The Road to Web 3.0: Empirical Insights into Full Node Workload in the Bitcoin Blockchain

Sayed Erfan Arefin
Department of Computer Science
Texas Tech University, Lubbock, Texas
saarefin@ttu.edu

Abdul Serwadda
Department of Computer Science
Texas Tech University, Lubbock, Texas
abdul.serwadda@ttu.edu

***Abstract*—The shift from Web 2.0 to Web 3.0, known as the"decentralized web", hinges crucially on end-user engagement, particularly in the hosting of blockchain nodes. This study delves into the operational requirements for maintaining Bitcoin *full nodes* by comparing the experiences of two distinct setups: a high-specification desktop and a low-resource Single Board Computer. We meticulously analyze the implications of these setups on end-user resources, focusing on power consumption, bandwidth usage, and data traffic patterns. The findings not only highlight the practical challenges users will encounter in the evolving digital landscape but also provide new insights into the dynamics of the Bitcoin blockchain, particularly during a period of significant market volatility.**

***Index Terms*—Blockchain, Bitcoin, Web 3.0, Network Traffic Analysis**

I. Introduction

Web 3.0, often called the"decentralized web", represents a transformative stage in internet evolution, marking a shift from the centralized databases of major tech firms to a structure underpinned by blockchain technology. This shift aims to restore data ownership and control to individuals, a move that has drawn considerable attention amidst escalating concerns about data privacy and the dominance of tech conglomerates. Despite its promise, the transition to Web 3.0 faces a wide range of challenges. Among the most formidable is achieving widespread adoption, a multifaceted endeavor that encompasses everything from the implementation of supportive government policies and the development of innovative applications by companies, to the active involvement of end-users with a variety of relevant nodes and tools.

Central to this paper is the concept of active user engagement. Envisioning a scenario where various public blockchain-based applications operate on the internet, it's anticipated that end-users will manage different types of nodes, particularly the resource-intensive full nodes. These nodes perform numerous blockchain functions, such as maintaining complete blockchain records and processing various transactions, crucial for blockchain's robustness and security. However, operating such a node is demanding; it requires a computer with substantial specifications (e.g., storage, processing power, RAM), consumes significant bandwidth, leading to high data costs in certain regions, and requires constant power to remain operational.

In this study, we establish two full nodes on the Bitcoin blockchain and analyze the impacts on the end-user's system. One node operates on a high-specification Windows desktop, simulating an end-user capable of dedicating a powerful machine to blockchain operations. The second node runs on a Single Board Computer, akin to the well-known Orange Pi, representing an end-user with minimal computing resources, possibly from less developed regions where high-powered desktops are less accessible. For both nodes, we meticulously detail the power consumption, bandwidth usage, packet-level patterns, and even the geographical origins of the requests encountered by the system. Our study not only highlights the demands these full nodes place on end-users but also provides insights into the recent state of the Bitcoin blockchain, derived from data collected in the latter part of 2023. This period, marked by a significant decline in Bitcoin's value, offers a unique perspective into the blockchain's latest traffic dynamics.

II. Experiment Details

Having installed the Bitcoin Core software on each of our nodes, we proceeded to make observations both during and after the synchronization phase. Details are provided next.

A. Stage One: Network Packet Analysis with Wireshark

1) Phase 1: During Node Synchronization: The initial phase of our study was instrumental in shedding light on the initial synchronization process of the Bitcoin node. To fully understand this, we needed to capture the packets related to the Bitcoin protocol. We used Wireshark [1], a network protocol analysis software, to monitor port 8333, the standard port designated for Bitcoin nodes. We meticulously captured all relevant packets transmitted during this phase, compiling an extensive dataset for further in-depth analysis. Moreover, we precisely documented the start and end times of the Bitcoin node's initial synchronization. These timestamps are vital for our subsequent analysis.

2) Phase 2: After Node Synchronization: Following the initial synchronization, the Bitcoin node transitions to a post-synchronization phase. In this phase, the node primarily engages in verifying newly generated transactions and, in some instances, disseminates information to other nodes as per the Bitcoin protocol's logic. Subsequently, our experiment

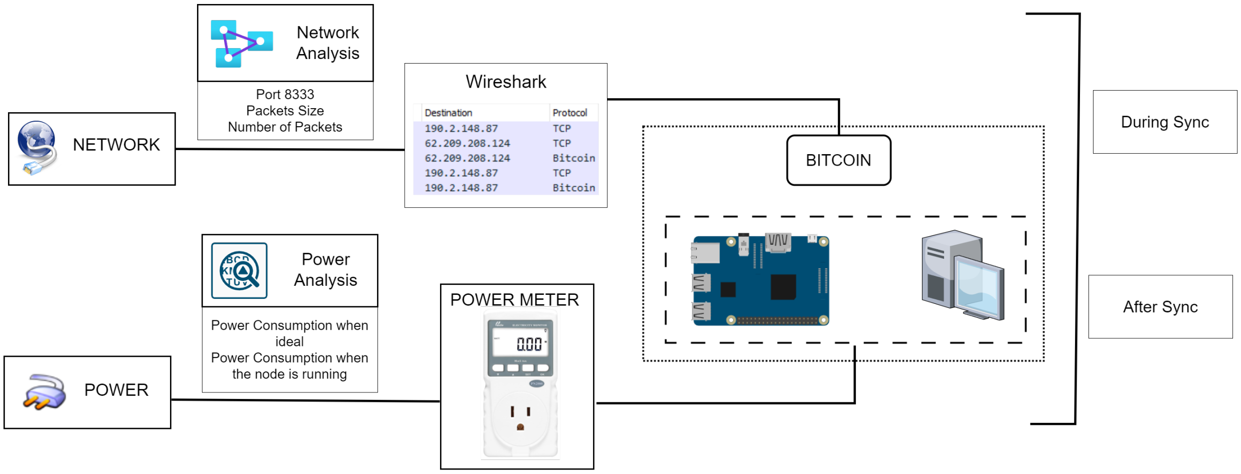


Fig. 1: Full Experiment Setup.

entered its second phase, extending the observation period by additional days in what we designated as the "After Node Synchronization" experiment.

B. Stage Two: Power Consumption Measurement

In Stage Two of our experiment, we focused on assessing the power consumption of Bitcoin nodes. To do this, we used a high-precision power monitor, as shown in Figure 2. This monitor, the Poniie PN2000 Watt Meter, is equipped with a built-in, highly accurate current sensor. The device has resolutions of 0.01W, 0.01V, and 0.001A.

For power data, the study used Tesseract, a package used for OCR operations. This process was essential to facilitate the subsequent analytical evaluation of the power data.

To record power readings, we employed a smartphone camera, capturing images of the meter every 10 minutes. These images were then processed using Optical Character Recognition (OCR) technology to digitize and extract the text data for analysis using the Tesseract library. This library is used for OCR operations. The power consumption experiment was conducted in two phases, mirroring the structure of Stage One: during the synchronization of the Bitcoin node and after the synchronization was complete.

Furthermore, this experiment was carried out on two distinct types of nodes: a fully-fledged desktop computer and a single-board computer. This dual approach was adopted to account for the interest in running Bitcoin nodes on more compact and less power-intensive devices. The desktop computer used in the experiment was equipped with an Intel Core i7 processor, 32GB of RAM, a 1TB HDD, and a 450W power supply. On the other hand, we opted for an alternative single-board computer, the Orange Pi 4 LTS. This device features a Rockchip RK3399 Six-Core ARM 64-bit processor, 3GB LPDDR4 RAM, and 16GB EMMC storage [2]. For our experiment, the Orange Pi 4 LTS was outfitted with Armbian, a Debian-based operating system [3]. Meanwhile, the desktop computer ran Microsoft Windows 11 operating system.

In both setups, we utilized the executable packages for Bitcoin nodes available on the official Bitcoin Core website, aimed at consumer usage [4]. This allowed us to conduct our experiments under conditions that closely mimic real-world user environments. Similarly to the first stage, this stage also employed two phases, "During Node Synchronization" and "After Node Synchronization".

Fig. 2: Power Meter PN2000 by Poniie

III. Data processing

In this study, multiple data processing methodologies were employed, specifically focusing on two key areas: network data processing using Wireshark and power data collection through Optical Character Recognition (OCR).

The network data was captured methodically using Wireshark software, which monitored port 8333, the designated port for the Bitcoin protocol. The software was configured to export captured packets in pcap format at ten-minute intervals. This interval was strategically chosen to mitigate processing difficulties associated with overly large pcap files. Later, these files were converted to tabular data using the Tshark utility [5] and the Pandas library. Additionally, the study involved the identification of packet direction (incoming or outgoing) based on the node's IP address, compared against source and destination IP addresses. Using the API provided by ipinfo.io [6], location data was acquired for each IP address. The API responses included approximate geographical coordinates

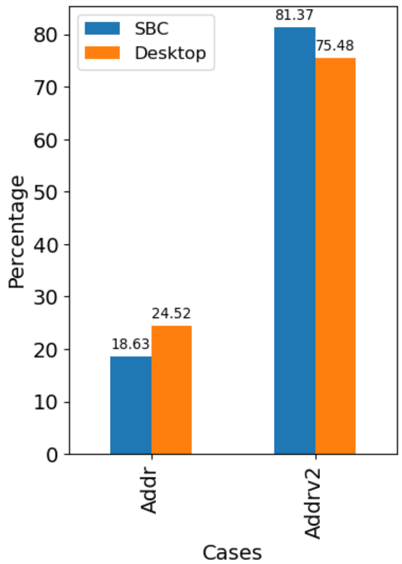


Fig. 3: Percentage of Addr and Addr v2 used in the whole experiment by Desktop Computer and Single Board Computer (SBC).

(latitude and longitude), city, state, and country, which were subsequently integrated into the CSV file to enrich the analysis.

IV. Experimental Results

In this section, we discuss several aspects of our experimental results. Initially, we examine the bandwidth consumption and packet statistics. Subsequently, we extract various information from the packets collected throughout our experimental procedures. Finally, we assess the power consumption of the nodes for different operations.

A. Bandwidth Consumption and Packet Statistics

In this section of the study, we outline the methodology employed for data collection. The focus was on capturing data specific to the Bitcoin protocol via port 8333, which is designated for this purpose. Our primary objective was to analyze the data volume involved in the blockchain's synchronization process (initial download) and the data exchange that occurs when the system continues to run postinitial sync. During our experimentation, the Bitcoin blockchain was approximately 450 GB in size. However, our monitoring was limited to packets transmitted through the designated port, revealing that the actual size of the blockchain differs from the data transferred over the network. The Bitcoin full node interacts with its peers to receive necessary commands, facilitating the construction of Merkle trees essential for accurately reconstructing the blockchain. We plotted the volume of packets accumulated during the hours of the initial sync process in Figure 4a for both the desktop computer and the SBC. After the sync process, the node transmits data to its peers and receives new updates from its peers. During this period of the experiment, we observe data consumption, which is plotted in Figure 4b. For the initial data sync process, note that the major data transfer occurs within the first 10 hours. The total bandwidth consumed by the Desktop computer node during the synchronization process is 82.25 Giga Bytes and in the case of SBC node, it is 78.88 Giga Bytes. After the synchronization process is complete we observed that the Desktop node and SBC node 47.52 Giga Bytes and 44.32 Giga Bytes consequently.

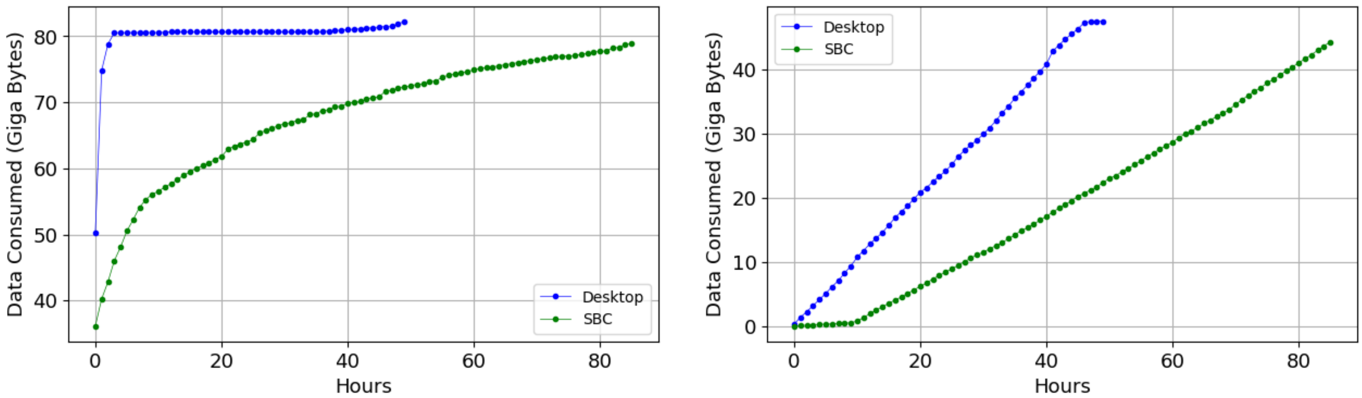
Figure 5 illustrates the average packet size for each Bitcoin command identified during our experiments, where we encountered 19 distinct Bitcoin commands.

B. Geographical Distribution of Nodes Interacting with our Node

In the Bitcoin protocol, ”ping” and ”pong” messages are crucial for sustaining active connections and ascertaining the responsiveness of peer nodes. The process involves a ”ping” message, which includes a nonce, eliciting a ”pong” response from its recipient. This interaction is pivotal in verifying the active and responsive status of network participants, thereby ensuring the network's integrity.

Throughout our investigation, we utilized Wireshark to capture all packet exchanges both during and subsequent to the initial blockchain synchronization. Our analysis of these packets revealed the occurrences of ”ping” and ”pong” commands within the Bitcoin network. These commands are instrumental for nodes to periodically affirm the peer-to-peer network's operational health. Notably, our node received multiple ”ping” messages, identifiable through the scrutiny of source and destination IP addresses within the captured packets. These inbound ”ping” messages, originating from other nodes, serve a mutual objective of sustaining network health. Observations confirmed that our node dispatched ”pong” messages in response, along with issuing its own ”ping” messages and acknowledging received ”pong” responses. However, a subset of nodes did not reciprocate with ”pong” messages to our node's ”ping” inquiries, prompting us to categorize these failed interactions. Initial categorization of unreciprocated ”ping” messages highlighted that, during the synchronization process, 326 nodes and post-synchronization, 1669 nodes (for the single board computer scenario) failed to respond. In contrast, the desktop node experienced failures from 304 nodes during synchronization and 94 nodes thereafter, with other failure rates negligible (below 10 nodes), which can be observed in Table II.

Leveraging the ipinfo API [6], we decoded IP addresses to approximate the geographical locations of nodes that engaged our node with ”ping” messages. These locations, expressed in latitude and longitude, facilitated the creation of a heat map using the Folium library. Figure 6 illustrates the nodes discovered via ”ping-pong” exchanges throughout and after the initial download phase. The resultant heat maps indicated an increase in node activity post-synchronization. For clarity, heat maps derived solely from inbound ”ping” messages to the desktop node were presented.



(a) During Initial Sync process.

(b) After Initial Sync process.

Fig. 4: Approximate Accumulated Data Consumption by the Devices During and After the Initial Sync Process.

TABLE I: Comparative analysis of the collected data. It is represented as percentages by the number of bitcoin commands for Desktop and SBC devices during and after initial synchronization

Packet Name	Bitcoin Command	Category	Desktop Computer				Single Board Computer			
			After Sync (%)		During Sync (%)		After Sync (%)		During Sync (%)	
			Incoming	**Outgoing**	**Incoming**	**Outgoing**	**Incoming**	**Outgoing**	**Incoming**	**Outgoing**
Ping Message	ping	Networking	0.55	0.94	0.48	0.79	1.51	1.63	0.71	1.55
Pong Message	pong		0.54	0.93	0.48	0.78	1.54	1.62	0.29	1.53
Get Address	getaddr		0	0	0	0	0	0.02	0	0.17
Send Address V2	sendaddrv2		0.01	0.02	0.04	0.03	0.03	0	0.24	0
Address	addr		0.05	0.12	0.02	0.09	0.04	0.17	0.06	0.13
Address V2	addrv2		0.25	0.64	0.32	0.61	1	1.33	0.59	1.67
Version	version		0.02	0.03	0.08	0.11	0.05	0.07	0.47	0.69
Version Ack	verack		0.03	0.03	0.08	0.04	0.06	0.05	0.51	0.2
Inventory	inv	Transaction	6.98	27.29	5.04	20.16	22.96	44.18	0.64	0
Get Data	getdata		7.57	10.24	8.54	15.91	5.08	17.84	0	57.98
Transaction	tx		18.01	25.22	14.93	30.87	0	0	0	0
Block	block	Block Operations	0	0	0	0	0.01	0	30.81	0
Headers	headers		0.05	0.13	0.08	0.11	0.17	0.15	0.19	0.09
Send Headers	sendheaders		0.02	0.02	0.02	0.03	0.04	0.03	0.12	0.08
Get Headers	getheaders		0.02	0.03	0.02	0.05	0.03	0.04	0.14	0.07
Compact Block	cmpctblock		0.03	0.01	0.02	0.01	0	0	0	0
Send Compact	sendcmpct		0.03	0.05	0.04	0.06	0.06	0.02	0.25	0.1
Transaction Relay	txidrelay	Management	0.01	0.02	0.03	0.03	0.03	0.04	0.24	0.17
Fee Filter	feefilter		0.06	0.05	0.07	0.04	0.17	0.04	0.21	0.09

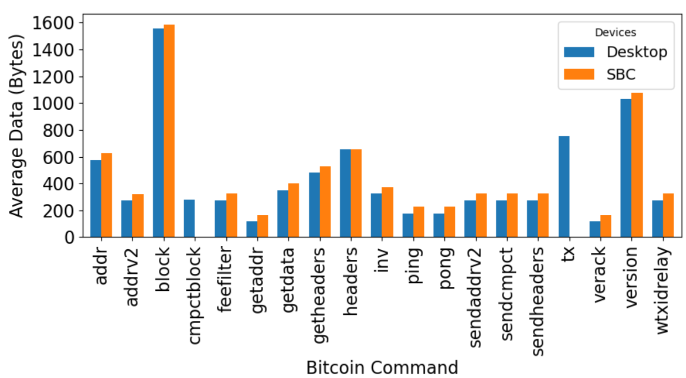


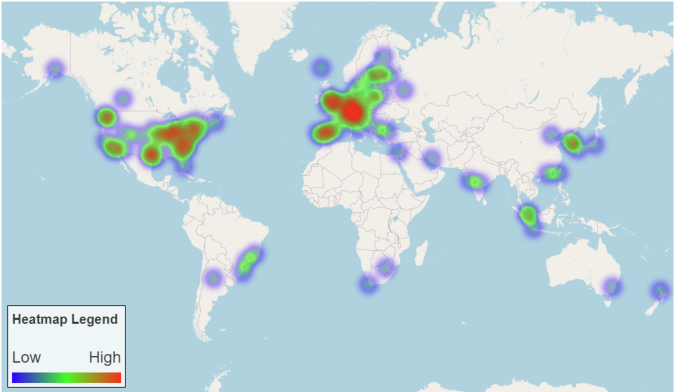
Fig. 5: Comparative Sizes of Packets Received by Desktop and SBC Devices During the Full Experiment

C. Observation for Bitcoin Commands

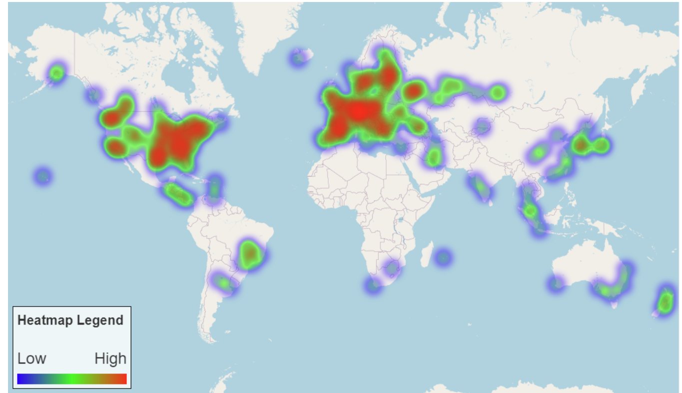
In this part of the study, we observed all other bitcoin commands and found that Bitcoin Core 0.21.0 introduced a new version for the "Addr" bitcoin command. This was added to the Bitcoin p2p network protocol in 2021. The newly added type of "Addr" can support larger address types like the 256-bit v3 Tor hidden service addresses [7]. We wanted to see how many nodes are now using Addr v2 and how many are still using the old address version. We observed that during the SBC experiment we found that 81.37% nodes are using Addr v2. On the other hand, during the experiment on Desktop Node, we observed that 75.48% nodes are using Addr v2. This

TABLE II: Table of Binned Failure Rates for Ping-Pong Messages, Indicating Correspondences Between Incoming Pings and Outgoing Pongs, and Vice Versa

Failure(%)	SBC Desktop		Desktop Node	
	During Sync	After Sync	During Sync	After Sync
0-20	304	94	326	1669
20-40	3	1	0	0
40-60	3	6	0	1
60-80	1	0	0	0
80-100	0	0	0	0



(a) Desktop Node During Sync process.



(b) Desktop Node After Sync process.

Fig. 6: Heatmap Visualization on Maps, Showcasing Geolocated Incoming Ping Messages Captured by a Bitcoin Node, with IP Addresses Reverse Geo-coded to Latitudes and Longitudes on the Desktop Node.

can be observed in Figure 3.

The percentage of packets of bitcoin commands among all captured packets is categorized and organized in Table I, based on the node type and also the two phases of the experiments described in Section II.

We observed an interesting behavior when comparing the different types of nodes. The Desktop computer exclusively received "TX" Bitcoin commands, which are utilized to convey bitcoin transactions within transaction packets. In contrast, the SBC did not encounter any packets of this nature. A similar pattern was observed for the "cmpctblock" command, which is integral to the Bitcoin protocol for transmitting "compact block" messages. These messages aim to minimize bandwidth usage by distributing a streamlined block version, enabling nodes with pre-existing transaction data in their "mempool" to efficiently reconstruct the complete block. This approach significantly improves the efficiency of data transmission across the network.

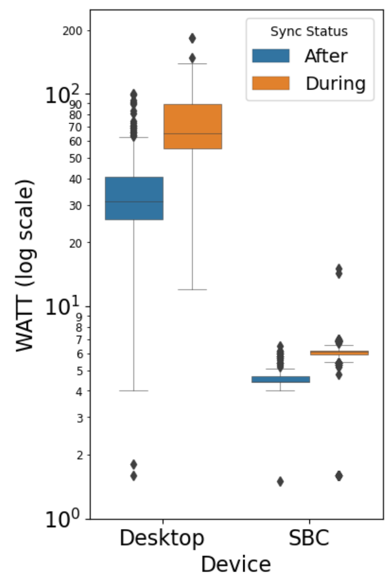


Fig. 7: Box plot for Desktop and SBC Power Usage in Watts

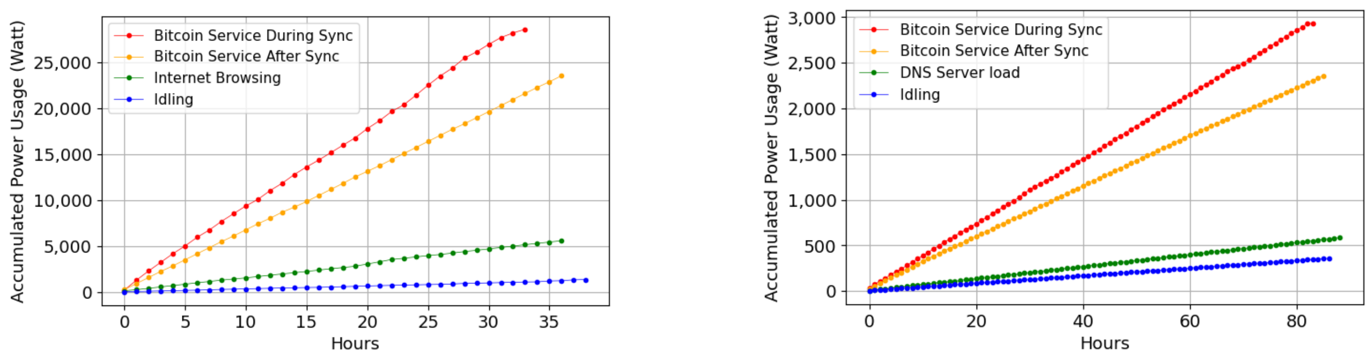
D. Power Consumption Analysis

In this segment of our research, we meticulously measured the power usage of both a Single Board Computer (SBC) and a Desktop Node, focusing on their energy consumption patterns during and after the initial blockchain synchronization process. The initial findings on power consumption are visualized in Figure 7 through a box plot, which illustrates the distribution of power usage between the SBC and the Desktop Computer. Given the significantly lower power consumption of the SBC compared to the Desktop Computer, we employed a logarithmic scale for the y-axis, which denotes power in watts, to facilitate a direct comparison within a single subplot.

Our objective was to contextualize the power usage of these nodes by comparing it with their consumption during typical activities. For instance, on the Desktop Node, we evaluated energy consumption while engaging in routine tasks such as streaming videos on YouTube, alongside periods where the computer remained idle without running any specific services. These comparative analyzes of power consumption, illustrated in Figure 8a, were depicted over time in an accumulated fashion, quantified in hours.

For the Single Board Computer, acknowledging that activities like streaming YouTube might not be as prevalent, we opted for a scenario more befitting its usage profile. We deployed a custom DNS server on the SBC, a task more suited to its capabilities and common use case, and compared the power usage in this scenario with others. The results of this comparison are detailed in Figure 8b.

It is also crucial to highlight the disparity in synchronization times between the two devices. The SBC required approximately 84 hours to complete the initial sync process, whereas the Desktop Node completed the same task in about half that time, around 49 hours. This discrepancy served as a baseline for evaluating power consumption in different scenarios for both platforms.



(a) Desktop Power Consumption Comparison.

(b) SBC Power Consumption Comparison.

Fig. 8: Approximate Accumulated Power Consumption for Different Scenarios.

Our findings revealed that the power consumption for both devices was higher during the sync process compared to after the sync process had been completed. We observed that to complete the entire initial synchronization process, the Desktop computer node consumed 28,543.79 watts and the SBC node consumed 2,933.21 watts. On the other hand, after the synchronization process is complete, the Desktop and SBC nodes consumed 23,497.70 watts and 2,358.42 watts consequently. On the other hand, running YouTube videos for that amount of time on a Desktop computer will consume 5,588.03 watts. In contrast, running a DNS server on the SBC node consumed 582.14 watts. This analysis offers a detailed view of operating a Bitcoin node, emphasizing the significant impact of electricity costs on power consumption.

V. Related works

The research conducted by Parlikar et al. [8], studies the feasibility of energy-efficient blockchains by using the proof-of-stake (PoS) networks instead of the proof-of-work (PoW) chains like Bitcoin's. They discuss the negative effects of running blockchain on proof-of-work, as it requires an enormous amount of computation to verify a work. Introduction to proof-of-stake can reduce the use of large computational requirements. Another study conducted by Tomatsu et al. [9], focuses on the energy consumption of Bitcoin mining, including its mechanisms, energy consumption, mining sites, and the potential for renewable energy use. The research discusses using renewable energy for Bitcoin mining. The research conducted by Taherdoost et al. [10], examines the adoption of blockchain technology in healthcare. It presents an objective evaluation of blockchain's advancement in healthcare and suggests that blockchain technology can significantly increase efficiency and cost-effectiveness in this sector by highlighting the decentralization, immutability, and transparency of the technology.

How we differ from these works: Our work focused on the study of the Bitcoin core module, which is a fundamental part of the Bitcoin Network. This differs from studies conducted on the mining nodes. To our knowledge, we did not find any study that focuses on the resource usage of running such a core module.

VI. Conclusions

In our study with the broader context of Web 3.0's evolution, we've navigated the operational intricacies of Bitcoin Core full nodes across diverse hardware. We ran experiments with the Bitcoin core full node from a different angle where we studied the Bitcoin packets, in different types of devices such as a full desktop computer and a Single Board Computer(orange pi). We also study the power consumed by the two nodes in different situations (During and After the initial synchronization process). We also look into the geographical positions of the nodes available in the network in the p2p bitcoin network by reverse engineering the IP addresses. By delving into packet patterns, global distribution of nodes and power consumption. This research provides a unique perspective on the real-world challenges of blockchain.

References

[1] Wireshark, "Wireshark tool," https://www.wireshark.org/download.html, (accessed November 30, 2023).

[2] Orange Pi, "Orange Pi 4 Product Page," https://orangepi.com/index.php?route=product/product&product_id=895, accessed: [Insert Date Here].

[3] "Armbian Documentation," https://docs.armbian.com/, accessed: December 10, 2023.

[4] Bitcoin, "Bitcoin core," https://bitcoin.org/en/bitcoin-core/, (accessed January 30, 2024).

[5] "Tshark Documentation," https://wireshark.org/docs/man-pages/tshark.html, accessed: December 10, 2023.

[6] "Ipinfo.io developer Documentation," https://ipinfo.io/developers, accessed: December 10, 2023.

[7] B. Optech, "Addr v2," https://bitcoinops.org/en/topics/addr-v2/, (accessed January 30, 2024).

[8] M. Parlikar, "More energy-efficient blockchains are possible. here's how," *CoinDesk*, 2021, accessed: 2024-04-01.

[9] Y. Tomatsu and W. Han, "Bitcoin and renewable energy mining: A survey," *Blockchains*, vol. 1, no. 2, pp. 90–110, 2023, accessed: 2024-04-01.

[10] H. Taherdoost, "Blockchain and healthcare: A critical analysis of progress and challenges in the last five years," *Blockchains*, vol. 1, no. 2, pp. 73–89, 2023, accessed: 2024-04-01.