Blockchain Energy Consumption: Unveiling the Impact of Network Topologies

Vincenzo P. Di Perna [*], Valerio Schiavoni [†], Francesco Fabris [‡], Marco Bernardo [§]
[*]University of Urbino, Urbino, Italy, vincenzo.diperna@unicam.it
[†]University of Neuchâtel, Neuchâtel, Switzerland, valerio.schiavoni@unine.ch
[‡]University of Trieste, Trieste, Italy, ffabris@units.it
[§]University of Urbino, Urbino, Italy, marco.bernardo@uniurb.it

***Abstract*—Blockchain is one of the most promising emerging technologies, but its large-scale deployment raises sustainability issues, specifically in terms of energy consumption. In addition to consensus algorithms, energy efficiency is influenced by several other parameters, including network topology. We present a systematic analysis of the impact of five distinct network topologies (fat-tree, full mesh, hypercube, scale-free, and torus) on the energy consumption of five blockchain platforms (Algorand, Diem, Ethereum Clique, Quorum IBFT, and Solana) under three different workloads (PayPal and VISA transaction processing and GAFAM smart contract execution). Our results reveal that fat-tree and full mesh are the most energy-efficient network topologies, in particular under heavy workloads. Moreover, they show that Algorand and Diem exhibit the lowest energy consumption per transaction, while Ethereum Clique features the highest energy consumption – regardless of the used topology – and Quorum IBFT and Solana lead to higher energy costs under intensive workloads as the network size increases.**

***Index Terms*—blockchain, energy consumption, network topology, emulation, sustainability.**

I. Introduction

Blockchains are distributed ledgers that can be public (permissionless) or private (permissioned). Their primary goal is to facilitate the recording of transactions in an immutable and transparent manner. While public blockchains aim for decentralization and the elimination of a central authority, private blockchains may retain some level of central control for specific use cases [1]. Achieving these properties necessitates a self-governing and computationally intensive validation process executed by participating nodes, called the consensus protocol [1]. Two notable consensus protocols for public blockchains include Proof of Work (PoW) [2] and Proof of Stake (PoS) [3], which are designed to synchronize node operations and resolve conflicts within the blockchain network.

Validator nodes rely on high-performance hardware such as GPUs, FPGAs, and ASICs to solve mathematical proofs for block validation and rewards. Despite their efficiency, these devices incur significant economic and environmental costs. Consequently, the rapid adoption of blockchain has driven energy demands, with Bitcoin mining in 2021 consuming nearly six times the amount of energy it consumed in 2017 and matching the annual energy consumption of countries like Finland and Argentina [4]. This has sparked global concern and efforts to mitigate blockchain environmental impact. Key contributors to energy inefficiency include the consensus protocol in use and the hardware employed by participating nodes. While research has focused on optimizing these factors – such as transitioning from PoW to PoS [5] or developing energy-efficient hardware like ASICs – other approaches have considered the adoption of renewable energy sources to mitigate carbon emissions and methods for data optimization (*e.g.*, data sharding) [6]. Additionally, a range of tools is available for analyzing blockchain energy consumption, using both predictive evaluations (*e.g.*, via testbed and benchmarking frameworks) and post-deployment analysis (*e.g.*, by means of visualization tools and measures [7]).

Despite such significant efforts and proposals to address excessive energy consumption, the role of network topology – affecting workload distribution, communication latency, and overall blockchain efficiency [8] – remains largely overlooked. This limitation hinders the development of holistic strategies to enhance blockchain sustainability. To address this gap, we provide a systematic and comprehensive analysis of how network topologies impact blockchain energy consumption under varying workloads, including both transaction processing and smart contract execution. Specifically, we evaluate five distinct network topologies – fat-tree, full mesh, hypercube, scale-free, and torus – which represent different real-world blockchain network configurations, ranging from public Internet networks to private data center infrastructures [9]. We investigate their impact on the energy consumption of five public/private blockchains: Algorand [10], Diem [11], Ethereum Clique [12], Quorum IBFT [13], and Solana [14]. We show how the choice of the network topology plays a critical role in determining the energy consumption of blockchain protocols. Specifically:

- Fat-tree and full mesh are generally the most energy-efficient topologies across all blockchains, particularly in handling intensive workloads.
- Hypercube performs well for transaction processing workloads, especially for Algorand and Diem, while scale-free and torus topologies show inefficiencies with certain types of transactions.
- Torus underperforms in energy efficiency, particularly in Ethereum and Solana, due to conflicts with increasing network size.
- Algorand and Diem benefit significantly from topologies like full mesh and hypercube, maintaining a low energy consumption per transaction.

- Ethereum Clique shows the highest energy consumption per transaction, regardless of the underlying topology.
- Quorum IBFT experiences increased energy consumption with more demanding workloads, especially under fat-tree, hypercube, and torus topologies.
- Solana demonstrates high energy demands and operational failures in larger node setups.

For our experimental framework, we enhanced *Diablo* [8], a benchmark suite for blockchains, by incorporating *Kollaps* [15], a state-of-the-art network topology emulator, along with *Intel Running Average Power Limit* (RAPL) energy indicators to measure energy consumption. This integration enabled the generation of custom network topologies, to emulate realistic communication patterns and measure energy consumption under varying conditions. To align with industry benchmarks, our workloads are derived from prior studies [8]. We modeled PayPal and VISA workloads based on their claimed transaction capacities, while the GAFAM workload simulated burst requests for Google, Apple, Facebook, Amazon, and Microsoft stock trades using smart contracts.

Roadmap. §II covers background on blockchains and energy awareness. §III presents related work. §IV explains our evaluation methodology. §V describes our results. Finally, §VI concludes the paper by discussing insights and limitations.

II. Background

A blockchain is a particular distributed digital ledger that records transactions across multiple nodes in a secure, transparent, and tamper-proof manner. Each "block" contains a set of transactions, cryptographically linked to the preceding block, thus forming a "chain". Smart contracts are specialized self-executing blockchain programs, with predefined conditions, written in high-level programming languages [16]. These contracts automate processes without intermediaries, guaranteeing trust and efficiency.

At the core of blockchain operations lies the consensus protocol, which ensures agreement among participants on the state of the ledger. Various consensus mechanisms exist, *e.g.*, PoW [2], used by Bitcoin, which relies on computationally intensive tasks, and PoS [3], which selects validators based on the amount of cryptocurrency they hold. Alternative protocols, like Delegated Proof of Stake (DPoS) [17] and Practical Byzantine Fault Tolerance (PBFT) [18], aim to improve efficiency and scalability.

Blockchains fall into three categories: public, private, and consortium [19]. Public blockchains (*e.g.*, Bitcoin), are open to anyone and without a central authority. Private blockchains restrict participation to authorized users and retain some level of central control. Consortium blockchains are jointly managed by multiple organizations. To study the impact of network topology, we examine five blockchains: three public ones (Algorand [10], Ethereum [12], Solana [14]), a private one (Diem [11]), and a consortium one (Quorum [13]).

Energy Consumption in Blockchains. Blockchain networks, especially those using PoW, face criticism for high energy demands driven by competitive mining and resource-intensive cryptographic puzzles. Key factors influencing energy consumption include hash functions, which are computationally intensive tasks [4]. Despite the millions of participants racing to solve the puzzle, only one is ultimately successful, leaving the others' computational efforts effectively wasted [4]. While validating a transaction in any blockchain network involves two main energy components – local computation by a node and communication energy for packet transmission between nodes – the computational demands of PoW are so high that the energy costs of communication become negligible [20].

While Bitcoin's PoW consumes between 200 and 950 kWh per transaction, Ethereum, before transitioning to PoS, required approximately 75 kWh per transaction [20], highlighting the need for more energy-efficient consensus mechanisms. One of the most impactful solutions to address PoW inefficiencies is the adoption of PoS consensus, which can reduce energy consumption by up to 99.5% [4].

III. Related Work

The literature reveals a strong emphasis on solutions and methods for studying and analyzing energy consumption in blockchains [7], [20]–[22]. These approaches primarily focus on post-deployment analysis. For example, the *Cambridge Blockchain Network Sustainability Index* (CBNSI) [7] is a recent a posteriori tool offering insights into Bitcoin and Ethereum's energy consumption.

Conversely, predictive methods, which rely on a testing environment, have received significantly less attention. Although several blockchain benchmarking tools have been developed [8], [23], they lack an efficient way of measuring energy consumption. Instead, it is typically measured manually or through external methods such as estimations. For instance, the authors of [24] leverage a highly effective benchmarking tool (BCTMark [23]) that measures energy consumption by relying on both external instrumentation and estimates based on Ethereum gas consumption.

In addition, a significant limitation in existing research lies in the evaluation of network variations, encompassing not only different topologies but also diverse network configurations that can characterize a distributed network of nodes such as a blockchain. Most of these predictive studies on energy consumption in blockchains focus on IoT and SDN scenarios. The authors of [21] propose an FPGA-based testbed for estimating Bitcoin's energy consumption. The research in [22] highlights the critical role of the underlying network in determining the energy efficiency of a blockchain, emphasizing how much it depends on broadcast protocols and network size – an issue even more pronounced in IoT contexts. The study highlights that blockchain peers often use a random neighbor selection mechanism to decide which peers to exchange data with, which can lead to suboptimal communication links.

DistBlockNet [25] and Blockchain Security over SDN (BSS) [26] lack an evaluation of energy consumption. This gap could potentially introduce security challenges within the architecture. The authors of [27] perform an energy comparison between different routing protocols. However, the consensus protocol was offloaded from the IoT devices, leading to a significant reduction in actual energy consumption.

In [20], the energy consumption of blockchain systems is analyzed through a model that also accounts for network-related aspects, *i.e.*, the number of messages exchanged per transaction. However, it was not possible to test networks of varying sizes due to resource constraints. This practical approach involved the use of a dedicated testbed and monitoring devices to measure energy consumption during the experiments. The study found that in Ripple [28] and Stellar [29] the majority of energy costs are due to packet transmission rather than the consensus mechanism itself. In contrast, for PoW-based systems, the consensus mechanism was identified as the primary source of energy consumption.

In [24] the focus is on the evaluation of applications implemented via smart contracts. It highlights that the highest energy consumption stems from call replications across the entire network. This underscores the importance of accounting for network variations in this context. However, despite leveraging tools like EnosLib [30] using Linux TC [31] for network modeling, the framework is unable to fully emulate or replicate a complete network topology.

By integrating Kollaps with Diablo, we are able to emulate comprehensive and configurable topologies with respect to various properties (*e.g.*, latency, bandwidth, packet drop, jitter, *etc.*) with greater accuracy than other similar tools, as demonstrated in [15]. Furthermore, we utilize Intel RAPL energy counters to obtain measurements without the need for external or manual instrumentation.

IV. Evaluation Methodology

This section describes the experimental evaluation with details on the emulated environment as well as experiment configurations, blockchains under test, used workloads, and considered topologies.

A. Experimental Settings

Our cluster consists of 7 Dell PowerEdge R630 servers, each equipped with two 16-core Intel Xeon E5-2683v4 CPUs (2.10 GHz) and 128 GB RAM, interconnected via a Dell S6010-ON 40 GbE switch. All nodes operate on Ubuntu Linux 22.04 LTS with kernel version 5.15.0-107-generic.

B. Framework

We conduct our experiments by using a blockchain benchmark suite, Diablo [8], and a network topology emulator, Kollaps [15]. These tools were selected as leading state-of-the-art solutions in their fields.

In particular, Diablo enables distributed experiments on platforms like the cloud through an emulated environment, thus ensuring reproducibility, versatility, observability, and portability, along with user-friendly features. It can manage a distributed workload generation mechanism with nodes interacting with the specific blockchain via a dedicated client interface, guaranteeing synchronized evaluations. Additionally, Diablo automates the infrastructure setup process and can inject realistic workloads (*e.g.*, smart contracts and transfer

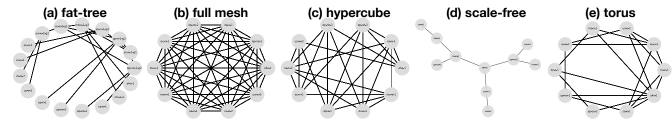

Fig. 1: Real-world network topologies

transactions) with varying volumes and complexities, natively supporting *Amazon Web Services* (AWS) deployments.

Kollaps, on the other hand, is a cutting-edge technology for emulating complex networks [15]. It can simulate large-scale distributed applications by modeling end-to-end properties such as latency, bandwidth, and packet loss. Kollaps supports native processes and virtual machines while integrating with container orchestrators like Docker Swarm and Kubernetes.

To collect energy data, we relied on Intel RAPL indicators [32]. This hardware feature allows us to monitor energy consumption across various domains of the CPU package and its components, as well as the DRAM memory managed by the CPU. These are incremental energy counters that provide measurements in microjoules. In our cluster setup, the machines are equipped with two sockets (each with 16 cores/32 threads), so the recorded values represent the cumulative energy consumption across both sockets. Specifically, we queried the available energy counters by using the `sysfs/powercap` interface. To convert the energy consumption from microjoules to kilowatt-hours (kWh), we first divide the values by 1,000,000 to obtain joules and then use the conversion factor $1\text{J} = 2.7778 \times 10^{-7}\text{kWh}$.

C. Topologies

Our experiments rely on five distinct network topologies (see Fig. 1), designed to simulate 10 geographically distributed regions worldwide: Cape Town, Tokyo, Mumbai, Sydney, Stockholm, Milan, Bahrain, Sao Paulo, Ohio, Oregon.

The **fat-tree** topology, widely employed in data centers, consists of multiple hierarchical layers of interconnected gateways. For our emulation, 20% of the gateways (2 per region) are assigned to the first layer, while 50% (5 regions) are assigned to the second layer. Gateway selection for these layers is based on the lowest latency values from the dataset; connections between layers are randomized to reduce dynamic routing and optimize data transmission. Blockchain nodes are connected to second-layer gateways.

The **full mesh** topology ensures that all nodes are directly connected to one another, mirroring the high connectivity typical of cloud-based environments.

In the **hypercube** topology, nodes connect in a binary pattern, forming a multidimensional structure often found in parallel computing and IoT systems. For our use case, we employ a 4-dimensional hypercube, which can accommodate up to $2^4 = 16$ nodes, sufficient for our 10-region setup.

The **scale-free** topology features nodes with highly uneven connectivity, where certain nodes maintain significantly more links than others [33]. This structure is generated by using the preferential attachment algorithm [34], which prioritizes connecting new nodes to those with high connectivity. This design is reminiscent of large-scale networks (WAN), such as the Internet [33].

TABLE I: Blockchains main characteristics

	Type	Consensus	Virtual Machine	DApp Language
Algorand	Public	BA [10]	AVM	PyTeal
Diem	Private	HotStuff [36]	MoveVM	Move
Ethereum	Public	Clique [37]	geth	Solidity
Quorum	Consortium	IBFT [38]	geth	Solidity
Solana	Public	TowerBFT [39]	Sealevel	Solang

Finally, the **torus** topology is a grid-like structure where nodes connect to adjacent ones in a wrap-around manner, thus enabling efficient data transfer. As is common in data centers and supercomputing environments [35], we implement a 2D torus with 2 rows and 5 columns to model our 10 regions.

D. Blockchains under Test

Below is an overview of the blockchains analyzed in our study, with their core characteristics summarized in Table I.

These blockchains represent diverse system architectures, varying in their network types (*e.g.*, public, private, consortium), smart contract programming languages (*e.g.*, Solidity, PyTeal, Move), execution environments (*e.g.*, AVM, MoveVM, Geth, Sealevel), and consensus protocols (*e.g.*, BA, HotStuff, Clique, IBFT, TowerBFT). To ensure consistency, we adopted the blockchain configurations used in [8]. Specifically, for each blockchain, we utilized a specific version indicated by a repository commit.

Algorand [10] operates using a pure PoS (Proof of Stake) consensus mechanism, where blocks are proposed by a randomly selected subset of nodes. During testing, we optimized performance by polling the blockchain only after new blocks were added, which enhanced transaction commit detection. Experiments were conducted by using Algorand at commit `116c06e` [40].

Diem [11] employs an adaptation of the HotStuff consensus protocol [36], achieving deterministic finality with reduced communication overhead. Although Diem is no longer actively developed, we used its testnet branch at commit `4b3bd1e` [41]. Despite its discontinuation, Diem remains a valuable case study for understanding permissioned cryptocurrency systems and the broader design considerations for future digital currencies.

Ethereum [12] serves as a public blockchain for decentralized applications (DApps) and smart contracts. Our experiments utilized the Clique PoA (Proof of Authority) protocol [37], chosen for its ability to avoid inherent throughput bottlenecks. Blocks are validated and added sequentially by pre-defined validators in a round-robin manner, with a minimum block interval of 1 second in our setup. To account for Ethereum's London upgrade (August 2021), which introduced dynamic gas fees, our benchmarks required real-time fee adjustments and transaction signing. Testing was performed using the Go implementation of Ethereum (Geth), commit `72c2c0a` [42].

Quorum [13], [43], an enterprise-focused version of Ethereum, is designed for permissioned networks. Following the configurations outlined in [8], we deployed Quorum with the IBFT consensus protocol [38], commit `919800f` [44]. This setup addresses common challenges in PoA systems, such as message delays and resilience to arbitrary faults [37].

TABLE II: Workloads description

Workload	Type	Duration (s)	Scenario	TPS
PayPal	Transfer Tx	300	Constant rate	200
VISA	Transfer Tx	300	Constant rate	1,800
GAFAM	Smart contract	180	Burst	20,000 down to 100

Solana [14] leverages the TowerBFT consensus protocol [39], which combines features from both BFT and PoS to enhance scalability and throughput. A key innovation in Solana is its use of Proof of History (PoH) [45] for timestamping, which allows for fast and efficient transaction ordering. Unlike Ethereum, Solana does not use Merkle Patricia Trie for its data structure, instead it opts for an alternative approach designed for high throughput and scalability. Transactions are finalized after 30 confirmations and blocks are added every 400 milliseconds. In our experiments, the API's ability to set commitment levels and monitor blocks was utilized. Given the short timeframes typical for realistic DApps (around 120 seconds), we periodically fetched the latest block hash during evaluations. Testing was conducted by using commit `0d36961` [46].

E. Workloads

We consider three workloads modeling transfer transactions and smart contracts (see Table II), as detailed next.

PayPal payment system workload averages 193 transactions per second (TPS) [47]. For simplicity, we modeled it as a constant workload of 200 TPS over 5 minutes in our tests.

VISA payment system workload averages 1,700 TPS [47]. For simplicity, we modeled it as a constant workload of 1,800 TPS over 5 minutes in our tests.

GAFAM implements a financial market smart contract that enables users to purchase and check the availability of stocks of Google, Apple, Facebook, Amazon, and Microsoft. Based on the values taken from [48], the system operates for 3 minutes, peaking at 19,800 TPS before stabilizing between 25 and 140 TPS. For simplicity, we have rounded the peak to 20,000 TPS.

V. Measurement Results

We investigate the impact of the five considered topologies on the energy consumption of the five examined blockchains under the three aforementioned workloads. To ensure robustness, each experiment was repeated three times. The average energy consumption across these runs is reported in Figs. 2 to 7. In Figs. 2 to 6 we present a multifaceted analysis for each blockchain, including the total network energy consumption (left), the average energy consumption per node (middle), and the average energy consumption per committed transaction (right). We include the specific *Commit Transaction Number* (CTN) for each experiment in the bar plot, as this is essential for energy consumption analyses. Furthermore, for the average energy consumption per node, we account for energy variability as the network size increases (denoted by *average · 10/40* in the legends of Figs. 2 to 6).

Specifically, we illustrate the energy variation between the observed values for the 40 nodes configuration and the 10 nodes configuration, comparing these results to the expected linear trend. For example, assuming linear scalability, a consumption of 2 kWh per node with 10 nodes would translate

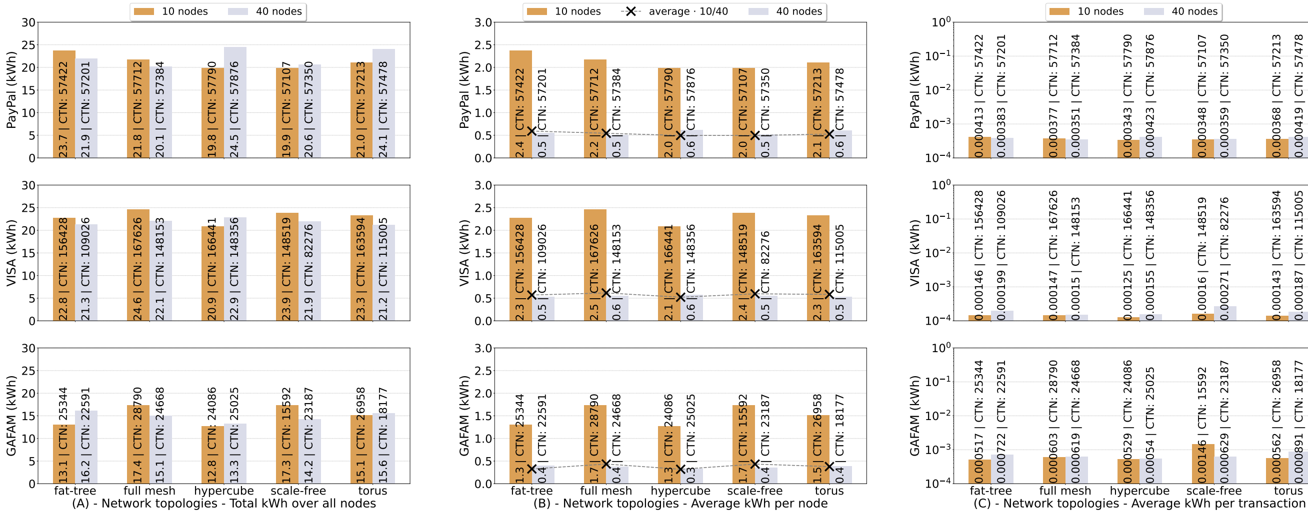

Fig. 2: Algorand energy consumption (kWh): total over all nodes (A), average per node (B), average per transaction (C).

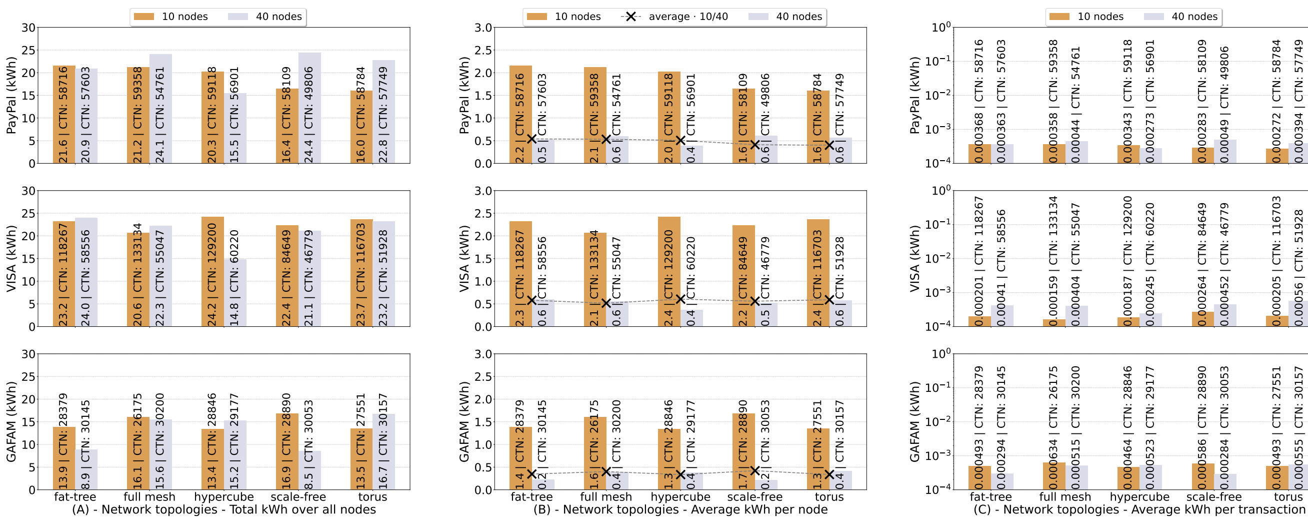

Fig. 3: Diem energy consumption (kWh): total over all nodes (A), average per node (B), average per transaction (C).

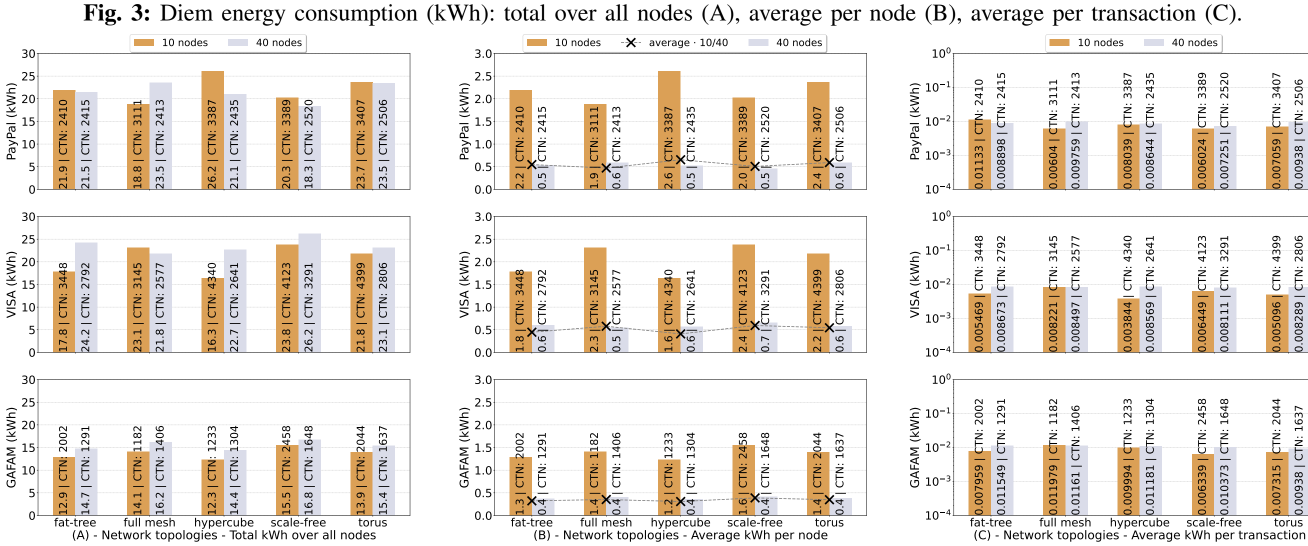

Fig. 4: Ethereum Clique energy consumption (kWh): total over all nodes (A), average per node (B), average per transaction (C).

to 0.5 kWh per node with 40 nodes. In Fig. 7, we focus on the same topology across all blockchains. We examine the average energy consumption per transaction, comparing it across all blockchains for the three workloads and each topology.

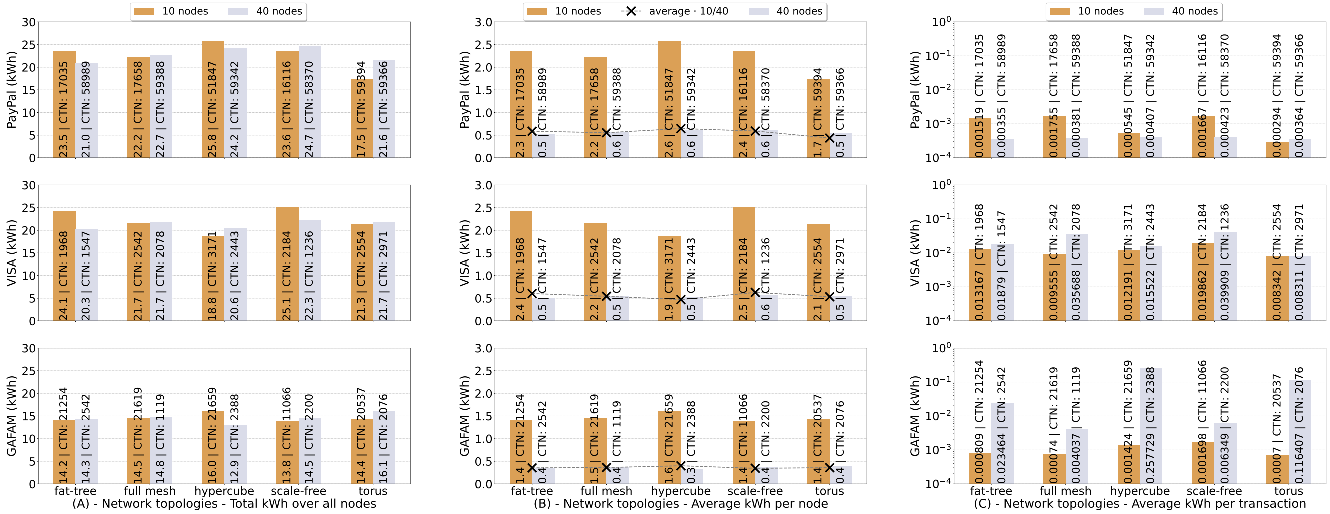

Fig. 5: Quorum IBFT energy consumption (kWh): total over all nodes (A), average per node (B), average per transaction (C).

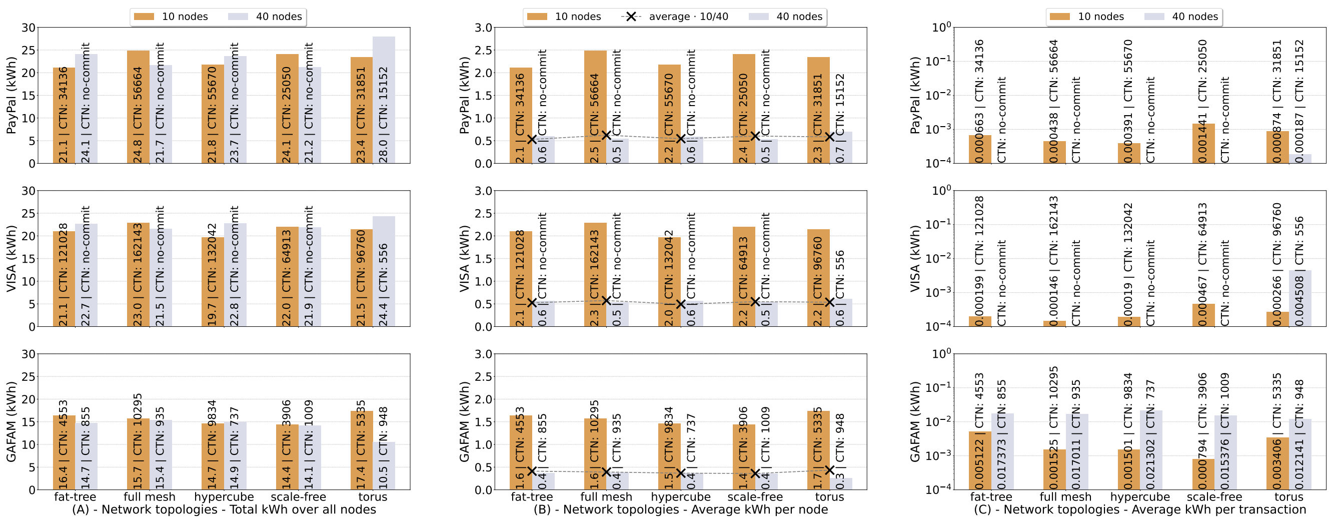

Fig. 6: Solana energy consumption (kWh): total over all nodes (A), average per node (B), average per transaction (C).

A. Energy Variability as the Network Expands

Let us analyze how energy consumption varies as the system scales in size. Interestingly, the data show that increasing the number of nodes in a network does not necessarily lead to a proportional rise in the overall energy consumption. In many cases, the network topology significantly and positively impacts the blockchain energy efficiency, although its effectiveness varies depending on the workload being analyzed. For Algorand (see Fig. 2 A and B), the full mesh topology stands out for its ability to reduce energy consumption (-17%) as the network grows, across all workloads, followed by the fat-tree topology. However, fat-tree reacts poorly to smart-contract-based workloads, with energy consumption per node increasing by over 20%. Similarly, hypercube and torus also exhibit inefficiencies under transaction processing workloads. In the case of Diem (see Fig. 3 A and B), the fat-tree topology is, on average, the most efficient across all workloads, with energy savings up to 35%. Hypercube also performs well for transaction processing workloads, reducing energy consumption by 40%, a trend that echoes the behavior observed in Algorand. However, the scale-free topology excels under smart contract workloads (-50%) but shows high consumption ($+45\%$) for less intensive transaction processing workloads, such as those resembling PayPal. Ethereum (see Fig. 4 A and B), on the other hand, struggles to achieve energy efficiency improvements as the network grows, regardless of the used topology. Scale-free and torus perform poorly across all three workloads, while hypercube and fat-tree are the least efficient for intensive transaction processing workloads like VISA ($+40\%$). For smart contract workloads, all topologies contribute to a consumption increasing between 10% and 20%. For Quorum (see Fig. 5 A and B), the fat-tree topology proves to be the most energy efficient across all three workloads, particularly for transaction processing workloads (-10% to -20%), followed by full mesh, which does not bring significant improvements or losses in energy consumption. Conversely, torus is the least efficient, with consumption increases ranging from 15% to 25%. Hypercube performs well with smart contract workloads ($+20\%$) but struggles with intensive transaction processing workloads like VISA, showing a $+15\%$ increase in energy consumption. Finally, Solana (see Fig. 6

A and B) faces severe scalability issues, with committed transactions dropping to zero in many 40-node configurations, which makes energy consumption irrelevant in most practical applications. However, for transaction processing workloads, the torus topology still emerges as inefficient ($+20\%$). A broader analysis highlights some common trends among these blockchains. Full mesh and fat-tree topologies tend to be the most energy efficient overall, adapting well to a variety of workloads. In contrast, topologies like torus and scale-free, while occasionally offering advantages in specific scenarios, are generally less reliable and often responsible for higher energy consumption, particularly in extended network configurations or intensive transaction processing workloads.

B. Network Topology Impact on Energy per Transaction

The relationship between energy consumption and committed transactions reveals that higher energy usage does not necessarily translate into greater transactional throughput. Instead, there is a clear negative correlation between energy per transaction (kWh/commit) and the number of committed transactions (see Figs. 2 to 6), which suggests that protocols capable of handling higher transaction volumes tend to achieve greater energy efficiency on a per-transaction basis. This highlights the importance of designing scalable network topologies that optimize energy use as the network grows. Additionally, Fig. 7 allows us to individually analyze each topology across all blockchains. Protocols like Algorand and Diem exemplify this efficiency, consistently maintaining low energy consumption per transaction (in the 10^{-4} order) under topologies such as hypercube and full mesh (see Figs. 2 and 3 C). These topologies enable both blockchains to process significant transaction volumes with minimal energy overhead. However, torus is less efficient in comparison, highlighting how the choice of topology can influence energy dynamics even within otherwise efficient systems. Ethereum, in stark contrast, exhibits the highest energy consumption per transaction (ranging from the 10^{-3} to 10^{-2} order) among the analyzed blockchains, regardless of the used network topology. This uniform inefficiency across all topologies points to inherent limitations in Ethereum's protocol design that restrict energy efficiency, especially as transaction volumes grow. Quorum, while generally more adaptable, experiences significant increases in energy consumption as workloads intensify, particularly when transitioning from simpler PayPal-like workloads to more demanding scenarios. This trend becomes even more pronounced under smart contract workloads, with topologies such as fat-tree, hypercube, and torus demonstrating energy consumption values in the 10^{-1} order range. As the network scales, Quorum's energy costs escalate further, indicating potential inefficiencies in adapting to high-demand workloads. Solana, on the other hand, shows energy costs per transaction comparable to those of Algorand and Diem under similar conditions, positioning it among the most energy-efficient protocols. However, network size issues within Solana hinder accurate quantification of its energy consumption as the network grows. Observations about the torus topology suggest that network conflicts in larger configurations would likely lead to increased energy consumption per transaction, undermining its otherwise competitive energy efficiency.

C. Workload-Specific Insights

The impact of network topology on energy efficiency becomes even more apparent when analyzing workload-specific behaviors across different blockchain protocols.

The PayPal workload, characterized by 200 TPS distributed over 300 seconds, is relatively low in intensity compared to the other two workloads. Under this workload, both Algorand and Diem maintain low energy consumption per transaction, with topologies such as full mesh and hypercube performing well. On the other hand, topologies like torus show slight inefficiencies, particularly for Ethereum and Quorum, where energy consumption rises moderately. For PayPal-like workloads, fat-tree and full mesh topologies are typically the most energy efficient.

The VISA workload, featuring 1,800 TPS for 300 seconds, significantly increases the demand on the system. Here, Algorand and Diem again show superior efficiency, with hypercube and full mesh ensuring energy efficiency even under high transaction volumes. In contrast, Ethereum faces substantial energy inefficiencies, with energy consumption rising across all topologies. Fat-tree and hypercube emerge as the least efficient under this workload, especially for Ethereum, where consumption can rise by over 40%. Similarly, Quorum exhibits an increase in energy usage under transaction processing workloads like VISA, especially with fat-tree and torus topologies, as the network scales.

The GAFAM workload, which executes smart contract calls over 180 seconds, presents a unique challenge due to its burst nature—initially reaching 20,000 TPS and then stabilizing at 100 TPS. For this workload, Algorand and Diem continue to demonstrate low energy consumption per transaction, particularly under hypercube and full mesh topologies. Torus again shows inefficiencies, especially with Ethereum, where energy usage rises sharply as the burst phase of the GAFAM workload demands high throughput. Fat-tree shows moderate efficiency for this workload, but its energy efficiency suffers compared to other topologies when dealing with high bursts of transactions.

VI. Conclusions

This study provides an innovative contribution to understanding energy consumption in blockchain systems, emphasizing the importance of network topology as a key factor. The findings underscore the need to integrate topological analysis into the design process to foster the development of more sustainable blockchains.

Network topology assumes a fundamental role in determining blockchain energy efficiency, with fat-tree and full mesh topologies emerging as the most efficient for increasing workloads. Algorand and Diem demonstrate superior efficiency, achieving low energy consumption per transaction, particularly under full mesh and hypercube configurations. However, the differences between Algorand and Diem are noteworthy. Algorand, as a fully decentralized blockchain, must overcome the complexity of consensus without central

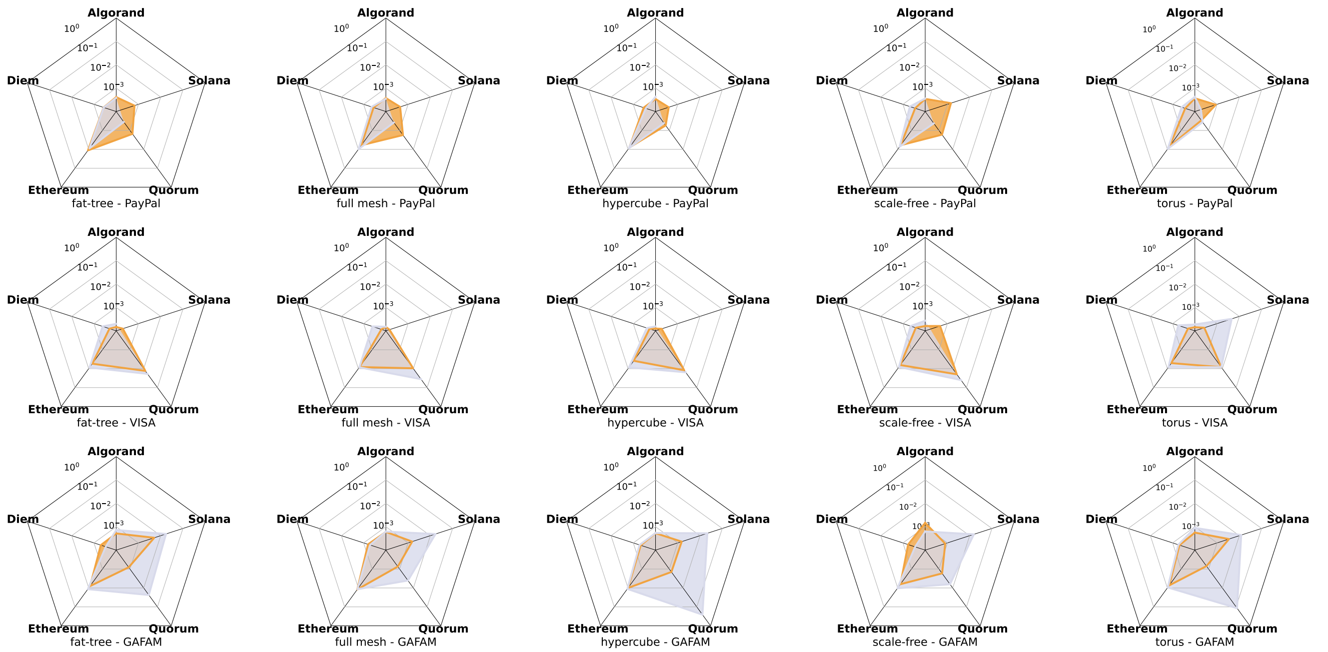

Fig. 7: Comparison of average energy consumption (kWh) per transaction across blockchains, stratified by topology (10 nodes , 40 nodes).

authority. Instead, Diem, with its centralized governance, inherently operates with reduced energy requirements. While Diem's efficiency is expected given its design, Algorand's ability to rival it highlights its advanced protocol optimization.

In stark contrast, Ethereum Clique features high energy consumption across all topologies and workloads, particularly under intensive scenarios like VISA and GAFAM.

Quorum faces scalability challenges, with rising energy costs under demanding workloads, especially in fat-tree, hypercube, and torus topologies. Solana's scalability issues obscure accurate energy consumption assessments, as the torus topology amplifies inefficiencies with network growth.

Workload-specific analyses further emphasize that full mesh and hypercube topologies perform best under lighter workloads (*i.e.*, PayPal), while Algorand and Diem remain efficient even under high-throughput tasks (*i.e.*, VISA). However, burst-heavy workloads (*i.e.*, GAFAM) challenge all networks, exposing Ethereum's inefficiencies and reinforcing the robustness of Algorand and Diem.

Overall, our results highlight that blockchain design must align with optimal topologies to enhance energy efficiency. While some public blockchains, like Ethereum, naturally align with scale-free structures and private ones, like Diem, with fat-tree or hypercube, our study suggests that node reconfigurations could further optimize efficiency while preserving their fundamental characteristics. The findings point to a critical call for Ethereum to address its limitations and a broader need for continued innovation in blockchain infrastructure to ensure sustainability and scalability in decentralized economies.

Limitations and Future Work. We conducted experiments on a 40-node blockchain network, offering insights into small-scale benchmarking [49]. While we did not explore large-scale dynamics, prior research [8] supports the representativeness of a 40-node setup. We anticipate that scaling up would further reinforce observed topology efficiency trends (Fig. 2-7). Future work will expand network size by using multiple clusters and heterogeneous machines for energy assessments.

We collected data from all three runs, allowing variance and confidence interval calculations, though these were omitted from figures for readability. Energy consumption was measured at the machine level for practicality, but finer-grain container-level analysis is possible [50]. Kollaps deploys containers based on resource availability, with minimal overhead from additional processes. While containerization and emulation (Docker and Kollaps) introduce differences from real-world deployments, they enable energy-efficient evaluations while capturing topology impact.

Electricity costs vary by region, so we report results in kWh, providing a basis for cost-effective topology selection and environmental impact assessment. Though this study focuses on static networks, our framework supports dynamic simulations of real-world events (e.g., node churn, connectivity changes).

To ensure reproducibility, we analyzed blockchain platforms with distinct characteristics (§IV-D), aligning with Diablo [8], but additional blockchains, topologies, and complex workloads can be considered.

Reproducibility. We support the reproducibility of our experiments by providing the repository link https://doi.org/10.5281/zenodo.11409100 along with instructions (file `README_ICBC25.md`) and datasets to replicate all experiments and results in this paper.

ACKNOWLEDGMENTS

This research has been supported by the PRIN 2020 project NiRvAna – Noninterference and Reversibility Analysis in Private Blockchains. The scholarship of the first author at the Italian PhD Program in Blockchain and Distributed Ledger Technology is funded by PNRR – Piano Nazionale di Ripresa e Resilienza according to D.M. 351/2022. This work was partially executed within the context of the REDONDA project (CHIST-ERA-22-SPiDDS-05).

References

[1] J. Xu, C. Wang, and X. Jia, "A Survey of Blockchain Consensus Protocols," *ACM Comput. Surv.*, vol. 55, no. 13s, pp. 278:1–278:35, 2023. [Online]. Available: https://doi.org/10.1145/3579845

[2] A. Gervais, G. O. Karame, K. Wüst, V. Glykantzis, H. Ritzdorf, and S. Capkun, "On the security and performance of proof of work blockchains," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '16. New York, NY, USA: Association for Computing Machinery, 2016, p. 3–16. [Online]. Available: https://doi.org/10.1145/2976749.2978341

[3] C. T. Nguyen, D. T. Hoang, D. N. Nguyen, D. Niyato, H. T. Nguyen, and E. Dutkiewicz, "Proof-of-stake consensus mechanisms for future blockchain networks: Fundamentals, applications and opportunities," *IEEE Access*, vol. 7, pp. 85 727–85 745, 2019.

[4] M. R. Islam, M. M. Rashid, M. A. Rahman, M. H. S. B. Mohamad, and A. H. B. Embong, "A comprehensive analysis of blockchain-based cryptocurrency mining impact on energy consumption," *International Journal of Advanced Computer Science and Applications*, vol. 13, no. 4, 2022. [Online]. Available: http://dx.doi.org/10.14569/IJACSA.2022.0130469

[5] C. T. Nguyen, D. T. Hoang, D. N. Nguyen, D. Niyato, H. T. Nguyen, and E. Dutkiewicz, "Proof-of-stake consensus mechanisms for future blockchain networks: Fundamentals, applications and opportunities," *IEEE Access*, vol. 7, pp. 85 727–85 745, 2019.

[6] R. Vaishnavi, C. Athira, C. Pradeep, Sankalpkumar, E. Prasanna, and V. Dr Srikanth, "Energy efficiency in blockchain social networks," *International Journal of Research Publication and Reviews*, vol. 5, pp. 819–825, 03 2024.

[7] Cambridge Center for Alternative Finance, "Cambridge blockchain network sustainability index (cbnsi)," accessed: 2025-10-01. [Online]. Available: https://ccaf.io/cbnsi/cbeci

[8] V. Gramoli, R. Guerraoui, A. Lebedev, C. Natoli, and G. Voron, "Diablo: A benchmark suite for blockchains," in *Proceedings of the Eighteenth European Conference on Computer Systems*, ser. EuroSys '23. New York, NY, USA: Association for Computing Machinery, 2023, p. 540–556. [Online]. Available: https://doi.org/10.1145/3552326.3567482

[9] B. Toshniwal and K. Kataoka, "Comparative performance analysis of underlying network topologies for blockchain," in *2021 International Conference on Information Networking (ICOIN)*, 2021, pp. 367–372.

[10] Y. Gilad, R. Hemo, S. Micali, G. Vlachos, and N. Zeldovich, "Algorand: Scaling byzantine agreements for cryptocurrencies," in *Proceedings of the 26th Symposium on Operating Systems Principles*, ser. SOSP '17. New York, NY, USA: Association for Computing Machinery, 2017, p. 51–68. [Online]. Available: https://doi.org/10.1145/3132747.3132757

[11] M. Baudet, A. Ching, A. Chursin, G. Danezis, F. Garillot, Z. Li, D. Malkhi, O. Naor, D. Perelman, and A. Sonnino, "State machine replication in the libra blockchain," *The Libra Assn., Tech. Rep*, vol. 7, 2019.

[12] G. Wood *et al.*, "Ethereum: A secure decentralised generalised transaction ledger," *Ethereum project yellow paper*, vol. 151, no. 2014, pp. 1–32, 2014.

[13] T. Espel, L. Katz, and G. Robin, "Proposal for protocol on a quorum blockchain with zero knowledge," *Cryptology ePrint Archive*, vol. 2017, p. 1093, 2017.

[14] A. Yakovenko. (2021) Solana: A new architecture for a high performance blockchain v0.8.13. Accessed: 2025-10-01. [Online]. Available: https://solana.com/solana-whitepaper.pdf

[15] P. Gouveia, J. Neves, C. Segarra, L. Liechti, S. Issa, V. Schiavoni, and M. Matos, "Kollaps: Decentralized and dynamic topology emulation," in *Proceedings of the Fifteenth European Conference on Computer Systems*, ser. EuroSys '20. New York, NY, USA: Association for Computing Machinery, 2020. [Online]. Available: https://doi.org/10.1145/3342195.3387540

[16] J. Zakrzewski, "Towards verification of ethereum smart contracts: A formalization of core of solidity," in *Verified Software. Theories, Tools, and Experiments: 10th International Conference, VSTTE 2018, Oxford, UK, July 18–19, 2018, Revised Selected Papers*. Berlin, Heidelberg: Springer-Verlag, 2018, p. 229–247. [Online]. Available: https://doi.org/10.1007/978-3-030-03592-1_13

[17] J. Mišić, V. B. Mišić, and X. Chang, "Toward decentralization in dpos systems: Election, voting, and leader selection using virtual stake," *IEEE Transactions on Network and Service Management*, vol. 21, no. 2, pp. 1777–1790, 2024.

[18] M. Castro and B. Liskov, "Practical Byzantine Fault Tolerance," in *Proceedings of the Third Symposium on Operating Systems Design and Implementation*, ser. OSDI '99. USA: USENIX Association, 1999, p. 173–186.

[19] G. Tripathi, M. A. Ahad, and G. Casalino, "A comprehensive review of blockchain technology: Underlying principles and historical background with future challenges," *Decision Analytics Journal*, vol. 9, p. 100344, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2772662223001844

[20] R. Cole and L. Cheng, "Modeling the energy consumption of blockchain consensus algorithms," in *2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, 2018, pp. 1691–1696.

[21] V. Hayashi, F. Almeida, and A. Komo, "LabBitcoin: FPGA IoT Testbed for Bitcoin Experiment with Energy Consumption," in *Anais Estendidos do XXI Simpósio Brasileiro de Segurança da Informação e de Sistemas Computacionais*. Porto Alegre, RS, Brasil: SBC, 2021, pp. 90–97. [Online]. Available: https://sol.sbc.org.br/index.php/sbseg_estendido/article/view/17344

[22] R. Antwi, J. D. Gadze, E. T. Tchao, A. Sikora, H. Nunoo-Mensah, A. S. Agbemenu, K. O.-B. Obour Agyekum, J. O. Agyemang, D. Welte, and E. Keelson, "A survey on network optimization techniques for blockchain systems," *Algorithms*, vol. 15, no. 6, 2022. [Online]. Available: https://www.mdpi.com/1999-4893/15/6/193

[23] D. Saingre, T. Ledoux, and J.-M. Menaud, "Bctmark: a framework for benchmarking blockchain technologies," in *2020 IEEE/ACS 17th International Conference on Computer Systems and Applications (AICCSA)*. Los Alamitos, CA, USA: IEEE Computer Society, nov 2020, pp. 1–8. [Online]. Available: https://doi.ieeecomputersociety.org/10.1109/AICCSA50499.2020.9316536

[24] ——, "Measuring performances and footprint of blockchains with bctmark: a case study on ethereum smart contracts energy consumption," *Cluster Computing*, vol. 25, no. 4, p. 2819–2837, Aug. 2022. [Online]. Available: https://doi.org/10.1007/s10586-021-03441-x

[25] P. K. Sharma, S. Singh, Y.-S. Jeong, and J. H. Park, "Distblocknet: A distributed blockchains-based secure sdn architecture for iot networks," *IEEE Communications Magazine*, vol. 55, no. 9, pp. 78–85, 2017.

[26] S. R. Basnet and S. Shakya, "Bss: Blockchain security over software defined network," in *2017 International Conference on Computing, Communication and Automation (ICCCA)*, 2017, pp. 720–725.

[27] A. Yazdinejad, R. M. Parizi, A. Dehghantanha, Q. Zhang, and K.-K. R. Choo, "An energy-efficient sdn controller architecture for iot networks with blockchain-based security," *IEEE Transactions on Services Computing*, vol. 13, no. 4, pp. 625–638, 2020.

[28] 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

[29] D. Mazieres, "The stellar consensus protocol: A federated model for Internet-level consensus," *Stellar Development Foundation*, vol. 32, pp. 1–45, 2015.

[30] R.-A. Cherrueau, M. Delavergne, A. van Kempen, A. Lebre, D. Pertin, J. R. Balderrama, A. Simonet, and M. Simonin, "Enoslib: A library for experiment-driven research in distributed computing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 33, no. 6, pp. 1464–1477, 2022.

[31] S. T. Borda and J. Ermont, "An evaluation of software-based tsn traffic shapers using linux tc," in *2022 IEEE 18th International Conference on Factory Communication Systems (WFCS)*, 2022, pp. 1–4.

[32] K. N. Khan, M. Hirki, T. Niemi, J. K. Nurminen, and Z. Ou, "Rapl in action: Experiences in using rapl for power measurements," *ACM Trans. Model. Perform. Eval. Comput. Syst.*, vol. 3, no. 2, Mar. 2018. [Online]. Available: https://doi.org/10.1145/3177754

[33] Q. Chen and D. Shi, "The modeling of scale-free networks," *Physica A: Statistical Mechanics and its Applications*, vol. 335, no. 1, pp. 240–248, 2004. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0378437103011257

[34] F. Chen, Z. Chen, X. Wang, and Z. Yuan, "The average path length of scale free networks," *Communications in Nonlinear Science and Numerical Simulation*, vol. 13, no. 7, pp. 1405–1410, 2008. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1007570406002383

[35] T. Wang, Z. Su, Y. Xia, B. Qin, and M. Hamdi, "Novacube: A low latency torus-based network architecture for data centers," in *2014 IEEE Global Communications Conference*, 2014, pp. 2252–2257.

[36] M. Yin, D. Malkhi, M. K. Reiter, G. G. Gueta, and I. Abraham, "Hotstuff: Bft consensus with linearity and responsiveness," in *Proceedings of the 2019 ACM Symposium on Principles of Distributed Computing*, ser. PODC '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 347–356. [Online]. Available: https://doi.org/10.1145/3293611.3331591

[37] P. Szilágyi, "Eip-225: Clique proof-of-authority consensus protocol," Ethereum Improvement Proposals, March 2017, accessed: 2025-10-01. [Online]. Available: https://eips.ethereum.org/EIPS/eip-225

[38] R. Saltini, "IBFT Liveness Analysis," in *2019 IEEE International Conference on Blockchain (Blockchain)*, 2019, pp. 245–252.

[39] A. Yakovenko, "TowerBFT: Solana's high performance implementation of PBFT," Feb 2020. [Online]. Available: https://medium.com/solana-labs/tower-bft-solanas-high-performance-implementation-of-pbft-464725911e79

[40] Algorand, "Support transaction arguments for 'goal app method' (3233) · algorand/go-algorand@116c06e," Nov 2021, accessed: 2025-10-01. [Online]. Available: https://github.com/algorand/go-algorand/commit/116c06e00ad4f0dfa0c22577b733d8b0d933be0a

[41] Diem, "[deps] update hyper to fix two rustsecs · diem/diem@4b3bd1e," Aug 2021, accessed: 2025-10-01. [Online]. Available: https://github.com/diem/diem/commit/4b3bd1e1b92d4aca5dfcc5294941179348c0be8e

[42] Ethereum, "Cmd/geth, console: Support interrupting the js console (23387) · ethereum/go-ethereum@72c2c0a," Dec 2021, accessed: 2025-10-01. [Online]. Available: https://github.com/ethereum/go-ethereum/commit/72c2c0ae7e2332b08d3e1ebfe5f850a92e26e8a1

[43] A. Baliga, I. Subhod, P. Kamat, and S. Chatterjee, "Performance evaluation of the quorum blockchain platform," *CoRR*, vol. abs/1809.03421, 2018. [Online]. Available: http://arxiv.org/abs/1809.03421

[44] Consensys, "Added test to confirm that there is no deadlock when testqbftblock is... · consensys/quorum@919800f," Nov 2021, accessed: 2025-10-01. [Online]. Available: https://github.com/Consensys/quorum/commit/919800f019cc5d2b931b5cd81600640a8e7cd444

[45] J. Sliwinski, Q. Kniep, R. Wattenhofer, and F. Schaich, "Halting the Solana Blockchain with Epsilon Stake," in *Proceedings of the 25th International Conference on Distributed Computing and Networking*, 2024, pp. 45–54.

[46] Solana-Labs, "Add script for the traceability (23626) · solana-labs/solana@0d36961," Mar 2022, accessed: 2025-10-01. [Online]. Available: https://github.com/solana-labs/solana/commit/0d369616e7e663355c682a156071ce66ba5a434a

[47] D. Mechkaroska, V. Dimitrova, and A. Popovska-Mitrovikj, "Analysis of the possibilities for improvement of blockchain technology," in *2018 26th Telecommunications Forum (TELFOR)*, 2018, pp. 1–4.

[48] 2022, accessed: 2025-10-01. [Online]. Available: https://www.nasdaq.com/

[49] A. Lebedev and V. Gramoli, "On the relevance of blockchain evaluations on bare metal," in *Distributed Ledger Technology*, N. Dong, B. Pillai, G. Bai, and M. Utting, Eds. Singapore: Springer Nature Singapore, 2024, pp. 22–38.

[50] M. Warade, K. Lee, C. Ranaweera, and J.-G. Schneider, "Monitoring the energy consumption of docker containers," in *2023 IEEE 47th Annual Computers, Software, and Applications Conference (COMPSAC)*. IEEE, 2023, pp. 1703–1710.