

# Energy Consumption Analysis of XRP Validator

Crystal Andrea Roma\* and M. Anwar Hasan†  
Department of Electrical and Computer Engineering  
University of Waterloo  
Waterloo, Canada

\*caroma@uwaterloo.ca, †ahasan@uwaterloo.ca

**Abstract**—The XRP Ledger is an open-source payment system with a primary focus on providing an efficient cross-border payment network. The network achieves distributed consensus and prevents double-spending all while remaining decentralized. Digital currencies which rely on proof-of-work strategies generally consume a great deal of energy. The XRP Ledger, on the other hand, uses the XRP Ledger Consensus Protocol, a unique process which is based on Byzantine agreement. The advancement of the XRP Ledger relies on the agreement of validator nodes which propose candidate transaction sets to be included in the next ledger through an iterative consensus process. Unlike Bitcoin miners, there is no equivalent reward scheme to incentivize volunteers to run XRP validator nodes. In this article, the energy consumption of an XRP validator is studied. Specifically, both the wall socket energy and the CPU package energy are measured using a software-based energy profiler and external measurement tools across 30 minute intervals. Various parameters in the configuration settings are also modified to study whether there is a significant change in the energy consumed. Using these results, estimated annual electricity costs are compared.

**Index Terms**—Blockchain, cryptocurrency, XRP, energy consumption, software profiling

## I. INTRODUCTION

In terms of market capitalization, Ripple’s XRP is currently the third-largest cryptocurrency, behind Bitcoin and Ethereum. The XRP Ledger is a decentralized cryptographic ledger powered by a network of peer-to-peer servers running the *rippled* software. The network uses the XRP Ledger Consensus Protocol (XRP LCP) which is fundamentally different from proof-of-work or proof-of-stake algorithms. This system is based on a Byzantine Fault Tolerant protocol through which the XRP Ledger is guaranteed to advance even with partial agreement by participants and the presence of malicious actors [1]. In this protocol, users develop a Unique Node List (UNL) of trusted validators they believe will not collude with one another. As long as a minimum overlap threshold is maintained between UNLs, the network state will remain consistent without a central authority [2].

Proof-of-work protocols used in decentralized open-source payment systems are known to be very greedy in terms of energy consumption, even consuming as much energy as entire countries such as Ireland and Austria [3]. The XRP LCP, on the other hand, has been shown to be very efficient in terms of transaction settling time and energy consumed.

The basis of the XRP LCP is to create a set of nodes which will not conspire. By consequence, it is important

to have diversity in the available validators to promote the stability and robustness of the decentralized network [4], [5]. Although the XRP Ledger network is quite energy efficient, it relies on the participation of validator nodes, akin to Bitcoin miners, in order to advance the ledger. Yet, unlike its Bitcoin counterparts, XRP validators are not rewarded newly minted XRP for their participation in the XRP LCP; in fact, all of the XRP available in the network have already been mined at its inception. This gives very little incentive to run a validator node. To the best of our knowledge, there has been no work that empirically studies the energy required to run an XRP validator node. The aim of this paper is to study the energy required to run a validator in order to quantify the cost of one’s participation. Further, it is explored whether changes in the build configuration settings have an impact on the energy expelled.

The rest of this report is organized as follows. In Section II, related work is explored. Section III provides a brief introduction to the rippled codebase, XRP server nodes, and the default configuration. In Section IV, the methodology by which the energy measurements are obtained is described. The results acquired from the default configuration and a comparative analysis between the energy measurements obtained when running an XRP validator node under different configurations are provided in Section V. Concluding remarks are provided in Section VI.

## II. RELATED WORK

Due to the large energy footprint of Bitcoin, many works have emerged to study the energy cost associated with this cryptocurrency. In [3], the author’s discuss different approaches which can be used to estimate the electricity consumption of the Bitcoin network. Project’s such as the Cambridge Bitcoin Electricity Consumption Index [6] and Digiconomist’s Bitcoin Energy Consumption Index [7] have been created to provide estimates and models to forecast the energy consumption of the Bitcoin network. The latter has also created a similar index for the Ethereum network [8]. In a Stanford and Stockholm University study, the energy consumed by the Visa, Bitcoin, Ethereum, and XRP Ledger networks are compared, where it is shown that the annual electricity consumption and electricity consumed per transaction characteristic to the XRP Ledger is the most sustainable of the four [9]. Unfortunately, the results of this study were founded on the assumption that an XRP validator consumes the same

amount of energy as a mail server [4] and were not obtained empirically [9].

### III. PRELIMINARIES

#### A. An Overview of *rippled*

The XRP Ledger is powered by the *rippled* software, an open-source codebase which is available for public download under the ISC open-source license [10]. It consists of a number of sub-modules which work together to support the overall functionality of the XRP Ledger network. This includes components responsible for the consensus algorithm, cryptographic functions, the ledger and database storage, as well as the overlay logic to support the peer-to-peer network. For more information regarding the project and its source code, the reader is referred to the XRP Ledger’s official documentation page [11] or to [12] for an independent audit of the *rippled* codebase.

Each *rippled* instance, once launched, consults a configuration file which contains information regarding the server, peer protocol, database, and other settings related to the *rippled* process. An example configuration file is provided with the installation. These settings can be fine-tuned based on the capabilities of the target hardware, the resources one is willing to dedicate to the *rippled* server, the mode in which one wishes to operate, and the desired performance. *Stock server* nodes follow the network and keep a local copy of the ledger. They can be further customized to run as a public hub to support the connectivity of the network or even to store a full history of the ledger. *Validator* nodes can perform all of the same tasks as a stock server and will also participate in consensus by signing and publishing their own validations [13]. For guidelines on network, hardware, and configuration recommendations, the reader is referred to [14].

#### B. Default Installation Details

During operation, a *rippled* server creates the NodeDB—a persistent database which stores ledger objects, transactions, and account data. The backend key/value database through which this is implemented can be chosen at runtime. The default implementation uses RocksDB, a general purpose persistent key/value store maintained by the Facebook Database Engineering team which can be used both on rotational and solid-state disks (SSD) [15]. Compared to other databases supported by *rippled*, RocksDB uses less disk storage and has better I/O latency but requires more RAM [16]. Other supported databases include NuDB, an append-only key/value store designed for *rippled*. NuDB requires an SSD to operate and unlike RocksDB, its performance does not degrade as the amount of stored data increases [14]. It is recommended that validators use RocksDB as it is not necessary for these nodes to store a large amount of ledger history.

The amount of disk space required to store the XRP Ledger will continue to grow as time passes, accruing approximately 12GB per day [14]. Fortunately, it is not necessary to store the full ledger’s history to participate in the network. A *rippled* server has the capability of automatically deleting data which

it no longer requires, a feature known as online deletion [17]. The number of ledgers a user wishes to keep can be specified in the configuration file. Guidelines on the amount of disk space required to store variable amounts of ledger data and their real time equivalents are available on the XRP Ledger capacity planning site [14]. The default online deletion parameter is set to 2000, which is approximately 2 hours worth of history.

Currently, the *rippled* software uses both Ed25519 and secp256k1 elliptic curve libraries for digital signatures; however, their support varies. In particular, validators can only use secp256k1 to sign messages and validations. This curve is known to be very efficient in practice due to the special structure of the underlying finite field which promotes fast arithmetic operations [18]. Additionally, the secp256k1 elliptic curve also possesses an efficiently computable *endomorphism* which can significantly improve the efficiency of scalar point multiplication—a critical operation in the elliptic curve digital signature algorithm (ECDSA). This feature is available in the secp256k1 library that *rippled* uses but is disabled by default and consequently, is not used within the default *rippled* installation [10].

### IV. METHODOLOGY

#### A. Measurement Tools

*CPU Energy Meter (Process-Level Energy)*: CPU Energy Meter is a linux-based tool which reads power statistics from the Running Average Power Limit (RAPL) domain and thus, requires no extra hardware [19]. The RAPL interface is a feature introduced by Intel’s Sandy Bridge-based processors and provides access to energy and performance counters. Depending on the platform, energy measurements from the system’s sockets (package), the CPU cores (power plane 0), the GPU (power plane 1), or the sum of energy consumed by memory in a given socket (DRAM) may be available for sampling [20]. The CPU Energy Meter also determines the maximal possible measurement interval at which to take measurements in order to minimize the overhead incurred by the tool itself.

*IgProf (Function-Level Energy)*: The aforementioned tool does not provide function-level energy analyses; rather, it generates energy consumption measurements for the duration of a specified command. This makes it difficult to perform optimization as it gives no insight into which portions of the software may be contributing most to the total energy consumption metric. IgProf is a profiling tool that provides function-level timing, memory usage, and energy consumption of software [21], [22]. The energy consumption profiler operates on the basis of statistical sampling and does not require manual instrumentation or recompilation of the software of interest, an important feature when profiling a large codebase such as *rippled*. The application uses the PAPI library which gives a user-friendly interface through which the RAPL counters can be sampled. Unfortunately, the energy profiler is still limited and does not account well for multiple threads of an application [21]. Thus, the IgProf profiles are used to give an

indicative estimate into which functions contribute most to the energy consumed to suggest configuration changes.

*KINCREA Meter (Wall Energy)*: To measure the wall socket energy, an external measurement tool is necessary. The KINCREA Electricity Usage Monitor Plug Power Meter is used [23]. The tool plugs directly into the wall socket and has an outlet into which the computer chord is inserted. The total accumulated energy since reset is reported with a display range of 0.000kWh-999kWh and with an error rate of  $\pm 1\%$ .

### B. Experimental Setup

A rippled instance (version 1.4.0) was built on a 64-bit processor Intel i5-2320 CPU @ 3.00GHz running Ubuntu 16.04 LTS with 16GB of RAM and 500GB SATA SSD storage (WD Blue 3D NAND). The CPU frequency governor is set to performance. The rippled configuration file uses all of the default configuration settings available at the time of installation and the log level is set to failure. A bash script was run which would initiate the rippled node and terminate it after a period of 30 minutes. Measurements from all three measurement tools are obtained simultaneously. This is repeated for 25 consecutive measurements, after which time the stored database is deleted. A total of 50 runs are acquired for each configuration. The various configurations tested are summarized in Table I. Configuration 1 refers to the default configuration as provided by the rippled installation and as described in Section III-B.

## V. EXPERIMENTAL RESULTS

### A. Default Installation

The average CPU package and wall energy obtained for the default installation (Configuration 1) are shown in the first row of Table II. At the time of writing, the on-peak electricity cost in Ontario is \$0.208/kWh [24]. Using this rate and the average wall energy consumption of 24.38Wh over 30 minutes of operation, it is estimated that the annual electricity cost for this particular configuration would be roughly \$88.84.

The measurements obtained from the default installation are chronologically plotted in Fig. 1. The package energy is obtained by the CPU Energy Meter and the wall energy is reported by the KINCREA monitor; all energy measurements have been converted to watt-hours (Wh). A clear pattern in the energy trace is observed, where periodic increases in the energy consumption are seen about every 4th measurement. As each run was sampled over a 30 minute period, this implies approximately 2 hours of operation. Interestingly, this time frame aligns with the interval at which online deletion is

TABLE I  
A SUMMARY OF THE DIFFERENT RIPPLED CONFIGURATIONS EXPLORED.

Configuration	NodeDB	Use Endomorphism	Online Delete
1	RocksDB	No	2000
2	RocksDB	Yes	2000
3	RocksDB	No	4000
4	NuDB	No	2000
5	NuDB	Yes	2000
6	NuDB	No	4000

TABLE II  
AVERAGE CPU PACKAGE ENERGY (Wh) AND WALL ENERGY (Wh) OVER 30 MINUTES AND ESTIMATED ANNUAL ELECTRICITY COST (\$).

Config.	CPU Package (Wh)	Wall (Wh)	Annual Cost <sup>a</sup> (\$)
1	6.29	24.38	88.84
2	6.24	24.16	88.04
3	6.13	24.08	87.75
4	6.22	24.36	88.78
5	5.80	23.74	86.51
6	5.61	23.46	85.49

<sup>a</sup>Based on average wall energy. Uses electricity rate of \$0.208/kWh.

scheduled to occur. This conclusion is further supported by the energy profiles obtained by IgProf where the presence of functions necessary for the online deletion process, such as those responsible for copying nodes and freshening the caches, are also observed.

Regarding IgProf results, the tool provides both cumulative and self energy reports for each trial run. The cumulative results report the total energy consumed by a specific function and its sub-routines whereas the self report indicates the energy consumed by that function alone. Analysis of the median energy data reveals that the `ripple::doFetch` function, which is responsible for fetching data from the stored database, is one of the most energy consuming sub-processes, consuming around 13.3% of the total energy in the cumulative energy reports. Similarly, the `ripple::verifyDigest` function, responsible for `secp256k1` verification of incoming messages, consumes 13.8% of the total energy of the application according to the cumulative reports. The presence of functions related to the `secp256k1` library within the top energy-consuming cohorts is also demonstrated in the self profile, where functions related to finite field arithmetic necessary for ECDSA operations, such as `secp256k1_fe_mul_inner` and `secp256k1_fe_sqr_inner`, individually comprise 4.58% and 3.48% of the total energy consumption, respectively.

### B. Additional Configurations

The observations made when analyzing the default installation's energy consumption are used as a reference on which to base changes made in the configuration. Particularly, it is studied whether changing the database backend from RocksDB to NuDB, enabling the use of efficiently computable endomorphisms, and changing the online deletion interval influences the total energy consumption of the XRP validator. The average CPU package and wall energy obtained for the additional configurations studied are displayed in

TABLE III  
IMPROVEMENT IN ENERGY FROM CONFIG. A TO CONFIG. B WHEN ONE PARAMETER IS CHANGED AND ALL OTHERS ARE FIXED.

Config. A vs. Config B	Config. A	Config. B	Package	Wall
RocksDB vs. NuDB	1	4	1.0%	0.1%
	2	5	7.1%	1.7%
	3	6	8.5%	2.6%
No Endo. vs. Endo.	1	2	0.8%	0.9%
	4	5	6.8%	2.5%
2000 vs. 4000	1	3	2.5%	1.2%
	4	6	9.9%	3.7%

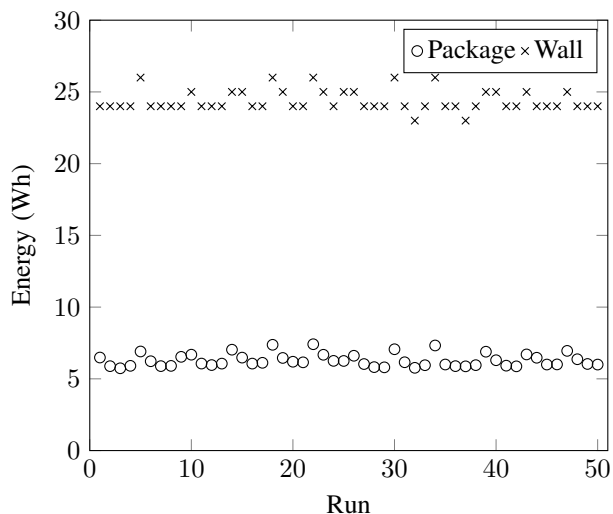


Fig. 1. XRP validator energy under Configuration 1 (online delete 2000). Each run represents the energy accumulated over a 30 minute interval.

the remaining entries of Table II. Again, the average wall energy consumption over the 30 minute interval of tests is used to estimate the annual energy usage from which the annual electricity cost for each configuration can be quantified. Depending on the configuration, the yearly cost for these additional configurations ranges between \$85.49 and \$88.78; this is less than the estimated \$88.84 obtained for the default configuration.

The first section of Table III depicts the percent decrease in energy across the configurations which only differ by the NodeDB setting to study the influence of changing the database from RocksDB to NuDB. The first row compares those configurations where only the NodeDB setting has changed (Configurations 1 and 4), the second compares those configurations which use endomorphisms (Configurations 2 and 5), and the last entry compares the configurations which have doubled the online deletion interval (Configurations 3 and 6). In all cases, the implementations which use NuDB consume less energy than their RocksDB counterparts. A decrease in energy consumption from 1.0-8.5% is observed in the CPU package energy and 0.1-2.6% in the wall energy.

The remaining sections of Table III provide an analysis of the improvement achieved when altering the installation to use either endomorphisms or a larger online deletion interval. It is observed that the endomorphism build can in fact reduce the energy consumed. The lowest energy consumption is achieved when an online delete interval of 4000 is used in both the NuDB and RocksDB cases. As compared to the default configuration provided by rippled, a 2.5% improvement in the average CPU package energy consumption is observed in Configuration 3 over Configuration 1 and similarly, a 9.9% improvement is seen between its NuDB counterpart (Configuration 6 over Configuration 4). Furthermore, the influence of doubling the online deletion interval can be seen in Fig. 2 where the results of Configuration 3 are plotted. A similar pattern to that seen in Fig. 1 is observed; however, the

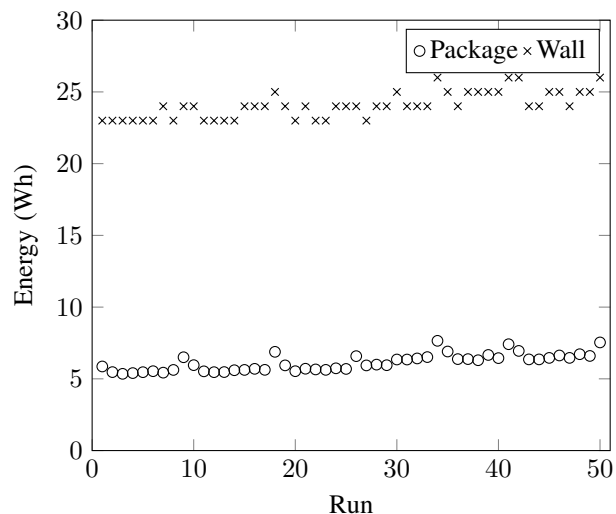


Fig. 2. XRP validator energy under Configuration 3 (online delete 4000). Each run represents the energy accumulated over a 30 minute interval.

frequency at which the energy spikes has noticeably decreased by about half.

## VI. CONCLUSION

In this work, the energy consumption of rippled's default configuration was analyzed. The average energy consumption for each configuration was obtained over 30 minutes which was the basis of calculating the annual energy cost to run a validator. It was found that the default configuration uses an average of 6.29Wh at the CPU package and draws around 24.38Wh of energy from the wall socket. By extrapolating the average wall energy, it is estimated the annual electricity cost for this particular configuration would be roughly \$88.84. Additionally, it is demonstrated that changing the database setting from RocksDB to NuDB, using efficiently computable endomorphisms in the secp256k1 library implementation, and increasing the time frame for online deletion can lead to a decrease in the energy consumed. Although it is shown that the energy consumption can be improved under these different configurations, it should be noted that these results can vary. As the XRP Ledger network is dynamic, individual results may differ as the behaviour of the server changes. Users should always customize their installation based on their desired performance and use case. It is hoped that this work will demonstrate the low electricity costs required to run an XRP validator and encourage greater participation in the XRP Ledger network by the general public and private businesses.

## ACKNOWLEDGMENT

We acknowledge the support of the Natural Sciences and Engineering Research Council of Canada (NSERC) and the Silicon Valley Community Foundation. A modified version of the present paper will appear in the proceedings of the 2020 IEEE International Conference on Blockchain and Cryptocurrency (ICBC 2020).

## REFERENCES

- [1] B. Chase and E. MacBrough, "Analysis of the XRP ledger consensus protocol," *CoRR*, vol. abs/1802.07242, 2018. [Online]. Available: <http://arxiv.org/abs/1802.07242>
- [2] XRP Ledger Project. (2020) Introduction to consensus. [Online]. Available: <https://xrpl.org/intro-to-consensus.html>
- [3] A. de Vries, "Bitcoin's growing energy problem," *Joule*, vol. 2, no. 5, pp. 801–805, 2018.
- [4] XRP Ledger Project. (2020) Technical FAQ. [Online]. Available: <https://xrpl.org/technical-faq.html>
- [5] XRP Ledger Project. (2020) Run a rippled validator. [Online]. Available: <https://xrpl.org/run-a-rippled-validator.html>
- [6] M. Rauchs, A. Blandin, and A. Dek. (2020) Cambridge Bitcoin energy consumption index. [Online]. Available: <https://www.cbeci.org/>
- [7] Digiconomist. (2020) Bitcoin energy consumption index. [Online]. Available: <https://digiconomist.net/bitcoin-energy-consumption>
- [8] Digiconomist. (2020) Ethereum energy consumption index (beta). [Online]. Available: <https://digiconomist.net/ethereum-energy-consumption>
- [9] S. J. Leopold and N. Englesson. (2017, Nov) How eco friendly is our money and is there an alternative? [Online]. Available: <http://papers.netrogenic.com/sid/eco-friendly-money.pdf>
- [10] "rippled v1.4.0," Nov 2019. [Online]. Available: <https://github.com/ripple/rippled>
- [11] XRP Ledger Project. (2020) XRP ledger documentation. [Online]. Available: <https://xrpl.org/docs.html>
- [12] Dev Null Productions. Rippled audit. [Online]. Available: <http://wipple.devnull.network/research/rippled.html>
- [13] XRP Ledger Project. (2020) rippled server modes. [Online]. Available: <https://xrpl.org/rippled-server-modes.html>
- [14] XRP Ledger Project. (2020) Capacity planning. [Online]. Available: <https://xrpl.org/capacity-planning.html>
- [15] Facebook Open Source. (2020) RocksDB a persistent key-value store for fast storage environments. [Online]. Available: <https://rocksdb.org/>
- [16] V. Falco. (2020) NuDB. [Online]. Available: <https://github.com/vinniefalco/NuDB>
- [17] XRP Ledger Project. (2020) Configure online deletion. [Online]. Available: <https://xrpl.org/configure-online-deletion.html>
- [18] Certicom Research. (2010) Standards for efficient cryptography 2 (sec 2). Recommended elliptic curve domain parameters (version 2.0). [Online]. Available: <http://www.secg.org/sec2-v2.pdf>
- [19] P. Wendler, "CPU energy meter v1.2," Aug. 2019. [Online]. Available: <https://github.com/sosy-lab/cpu-energy-meter>
- [20] *Intel 64 and IA-32 Architectures Software Developers Manual*, 3rd ed., Intel, Oct. 2019. [Online]. Available: <https://software.intel.com/en-us/download/intel-64-and-ia-32-architectures-sdm-volume-3b-system-programming-guide-part-2>
- [21] K. N. Khan, F. Nybäck, Z. Ou, J. K. Nurminen, T. Niemi, G. Eulisse, P. Elmer, and D. Abdurachmanov, "Energy profiling using IgProf," in *2015 15th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing*. IEEE, May 2015, pp. 1115–1118.
- [22] "IgProf the ignominious profiler v5.9.16," 2015. [Online]. Available: <https://github.com/igprof/igprof>
- [23] KINCREA. (2020) KINCREA electricity usage monitor plug power meter energy watt voltage amps meter with digital LCD display JKW47. [Online]. Available: <https://www.amazon.ca/KINCREA-Electricity-Monitor-Voltage-Digital/dp/B07Q121ZQY>
- [24] Ontario Energy Board. (2020) Electricity rates. [Online]. Available: <https://www.oeb.ca/rates-and-your-bill/electricity-rates>