in v's quorum slice has voted for or accepted S, or
  - v's blocking set has accepted S (if v has previously voted for a statement that contradicts S, forget about that vote and proceed with S)
- **Confirm** S iff every node in one of v's quorum slice has accepted S

The states and the transitioning between these states is visualized in figure 2.3.

It is worth emphasizing the need for the confirmation step in order to obtain optimal safety. If it was omitted, and consensus was reached simply by accepting statements, a node's single blocking set could convince the node to accept any arbitrary statement. This would be catastrophic, as we cannot be sure that all blocking sets are honest. With the additional confirm step, a node can only agree to a statement iff every node in its quorum also has accepted that statement [39].

This mechanism of federated voting constitute a fundamental building block of SCP's consensus round.

[4] if and only if

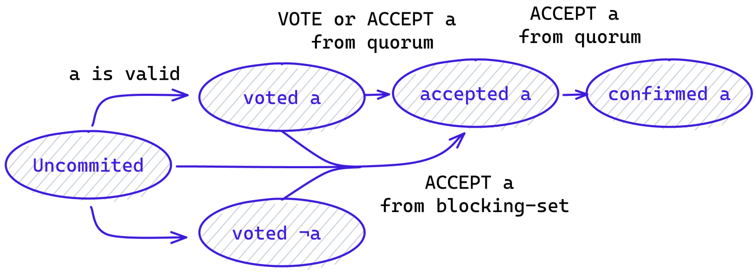

Figure 2.3: The stages of federated voting (adapted from [1, Fig. 1])

2.4.2 The two stages of a consensus round

The SCP consensus round, through which a block of operations is added to the ledger, consists of distinct stages with separate protocols: the nomination and the ballot protocol. During the nomination stage, the nodes select candidate sets of operations to include in the ledger. As soon as a candidate set has been nominated, the ballot protocol is initiated in parallel, which tries to ensure that the protocol can unanimously accept, confirm and apply a nominated operation set.

The funnel

Each consensus round can be described as a funnel, with the goal of selecting, committing to, and applying a distinct set of operations from a large set of possible operations, as depicted in figure 2.4.

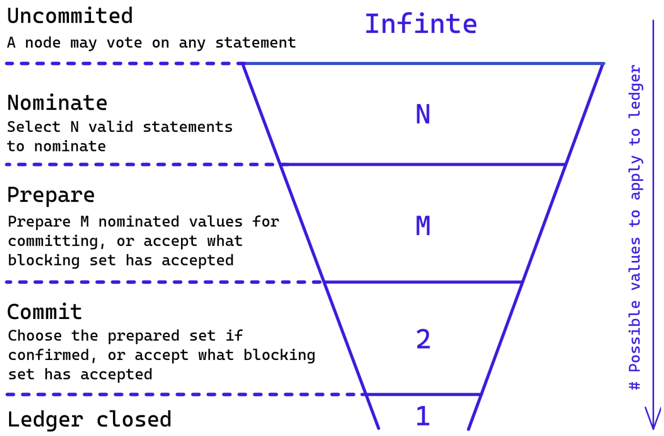

Figure 2.4: An intuitive depiction of SCP is as a funnel, where the protocol successively reduces the number of possible operations to include in a block to reach one distinct set.

A node v participating in SCP begins in an uncommitted state, with the possibility of agreeing to any transaction. It proceeds to nominate statements, with the aim of reaching a (ideally small) set N of valid candidate values through the nomination protocol. Once the voting on a candidate set to nominate is successful, the node initiates the ballot protocol on the candidate set. Through the ballot protocol, a smaller set of statements $M \subset N$ ($|M| \ll |N|$) is prepared to the commitment phase. Finally, the node is ready to commit to a single value, either choosing one of the statements prepared after the nomination phase, or accepting what the blocking set has accepted, after which a single set of operations is applied to the ledger.

Nomination protocol

The nomination protocol is responsible for the first part of the funnel, reducing an unbounded number of transaction candidates to a limited set. It applies federated voting on statements in the shape of "Nominate transaction set x".

- NOMINATE x — x is a valid decision candidate.

If the vote succeeds according to the set of federated voting rules above, the set of transactions is considered eligible for the next stage: balloting.

Nodes may vote to nominate multiple different values, as two NOMINATE statements are not contradictory. But, once a node confirms a NOMINATE statement, it ceases voting to nominate any new transaction sets. It may still both accept and confirm other nominate statements that were introduced before, e.g. if it learns that the blocking set has accepted some new value [1, 39]. This important criteria guarantees convergence on a candidate set — as every node on the network stops introducing new transaction sets to nominate, eventually all nodes will end up with the same set of nominated candidates.

The output of the nomination protocol, which in turn is the input to the ballot protocol, is called the composite value [1]. Important to note is that a node may start the ballot protocol on a composite value as soon as it confirms a candidate, introducing some asynchrony between the two protocols.

Balloting protocol

The aim of the balloting protocol is for the network to safely commit to the composite value through a series of numbered **ballots**. The protocol describes the steps taken in each **ballot**. A ballot starts in a prepare phase, where nodes try to submit a value that does not contradict any previous decision. This is followed by a commit phase, where nodes try to make a decision on the prepared value. This is achieved through federated voting on two types of statements: [1]

- PREPARE $\langle n, x\rangle$ — no value other than x was or will ever be decided in any ballot $\leq n$
- COMMIT $\langle n, x\rangle$ — x is decided in ballot n

A node begins ballot n by initiating federated voting on statement PREPARE $\langle n, x\rangle$, where x is a set of transactions. Iff a node successfully confirms the prepare-statement, it attempts federated voting on COMMIT $\langle n, x\rangle$. If in turn the vote on that statement succeeds, the node applies the set of transactions and there is consensus. If nodes fail to reach a decision in ballot n, nodes time out and try again in ballot $n + 1$, with successively longer timeout periods [1].

2.4.3 Practical properties of the network

Network configuration While the protocol strictly defines a number of characteristics and state transitions, there are numerous properties that are configurable on a node or network basis. One of these network-wide configurations is the number of operations to include in a block. This is currently capped at 1,000 operations, a pragmatic choice rather than an intrinsically determined limit. At the moment, the bandwidth has not been a limiting factor for the network, especially as much of the network is occupied by semantically dubious traffic, mostly in the form of arbitrage bots trying to win profits [40]. The network can at any time choose to vote on increasing the limit, but with the current state of the network, any increases in bandwidth risk being exhausted by these arbitrage bots. Increases in the transaction base fee, currently at 0.00001 XLM, is likely to deter the so-called arbspam. Indeed, part of the community discussed increasing the base fee 10 or 100-fold in June 2020, but decided to postpone the action [41].

The limit of 1,000 operations per block, together with a ledger close time of 5 seconds, means that the network currently has a theoretical capacity of 200 operations per second, or 720,000 operations in an hour. There are currently 21 different operations [42], such as "Create an account", "Payment", "Create passive path payment", and more. In the network, a variable number of operations (about 1–100) can be grouped together to form a transaction. As such, the network capacity is better measured in rate of operations than in transactions.

Validator Quality Hierarchy Within the above explained concept of quorums and quorum sets, SCP introduces nested quorums through the notion of organizations, where each organisation is a set of validator nodes labelled with a trust classification TRUST $\in$ { LOW, MEDIUM, HIGH }. Certain requirements are placed on organizations depending on their trust level. For example, HIGH-trust organizations are required to publish history and to run at least three validator nodes in a group, for increased redundancy. A number of reliable HIGH-trust organizations are labelled as "Tier 1 Organizations", with which SDF work closely to maintain the health of the network. For example, the Tier 1 organizations coordinate updates to avoid network outages, and a selection of Tier 1 organizations is often included in the quorum set of other nodes as an acknowledgement to their reliability.

2.4.4 Relevance for this study

For the sake of this study, there are a few things worth emphasizing regarding SCP as a consensus protocol.

Firstly, it's fundamentally based on message passing and the intention of some nodes to stay in agreement with others, a property that transitively connects the entire network. As such, computational work is not inherent to the protocol itself to guarantee safety. Surely, some computationally intensive work is needed, such as for generating cryptographic signatures and for (de)marshaling data sent on the wire. But, unlike proof-of-work algorithms where efficiency improvements will lead to the protocol self-regulating to increase the difficulty, the computational work of SCP is not core to the consensus mechanism. It is rather a side effect. As such, there is room for optimization, as we will see later.

Secondly, with the message passing being an integral part of the protocol, and the consensus round time being as short as 3–5 seconds, a large number of messages will need to be transmitted, received, and possibly acted upon in a small window of time. On the Stellar network, all messages are encoded to XDR [43] (External Data Representation) before being sent, and decoded once received. Additionally, cryptographic signatures and hashes are used to verify the authenticity and integrity of messages. Encoding/decoding (or marshaling/demarshaling) messages as well as constructing and verifying signatures can be computationally intensive in large numbers.

Finally, as with most blockchain implementations, adding more validator nodes to the network does not increase the bandwidth. If anything, it may even negatively impact the bandwidth as the quorum system grows more intricate. Instead, well-behaving nodes contribute by adding more trust and safety to the system by decentralizing the algorithm. An organization relying on the Stellar network, such as a bank running an anchor node or a financial service managing an exchange marketplace, may want to run a validator node to contribute to the network's decentralization, redundancy, and to yourself add a layer of security, if the network should be malfunctioning. For instance, an anchor organization that issues a real-world asset on the Stellar network can set up their own validator node, and decide to only honour transactions or redemptions on ledgers signed by their node. In doing so, they themselves become the final arbiter of truth for their issuing of assets.

2.5 Estimating electricity consumption of software services

A significant part of this study was devoted to studying related works developing a method of estimating the electricity consumption of software services, with the added complexity of Stellar being a decentralized network where every node-runner has the freedom to choose how and where they run their node.

In April 2020, Sommer et al. published an in-depth blog post detailing their work on estimating how much electricity the cloud-hosted software services of the company Etsy consume [2]. Their work was based on the assumption that power usage could be attributed to four properties: computation (CPU), memory (RAM), storage and networking. From the cloud provider dashboard, they were able to extract usage data on the virtual CPU usage, how much memory was reserved for their servers, how much data they had stored for how long, and how much networking traffic they were transmitting and receiving for. Knowing what hardware the servers were running, they came up with general estimates of the watt-hours that compute, storage, and networking consumed in a cloud environment, which they could then apply to their usage data. An important aspect of their estimates was adjusting for PUE (Power Usage Effectiveness), a ratio of how much of the total electricity used by data centres is used for the servers as opposed to cooling etc (described by Figure 2.5).

$$\text{PUE} := \frac{\text{All electricity used by data centre}}{\text{Energy used by servers and computer hardware}}$$

Figure 2.5: PUE is a ratio of the total usage of the data centre to the power used by servers and computer hardware, as opposed to cooling, lighting, etc [2].

As Sommer et al. notes themselves, they lack precision in their estimates for memory usage, as their cloud provider (GCP, Google Cloud Platform) does not provide data on the memory load. As such, the lack of confidence on memory usage caused them to leave it out of the aggregated electricity consumption entirely.

In two articles, published December 2020 and March 2021 respectively, Benjamin Davy at Teads Engineering shares the steps they took to estimate the power consumption of their services running on Amazon Web Services (AWS) [44, 45]. The first article focuses on theory and methodology, reviewing different approaches for estimating the climate impact of cloud services. While a substantial part of the work is focused on the translation of electricity consumption into CO_2e, which is outside the scope of this study, the steps applied to estimate the electricity consumption are of relevance.

Davy generalizes the approaches to obtaining electricity estimates to two options: Estimation based on hardware specifications, and estimating consumption based on software metering. While the first approach is adapted for software running on dedicated servers, and can include a larger range of the hardware's lifecycle, the second approach instead relies on the (albeit sparse) data available from cloud providers. Since services on cloud platforms tend to run on several types of hardware, and that some cloud vendors run specialized hardware, the second approach is better suited for cloud- and distributed computing — similarly to what Sommer et al. concluded. Indeed, figure 2.6 below shows the complex web of components that constitute cloud computing and make electricity usage complicated to estimate.

Moreover, Davy pays interest to research by Hendersen et al. In their study, Hendersen et al. applies Intel's RAPL (Running Average Power Limit) interface to collect CPU and RAM power consumption [46]. As their approach focuses on execution of machine learning algorithms, they neglect storage but instead include GPU into their core primitives of electricity consumption. They express their electricity estimate as a PUE-adjusted sum over the set of measured processes, where each process is estimated as the total electricity consumption of each source

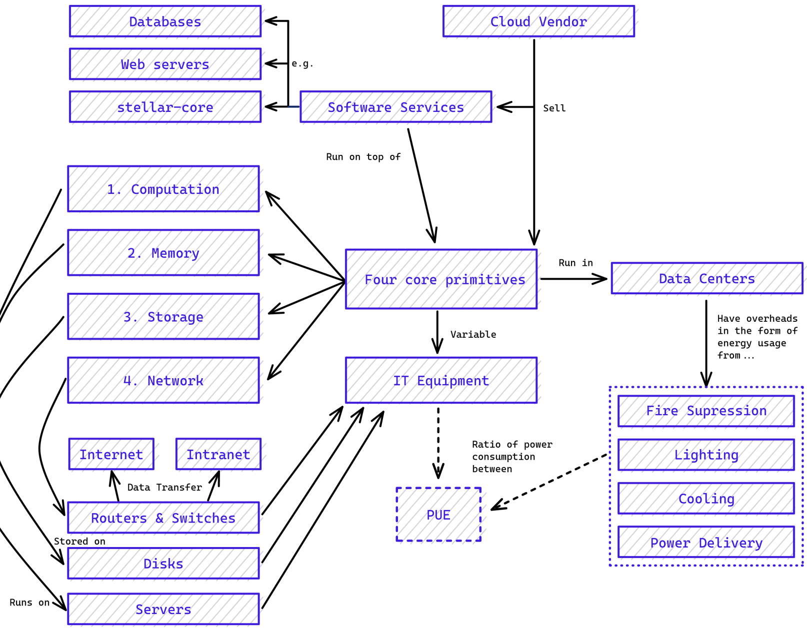

Figure 2.6: Public cloud components showing the complexity of assessing power consumption of cloud infrastructure, adapted from [3, Fig. 1].

of electricity consumption (RAM, CPU, GPU) adjusted by percentage of how much of the total resource type is attributed to the specific process. In doing so, they adjust for other system processes such as background updates.

Davy emphasizes the relevance of the RAPL-based approach, supported by research by Fahad et al. showing strong correlation between the readings of system power meters and RAPL [47] as well as research by Nizam Khan et al. demonstrating success with RAPL-readings on AWS EC2 instances [48]. The main limitation to the approach lies again in profiling virtual machines (VMs), which is how cloud platform software is commonly run. Indeed, RAPL readings apply to consumption on a processor level, rather than at a thread or core level. As such, when software is running in a virtualized cloud environment, sharing the system resources with other co-running user instances, there will be an impact of other co-running user instances.

The second article is dedicated to evaluating the feasibility of measuring the power consumption of software running in a shared AWS instance using Intel RAPL readings, despite RAPL readings only reporting system-total power consumption. Davy describes the intricate process of constructing a model for estimating the power usage of their cloud-running software services. Indeed, through rigorous measurements on a dedicated server, at different workloads and comparisons with SPEC power profiles, Davy derives estimates for the power consumption based on system resource usage. In the process of doing so, Davy discusses the validity of the methods chosen, and what assumptions underpin it.

Supported by research from Roose et al. [49], Davy concentrates on the electricity consumption of computation and memory usage, stating that these factors are responsible for a majority of the electricity consumption of servers (at least in traditional servers without a GPU). Davy also describes how external factors can significantly impact CPU power consumptions, e.g. manufacturing inconsistencies or ambient temperature in the server halls. Moreover, the workload as well as the type of instruction executed by the CPU impact the power consumption. Davy brings up unpublished work from Guermouche et al. demonstrating that so-called AVX-512 instructions (commonly used for High-performance computing) has a strong impact on power consumption, and proceeds by experimentally verifying that the type of instructions can have a great impact on the power consumption.

Davy also discusses and demonstrates the importance of including memory in power consumption estimates, something which the work by Sommer et al. were unable to do. In the stress tests conducted by Davy, it is found that power consumption of memory may exceed CPU consumption under certain types of workloads. Interestingly, the power profiles obtained by Davy closely match the slope of those found in the SPEC power report for similar sized servers. As the SPEC power report measures the entire server with a power meter, while Davy's method is limited to CPU and RAM, this indicates that CPU and RAM is a relatively strong indicator of the total power consumption.

All in all, the articles by Sommer et al. and Davy describe the many factors that contribute to power consumption of software services, and the intricate process that is today required to obtain accurate estimates. The variation in power consumption depending on factors such as the type of workload motivate the choice to perform measurements using the Intel RAPL interface, complemented with coefficients for network and storage, as opposed to only basing power estimates on the CPU load. Additionally, while the three articles focus on software running as cloud services and in shared environments — something that is not necessarily true for Stellar validator node runners — they discuss many of the challenges with estimating the power consumption where one cannot run physical measurements on the servers.

Chapter 3

Method

The aim of the experiment was to obtain a generalized estimate for a transaction on the Stellar network, and to be able to identify which component system has the highest electricity consumption. As the network constitutes a decentralized platform, where Stellar does not have control of all nodes, we could not perform measurements on the nodes directly. Instead, an estimate had to be obtained, for which a multi-step process was designed. First, an approximation of electricity consumption for a typical validator node was made. This was done through setting up a dedicated node and using measurements from the Intel RAPL sensor, from which to construct an electricity estimate. Additionally, a community survey was constructed to evaluate how typical the node setup chosen was, helping determine the validity of the estimate obtained. From this, approximations on the electricity consumption of the entire network as a whole was constructed.

Much like the work by Sommer et al. [2], Davy (2021) [45], Wei Wei [50], and Mytton [3], the energy of the server was estimated through a few core criteria, rather than including all factors of the entire cloud infrastructure as illustrated in figure 2.6. These factors are computation (CPU), memory (RAM), storage and network, described by equation 3.1.

$$e_{total} \approx e_{CPU} + e_{RAM} + e_{storage} + e_{network} \tag{3.1}$$

3.1 Estimating electricity consumption

To estimate the current electricity consumption of a single node on Stellar's Payment Network, a single validator node was set up on a dedicated server. This allowed accessing measurements from the Intel RAPL tool of the processor, providing more accurate estimates of power consumption from CPU and RAM, much like in the work by Davy [44].

To monitor and aggregate the electricity usage, a script reading from the Intel RAPL interface (via the Linux kernel's `powercap` interface) was written. While the RAPL interface exposes estimates of the consumption from the CPU and the RAM, it does not include estimates for network nor for storage.

Hence, conversion factors between bytes of data and electricity consumption were used to account for these two factors.

3.1.1 Selection of factors

Numerous of studies have been conducted on the electricity consumption of ICT infrastructure, with varying scopes and methods, arriving at significantly different estimates for kWh consumption per GB of network traffic. Additional complexity was added by rapid improvements in efficiency having been made over recent years, making comparative studies more difficult.

For example, in a study evaluating the energy consumption of mobile data transfer, Pihkola et al. (2018) arrives at an estimate of 2.9 kWh per GB for the transfer of mobile data between 2010–2017 [51]. On the other hand, Schien et al. arrives at 0.052 kWh/GB for the core network (edge, metro and long haul network, but excluding access network) [52]. As illustrated by the variance in these factors, it was essential to base our estimate on a study that only includes the intended network layers. For example, for the purpose of this study, an estimate that does not include the wireless access networks (e.g. the 3G network) is more relevant than one that does.

With this in mind, the elaborate review study conducted by Aslan et al. [53] was of great value. In their article, the authors compare a number of other studies in light of their system boundaries, assumptions underlying the methodology, and years to which data applies. Based on the results and discussions from the review study above, 0.06 kWh/GB was chosen as a relevant and well-supported coefficient for network traffic, while keeping in mind that it may very well be an over-estimate. This, especially as the previous increase in efficiency seem to have continued between the construction of the estimate to today [16].

$$e_{network} \approx 0.06\,\text{kWh/GB} \times \text{amount of data transmitted (GB)} \tag{3.2}$$

The coefficient used for mapping storage capacity to power consumption was derived from the United States Data Center Energy Usage Report (2016) [54]. In their report, Shehabi et al. arrive at estimates of electricity usage of storage as a function of disk type and storage capacity. They conclude that the power consumption of HDD drives is relatively independent of disk capacity, but that the technology has become more efficient in the years leading up to the study, something they project will continue to at least 2020. For SSD units, the power consumption was found to be more closely related to disk capacity, but is still reported on a per-disk level [54, p. 14].

While Sommer et al. derive a continuous W-per-TB function from the above-mentioned study through combining the average power drain with the average disk capacity, we instead mainly opt for a discrete W-per-drive mapping. This, as we hypothesize that a typical node will run in a dedicated environment, and we as such arrive at equation 3.3. Put simply, we attribute 6.5 W per 10 TB slot.

$$P_{storage,dedicated} \approx \lceil \text{storage used (TB)}/10\text{TB} \rceil \times 6.5\text{W} \tag{3.3}$$

This could be complemented with a continuous function of W-per-TB for archive storage, in a similar vein as Sommer et al., as archive storage is more commonly handled by a cloud, flat-file service such as AWS S3 or Microsoft Azure Blob Storage. For this, we use the default server HDD size of 10 TB coupled with the average disk power consumption of 6.5 W, to arrive at a power consumption per TB stored:

$$P_{storage,shared} \approx 0.65\text{W / TB} \tag{3.4}$$

3.1.2 Community survey to verify validity

In order to verify the validity of the assumptions made on the hardware, a community survey was conducted. The survey was sent out to a number of mailing lists held by the SDF, directed at the administrators and developers running verifier nodes.

3.2 Setting up and configuring a Stellar validator node

To obtain measurements from a typical Stellar node, a dedicated server was rented and configured to run the experiments. See table 3.1 for specifications. The server was accessed over `ssh`, and a repository summarizing the scripts and packages used to run the node can be found at https://github.com/wanecek/eitl01-scripts.

Component	Model
CPU	Intel Core™ i7-7700 CPU @ 3.60GHz
RAM	2 × 16 GB DIMM DDR4
Storage	2 × 2 TB HDD
Network	1 GBit/s ethernet connection
Operating System	`Ubuntu 20.04.2`
Linux Kernel	`5.4.0-67-generic`

Table 3.1: Specification of dedicated server where experiments was run.

3.2.1 Setting up and configuring the Stellar Node

A node can be connected to the Stellar network through the package stellar-core, developed and maintained by SDF. It's a C++ application implementing the Stellar Consensus Protocol, with cross-platform support. It can be run in a docker-container, through pre-compiled packages, or by building the project from source.

For a balance between production-level performance and ease of setup and reproducibility, this experiment used the pre-compiled package, with `stellar-core` running as a system service.

Setting up a database using PostgreSQL One of the requirements for running a validator node is access to a database where the entries of the ledger are stored. In fact, stellar-core stores the ledger twice simultaneously — on disk, in so-called buckets, as well as in a database. For this experiment, a PostgreSQL database was set up. PostgreSQL is industry-standard for managing databases, and some other Stellar services (e.g. Horizon, Stellar's API server) requires PostgreSQL specifically. One alternative would have been an in-memory database such as SQLite, but PostgreSQL was chosen because it is a common and recommended choice that reflects a typical node.

Connecting to the test network The first goal was to configure the node to connect it to the Stellar Test Network, where it's easier to experiment with different configurations. SDF provides example configurations for a node to connect to the test network, which were used as a starting-point. The main steps taken from the predefined configuration was to generate and enter a key pair (or seed) using `stellar-core gen-seed`, and ensuring that the connection to the PostgreSQL instance was working.

Additionally, a TOML file had to be created and placed on a publicly facing domain. The `stellar.toml` describes the validator nodes that a single organization runs (including their public keys), and acts as a second layer of authentication. If validator A wants to trust our node and include it in their quorum configuration, they provide our public key as well as our home-domain. Stellar-core can then, in addition to verifying that our node has the secret key matching our public key, check the file hosted on our home-domain to ensure that the public key there provided matches our `stellar.toml`-file, acting as a second factor. As such, should we want to deprecate our node, or if our secret key was compromised, we can tell other nodes using our `stellar.toml` file on our domain.

The `stellar.toml` file was placed in the `/.well-known` directory of our domain, with the appropriate CORS header (`Access-Control-Allow-Origin: *`), on a public domain.

Connecting to the public network Once the `stellar-core` node was connected to the network and successfully validating, it was taken down and reconfigured to instead connect to the public network. The trusted test validator nodes in our quorum set were then replaced with the Tier 1 validator nodes, at the time of writing including 23 nodes. Their home-domains were added to the configuration, and the network passphrase was changed from `Test SDF Network ; September 2015` to `Public Global Stellar Network ; September 2015`.

Throughout the process of configuring the node, several minor roadblocks were hit, ranging from mistyped configuration fields to requirements on the `stellar.toml` file that was not clearly listed in the developer documentation resources. As `stellar-core` was managed as a system service using `systemctl`, tailing the logs of `journalctl` (e.g. through `journalctl --unit=stellar-core -f`) was a great way to debug any issues. The error messages from stellar-core were generally very descriptive and helpful. Moreover, the stellar-core info command, reporting back the overall status of the node as a JSON object was helpful to track the status once the node started catching up to the network (`stellar-core-cmd info`).

3.2.2 Monitoring CPU and RAM Power usage

Scaphandre The initial plan was to use the open-source tool scaphandre. It is an open source software written in Rust with low overhead, relying on RAPL readings using the powercap interface. Is uses the `proc` filesystem to determine how much of the CPU every given process is using, which it uses to infer an estimate of the power consumption of each individual process from the total power consumption reported by Intel RAPL. This per-process power consumption distinguishes it from many other tools.

However, despite a promising start, a number of bugs were encountered. Memory power consumption was reported as 0 W. Being an open-source piece of software, I managed to debug the program and submit a fix (https://github.com/hubblo-org/scaphandre/issues/108).

Unfortunately, once this was addressed, it was discovered that the power consumption reported by Scaphandre did not add up — the total consumption from the sockets greatly exceeded the reported total consumption. The same applied when comparing the consumption of consumers (processes, e.g. `stellar-core`) and the total, host power. Because of this, the tool was abandoned. However, once the tool stabilizes, future studies are encouraged to look into using Scaphandre for software-based power measurements.

Shell script reading from powercap interface Instead, a shell-script was written that directly read from the powercap interface. While it does not infer the power consumption of individual processes, it does divide the power consumption into three different sockets: CPU, Core and RAM. The script read the powercap sensors, sleeps for one second, and then reads again, computing the change in electricity consumption between the two measurements. The majority of the logic for reading powercap sensors and computing the difference was inspired by the open-source GitHub repository powerapi-ng/energy-scripts, authored by researchers at the University of Lille [55].

Between each measurement, a random delay $\in (0, 5]$ seconds was added, spreading the measurements out over the 5-second ledger closing time. The measurements are appended to a locally stored CSV file — one CSV file per day. The shell-script was registered as a `systemd` service, ensuring that it is restarted upon exiting.

Monitoring CPU usage To be able to analyze the power consumption in relation to system resource usage, a `systemd` service monitoring CPU usage was written. To avoid unnecessary and unpredictable resource usage from this process that could skew the results, the service collects samples for 60 seconds once every hour. The service uses a combination of the well-known UNIX system tool `top` and the `sysstat` subpackage `sar` [56].

3.2.3 Monitoring network traffic

To estimate the network traffic over an extended period of time, we needed a tool with low overhead collecting statistics on the network traffic — both receiving and transmitting. A number of tools were evaluated and tested, including `collectl`, `nethogs`, `iptraf`, and `ifstat`. Most of the software found focus on monitoring momentary traffic, e.g. when debugging a spike in network traffic. For this experiment, it was instead needed something which collected statistics over a longer period of time, for which we finally settled with vnstat, which too runs as a `systemd` service.

Much like other tools, `vnstat` monitors both transmitting and receiving traffic. While this may be relevant when studying the power consumption of a single node, including both factors would introduce duplicate measurements when looking on a network-level (what one node transmits, another will receive). To account for this, we base the network activity on an average between transmitting and receiving data.

Chapter 4

Results

In this section, the results from the measurements and the survey are presented.

4.1 Measurements from a single node

Through running software-based measurements on a dedicated server, estimates of average electricity consumption of running a validator node were obtained. The results suggest that with the selected coefficients, network is by far the largest source of power consumption, on average responsible for 94 % of power consumption. Recall the division of the electricity estimate into four core primitives: CPU, Memory, Storage and Network, where the two former factors are estimated using readings from the Intel RAPL sensors, whereas storage and network are estimated using conversion factors.

4.1.1 CPU and Memory

The software-based RAPL readings on computation- and memory power consumption indicate low power consumption from both CPU and RAM. While the power consumption of memory remains relatively constant, the CPU power consumption fluctuates significantly, as can be seen in table 4.1 together with figure 4.1.

	CPU	RAM
10% quantile	1.348 W	1.079 W
Mean	2.816 W	1.099 W
90% quantile	6.060 W	1.118 W

Table 4.1: Mean as well as 10 % and 90 % quantile for RAPL power measurements.

To better understand the reason for these fluctuations, the power consumption was plotted together with several other factors, such as the system resource usage (overall as well as for `stellar-core` and `PostgreSQL` specifically), the number of operations and transactions on the Stellar payment network, as well as the volume

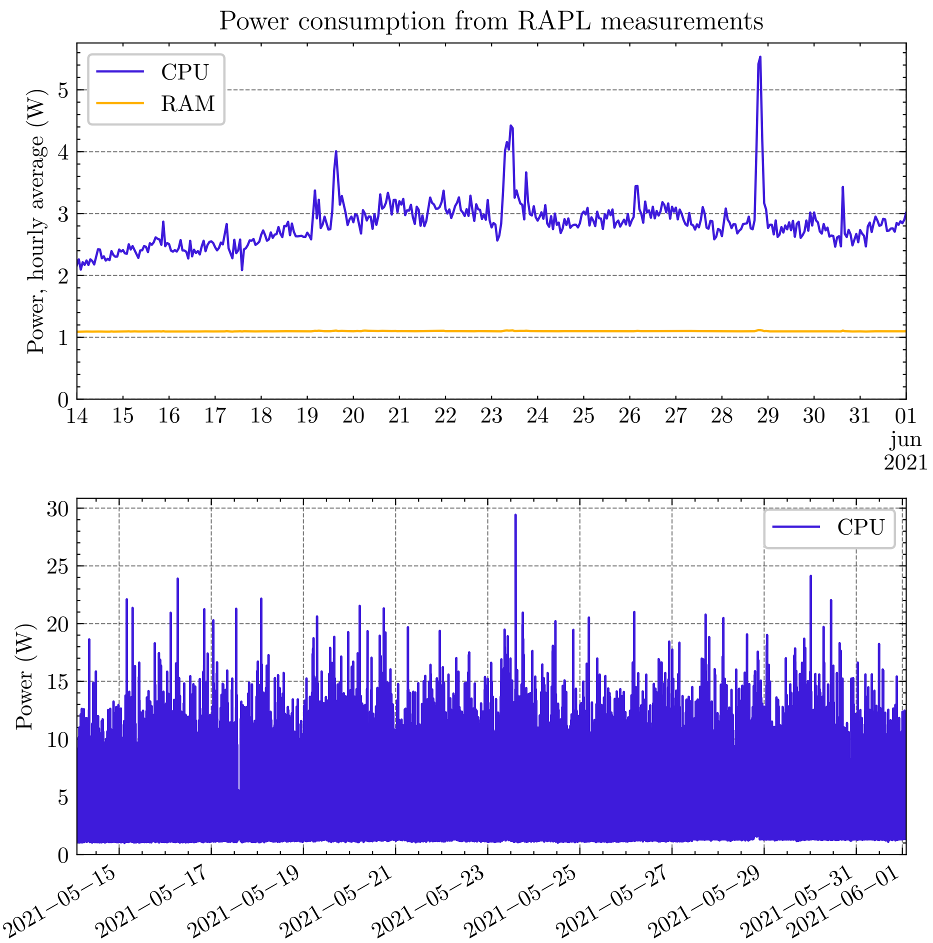

Figure 4.1: Power consumption of CPU and RAM, in the topmost subfigure as an hourly average and in the bottommost as instantaneous measurements.

of network traffic. This is visualized in figure 4.2. Additionally, the consumption from CPU only was plotted against the number of transactions and operations in Figure 4.3, to visualize the existence or lack of correlation between the two.

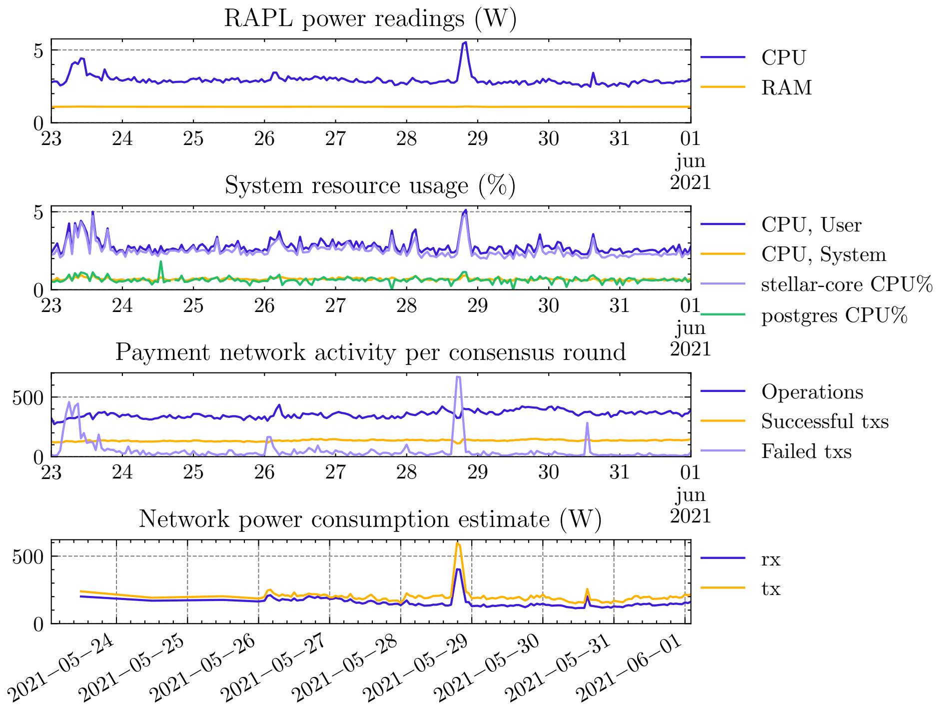

Figure 4.2: Power consumption of CPU and RAM, compared with system resource usage, Stellar operation/transactions per ledger, and network traffic.

Indeed, the RAPL power readings in 4.2 show clear correlation with the system resource usage, and especially that of the CPU usage of `stellar-core`. Moreover, there exists a correlation between the network activity and the system resource usage — a distinct spike in the number of transactions (successful or failed) seem to cause increased power usage.

However, the scatter plot in figure 4.3 indicates a lack of direct correlation between the number of successful operations and the electricity consumption, at least within a certain bounded range, as the hourly electricity consumption varies with roughly 100 Wh for a very similar number of successful transactions.

4.1.2 Storage

When investigating the storage usage, the distinction between basic validator nodes and full validator nodes is of importance. While a basic validator node only needs to store the current ledger, a full validator offers public archive, featuring a larger snapshot of the ledger. Such a public archive requires more storage,

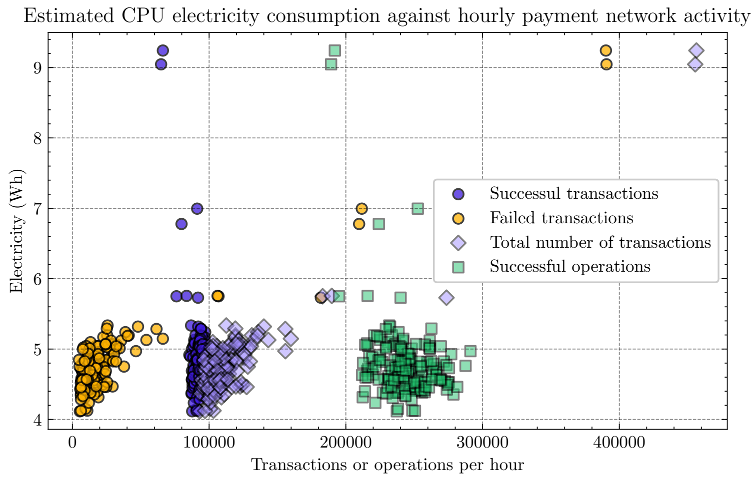

Figure 4.3: Hourly electricity consumption of CPU against the number of transactions and operations, against the number of transactions and operations in the same period.

which may entail a higher power consumption.

In our experiments, our basic validator node uses a total of 19 GB (!) of data, a very sparse amount. Out of these 19 GB, 5.0 GB are log files and 6.3 GB are used for storage of the ledger. In total, 4.0 GB can be attributed to the ledger stored in the database, and 2.4 GB to a copy of the ledger stored in flat XDR files (so-called buckets). As such, our experimental setup of dual 2 TB HDD was a great exaggeration in relation to what was necessary.

With an average storage capacity of 10 TB in 2020 [54], something only expected to increase, the storage requirements of running a validator node is with great margin satisfied by a single HDD disk. As such, we attribute 6.5 W to power consumption from storage, the estimated average power consumption of a disk in 2020.

Indeed, the responses from the survey indicate similar storage requirements for other nodes. The respondents who run basic nodes report "Less than 50GB", and "45 GB" for their nodes. The difference between these measurements and our 19 GB could be attributed to log-files, monitoring software, and how precise one is with what to include (e.g. rest of operating system). Regardless, it is still sufficiently little to fit on a single storage drive without any issues.

Similarly, full validator nodes report using "> 1.5 TB" and "between 3–4 TB", an amount that still fits well within the average drive capacity referred to above. However, it is standard practice to instead use public facing flat file storage service, such as AWS S3 or Microsoft Azure Blob storage, rather than storing the history data on the same. Due to lack of precise data on the storage requirements of running a full validator, as well as it being outside the strict requirements of running a validator node, power consumption attributed by this additional storage was not included in the results.

4.1.3 Network

Through monitoring the network traffic (receiving as well as transmitting, or rx and tx), results indicate that the network fluctuates drastically. Indeed, as can be seen in figure 4.4, where the network traffic ranges from just over 45 GB per day to 100 GB per day, more than a 100 % increase.

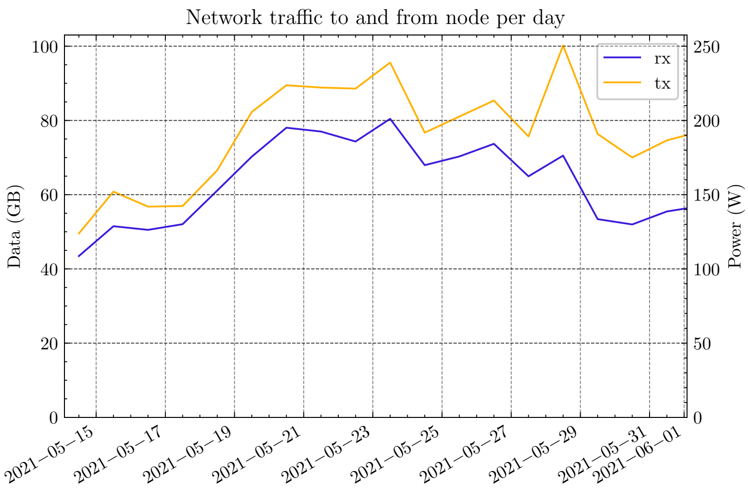

Figure 4.4: The daily network traffic measured network traffic of a single node, divided into transmitting (tx) and receiving (rx). See left y-axis for data in Gigabytes, and right axis for the corresponding estimated power consumption.

In the survey sent out to node maintainers, precise responses to the question on network traffic was sparse. Two Tier 1 validator providers responded in text to the question, stating an average of 76GB per day and roughly 3–4 TB per month respectively (corresponding to 96–129 GB per day). Additionally, two other Tier 1 node providers sent graphs on traffic usage from their validator nodes — one of the two providers running three nodes. See figure B.1 and figure B.2 respectively. These show similar network usage as our experimental node, suggesting that it is reasonable to assume that the network traffic is not significantly impacted by the position in the network.

4.1.4 Total Energy usage

Through combining the factors above, it is clear that the network traffic is by far the largest contributing factor to the overall power consumption. Table 4.2 summarizes the energy consumption of the different components, also illustrated in figure 4.5.

	CPU	RAM	Network	Storage	Total
10 % quantile	1.348 W	1.079 W	116.273 W	—	118.699 W
Mean	2.816 W	1.099 W	169.129 W	6.500 W	179.543 W
90 % quantile	6.060 W	1.118 W	209.397 W	—	216.575 W

Table 4.2: Energy usage of different components. Mean value complemented with the 10% and 90% quantiles.

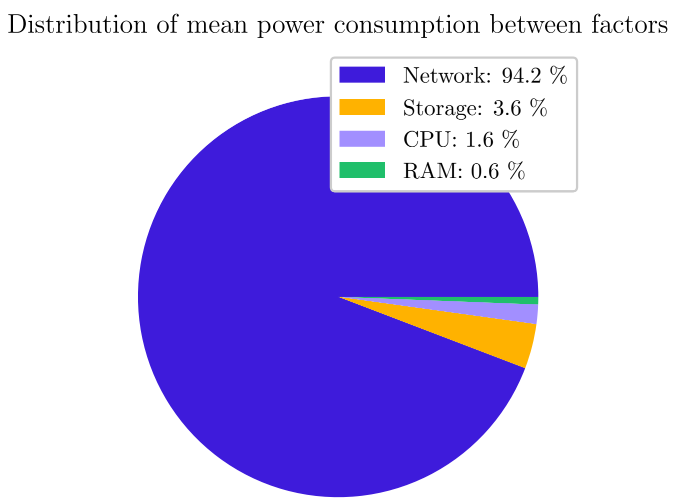

Figure 4.5: Comparison of average power consumption between different factors.

Lastly, plotting the power consumption (adjusted to PUE of 1.67, the 2020 average PUE [18]) against the number of transactions, we can visualize how the number of transactions (successful, failed, and total) correlates with the power consumption. See Figure 4.6.

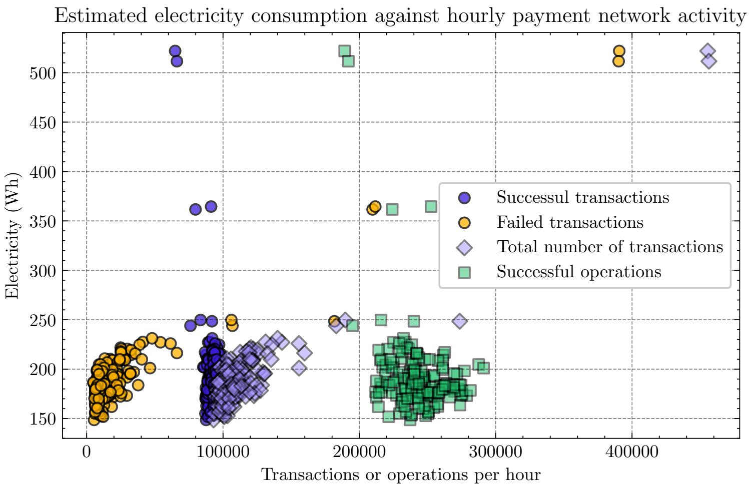

Figure 4.6: The number of successful, failed and total transactions against the estimated power consumption of a single node, per hour.

4.2 Survey responses on node configuration

In total, eight organizations replied to the survey, together representing 24 validator nodes on the network most which are so-called Tier 1 nodes. Since the survey covered a number of questions related to this work as well as to other Stellar services, with all questions being optional, not all respondents filled out every question. With that being said, there were a number of relevant results that arose from the survey.

More specifically, relevant for this study was the choice of hosting provider and the distribution between full and basic nodes (see table 4.3), if organizations run on dedicated or shared hosts (see table 4.4), and the type of hardware that nodes run on (see table 4.5).

Hetzner was the most common hosting provider, and the majority of nodes were run on dedicated servers. 5 out of 8 ran only `stellar-core` on their server, and 2 reported running other services with `stellar-core` still being the main reason for hosting the server.

There were two noteworthy outliers in the survey. One was that of an individual running their own basic validator from home, on a Raspberry Pi. This can be seen in the higher CPU-load, as well as being the only self-hosted node. A second

Hosting provider	# Nodes
Hetzner	12
OVHCloud	5
Microsoft Azure	3
Digital Ocean	2
Google Cloud Platform	1
Self-hosted*	1

Node type	# Nodes
Full validators	17
Basic Validators	5
Watcher nodes	2

Table 4.3: Responses from survey on what type of node and what hosting provider is used.

Are you running Stellar on a dedicated server?		Are you running other software services on the same instances/servers?	
Yes, all of our nodes run on one or more dedicated servers	6	Yes, but Stellar is the main reason we run the instance/server.	2
Yes, some of our nodes run on a dedicated server, and some in a shared environment	1	Yes, and other software services are our main priority.	1
No, none of our nodes run on dedicated servers	1	No, we only run stellar-core on the concerned instance/server.	5

Table 4.4: Responses from survey discerning if respondents are running stellar-core as a primary or secondary service, and if they are running stellar-core in a shared environment or on a dedicated server.

CPU	RAM	Storage	Avg CPU %	Avg RAM usage
AMD Ryzen 7 3700X Octa-Core	64 GB	2 × 1 TB, SSD	< 10 %	≈1GB
Intel Xeon, 4 vCPU	28 GB	200 GB, SSD	12 %	—
Intel Xeon Skylake	16 GB	160 GB	—	—
Raspberry Pi 4	—	—	80 %	≈1GB
Intel Xeon-D 1521	16 GB	500GB SSD + 4×4TB HDD	—	4GB
Digital Ocean CPU Optimized	8 GB	50 GB, SSD	< 10 %	—

Table 4.5: Results from survey on the type of hardware and respective system load.

outlier in the data was the one node that was not running on a dedicated server, but instead in a shared cloud environment. This respondent reported representing an anchor service, and was running the validator in conjunction with an API service, necessary to support their anchor service with the relevant data.

4.3 Extrapolating to the entire network

At the time of writing, there are about 130 active nodes on the public network, out of which 39–41 are basic validators and 16–18 are full validators. The rest of the nodes are so-called watcher nodes, watching the traffic but not participating in consensus.

This number was retrieved from using the StellarBeat API using a script found in the project repository.

Type of node	Recorded min	Recorded max
Full Validator	39	41
Basic Validator	16	18
Watcher	75	76

Table 4.6: Number of active nodes on the network, from measurements between 2021–05–14 and 2021–05–31.

The survey suggests that the network traffic is about the same for all nodes, regardless of whether they are a full or basic validator. This may not come as a surprise, as the separation of basic and full validators lie in how much data they store and not what data they send/receive. Additionally, the network traffic was similar for the experimental node and a Tier 1 node. This is particularly interesting, as these point of measurements represent two drastically different extremes in the network topology. While a Tier 1 node will be selected as a trusted organization by many, i.e. commonly added to a quorum set, the majority of their own quorum sets will be limited to other Tier 1 nodes. On the other hand, our experimental node was not included in any other nodes quorum set, while we had in turn added the entire Tier 1 set. As such, they are at two ends of the spectrum of "how many nodes include this in their quorum set", suggesting that traffic is relatively uniformly spread throughout the network.

Because of this, we dare approximate the complete network traffic (and hence electricity consumption) by multiplying our estimates with the number of validator nodes. We make the assumption that this holds for a discrete point in time, while it may not hold true as the network grows.

Indeed, extrapolating the power consumption presented in table 4.2, adjusting for PUE for computational, memory and storage work, and using the mean values for the number of nodes above, gives us values as presented in 4.7.

Finally, with the above results in mind, we arrived at a total mean electricity consumption per transaction, described by the equation 4.1, of 0.222 Wh per transaction.

PUE: 1.67 [18] Nodes: 132		
	Power per node	Total power
10 % quantile	130.473 W	172.224 kW
Mean	185.787 W	245.239 kW
90 % quantile	231.468 W	305.538 kW

Table 4.7: Estimated power consumption of the entire Stellar Network, with computation, storage and memory usage multiplied by PUE.

$$
\begin{aligned}
e_{transaction} = \quad & \tfrac{\mathrm{PUE}\times(e_{CPU}+e_{RAM}+e_{storage})+e_{network}}{N} \\
\text{where} \quad & N : \text{transactions in an hour}, \\
& \mathrm{PUE} : 1.67, \\
& e_{CPU,RAM,storage,network} : \text{mean electricity during an hour}
\end{aligned}
\tag{4.1}
$$

Chapter 5

Discussion

The aim of the study was to investigate how the electricity consumption of the Stellar payment network can be measured, and to arrive at a generalized Wh-per-transaction estimate. The results suggested a total of 0.222 Wh/transaction (or 0.000 222 kWh), which can be better understood by relating[1] it to the consumption of Bitcoin, Ethereum, and VISA, as done in Table 5.1.

System	Electricity consumption per transaction
Bitcoin [57]	1 575.93 kWh
Ethereum [58]	107.75 kWh
VISA [59]	0.00092 kWh
Stellar (this study)	0.00022 kWh

Table 5.1: Comparison of electricity consumption per transaction between Bitcoin, Ethereum, VISA and the corresponding value derived for Stellar in this study.

The method chosen for this study — software-based power measurements combined with generalized conversion factors to account for storage and network usage — worked well for a dedicated host, albeit required a fair amount of work to research and implement, without being able to eliminate certain large sources of uncertainty discussed below. Indeed, Davy [45] and Sommer et al. [2] reach similar conclusions — that better tooling and research is required to obtain reliable and precise estimates. This section will discuss what conclusions can be drawn from the results presented above while critically reflecting on the methods applied, before putting the conclusions reached in a larger context.

[1] A comparison of Stellar to Bitcoin, Ethereum and VISA requires a disclaimer. Stellar is not primarily a cryptocurrency, and thus serves a different purpose than both Bitcoin and Ethereum. In a way, it is comparing apples to pears. Similarly, it may be argued that Horizon should be included in comparisons with VISA, as the estimate of VISA also includes the API-layer, which this study on `stellar-core` does not. The relevance of comparing does not lie in these being able to replace each other, but rather in providing an intuition of the magnitude of electricity consumption of Stellar.

5.1 A ghost in the wires?

The results suggest that the vast majority of electricity consumption in the operation of the Stellar network can be attributed to network traffic, and not computation, memory usage, nor file storage.

With understanding of the consensus algorithm's design, relying on message passing rather than computational work, it becomes clear that arbitrary, complex computational work is not going to contribute to power consumption like it does in proof-of-work algorithms. Instead, as the algorithm builds on several rounds of federated voting where each node needs to be aware of the votes from its quorum set, every five-second consensus round is characterized by a phase of intense message-passing before a single value is committed to. The results indicate that it is the passing of messages that most significantly contribute to the electricity consumption of the network.

The translation of data traffic to electricity consumption does come with a few noteworthy sources of uncertainty. First, the amount of data transmitted and received may vary between different nodes. The results from the survey suggest that the variance is indeed fairly small, as discussed in Section 4.3, but this is something that could benefit from further investigation. It may be the case that certain quorum arrangements lead to a network topology that causes substantially higher network traffic to or from one or many nodes. Second, and perhaps more significant, is the uncertainty in the generalized conversion factor between volume of traffic to electricity consumption. Indeed, as discussed in Appendix A, while serious improvements to the energy efficiency of global ICT infrastructure has been made in the past years, estimates of electricity consumption varies drastically between studies. Moreover, the electricity-per-GB factor chosen for this study is an estimate from 2016, which makes it five years old at the time of conducting this project. A discrepancy between the approximated value and the actual electricity cost of transmitting a GB of data could have a significant impact on the conclusions drawn in this study regarding the power consumption of the four factors. However, given the recent trends, it is unlikely that the electricity consumption of transmitting data has increased, and as such the overall conclusion of a low electricity consumption can be presumed intact. Additionally, if the global ICT infrastructure continues to improve in efficiency, the energy footprint of the Stellar network may decrease considerably. With that being said, Stellar may not need to wait for improvements in infrastructure to further reduce the electricity impact of the payment network, further discussed in Section 5.2.1 below.

The second most significant source of power consumption was found to be the storage, where 6.5 W was attributed per node — a fixed amount per disk, based on an assumption that a single HDD drive was the most common storage alternative. While this was true for some validators, it was not true for the respondents to the survey — an SSD drive was the more common alternative. With that being said, this does not considerably impact the resulting power consumption, given that the approximate estimate for an SSD drive is only 0.5 W less, something that in relation to the network power consumption makes little impact.

Given how common full validators turned out to be in relation to basic validator nodes, including the power consumption from storing archive history could have been a relevant addition for more precise electricity measurements. This was excluded from the scope of this study, something that can be argued to have had little, but maybe not insignificant, impact on the total energy impact. At least if using the continuous coefficients described in equation 3.4, as a storage of 2–4 TB would result in an additional power consumption of 1.3–2.6 W. The results also revealed a large over-capacity of some nodes, including our own experimental node, suggesting that a single drive or a shared drive may be more suitable for basic validators from an energy perspective. One source of uncertainty here is the factor used for the estimated power consumption of an HDD drive. The number is based on projections for 2020 from a 2015-study [54]. A more up-to-date study may more accurately take into account the development since then, as well as provide future projections.

Lastly, the electricity consumption of RAM and CPU turned out to be least significant from an electricity consumption perspective. In fact, the average power consumption was surprisingly low, especially considering a Thermal Design Power (TDP) of 65 W for the selected processor [60]. However, the low power consumption is supported by the low average CPU usage measured as well as reported by respondents to the survey. Indeed, the CPU was idle for over 95 % of the total CPU-time, and the memory consumption was rarely above 1 GB — a behaviour that other nodes reported sharing in the survey. What is interesting is that many nodes (including our experimental one) had significantly higher memory and CPU capacity. This seeming over-capacity can partly be justified by the spikes in CPU consumption, visualized in Figure 4.1. Indeed, given how the protocol operates, it is only towards the end of each consensus round that transactions are processed, the ledger-database is read to and written from, and signatures are verified as well as created. This explains sudden, short spikes in the CPU usage, in-between which the CPU is mostly idle. These spikes are something that needs to be accounted for when selecting processor model. The RAPL power measurements suggest that this results in very little power being drained on average, but it is not unthinkable that the software-based method underestimates the long-term average power consumption. More frequent power measurements, or the addition of a physical power meter, could validate if this is the case or not.

At a glance, Figure 4.1 suggests clear correlation in spikes between transaction count, network traffic, and computational work. Additionally, the scatter-plot of electricity cost against ledger activity in Figure 4.6 illuminates how failed and successful transactions together contribute to a total electricity usage. While the electricity consumption varied for a similar number of transactions per hour, as seen in the vertical cluster of successful transactions, the total number of transactions showed a stronger correlation with electricity consumption. Yet, it is worth noting that the electricity consumption varies quite significantly, ca 80 Wh in the measurements on a single node, for a similar number of transactions, suggesting that it is not only the number of transactions but also the contents of the transaction that impacts electricity consumption. Recall that a transaction is a bundle of operations, and that there exists 21 different operations [42]. It is then reasonable to assume that certain operations may incur higher electricity costs, e.g. if they cause more numerous or expensive operations to the ledger. It may also be the case that a larger number of ballot rounds were required for that specific set of transactions, causing more message passing for said consensus round, or that the network entered so-called surge pricing mode, which is a message-passing process that handles the case when more than the max number of operations are nominated to the ledger. Such factors would be of relevance to investigate in further research, as they could reveal potential areas of optimization. In a similar vein to that of successful transactions, there seems to be no clear correlation between the number of successful operations and the power consumption, further supporting that it is not only the number of successful operations that matter, but also other factors such as operation type.

5.2 Room for improvement

As previously discussed, the Stellar Consensus Protocol is not intrinsically dependent on power consumption to ensure safety, and as such, there is room for optimization that can decrease the total electricity consumption. Figure 4.2 shows that the number of operations averages around 300 in a block, far less than the allowed 1 000 operations per block. This, together with the prevalence of arbitrage spam, is part of the reason why SDF has not yet had the reason to focus their efforts on optimizing the resource usage for a higher capacity, but instead has focused on reliability and ease-of-use.

It is, however, worth emphasizing that minimizing the resource usage of running a core node has benefits beyond reducing the electricity consumption. A fundamental prerequisite of the Stellar ecosystem's health is that running a core node should not be too expensive, and as such cannot set too demanding hardware requirements. While the network is not designed such that everyone is to run their own validator node, a core principle is that anyone *could* run a node, or more precisely that it should not be restricted to entities with significant capital, as that would limit the democratic decentralization and possibly prevent e.g. non-profit organizations from participating in the network.

5.2.1 Naïve flooding

As shown in the results section, one area worth focusing optimization efforts on in order to reduce electricity consumption is that of network traffic. Indeed, today `stellar-core` employs a naïve peer-to-peer flooding algorithm, where traffic duplication is only inhibited on a per-link basis. The so-called overlay network through which messages are passed is a peer-to-peer virtual broadcast network organized using a crude "everyone sends everything to everyone" structure. A node will flood every message to all nodes to which it has not already sent said message to, or received said message from [61]. While this approach is robust, it does lead to enormous duplication of traffic — internal empirical measurements suggest quadratic duplication in relation to network node count. There is currently ongoing work in SDF to build sufficient network simulation infrastructure to test and validate other approaches.

Replacing the naïve flooding overlay implementation with a structured peer-to-peer protocol is suggested towards the end of the SCP white paper [1]. This study suggests that doing so could significantly reduce the total electricity consumption of the Stellar network, without having to alter the consensus mechanism. Indeed, a solution such as SplitStream [62] could potentially reduce the volume of network traffic significantly without infringing on the decentralization. Moreover, it is plausible that reducing the duplicate network traffic would decrease the overall system resource usage, as less data has to be marshaled or demarshaled when sending and receiving messages, respectively, and hence allow cheaper hardware for validator node runners.

5.2.2 Sharding

While the storage requirements of today are very low, the ledger has grown in length alongside increased network activity and more accounts. This is particularly true for the nodes running as a full validator or with a history archive. If Stellar is to facilitate a larger portion of the global financial infrastructure, it may at some point be necessary to partition the complete ledger if one is to avoid the storage requirements for a single node becoming too high. One way of achieving this is through a concept known as sharding. A sharding scheme enables horizontal scaling through dividing the processing of operations among smaller groups of nodes (called shards). This allows shards to work in parallel on a subset of transactions, which has the possibility of maximizing performance and improving throughput, as well as requiring less overhead in form of communication, computation and storage [63]. Sharding is one of the approaches promising scaling in decentralized blockchain systems, but is known to be difficult to implement without skewing trust. If sharding is implemented in the Stellar network, investigating its impact on electricity consumption would be of relevance.

5.3 Validity of extrapolation to the network

In this study, the power estimate of a single node was extrapolated to the entire network in a linear model, assuming that the power consumption is reasonably uniformly distributed. As the network consumption turned out to be the largest source of electricity usage, handled by a factor independent to the hardware choice of individual nodes, this assumption is justifiable within the limits of this study. It may, however, be the case that the power drain from hardware varies significantly between nodes. An extreme case on one end is that of the Raspberry Pi which one private validator node runner reported using, which has a significantly smaller power supply unit (PSU) than a normal server. On the other end of the spectrum lies the more common dedicated servers, with far more powerful hardware. A CPU operating closer to its TDP, commonly around 60 W, would greatly impact the total power consumption of a single node.

In an ideal scenario, software services would be able to access precise data on power consumption of their service as well as the server as a whole through a standardized interface. But until this is available, measurements from the Intel RAPL interface coupled with factors provide justifiable estimates, especially if verified with physical power meter readings.

To better understand the variances within the network, as well as how the electricity consumption is affected by growth in the network, Stellar could offer a standardized way of collecting, monitoring and publishing data that can contribute to modelling power consumption, e.g. through the standardized process of submitting a SEP (Stellar Ecosystem Proposal). While this study suggests that the most important factor to monitor today is the network traffic, more comprehensive data could reveal significance of other factors in specific situations, especially in relation to network growth. This could also help ensure that the electricity consumption of the network remains low over time.

Given the decoupling of extreme electricity consumption and achieving decentralized trust suggested by this study, the methods applied are not isolated to blockchain solutions. Instead, the concepts can be generalized to distributed software services on a broader scale. This connects Stellar to part of a larger question of the significant yet difficult-to-estimate energy consumption of the ICT sector. Arguably, an equally relevant context to place these energy estimates in is that of distributed peer-to-peer software services. In doing so, applying a framework such as that developed by Seo et al. [64] could provide rigour to the results obtained.

5.3.1 What happens as the network grows?

The growth of the Stellar network can imply several things — there are different directions in which it can scale. The three discussed here is a higher number of transaction per consensus round (higher bandwidth), a longer ledger (more accounts and a longer history), and a greater number of nodes (more safety).

More operations per consensus round

Increasing the maximum number of operations included in each consensus round serves the purpose of increasing the bandwidth of the payment network, something that may become necessary if the ledger activity continues growing. For reference, VISA has a transaction speed of 65 000 transactions per second, and Stellar's competitor Ripple supports 1 500 transactions per second, while Bitcoin can facilitate as little as an average of 7 transactions per second [65]. The maximum throughput of Stellar's payment network is today on average 200 operations per second. As such, if Stellar continues growing in popularity, it may be necessary to increase the capacity of the network.

As seen in the results section, an increased number of total transactions — successful and failed — seem to increase the total electricity consumption of the network. This, as more messages communicated across the network mean more signatures need to be verified, and the amount of work required to apply transactions to the ledger increases. However, the survey indicates that there remains a serious amount of unused capacity in the servers powering the network to increase the ledger size without requiring more powerful nodes, while still being able to handle spikes in traffic. If the overlay network is optimized further, the Stellar network may very well be able to handle increased capacity without an increased electricity consumption.

Important to note is also that an increased ledger capacity may cause the cumulative ledger history to increase at a higher rate.

Longer ledger history

A larger number of accounts would increase the storage capacity required to run a node. For full validators, who store a history archive instead of only the recent state of the ledger, the growth will be quicker than for basic validators. Using the model from this study, the storage requirements for a basic validator can increase many times before the contents cannot fit on a standard-sized storage drive. As such, up to a critical breaking point where the average basic validator would require two disks to operate, a longer ledger would not cause increased electricity consumption of storage drives on the Stellar network. However, once that threshold has been passed, the power consumption could increase by a fairly significant amount, doubling the power consumption from storage per validator.

For full validators and archive nodes, the effects of a longer ledger could have a larger and more continuous impact. A tenfold increase in the ledger length would indeed increase the relevance of studying the contribution of archive storage to electricity consumption. Whether gains in capacity and efficiency will continue to cancel out the effects of this scenario remains to be seen. Additionally, there are other ways in which Stellar could avoid this scenario, such as partitioning the network through sharding, discussed in 5.2.

Outside direct impacts of increased storage of data, the effects of a longer ledger on electricity consumption is not as clear. As long as Stellar continues to have checkpoints in the ledger instead of requiring that the state is computed through iterating over the entire history like in Bitcoin, impacts on memory consumption or computational requirements are unclear. One thing that is likely to increase is the data transmission required for newly started nodes and for API nodes, who need to reingest the history archive — something that was considered out of scope for this study.

More validating nodes

As validator nodes from a diverse set of organizations connect to the Stellar payment network, the overall security and decentralized trust may increase.

Increasing the number of validating nodes on the network is assumed to at least cause a linear growth in electricity consumption of a transaction. Each node will be powered by additional hardware, and comprise an additional node to which statements need to propagate in the overlay network. However, it may incur a greater increase than so. As discussed above, the data traffic has been suggested to have a near quadratic growth with the number of nodes. It is also possible that the taking more votes into account may increase the number of balloting rounds.

The exact impact on the computational work and memory consumption from the consensus algorithm caused by an increase in the number of nodes is not evident nor trivial, and will depend on a number of configuration parameters as well as the network topology.

5.4 Future research

In order to derive more precise and more relevant estimates of the electricity consumption of the Stellar payment network, a number of topics are interesting for future studies.

First and foremost, well-supported and up-to-date coefficients for the electricity consumption of transferring data on the global ICT infrastructure would greatly contribute to the rigour of the estimate. Similarly, suggesting a systematic way for Stellar validator nodes to collect and publish data that contribute to better understanding the electricity consumption across the distributed system would be of great value to better understand variances within the network. Perhaps this study can serve as an inspiration to such a proposal.

Moreover, it would be of interest to study what factors contribute to the variation in electricity consumption for similar number of transactions and operations. For example, isolating which type of operations or network topology cause higher electricity consumption may reveal areas to focus optimizations on. It would also be of great relevance to extend the scope of the estimate to include Horizon, Stellar's API client.

Lastly, simulation and theoretical models on the impact of network growth on the Stellar payment network's total electricity consumption should better account for Stellar's capacity to support a larger portion of the global financial infrastructure, and highlight where efforts to decrease the electricity consumption should be focused. In a world with an ongoing climate crisis requiring significant reductions in energy usage across all sectors, it is critical that the financial infrastructure actively reduces its energy consumption.

Chapter 6

Conclusions

The results in this study suggest a staggeringly low electricity consumption of the Stellar network, arriving at a generalized estimate of 0.222 Wh per transaction, with room for additional, significant optimization to further decrease the electricity consumption.

As such, the Stellar network is an example of a blockchain implementation decoupling high electricity consumption from achieving decentralized trust, breaking free of a serious adverse side-effect that Bitcoin and Ethereum has been characterized by. Yet, the method applied leaves several sources of uncertainty, something which improvements to estimates of the impact of distributed systems on electricity consumption could reduce. Additionally, Stellar may want to provide a way for node operators to contribute with statistics on system resource usage in general and electricity consumption specifically. Such information could improve understanding of the variance within the network as well as how the electricity consumption is affected by the network growing.

Through achieving an open-membership blockchain solution that decouples electricity consumption from decentralized trust in an open source software environment, Stellar is an example that other may want to look at when designing their own blockchain implementation in areas within and outside of fintech. In conclusion, this study indicates that the energy consumption will not be a limiting factor for Stellar's possibility to scale to the global financial infrastructure.

Bibliography

[1] M. Lokhava, G. Losa, D. Mazières, G. Hoare, N. Barry, E. Gafni, J. Jove, R. Malinowsky, and J. McCaleb, "Fast and secure global payments with stellar.," *Operating Systems Principles*, pp. 80 – 96, 2019.

[2] E. Sommer, M. Adler, J. Perkins, J. Thiel, H. Young, C. Mozen, D. Daya, and K. Sundstrom, "Cloud jewels: Estimating kwh in the cloud," Apr. 2020. https://codeascraft.com/2020/04/23/cloud-jewels-estimating-kwh-in-the-cloud/ Accessed on 2021-03-30.

[3] D. Mytton, "Assessing the suitability of the greenhouse gas protocol for calculation of emissions from public cloud computing workloads," *Journal of Cloud Computing*, vol. 9, p. 45, Aug. 2020.

[4] S. Pratima, J. Rajni, and B. Malaya Dutta, "Blockchain technology for cloud storage: A systematic literature review," *ACM Computing Surveys (CSUR)*, vol. 53, no. 4, pp. 1 – 32, 2020.

[5] N. Patelli and M. Mandrioli, "Blockchain technology and traceability in the agrifood industry," *Journal of food science*, vol. 85, no. 11, pp. 3670 – 3678, 2020.

[6] E. P. Fedorova and E. I. Skobleva, "Application of blockchain technology in higher education," *European Journal of Contemporary Education*, vol. 9, no. 3, pp. 552 – 571, 2020.

[7] I. Abu-Elezz, A. Hassan, A. Nazeemudeen, M. Househ, and A. Abd-Alrazaq, "The benefits and threats of blockchain technology in healthcare: A scoping review," *International journal of medical informatics*, vol. 142, p. 104246, 2020.

[8] CoinMarketCap, "All cryptocurrencies." https://coinmarketcap.com/all/views/all/.

[9] K. Caliskan, "Data money: The socio-technical infrastructure of cryptocurrency blockchains," *Economy and Society*, vol. 49, no. 4, pp. 540–561, 2020.

[10] R. Partington, “Bitcoin boom threatens to turn it into pure gold.” https://www.theguardian.com/technology/2021/jan/09/bitcoin-boom-threatens-to-turn-it-into-pure-gold, Jan. 2021. Accessed on 2021-05-10.

[11] L. C. Strauss, “Bitcoin hits a new high. what to know..” https://www.barrons.com/articles/bitcoin-hits-new-high-breaking-60k-barrier-51615652498, Mar. 2021. Accessed on 2021-05-10.

[12] Sveriges Riksbank, “E-krona,” Apr. 2021. Accessed on 2021-04-28. https://www.riksbank.se/en-gb/payments--cash/e-krona/.

[13] Sveriges Riksbank, “E-krona pilot phase 1,” Apr. 2021. Accessed on 2021-05-06. https://www.riksbank.se/globalassets/media/rapporter/e-krona/2021/e-krona-pilot-phase-1.pdf.

[14] S. Nakamoto, “Bitcoin: A peer-to-peer electronic cash system,” 2008. https://bitcoin.org/bitcoin.pdf.

[15] M. Drijvers, S. Gorbunov, G. Neven, and H. Wee, “Pixel: Multi-signatures for consensus.” Cryptology ePrint Archive, Report 2019/514, 2019. https://eprint.iacr.org/2019/514.

[16] IEA, “Data centres and data transmission networks,” 2020. Paris https://www.iea.org/reports/data-centres-and-data-transmission-networks.

[17] C. Koronen, M. Åhman, and L. J. Nilsson, “Data centres in future european energy systems—energy efficiency, integration and policy,” *Energy Efficiency (1570646X)*, vol. 13, pp. 129 – 144, Dec. 2019.

[18] R. Bashroush and A. Lawrence, “Beyond pue: tackling it’s wasted terawatts.,” tech. rep., Uptime Institute, 2020.

[19] A. Shehabi, S. J. Smith, E. Masanet, and J. Koomey, “Data center growth in the united states: decoupling the demand for services from electricity use,” *Environmental Research Letters*, vol. 13, p. 124030, Dec. 2018.

[20] E. Masanet, A. Shehabi, N. Lei, S. Smith, and J. Koomey, “Recalibrating global data center energy-use estimates.,” *Science (New York, N.Y.)*, vol. 367, no. 6481, pp. 984 – 986, 2020.

[21] S. Jiang, Y. Li, Q. Lu, Y. Hong, D. Guan, Y. Xiong, and S. Wang, “Policy assessments for the carbon emission flows and sustainability of bitcoin blockchain operation in china,” *Nature Communications*, vol. 12, p. 1938, Apr. 2021.

[22] A. de Vries, “Bitcoin boom: What rising prices mean for the network’s energy consumption,” *Joule*, vol. 5, Mar. 2021.

[23] C. Mora, R. L. Rollins, K. Taladay, M. B. Kantar, M. K. Chock, M. Shimada, and E. C. Franklin, “Bitcoin emissions alone could push global warming above 2 degrees C,” *Nature Climate Change*, vol. 8, pp. 932+, Nov. 2018.

[24] The World Bank, “Personal remittances, received (current us$),” 2021. World Bank Open Data. Accessed on 2021-04-23. https://data.worldbank.org.

[25] S. Cecchetti and K. Schoenholtz, “The stubbornly high cost of remittances,” Mar. 2018. Accessed on 2021-04-23. https://voxeu.org/article/stubbornly-high-cost-remittances.

[26] Stellar Development Foundation, “Anchor basics: Stablecoin/asset issuance & fiat on/off-ramp - stellar.” https://www.stellar.org/learn/anchor-basics, 2021. Accessed on 2021-05-10.

[27] StellarExpert, “Activity on stellar public network,” 2021. Accessed on 2021-05-15. https://stellar.expert/explorer/public/network-activity.

[28] Stellar Development Foundation, “Ukrainian ministry of digital transformation to develop virtual assets and to facilitate cbdc infrastructure with the stellar development foundation.” https://stellar.org/press-releases/ukrainian-ministry-of-digital-transformation-to-develop-virtual-assets-and-to-facilitate-cbdc-infrastructure-with-the-stellar-development-foundation, 2021. Accessed on 2021-05-14.

[29] N. Zivic, C. Ruland, and O. Ur-Rehman, “Addressing byzantine fault tolerance in blockchain technology,” in *2019 8th International Conference on Modeling Simulation and Applied Optimization (ICMSAO)*, pp. 1–5, 2019.

[30] K. Driscoll, B. Hall, M. Paulitsch, P. Zumsteg, and H. Sivencrona, “The real byzantine generals,” in *The 23rd Digital Avionics Systems Conference (IEEE Cat. No.04CH37576)*, vol. 2, pp. 6.D.4–61, 2004.

[31] K. Driscoll, B. Hall, H. Sivencrona, and P. Zumsteg, “Byzantine fault tolerance, from theory to reality,” in *Lecture Notes in Computer Science*, pp. 235–248, Berlin, Heidelberg: Springer Berlin Heidelberg, 2003.

[32] Blockchain.com, “Bitcoin charts — total hash rate.” https://www.blockchain.com/charts/hash-rate. Accessed on 2021-05-11.

[33] Z. Voell, “Bitcoin miners saw 33% revenue increase in december.” https://www.coindesk.com/bitcoin-mining-revenue-december-2020, 2021. Accessed on 2021-05-03.

[34] D. K. Gifford, “Weighted voting for replicated data,” in *Proceedings of the Seventh ACM Symposium on Operating Systems Principles*, SOSP ’79, (New York, NY, USA), p. 150–162, Association for Computing Machinery, 1979.

[35] R. H. Thomas, “A majority consensus approach to concurrency control for multiple copy databases,” *ACM Trans. Database Syst.*, vol. 4, p. 180–209, June 1979.

[36] R. Guerraoui and M. Vukolić, “Refined quorum systems,” *Distrib. Comput.*, vol. 23, p. 1–42, Sept. 2010.

[37] D. Malkhi and M. Reiter, “Byzantine quorum systems,” *Distributed Computing*, vol. 11, pp. 203–213, Oct. 1998.

[38] M. Naor and A. Wool, “The load, capacity, and availability of quorum systems,” *SIAM Journal on Computing*, vol. 27, no. 2, pp. 423–447, 1998.

[39] M. Lokhava, "Intuitive stellar consensus protocol." https://www.stellar.org/developers-blog/intuitive-stellar-consensus-protocol, Oct. 2019. Accessed on 2021-05-11.

[40] J. Rice, "Transaction submission, timeouts, and dynamic fees faq." https://stellar.org/developers-blog/transaction-submission-timeouts-and-dynamic-fees-faq, Feb. 2021. Accessed on 2012-05-21.

[41] "Should we increase the minimum fee?." Google Groups forum, https://groups.google.com/g/stellar-dev/c/_XUh7dmrjUc/m/ob4dFyj4CQAJ, 2021. Accessed on 2012-05-21.

[42] "List of operations." Stellar Developer Docs, https://developers.stellar.org/docs/start/list-of-operations/, 2021. Accessed on 2012-05-21.

[43] "XDR: External Data Representation standard." RFC 1014, June 1987. Accessed on 2021-06-02. https://rfc-editor.org/rfc/rfc1014.txt.

[44] B. Davy, "Evaluating the carbon footprint of a software platform hosted in the cloud," Dec. 2020. https://medium.com/teads-engineering/evaluating-the-carbon-footprint-of-a-software-platform-hosted-in-the-cloud-e716e14e060c Accessed on 2021-03-30.

[45] B. Davy, "Estimating aws ec2 instances power consumption," Mar. 2021. https://medium.com/teads-engineering/estimating-aws-ec2-instances-power-consumption-c9745e347959 Accessed on 2021-04-07.

[46] P. Henderson, J. Hu, J. Romoff, E. Brunskill, D. Jurafsky, and J. Pineau, "Towards the systematic reporting of the energy and carbon footprints of machine learning," Feb. 2020.

[47] M. Fahad, A. Shahid, R. R. Manumachu, and A. Lastovetsky, "A comparative study of methods for measurement of energy of computing," *Energies*, vol. 12, no. 11, 2019.

[48] K. Nizam Khan, M. Hirki, T. Niemi, J. Nurminen, and Z. Ou, "Rapl in action: Experiences in using rapl for power measurements," *ACM Transactions on Modeling and Performance Evaluation of Computing Systems (TOMPECS)*, vol. 3, Jan. 2018.

[49] P. Roose, M. Soltane, D. Makhlouf, and K. Okba, "Predictions & Modeling Energy Consumption For IT Data Center Infrastructure," in *Advances In Intelligent Systems And Computing*, vol. 912, (Tanger, Morocco), pp. 1–11, 2018.

[50] L. Weiwei, S. Fang, W. Wentai, L. Keqin, W. Guangxin, and A.-a. Mohammed, "A taxonomy and survey of power models and power modeling for cloud servers.," *ACM Computing Surveys*, vol. 53, no. 5, pp. 100 – 140, 2020.

[51] H. Pihkola, M. Hongisto, O. Apilo, and M. Lasanen, "Evaluating the energy consumption of mobile data transfer—from technology development to consumer behaviour and life cycle thinking," *Sustainability*, vol. 10, no. 7, 2018.

[52] D. Schien, V. C. Coroama, L. M. Hilty, and C. Preist, "The energy intensity of the internet: Edge and core networks," in *ICT Innovations for Sustainability* (L. M. Hilty and B. Aebischer, eds.), (Cham), pp. 157–170, Springer International Publishing, 2015.

[53] J. Aslan, K. Mayers, J. G. Koomey, and C. France, "Electricity intensity of internet data transmission: Untangling the estimates," *Journal of Industrial Ecology*, vol. 22, no. 4, pp. 785–798, 2018.

[54] A. Shehabi, S. J. Smith, D. A. Sartor, R. E. Brown, M. Herrlin, J. G. Koomey, E. R. Masanet, N. Horner, I. L. Azevedo, and W. Lintner, "United states data center energy usage report," tech. rep., Ernest Orlando Lawrence Berkeley National Laboratory, June 2016.

[55] R. Rouvoy and C. Belgaid, "energy-scripts." https://github.com/powerapi-ng/energy-scripts. Accessed on 2021-05-12 (a78094e).

[56] S. Godard, "sar manual page," Aug. 2020. Accessed on 2021-05-27. http://sebastien.godard.pagesperso-orange.fr/man_sar.html.

[57] Digiconomist, "Bitcoin energy consumption index." https://BitcoinEnergyConsumption.com. Accessed on 2021-04-28, 2021-06-10.

[58] Digiconomist, "Eteherum energy consumption index." https://EthereumEnergyConsumption.com. Accessed on 2021-06-10.

[59] A. A. Sori, M. Golsorkhtabaramiri, and A. M. Rahmani, "Cryptocurrency grade of green; iota energy consumption modeling and measurement," in *2020 IEEE Green Technologies Conference(GreenTech)*, pp. 80–82, 2020.

[60] intel.com, "Intel® core™ i7-7700 processor," Aug. 2021. Accessed on 2021-05-30. https://ark.intel.com/content/www/us/en/ark/products/97128/intel-core-i7-7700-processor-8m-cache-up-to-4-20-ghz.html.

[61] "stellar-core — src/overlay/floodgate.cpp," 2021. https://github.com/stellar/stellar-core/blob/ff56e7f207b918aae53710abcc001511ead7ece9/src/overlay/Floodgate.cpp#L86-L140.

[62] M. Castro, P. Druschel, A.-M. Kermarrec, A. Nandi, A. Rowstron, and A. Singh, "Splitstream: High-bandwidth content distribution in cooperative environments," in *Peer-to-Peer Systems II* (M. F. Kaashoek and I. Stoica, eds.), (Berlin, Heidelberg), pp. 292–303, Springer Berlin Heidelberg, 2003.

[63] M. Zamani, M. Movahedi, and M. Raykova, "Rapidchain: Scaling blockchain via full sharding," in *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, CCS '18, (New York, NY, USA), p. 931–948, Association for Computing Machinery, 2018.

[64] C. Seo, G. Edwards, S. Malek, and N. Medvidovic, "A framework for estimating the impact of a distributed software system's architectural style on its energy consumption," in *Seventh Working IEEE/IFIP Conference on Software Architecture (WICSA 2008)*, pp. 277–280, 2008.

[65] M. Laboure, "When digital currencies become mainstream," tech. rep., Deutsche Bank Research, Feb. 2021. Accessed on 2021-06-05. https://rfc-editor.org/rfc/rfc1014.txt.

[66] L. A. Barroso, U. Hölzle, and P. Ranganathan, "The datacenter as a computer: Designing warehouse-scale machines, third edition," *Synthesis Lectures on Computer Architecture*, vol. 13, no. 3, pp. i–189, 2018.

[67] W. Van Heddeghem, S. Lambert, B. Lannoo, D. Colle, M. Pickavet, and P. Demeester, "Trends in worldwide ict electricity consumption from 2007 to 2012," *Computer Communications*, vol. 50, pp. 64–76, 2014. Green Networking.

[68] M. Avgerinou, P. Bertoldi, and L. Castellazzi, "Trends in data centre energy consumption under the european code of conduct for data centre energy efficiency," *Energies*, vol. 10, no. 10, 2017.

[69] A. Andrae, "New perspectives on internet electricity use in 2030," *Engineering and Applied Science Letters*, vol. 3, pp. 19–31, June 2020.

Appendix A

Electricity consumption of data centres

Over thirty years since the birth of the World Wide Web, internet technologies continue to transform industries and cultures across the globe. The growing integration of internet services such as email, video streaming, web conferencing, social networks and search into everyday life have accelerated a trend toward server-side or "cloud" computing. Between 2010 and 2019, the global internet traffic grew by 1 100 % [16], a trend that was only accelerated by the pandemic outbreak of COVID-19 in 2020.

Modern internet-connected services depend on data centres, constituting complex systems. In the third edition of *The Datacenter as a Computer: Designing Warehouse-Scale Machines*, Barroso et al. describe the intricate complexities of the computing platforms that power cloud computing. They highlight how hardware and software is orchestrated to deliver reliable and performant internet services, abstracting the data centre itself to a warehouse-scale computer (WSC). Barroso et al. describes that it is this growth in both scale and complexity that has allowed infrastructure providers to drastically increase the efficiency of data centre computing in recent years [66].

While there is consensus on that data centres have become more efficient alongside the expanding demand for cloud computing, more exact estimates of global annual data centre electricity usage vary significantly. In 2020, the International Energy Agency (IEA) estimated data centres globally to account for 200 TWh of electricity usage, and data networks for 250 TWh [16], while a report from Bashroush and Lawrence (2020) estimate data centres to account for as much as 500 TWh of global electricity usage [18]. Additionally, a study from 2014 suggests that global data centre electricity was as high as 270 TWh in 2012 already [67], and yet another study from 2017 estimated that only the electricity consumption from data centres in the European Union would be as high as 104 TWh in 2020 only [68], making a global total of 200 TWh unlikely.

The uncertainty in current estimates of electricity usage seem to extend to predictions of future electricity usage and future efficiency gains. Shehabi and Masanet et al. [19, 20] are carefully optimistic, suggesting that if the structural trend of shifting from smaller traditional data centres to hyperscale data centres continues, the efficiency gains may continue keeping the growing energy demand in check, while at the same time acknowledging that it is notoriously difficult to predict long-term efficiency limits of IT infrastructure [19]. Andrae, on the other hand, presents a worst-case scenario where ICT electricity usage could contribute up to 23 % of globally released greenhouse emissions in 2030 [69].

In addition to the general concern of the exponentially growing demand for cloud computing, the specific issue of Bitcoin mining has raised further concerns of many. In 2018, Mora et al. projected the electricity demand from Bitcoin usage alone could push global warming above the 2 degree Celsius limit within a few decades, if it follows the rate of adoption of other popular technologies [23]. Similar fears are expressed by other scholars [21, 22], some warning that the hoped-for efficiency gains in ICT infrastructure may be cancelled out by the extreme growth of Bitcoin's miners.

With this in mind, it is important to attain transparency on the electricity consumption of current and future blockchain implementations and to ensure it is kept to a minimum. However, estimating the electricity consumption of a decentralized service is no trivial task, as hardware and external factors such as the server environment can drastically affect the electricity usage of running a given software service. Unlike centralized services, direct measurements are difficult or impossible, creating a dependence on estimates and extrapolation. Indeed, this applies to Stellar — an open-membership, distributed blockchain solution with a vision to support the global financial infrastructure.

Appendix B

Survey responses: Traffic usage of public nodes

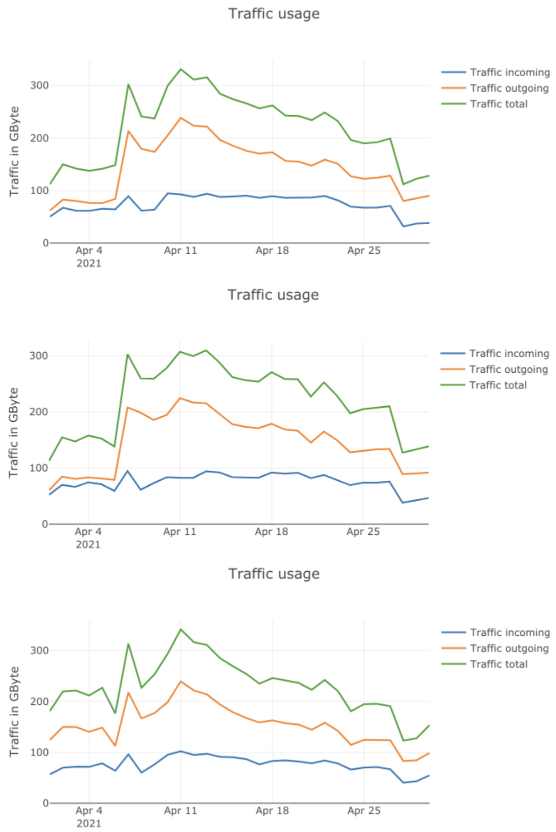

Figure B.1: Network traffic of three Tier 1 nodes belonging to the same organization, but each one on a separate server in different countries.

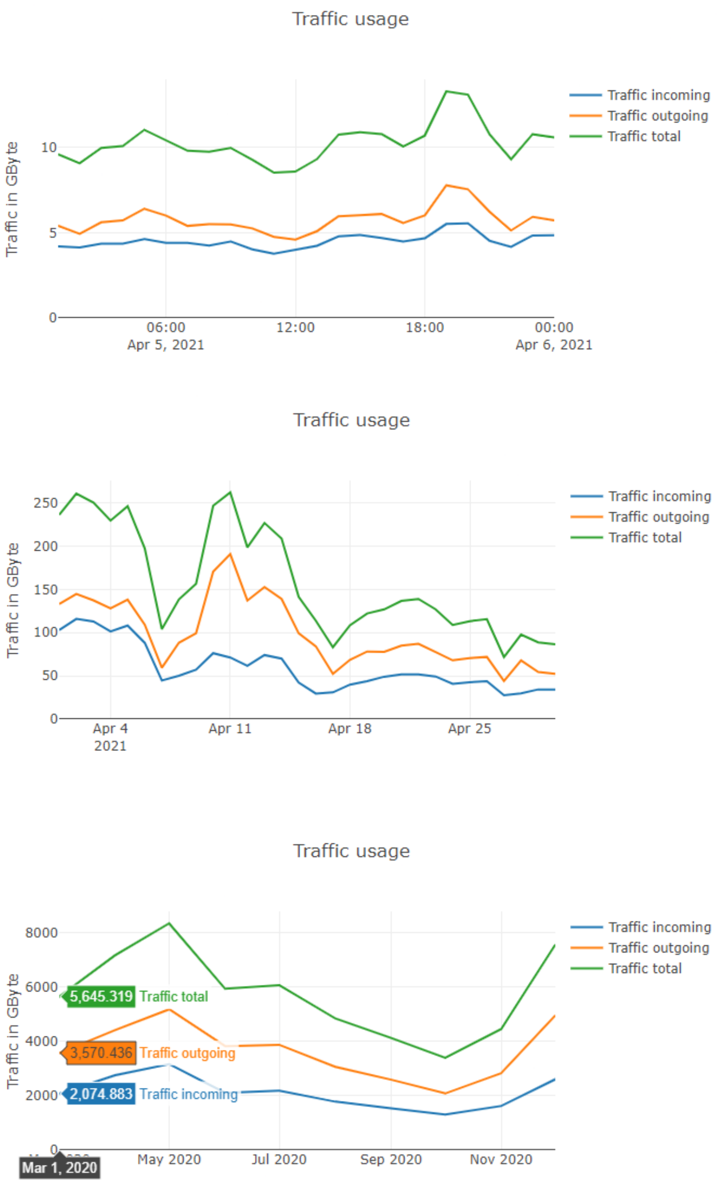

Figure B.2: Network traffic of a single Tier 1 node, showing traffic per hour, day and month, respectively.

Printed by Tryckeriet i E-huset, Lund 2021

LUND UNIVERSITY

Series of Bachelor´s theses
Department of Electrical and Information Technology
LU/LTH-EIT 2021-831
http://www.eit.lth.se