Università degli Studi di Camerino

School of Advanced Studies

Doctoral course in

BLOCKCHAIN AND DISTRIBUTED LEDGER TECHNOLOGY

Curriculum

ECONOMICS AND FINANCE

Cycle

XXXVIII

Scientific field of the dissertation (SSD)

INF/01

COMPARING BLOCKCHAINS: PERFORMANCE, ENERGY, AND ECONOMIC EFFICIENCIES

Ph.D. Student:

Vincenzo P. Di Perna *(UniUrb)*

Supervisor:

Prof. Marco Bernardo *(UniUrb)*

Co-supervisors:

Prof. Francesco Fabris *(UniTs)*

Prof. Valerio Schiavoni *(UniNe)*

Coordinator of the Ph.D. Program:

Prof. Flavio Corradini *(UniCam)*

Abstract

Blockchains are distributed ledgers that let mutually distrustful parties agree on an append-only transaction history without relying on a central authority. By combining cryptographic hashing, digital signatures, and consensus mechanisms, blockchains provide tamper evidence, auditability, and agreement among nodes. Modern blockchain systems significantly vary in consensus design (*e.g.,* Proof of Work, Proof of Stake, Proof of Authority, and Byzantine Fault Tolerance mechanisms), access model (open vs. permissioned), and execution layers (from simple asset transfers to expressive smart-contract virtual machines). The related architectural choices shape decentralization, fault tolerance, and the attainable latency-throughput envelope.

As blockchain deployments expand to payments, tokenization, decentralized finance, supply chain traceability, and digital identities, comparing these systems has become an urgent necessity. Unfortunately, rigorous blockchain evaluation remains difficult. On the one hand, measurements are confounded by fluctuating network conditions, heterogeneous infrastructures, and rapidly evolving software. On the other hand, results are too often collapsed to a single number (such as transactions per second) without dispersion or methodological details; economic assessments of crypto-assets lack a unified and interpretable index that captures the balance of core economic parameters and their trade-offs (usage, liquidity, stability, and security) rather than market price sentiment; and experimental studies rarely address the dimensions of experimental repeatability (same setup, same results) and performance predictability (stable expectation). The consequence is an evidence gap: how to assess and compare the efficiency of blockchains – spanning performance, energy, economics, and result stability – in different scenarios?

This dissertation aims at reducing this gap with a coherent yet modular approach that combines topology-controlled benchmarking with an orthogonal, entropy-based economic analysis, delivering four contributions. First, it introduces Lilith, a system-agnostic benchmarking framework that couples workload generation with network emulation to run controlled, repeatable experiments under explicit overlay topologies (*i.e.,* the logical peer-to-peer connectivity graphs) and link properties such as latency, bandwidth, and packet loss. Lilith orchestrates deterministic deployments (pinned artifacts, controlled boot order, and CPU core pinning and memory binding), integrates power probes, and provides a uniform client interface; this underpins a comparison based on typical performance metrics. Second, Lilith is employed to quantify blockchain energy consumption under realistic conditions. Third, Lilith is adopted for a network-controlled, multi-run measurement campaign to produce a public dataset. By combining dispersion metrics (*e.g.,* worst-case deviation) with analysis of variance and intraclass correlation, we quantify run-to-run variability and performance predictability across blockchains, topologies, workloads, and node-set sizes. Fourth, in addition to Lilith, the dissertation introduces the Entropy Balance index (EB-index), which aggregates heterogeneous on-chain indicators into a single, interpretable score of economic efficiency.

As for the first three contributions, we set up the experimental baseline by considering five network topologies (fat-tree, full mesh, hypercube, scale-free, torus) and five industry-grade blockchains (Algorand, Diem, Ethereum Clique, Quorum IBFT, Solana), exercised with transfer transactions and smart-contract workloads (DDoS, FIFA, GAFAM, gaming, PayPal, VISA) across two node-set sizes (10 and 40).

In the performance study, the network topology emerges as the primary factor determining throughput and latency. Full mesh, hypercube, and torus deliver higher performance under heavy load. The performance of Algorand and Diem is stable with respect to topology changes, while Ethereum is less sensitive but remains slower.

In the energy study, fat-tree and full mesh turn out to be the most energy-efficient topologies, especially at high load. Algorand and Diem exhibit the lowest energy per transaction, Ethereum Clique the highest across topologies; Quorum IBFT and Solana become costlier as workload intensity and network size increase.

The experimental repeatability and performance predictability study shows low performance variance (transactions per second, block latency, energy consumption) for Algorand and Diem and pronounced sensitivity for Solana and Quorum IBFT, especially as workloads, node-set size, and geo-latency conditions vary. The released dataset and the accompanying analysis templates, which are based on clusters instead of public-cloud testing, enable thorough checks that go beyond point estimates by quantifying dispersion and confidence in comparative results.

Finally, in the economic study, the EB-index aggregates heterogeneous on-chain indicators – such as user activity (transactions, active addresses), token distribution (balance concentration), and supply turnover/velocity – by using the normalized Shannon entropy and its weighted Beliş-Guiaşu variant. When applied to the capitalization-based leading crypto-assets Bitcoin, Ethereum, Ripple, USD Coin, Dogecoin, and Cardano, the EB-index separates volume-driven bursts from structurally balanced ecosystems and reveals differences that price, total value locked, or raw activity may blur.

Overall, this dissertation delivers a topology-aware blockchain benchmarking framework, empirical evidence that network structure materially affects performance and energy, a public multi-run dataset together with analysis templates that promote experimental repeatability and performance predictability, and an entropy-based index for assessing economic efficiency.

Key words: Blockchain benchmarking; Network topology; Energy consumption; Experimental repeatability; Performance predictability; Cryptocurrency efficiency; Economic index; Shannon entropy.

Acknowledgments

I would like to thank Prof. Marco Bernardo (University of Urbino) for being the kind of academic who turns craftsmanship into a method. His precision, organization, and care for the smallest details taught me that good research is the art of leaving no loose ends – and great research is making it unassailable. He was always available, fair, and open to compromise when it served clarity and rigor. I learned from him that high standards and empathy can – and should – coexist.

I am grateful to Prof. Francesco Fabris (University of Trieste): his calm, balanced voice has been a compass whenever discussions drifted. He has a gift for restoring equilibrium and moving teams forward. I still hear his rule of thumb – "a result is correct if it's verified three times" – guiding my experiments. I thank him for his patience and sense of humor as well as for moderating with grace when it mattered most.

I am thankful to Prof. Valerio Schiavoni (University of Neuchâtel) for turning a visit abroad into a home away from home. His availability, kindness, and enthusiasm made every meeting productive and every game of tennis a reminder that curiosity and play can live side by side. His hospitality and candid feedback helped me grow both as a researcher and as a person.

I wish to thank Christopher Jack (CCAF, Cambridge) because, from day one of my internship, he made room for dialogue. He helped me feel at ease in a new context and encouraged me to contribute, not just to fit in. I thank him for opening doors, asking the right questions, and making space for my work within a community I deeply respect.

I am grateful to the reviewers of this thesis, Prof. Andrea Marin (Ca' Foscari University of Venice) and Prof. Etienne Rivière (Université Catholique de Louvain), for their careful reading and incisive feedback. Their comments and suggestions substantially improved the clarity, rigor, and overall presentation of this manuscript.

I gratefully acknowledge the support of the PNRR scholarship of the University of Urbino, which made this doctoral journey financially sustainable and intellectually focused. That stability let me spend time where it mattered most: with data and code.

I also thank the PRIN 2020 project NiRvAna for supporting the "noninterference and reversibility" perspective that informed my thinking about correctness, observability, and security throughout this work. Its community and meetings sharpened my understanding of what "sound by construction" really means.

My thanks to project REDONDA for the resources and collaborative environment that helped me stress-test ideas at scale. The project's goals – bridging research and practice – kept me honest about external validity and practical relevance.

A special thanks to Prof. Flavio Corradini (University of Camerino) for his guidance and care – often behind the scenes – whenever administrative paths got complicated.

To all of them: many thanks for their trust, challenges, and patience. This work reflects their influence at every step.

Contents

List of Tables ix

List of Figures xi

List of Acronyms xiii

1 Introduction 1
- 1.1 Blockchain Technology . . . 1
- 1.2 How to Compare Blockchains . . . 3
- 1.3 Contributions of the Thesis . . . 5
- 1.4 Structure of the Thesis . . . 6

2 Background 9
- 2.1 Blockchains: Features and Classification . . . 9
  - 2.1.1 Toward Algorithmic Consensus . . . 10
  - 2.1.2 A Decentralized Perspective . . . 13
  - 2.1.3 Key Dimensions of Blockchain Systems . . . 16
  - 2.1.4 Representative Blockchains . . . 20
- 2.2 Network Topologies . . . 22
  - 2.2.1 Topology Families and Structural Properties . . . 22
  - 2.2.2 Overlay Networking and Gossip in Blockchains . . . 24
  - 2.2.3 Propagation and Latency . . . 25
  - 2.2.4 Common Deployments . . . 27
- 2.3 Typical Workloads . . . 29
  - 2.3.1 Evolution and Taxonomy of Workloads . . . 29
  - 2.3.2 Canonical Families . . . 31
- 2.4 Benchmarking Tools . . . 33
  - 2.4.1 Technical Frameworks . . . 34
  - 2.4.2 Energetic Approaches . . . 35
  - 2.4.3 Economic Indicators . . . 37
- 2.5 Open Challenges . . . 38
  - 2.5.1 Protocol Claims and Evidence-Based Validation . . . 39
  - 2.5.2 Standardization and Comparability . . . 39
  - 2.5.3 Variability and Cost of Uncontrolled Deployment Environments . . . 40
  - 2.5.4 Network Variability and Controllability . . . 41

2.5.5 Adversarial Dynamics . . . 42
2.5.6 Resource Management and Measurement Fidelity . . . 42
2.5.7 Missing a Unified Index . . . 42

3 Lilith: A New Blockchain Benchmarking Framework 45
3.1 Lilith Overview . . . 45
3.2 The Diablo Benchmark Suite . . . 48
3.3 The Kollaps Network Emulator . . . 48
3.4 Network Topology Generation . . . 48
3.5 Built-in Workloads . . . 50
3.6 Blockchains under Test . . . 52
3.7 Benchmark Execution . . . 54
3.8 Metrics Gathering . . . 55
3.9 Testing Configurations . . . 56

4 Performance Efficiency: Impact of Network Topologies 59
4.1 Performance Evaluation of Blockchains . . . 59
4.2 Evaluation Setup . . . 60
4.2.1 Experimental Assets . . . 61
4.2.2 Testing Configurations . . . 61
4.3 Measurement Results . . . 61
4.3.1 Small Deployment . . . 61
4.3.2 Scaling the Network . . . 62
4.3.3 Network Load . . . 63
4.3.4 Network Dynamics . . . 63
4.4 Discussion and Limitations . . . 65

5 Energy Efficiency: Impact of Network Topologies 69
5.1 Energy Consumption of Blockchains . . . 69
5.2 Evaluation Methodology . . . 71
5.2.1 Framework . . . 71
5.2.2 Experimental Settings . . . 72
5.3 Measurement Results . . . 72
5.3.1 Energy Variability as the Network Expands . . . 74
5.3.2 Network Topology Impact on Energy per Transaction . . . 75
5.3.3 Workload-Specific Insights . . . 76
5.4 Discussion and Limitations . . . 77

6 Experimental Repeatability and Performance Predictability: A Network-Controlled Approach 79
6.1 Repeatability and Predictability . . . 79
6.2 Methodology . . . 81
6.2.1 P0–P3 Labels, Predictability, and Network Isolation . . . 81

6.2.2 Controlled Testbed and Experimental Design 83
6.2.3 Statistical Treatment of Variability 83
6.3 Experimental Results 85
6.3.1 Typical Dispersion around the Mean 87
6.3.2 Worst-Case Behavior 87
6.3.3 Variability Sources 88
6.4 Discussion and Limitations 91
6.4.1 Positioning the Dataset and Study Scope 92
6.4.2 System/Topology-Level Variance Interpretation 93
6.4.3 Implications for Practice and Benchmark Design 95
6.4.4 Experimental Limitations 96

7 Economic Efficiency: An Entropy-Based Approach 99
7.1 Economic Efficiency 99
7.2 Efficiency of Cryptocurrencies: A Literature Review 101
7.3 The Quest for an Aggregated Economic Efficiency Index 101
7.3.1 The Entropy Measure 102
7.3.2 Economic Parameters 105
7.4 An Illustrative Example 109
7.4.1 EB-index Based on Unweighted Entropy: Good Balance 110
7.4.2 EB-index Based on Weighted Entropy: Performance or Importance 111

8 Conclusions 115
8.1 Summary of Results 115
8.2 Future Work 117

Bibliography 121

List of Tables

2.1 Comparison of major blockchain consensus mechanisms across key operational and architectural dimensions: finality, energy consumption, decentralization, scalability, and common deployment contexts (expressed in qualitative terms, not via measures). 12

2.2 Top 10 assets by market capitalization (USD). Source: CompaniesMarketCap, *Assets by Market Cap* (accessed Dec 2025); values are live and may fluctuate. 18

2.3 Blockchains considered in this thesis: type, consensus/finality, execution environment, and Decentralized Application (DApp) language. 21

2.4 Canonical structural counts for reference topologies. 29

2.5 Overview of supported features across blockchain benchmarking tools. Legend: data unavailable; ✗ not reported; ◑ partially reported; ✓ fully reported; "–" badge scheme not yet in place (ACM/IEEE since 2016/2019). Abbreviations used: Centr. = Centralized, Distr. = Distributed, Simul. = Simulation, Emul. = Emulation, SC = Smart Contracts, TT = Transfer Transactions. 36

2.6 Weekly cloud cost comparison for three scenarios: (i) 10 nodes, 30/60 vCPUs, 64/120 GB RAM; (ii) 200 nodes, 4 vCPUs, 8 GB RAM; (iii) 200 nodes, 8 vCPUs, 16 GB RAM. 42

3.1 Selected topologies (a), workloads (b), and blockchains (c). 53

6.1 Representative blockchain performance studies and their position in the P0–P3 ladder. Legend: ✓ variance reporting; ◑ some distributional analysis (*e.g.,* percentiles) but not under controlled repeated setups; data unavailable; ✗ single-run or averages only. $B \times T \times W \times S$ = Blockchains × Topologies × Workloads × Scales (validator-set sizes). 82

6.2 Overall blockchain-level variability. For each metric we report IQR% and Std%. Percentages are with respect to the per-configuration mean; absolute magnitudes are in Figures 6.1–6.3 (left). The most stable blockchains (minimum values) are highlighted in , the least stable ones in . Highlights select best/worst stability over all T, W, S per blockchain; generating configurations appear in the per-factor breakdown (Figures 6.1–6.3, Tables 6.5–6.7). 86

6.3 Worst-case run-to-run blockchain swings. For each metric we report: the maximum observed range (Δ_{abs}), the largest absolute drop below the mean ($\Delta^{\downarrow}$), the largest absolute increase above the mean ($\Delta^{\uparrow}$), and their percentage counterparts ($\Delta^{\downarrow}\%$, $\Delta^{\uparrow}\%$) relative to the configuration mean. The least stable values (WCD and WCD%, depending on the column) are highlighted in . All WCD metrics are computed over 10/10 successful runs; failure regimes (*NO* or m=0) are excluded. 87
6.4 Variance decomposition and run-to-run reliability. Each entry reports the percentage of variance explained by the corresponding factor or interaction in the factorial ANOVA; ε denotes the residual term (higher-order interactions and unexplained noise). The last column reports the intraclass correlation coefficient (ICC) for per-configuration runs. B: Blockchains; T: Topologies; W: Workloads; S Scales (validator-set sizes). 88
6.5 Experimental repeatability for each blockchain-topology pair (same conventions as Table 6.2). 92
6.6 Experimental repeatability for each blockchain-workload pair (same conventions as Table 6.2). 93
6.7 Experimental repeatability for each blockchain with 10- and 40-node scaling (same conventions as Table 6.2). 93

7.1 Wealth concentration, engagement, and asset dormancy of major cryptoassets from July 2010 up to April 2025. For wealth concentration we use a Gini coefficient adapted to average balances over class intervals. Engagement is defined as the share of addresses with positive balance over the total. Asset dormancy is calculated as one minus the ratio of active supply to circulating supply. Source: *Coin Metrics* [86]. 100
7.2 Basic parameters. 106
7.3 Key economic quality attributes. 107
7.4 Blockchain data intelligence platforms comparison. 107
7.5 *Coin Metrics* parameters selection. 108
7.6 Selected cryptocurrencies and their financial and technical characteristics as of April 2025. Source: *Coin Metrics*. 109
7.7 Derived quality attributes chosen for the two sets *Set1* and *Set2*. 110

List of Figures

2.1 Historical evolution of cryptocurrency market capitalization and Decentralized Finance (DeFi) Total Value Locked (TVL), illustrating the divergence between speculative asset valuation and protocol-level engagement. Source: TradingView at `https://www.tradingview.com/chart/` 15

2.2 Real-world network topologies. 27

2.3 Latency (ms) and throughput (Mbps) heatmaps for 2024 measurements (upper colored triangles); % difference 2024 vs. 2023 (lower gray triangles). Amazon Web Services (AWS) regions: *af-s-1* (Cape Town), *ap-ne-1* (Tokyo), *ap-s-1* (Mumbai), *ap-se-2* (Sydney), *eu-n-1* (Stockholm), *eu-s-1* (Milan), *me-s-1* (Bahrain), *sa-e-1* (Sao Paulo), *us-e-2* (Ohio), and *us-w-2* (Oregon). 40

2.4 Long-term standard deviation of daily-mean RTTs (Apr. '23–Jan. '25) across 10 Amazon Web Services (AWS) regions (see Figure 2.3). Each panel shows one source region against all destination regions; measurements use one Amazon Elastic Compute Cloud (EC2) virtual machine per region and cover all region pairs. 41

3.1 Architecture and execution flow of a Lilith experiment (dashed contour lines represent existing components). 46

4.1 Blockchain performance across various workloads using the 2023 AWS dataset (**Algorand**, **Diem**, **Ethereum**, **Quorum**, **Solana**). 64

4.2 Network load (Mbps) during workload execution (**Algorand**, **Diem**, **Ethereum**, **Quorum**, **Solana**). 65

4.3 Network dynamics, one node per region, full mesh topology (**Algorand**, **Diem**, **Ethereum**, **Quorum**, **Solana**). 66

4.4 Benchmark with increasing latencies (PayPal workload). Red lines mark latency variation events (see §4.3.4). 66

4.5 Benchmark with increasing latencies (GAFAM workload). Red lines mark latency variation events (see §4.3.4). 66

5.1 Algorand energy consumption (Kilowatt-hour (kWh)): total over all nodes (A), average per node (B), average per transaction (C). 72

5.2 Diem energy consumption (Kilowatt-hour (kWh)): total over all nodes (A), average per node (B), average per transaction (C). 73
5.3 Ethereum Clique energy consumption (Kilowatt-hour (kWh)): total over all nodes (A), average per node (B), average per transaction (C). 73
5.4 Quorum IBFT energy consumption (Kilowatt-hour (kWh)): total over all nodes (A), average per node (B), average per transaction (C). 73
5.5 Solana energy consumption (Kilowatt-hour (kWh)): total over all nodes (A), average per node (B), average per transaction (C). 74
5.6 Comparison of average energy consumption (Kilowatt-hour (kWh)) per transaction across blockchains, stratified by topology (10 nodes, 40 nodes). 75

6.1 Throughput deviation across workloads. For each panel and configuration, deviations (Δ) are computed against the mean m over the 10 runs: left = absolute swing (TPS); right: relative swing (% from mean). Right axis clipped at $\pm 100\%$, with out-of-scale values explicitly labeled. Panels show (a) 10 nodes and (b) 40 nodes. Failure regimes are explicitly labeled: *NO* denotes 10/10 non-operational runs, while m=0 denotes 10/10 runs where the workload was delivered but no blocks were committed. 89
6.2 Latency deviation (same conventions as Figure 6.1). 90
6.3 Energy deviation (same conventions as Figure 6.1). 91

7.1 Comparison between unweighted and weighted entropy for (a-b) *Set1* (DQ1, DQ3, DQ7, PQ7) and (c-d) *Set2* (DQ1, DQ2, DQ3, DQ4, DQ5, DQ6). For the weighted entropy the chosen weights are $w_i = r_i$. 110
7.2 Probability distribution of the selected attributes within each set, computed for each asset. 112

List of Acronyms

ADA Cardano coin

AI Artificial Intelligence

AMM Automated Market Maker

ASIC Application-Specific Integrated Circuit

AVM Algorand VM

AWS Amazon Web Services

BAT Basic Attention Token

BBA* Binary Byzantine Agreement

BFT Byzantine Fault Tolerance

BTC Bitcoin

CBDC Central Bank Digital Currency

CBNSI Cambridge Blockchain Network Sustainability Index

CEO Chief Executive Officer

CPU Central Processing Unit

DAG Directed Acyclic Graph

DAI DAI coin

DAO Decentralized Autonomous Organization

DApp Decentralized Application

DDoS Distributed Denial of Service

DeFi Decentralized Finance

DEX Decentralized Exchange

DHT Distributed Hash Table

DLT Distributed Ledger Technology

DNS Domain Name System

DOGE Dogecoin

DoS Denial of Service

DPoS Delegated Proof of Stake

EB-index Entropy Balance index

ECB European Central Bank

ETF Exchange-Traded Fund

ETH Ethereum coin

EVM Ethereum VM

FaaS Function as a Service

FPGA Field-Programmable Gate Array

GAFAM Google Apple Facebook Amazon Microsoft

GDP Gross Domestic Product

HPI Human Poverty Index

I/O Input/Output

IBFT Istanbul BFT

IoT Internet of Things

IPFS InterPlanetary File System

kWh Kilowatt-hour

L0 Layer 0

L1 Layer 1

L2 Layer 2

LINK Chainlink coin

MEV Maximal Extractable Value

MKR Maker

NFT Non-Fungible Token

OSPF Open Shortest Path First

P2P Peer to Peer

PBFT Practical BFT

PoA Proof of Authority

PoH Proof of History

PoS Proof of Stake

PoW Proof of Work

RAPL Intel Running Average Power Limit

RTT Round-Trip Time

SDN Software-Defined Networking

SEC U.S. Securities and Exchange Commission

SHA Secure Hash Algorithm

SOL Solana coin

SSI Self-Sovereign Identity

TPS Transactions Per Second

TVL Total Value Locked

UNI Uniswap

USD U.S. Dollar

USDC USD Coin

USDT USD Tether

UTXO Unspent Transaction Output

VM Virtual Machine

VRF Verifiable Random Function

WAN Wide Area Network

XRP Ripple coin

Chapter 1

Introduction

This chapter introduces the technological context of blockchain systems and their main operating properties (§1.1). It then sets the scope of the dissertation, which is the production of a reliable comparison of blockchains along four axes (§1.2): performance variability, energy consumption, experimental rigor, and economic efficiency. It explains why thorough comparisons must treat operating conditions like network topology and deployment heterogeneity as first-class parameters and motivates an economic index where systemic and economic data are aggregated coherently. Building on these principles, the chapter previews the dissertation's core contributions (§1.3) and closes by outlining its structure (§1.4).

1.1 Blockchain Technology

A blockchain system is a form of Distributed Ledger Technology (DLT), which denotes a family of protocols that records data securely, transparently, and in a tamper-evident way across a decentralized network of mutually untrusted nodes [88, 178]. Ledger entries – hereafter transactions – consist of digitally signed state updates that are validated against protocol rules and digital signatures, batched into blocks, and chained through cryptographic hashes, *i.e.,* fixed-length one-way digests that change unpredictably even after a single-bit input modification. Each block commits to its predecessor and the set of included transactions, enabling compact inclusion proofs and making tampering evident.

Unlike centralized databases, which are optimized for trusted operators, stable infrastructures, and low-latency transactions, blockchains are designed for adversarial environments and cross-organizational trust boundaries. Deployments can be public (open), private (permissioned), or consortium-based; they may couple incentives with verifiable execution of application logic.

These properties arise from the combination of a consensus mechanism – which establishes agreement on transaction validity and ordering across distributed nodes [335] – and a Peer to Peer (P2P) overlay where peers discover one another and propagate transactions and blocks through gossip – a push-pull exchange among randomly chosen peers without any central coordinator [2]. Consensus designs span probabilistic finality in longest/heaviest chain variants under Proof of Work (PoW) [235], carried out by miners,

or Proof of Stake (PoS) [244], carried out by validators, as well as deterministic finality in Byzantine Fault Tolerance (BFT) mechanisms [72, 280] once a quorum certificate – a cryptographic attestation by at least two thirds of the validators – is formed, in addition to hybrid designs. In permissionless P2P settings, PoW was introduced to address the double-spending problem without a trusted central operator: by tying block production to costly work and adopting the longest/heaviest-chain rule, rewriting history to spend the same coins twice becomes economically prohibitive unless an adversary controls a majority of the mining power [235]. This security model yields probabilistic finality: the probability of reversal (and thus of successful double spending) decreases as confirmations accumulate. The aforementioned combination of a consensus mechanism and a P2P overlay affects security assumptions, performance outcomes (block-confirmation latency, transaction throughput), and the energy profile.

Real-world public blockchain networks are geo-distributed (with validator and full-node footprints across continents [50, 52, 124]) and operate over heterogeneous paths with diverse per-link latency and bandwidth [3]. Nodes run on varied hardware – bare-metal servers, Virtual Machines (VMs), containers – where frequency scaling and multi-tenancy may introduce performance variability through shared caches and Input/Output (I/O) contention [5, 204, 257, 266]. In practice, most blockchain systems assume partial synchrony: after an unknown stabilization time, message delays become bounded [64, 114]. These factors shape propagation, block production, and the queueing and scheduling effects observed in measurements.

Many blockchains also add programmability through smart contracts, on-chain programs that execute deterministically when triggered by transactions meeting specified preconditions, thereby updating a shared global state [349]. The combination of programmability, verifiable auditability, and open participation has catalyzed adoption across diverse application domains that include cryptocurrencies (native digital assets for P2P payments and protocol incentive mechanisms, *e.g.,* Bitcoin and Ethereum [235, 331]), digital payments and stablecoins [33, 89], tokenization of real-world assets [127, 168, 234], Decentralized Finance (DeFi) [29, 216], supply chain traceability [275, 348], digital identities [351], healthcare [78], Internet of Things (IoT) [286], Function as a Service (FaaS) platforms [186], and governance mechanisms [332]. This expansion is accompanied by hundreds of active blockchains and a growing number of networks and assets: as of December 2025, major aggregators list about 19,200 coins and tokens, with the total crypto-market capitalization being about 3.2 trillion U.S. Dollars (USDs) [85].

An indication of the technological and economic advancements reached by blockchain applications is the increasing attention from regulators and institutional investors, as evidenced by: (*i*) the 2024 approval of Bitcoin (BTC) and Ethereum coin (ETH) Exchange-Traded Funds (ETFs)[1] [142] by the U.S. Securities and Exchange Commission (SEC)[2],

[1] ETFs constitute a type of investment fund traded on stock exchanges, which holds assets such as stocks, commodities, or cryptocurrencies. BTC and ETH ETFs allow traditional investors to gain exposure to these assets without directly holding or managing them.

[2] The SEC is the U.S. federal agency responsible for enforcing securities laws and regulating the securities industry, including stock and options exchanges. Its approval is often seen as a signal of institutional

pushing BTC over 100,000 USD [61]; (*ii*) the recent statements by Larry Fink, Chief Executive Officer (CEO) of BlackRock[3], in his 2025 annual letter to investors, in which he says "*What exactly is tokenization? It is turning real-world assets – stocks, bonds, real estate – into digital tokens tradable online*" [135]; and (*iii*) the anticipated use of stablecoins backed by the USD – like USD Coin (USDC) [82] and USD Tether (USDT) [310] – as the digital version of the USD itself, which can play the role of a Central Bank Digital Currency (CBDC) [293].

For end users, blockchain systems provide: lower barriers as opposed to trusted intermediaries or institutional access; self-custody of digital assets; programmable money and assets with policy-enforced constraints; cross-border P2P value transfer with daily availability and transparent fees; permissionless participation with only a smartphone and an Internet connection, thereby facilitating "*banking the unbanked*" [277]; fractional and programmable ownership of assets; composability of services (contracts invoking one another to create higher-level functionality); verifiable provenance and selective disclosure of credentials; and, depending on the governance model, varying degrees of censorship resistance or accountable control. These affordances widen the design space for financial and non-financial applications, while shifting some operational and security responsibilities to the network edge.

1.2 How to Compare Blockchains

A meaningful comparison treats a blockchain as a stack whose behavior emerges from the interaction of system logic, networking, and infrastructure, as well as the economic layer that drives demand. The aim is not a single headline number, but an agreeable mapping from assumptions to outcomes – performance, energy, and economic efficiencies – under a clear methodology that makes experimental parameters explicit and controllable.

A correct comparison must define the environment and uniformize semantics across systems. This includes network topology[4] and routing, latency and bandwidth, hardware blend, node density, transaction mix, block and gas[5] limits, and thresholds for confirmation and finality. Metrics should report technical performance – throughput in terms of Transactions Per Second (TPS) as well as end-to-end and block latency – and resource efficiency – energy per transaction expressed in Kilowatt-hour (kWh). The methodology should be transparent and repeatable: pinned software versions and configurations, traceable datasets, open pipelines, and consistent instrumentation. Finally, the economic layer should acknowledge diversity across assets, rather than price alone.

However, the growing heterogeneity of blockchain systems, consensus designs, and token economies complicates thorough evaluation. What was once a binary choice between Bitcoin and Ethereum has evolved into an ecosystem with different security assumptions,

acceptance and regulatory maturity.

[3] BlackRock is the world's largest asset management firm, offering investment and risk management services to institutional and retail clients globally. See: https://www.blackrock.com/.

[4] The structural arrangement of network nodes and their interconnections.

[5] Unit that quantifies metered computations and fees on Ethereum-like blockchains.

execution models, and usage patterns. Aligning workloads, semantics, and operating conditions across systems is difficult. The literature – covering performance, energy, decentralization, governance, cross-chain interoperability, and economic efficiency – is rich, yet methodologically inconsistent: system-level metrics like TPS and block latency are often reported in isolation and under synthetic workloads that deviate from real usage [144]; energy studies [164, 279] typically focus on consensus mechanisms while overlooking deployment factors such as topology, hardware mix, and node density; economic indicators like price dynamics [187], token velocity [103], and wealth distribution [276] capture emerging phenomena but remain model-dependent and chain-specific. Even with deterministic implementations and well-defined assumptions, experimental outcomes often exhibit non-negligible run-to-run dispersion: results are not always repeatable – consistent reproduction under fixed conditions – nor predictable – stable performance as controlled factors (*e.g.,* network size) vary in prescribed ways.

This fragmentation extends to methodology. Benchmarking frameworks abstract away the heterogeneity and unpredictability of real deployments. Financial analytics platforms like CoinGecko [85] and DeFiLlama [105] track capital, usage, and risk, yet are largely decoupled from node-level performance and network telemetry exposed by technical dashboards like ETHStats [120] and Solana Beach [300]. Empirical analyses based on network emulation, live monitoring, or case studies are more grounded but face repeatability challenges due to inconsistent experimental setups, heterogeneous infrastructures, limited observability, and non-standardized data collection. As a result, findings across studies are difficult to reconcile and operational insight is hard to extract.

Researchers therefore combine complementary approaches. Benchmarking tools like BlockBench [113] and Diablo [154] facilitate controlled evaluations of throughput and latency, isolating system behaviors under synthetic workloads. When targeting geo-distributed, real-world deployments, public cloud platforms like Amazon Web Services (AWS) [18] become an integral part of both the experimental setup and the benchmarking toolchain: they offer elastic scale, a global multi-region footprint, and production-like network conditions that are difficult to reproduce on-premise. This realism, however, couples results to best-effort Wide Area Network (WAN) dynamics and opaque multi-tenant effects – variable cross-traffic, shifting routes, time-of-day congestion, heterogeneous VM generations – which complicate attribution and erode run-to-run repeatability and performance predictability. Financial dashboards like DeFiLlama [105] and CoinGecko [85] track economic indicators like token movements, Total Value Locked (TVL)[6], and user activity, providing insight into the financial dynamics of the ecosystem. Real-time monitoring platforms (*e.g.,* ETHStats [120] and Solana Beach [300]) and emulation frameworks (*e.g.,* BlockEmu [172] and, more generally, container-based testbeds that use Docker or Kubernetes) can approximate real-world conditions while maintaining experimental control. Together, these efforts provide a richer view of blockchain systems, yet coordination, standardized workloads, and reproducible pipelines remain limited.

[6]The USD value of assets deposited in a protocol's smart contracts.

Bridging these gaps – by combining benchmarking precision with empirical realism – remains a key challenge.

We thus advocate a unifying, lightweight evaluation approach that: (*i*) treats the environment as a first-class parameter by reporting and, where possible, controlling topology, per-link latency/bandwidth, routing, and resource caps to approximate partial synchrony; (*ii*) standardizes workload families that reflect realistic demand and semantics; (*iii*) reports the exact experimental conditions and summary statistics (not just the mean) over repeated, independent runs, making run-to-run dispersion explicit; (*iv*) instruments energy alongside performance; (*v*) releases data and code for independent verification and longitudinal analysis; and (*vi*) combines heterogeneous economic indicators into a single, interpretable balance index rather than a price-based evaluation. These principles, instantiated through the benchmarking framework and the multi-run dataset developed in this dissertation, enable coherent comparisons without prescribing a single toolchain.

1.3 Contributions of the Thesis

This dissertation investigates the efficiency of blockchain systems in terms of performance, energy consumption, economic balance, and experimental repeatability plus performance predictability, with a specific focus on quantifying run-to-run variability.

Blockchain efficiency is frequently invoked, yet seldom defined with precision. Unlike traditional domains (*e.g.,* mechanical engineering or classical economics), where efficiency is a measurable relation between inputs and outputs, blockchain systems span overlapping layers of computation, resource expenditure, economic design, and emergent system dynamics. In such a context, efficiency cannot be reduced to a single performance indicator (*e.g.,* TPS) or to narrowly scoped experiments.

To address this gap, we propose a comprehensive and reusable methodology for evaluating efficiency in blockchain ecosystems that integrates technical, energetic, and economic factors into a context-first perspective grounded in operational realism. Heterogeneity – both systemic and economic – is treated as intrisic to deployments. Likewise, the evaluation process is structured to enable empirical insight, cross-platform comparison, and analytical depth. We examine these dimensions under controlled conditions designed to emulate real-world complexity, with particular attention to how external parameters – like network topology and heterogeneous economic attributes – influence internal dynamics and observable outcomes. The central research question is about the efficiency of blockchain systems when evaluations account for context-specific, real-world factors – related to both economic aspects and technical deployment conditions – rather than being optimized for idealized settings.

In this dissertation, efficiency links technical behavior to operating conditions along four complementary axes: (*i*) performance in terms of throughput and latency; (*ii*) energy consumption at node and system level; (*iii*) experimental repeatability (the ability to reproduce outcomes under identical experimental conditions [4]) and performance predictability (how well performance can be anticipated from a given execution configuration [307]);

and (*iv*) economic balance of the surrounding ecosystem.

Each analytical dimension maps to a distinct research contribution:

1. The design of Lilith, an experimental benchmarking framework for blockchain systems capable of emulating network topologies and executing configurable workloads, and its application to assess performance [108].

2. The use of Lilith to analyze energy consumption, so as to measure performance-energy trade-offs under realistic deployments [110].

3. The construction (with Lilith) of a network-controlled, multi-run dataset and accompanying analysis templates that quantify experimental repeatability and performance predictability by applying dispersion metrics (*e.g.,* worst-case deviation) and variance-aware techniques (analysis of variance and intraclass correlation) to separate configuration effects from exogenous noise [111].

4. The introduction of the Entropy Balance index (EB-index) – where entropy is a concept borrowed from information theory [287] that measures the uncertainty or disorder within a system – which aggregates heterogeneous economic indicators into a single entropy-derived measure of balance in crypto-asset ecosystems [109].

All results specify software versions, configurations, topologies, workloads, node-set size, and number of runs. Metrics are reported not only as means but also with dispersion summaries across repeated trials under identical conditions and, where applicable, with variance decompositions. We release code and datasets too, spanning multiple blockchains, versions, topologies, and workloads – with repeated trials – to enable independent verification and longitudinal analysis.

Our experiments emphasize controlled environments, particularly fixed software stacks, and emulated network conditions (*e.g.,* latency, bandwidth) that support repeatable, multi-run measurements. We do not model adversarial strategies in depth nor provide formal security proofs. Long-term state growth and software drift are considered when feasible, isolated where possible, and both are avenues for extended study. The focus is on evidence that links the considered mechanisms to measured variability under explicit, repeatable conditions.

1.4 Structure of the Thesis

This dissertation is organized as follows:

- Chapter 2 surveys blockchain features and families. It introduces operating conditions (overlay networking, topologies, routing) and workload semantics, outlines benchmarking methods for performance, energy, and economic evaluations, and highlights the fragmentation and methodological gaps that motivate the dissertation.

- Chapter 3 presents the Lilith benchmarking framework, a modular suite that emulates overlay/network topologies and resource caps. It executes standardized,

VM-agnostic workloads, orchestrates repeated runs under fixed configurations, and instruments end-to-end metrics (throughput, latency/finality, variance, and node/system-level energy). This chapter is based on our prior work in [108, 110].

- Chapter 4 illustrates LILITH at work to evaluate the performance of five blockchains – Algorand [76, 147], Diem [38], Ethereum Clique [308], Quorum Istanbul BFT (IBFT) [117], and Solana [342] – across five topologies – fat-tree, full mesh, hypercube, scale-free, and torus – under transfer transactions and smart contract requests – DDoS, FIFA, GAFAM, gaming, PayPal, and VISA – for two node-set sizes – 10 and 40. We show how topology and workload shape throughput and block latency and reveal bottlenecks overlooked by previous studies. This chapter is based on our prior work in [108].

- Chapter 5 extends the analysis to energy consumption by examining the interplay between network topology and workload composition. The results uncover non-trivial trade-offs between performance and energy efficiency, offering insights for designing energy-aware blockchain infrastructures. This chapter is based on our prior work in [110].

- Chapter 6 investigates experimental repeatability and performance predictability through a network-controlled, multi-run dataset of independent benchmarks across predefined experiments and configurations. It introduces dispersion metrics and variance-aware analysis (including analysis of variance and intraclass correlation) to characterize run-to-run variability, identify stable and volatile blockchain-topology pairs, and determine which controls are required for empirical stability. This chapter is the basis for a manuscript that is currently under review [111].

- Chapter 7 introduces the EB-index and applies it to six major cryptocurrencies – BTC [235], ETH [331], XRP (Ripple coin) [74], USDC [82], DOGE (Dogecoin) [220], ADA (Cardano coin) [170] – to characterize ecosystems through an interpretable notion of economic balance. This chapter is based on our prior work in [109].

- Chapter 8 summarizes the main contributions and outlines future research directions, emphasizing the need for multidimensional and rigorous benchmarking methodologies to support the comparison of blockchains based on different notions of efficiency.

Chapter 2

Background

This chapter outlines the foundational concepts of blockchain technology, including core principles, consensus mechanisms, and the rise of decentralized applications (§2.1). It then deepens the analysis along three pillars that shape (and confound) any evaluation: (i) overlay network topologies and message propagation (§2.2), (ii) workloads and their semantics (§2.3), and (iii) benchmarking tools and methods (§2.4). The chapter closes with an overview of the main benchmarking challenges (§2.5) – such as network-induced variability, workload opacity to non-portable toolchains, limited reproducibility, and the absence of a theoretically grounded, aggregation-ready economic measure for heterogeneous inputs – thus motivating the rigorous methodology proposed in the subsequent chapters.

2.1 Blockchains: Features and Classification

Before blockchains, mainstream digital finance and online infrastructure relied on centralized stacks that anchored trust in institutional intermediaries rather than protocol-level consensus. In these architectures, a single administrative domain – a bank, a government agency, or a large platform provider – controlled validation, data custody, and transaction processing, set access policies, and acted as the final arbiter of correctness and availability. This model delivers operational efficiency and simplified governance, yet concentrates control and creates critical vulnerabilities: single points of failure (*e.g.,* centralized servers targeted by cyberattacks, as in the Equifax data breach [189]), limited transparency (*e.g.,* opaque financial operations and audits [153]), censorship risk (*e.g.,* services or transactions suppressed by public or private actors [177]), and power asymmetries that can undermine user trust and system resilience (*e.g.,* the 2008 financial crisis, where a few institutions' decisions had global consequences [63]).

Advances in cryptography, networking, and distributed computing enabled blockchains as a foundation for the design of decentralized systems. This technology was conceived to address structural limitations of centralized infrastructures, offering a trust-minimized alternative that aims for transparency, integrity, and resilience without relying on a central authority. At its core, a blockchain is a distributed ledger: a replicated, append-only database shared across a network of nodes that need not trust one another. This trust

minimization follows from cryptographic primitives (*e.g.,* hash functions [271], digital signatures[1] [151], Merkle trees[2] [231]) as well as consensus mechanisms (distributed, computationally intensive agreement processes executed by nodes to validate transactions and keep the ledger consistent [335]). Together, these elements ensure that all honest participants converge on a consistent view of the ledger's state.

This architecture enables new socio-technical paradigms – permissionless access to data and services, P2P financial rails, and forms of decentralized governance – and reframes assumptions about trust and coordination in distributed environments.

2.1.1 Toward Algorithmic Consensus

Consensus is the system layer that establishes agreement on the ledger state in potentially adversarial networks. Unlike traditional replicated databases that rely on trusted coordinators, blockchains embed decentralized agreement mechanisms into the network logic to achieve state consistency. This approach enables trustless coordination among nodes that may not know or trust each other, making consensus the core enabler of blockchain's decentralization.

In a typical blockchain system, users submit transactions to request specific actions such as asset transfers or updates to application state. These transactions are broadcast to the network and collected by nodes participating in consensus. Participants validate transactions against system rules and agree on the next valid state of the ledger. Once agreement is reached, transactions are ordered and bundled into a block – or, in Directed Acyclic Graph (DAG)-based systems, into vertices – together with the necessary validity proofs; the resulting state transition is appended to a tamper-evident history.

Several consensus families have emerged, each with distinct assumptions, security models, and trade-offs:

- **Proof of Work (PoW)** [235] relies on computational effort. In PoW systems, consensus participants are typically called *miners*: they compete to solve cryptographic puzzles in order to append new blocks and claim rewards. PoW offers strong resilience to Sybil attacks due to resource-based voting[3], but consumes massive energy[4] and tends toward centralization as mining power concentrates in large pools.

- **Proof of Stake (PoS)** [244] selects consensus participants according to the amount of tokens they stake. In these systems, they are typically called *validators*: depending on the protocol, they may be selected to propose blocks, attest to them, or both. PoS is far less energy-intensive, yet introduces new risks. The so-called *nothing-at-stake*

[1] Digital signatures let a user sign a message with a private key; the corresponding public key suffices for anyone to verify authenticity and integrity.

[2] Merkle trees are binary trees in which each non-leaf node contains the hash of its two child nodes. This hierarchical structure enables efficient and secure verification of the contents of large datasets, as only a logarithmic number of hashes are needed to prove membership.

[3] PoW security assumes that acquiring a majority of the global hash power is prohibitively expensive, thereby making Sybil attacks – where one adversary uses many identities to gain influence – costly [62].

[4] Bitcoin mining in 2021 consumed nearly six times the amount of energy it did in 2017, matching the annual energy consumption of countries like Finland and Argentina [182].

problem arises in PoS systems when validators can propose blocks on multiple competing forks without incurring any penalty [62]. Without effective slashing mechanisms – which involve the loss of part or all of a validator's staked funds as a penalty for malicious behavior or unavailability [165] – rational validators may support all available forks to maximize potential rewards. Vulnerabilities chiefly concern naive longest-chain PoS variants; modern designs adopt slashing and checkpointing to provide economic finality after a small number of epochs. PoS also risks centralization due to wealth concentration among large stakeholders [180].

- **Proof of Authority (PoA)** [236] and related variants (*e.g.,* Clique [308], Aura [181]) rely on a fixed, permissioned set of *validators*, whose identities are known in advance and admitted according to organizational or trust-based criteria. Used primarily in private or consortium blockchains (*e.g.,* Hyperledger Fabric [23], Quorum [117]), they offer high transaction throughput and low block latency, but at the expense of decentralization since validator identities are predetermined.

- **Byzantine Fault Tolerance (BFT)** algorithms, such as Practical BFT (PBFT) [72], rely on a bounded set of known participants – commonly called *validators* or *replicas* – that exchange messages to agree on transaction ordering. They provide deterministic finality[5] and operate correctly if fewer than one-third of nodes are malicious. They are well-suited to permissioned systems with modest network size and low latency.

- **Directed Acyclic Graph (DAG)** mechanisms, as seen in IOTA [179] or Avalanche [284], which employs repeated subsampled voting with a DAG used for ordering/propagation, organize transactions in a DAG instead of a linear chain. This structure allows parallel validation and improved transaction throughput.

Table 2.1 provides a comparative overview of the above consensus mechanisms in blockchain systems, evaluating them across five core dimensions: finality type, energy consumption, decentralization, scalability, and adoption context. These criteria help highlight the trade-offs between security guarantees, performance efficiency, and architectural design choices. While PoW remains dominant in high-security public blockchains, PoS offers more scalable and energy-efficient alternatives. Meanwhile, BFT and DAG-based models cater to permissioned or high-throughput scenarios with different decentralization assumptions.

To illustrate how consensus mechanisms shape the architecture and operational roles within blockchain networks, we describe two widely deployed paradigms: PoW (Bitcoin) and PoS (Ethereum).

[5] Finality is guaranteed as soon as consensus is reached, with no chance of forks – providing strong safety but limited scalability due to $O(n^2)$ message complexity in classical mechanisms [22]; modern variants (Tendermint, HotStuff) streamline rounds and pipelining.

Table (2.1) Comparison of major blockchain consensus mechanisms across key operational and architectural dimensions: finality, energy consumption, decentralization, scalability, and common deployment contexts (expressed in qualitative terms, not via measures).

Consensus	**Finality Type**	**Energy Use**	**Decentralization**	**Scalability**	**Adoption Context**
PoW	Probabilistic	Very High	Low–Medium (mining pools)	Medium (slow block times)	Public blockchains with high security needs
PoS	Family-dependent: probabilistic in chain-based PoS; (economic) finality via checkpoints/quorum in BFT-style PoS	Low	Medium (depends on stake distribution)	High (faster confirmations; finality mechanism-dependent)	Public and hybrid blockchains
PoA	Deterministic	Low	Low (fixed validators)	High (low latency, high throughput)	Consortium and private blockchains
PBFT	Deterministic	Low	Medium (assumes $< 1/3$ faults)	Low–Medium (classical $O(n^2)$; streamlined in modern variants)	Permissioned settings
DAG-based systems	Probabilistic or Adaptive	Medium	Protocol-dependent (some centralized coordination)	High (parallel processing)	IoT, DeFi, and experimental base-layer blockchain systems

Proof of Work (PoW)

Bitcoin's consensus model is grounded in a permissionless and resource-intensive process where distinct node types coordinate to maintain the ledger. Full nodes verify blocks and transactions; miners compete to find a valid nonce such that the double Secure Hash Algorithm (SHA)-256 hash of the candidate block header falls below the network difficulty target [235]. Upon discovery, the block is propagated and independently validated by peers; if accepted, the miner earns a block reward – the block subsidy (newly minted BTC) plus all transaction fees from the transactions included in the block – via a special coinbase transaction. Coinbase outputs become spendable only after 100 additional blocks have been added on top of the block that contains them [45]. As of April 2024 halving (block 840,000), the block subsidy is 3.125 BTC and is programmed to halve every 210,000 blocks ($\approx$ 4 years) [46]. In practice, this mechanism has driven significant hardware

specialization, with mining dominated by Application-Specific Integrated Circuit (ASIC)[6] deployed in large-scale facilities, often in regions with low electricity costs [305]. Mining pools have further consolidated coordination power, with a small number of pool operators accounting for most mined blocks [128, 141]. While each pool aggregates hash power from many independent miners, block production is effectively coordinated by a small set of pool operators (*e.g.,* via block-template construction and payout policies), raising concerns over the system's practical decentralization.

Proof of Stake (PoS)

Ethereum post-Merge PoS mechanism [274] eliminates mining (PoW) in favor of stake-weighted validator duties coordinated by the Beacon Chain [345]. Time is divided into 12-second *slots*, grouped into *epochs* of 32 slots each [125]; validators are selected pseudorandomly to propose new blocks and to attest to others' proposals, weighted by the amount of ETH staked. Honest behavior is rewarded with small ETH incentives, while misconduct or extended inactivity results in penalties or slashing. Unlike Bitcoin, which relies on external hardware investments, Ethereum's PoS encourages capital-based participation. However, in practice, much of the staked ETH is controlled by a handful of liquid staking platforms and exchanges, raising concerns about validator concentration and governance power [99]. Short-term confirmations are probabilistic; checkpoints provide economic finality after a small number of epochs.

These two examples reflect the ongoing tension in consensus design between performance, security, and decentralization – often referred to as the *blockchain trilemma* [138]. While consensus mechanisms aim to strike a balance among these objectives, no single approach optimally satisfies all three in practice. More broadly, the defining innovation of blockchain lies in how consensus is embedded into the protocol layer itself.

2.1.2 A Decentralized Perspective

Beyond technical implementation details, blockchains change how we conceive and govern digital systems. By removing intermediaries and enabling collective verification, blockchains support transparent-by-design applications whose auditability and censorship-resistance can be verified from system artifacts.

Originally conceptualized in the early 1990s as a method for ensuring the integrity of digital documents through cryptographically linked timestamps, blockchain-like structures were first proposed by S. Haber and W. S. Stornetta [160]. Their work introduced the idea of chaining blocks of timestamped records to prevent backdating or tampering. This concept was later extended with the incorporation of Merkle trees [231], enhancing efficiency and data verification, laying the foundation for blockchain's core design. However, it was not until the publication of the Bitcoin whitepaper by Satoshi Nakamoto in 2008 [235]

[6] ASICs are custom-designed hardware optimized for specific tasks – in this case, solving the SHA-256 hash puzzle used in Bitcoin's PoW. Their superior efficiency over general-purpose Central Processing Units (CPUs) and GPUs has led to the industrialization and geographic concentration of mining operations.

and the subsequent launch in 2009 that the first fully operational blockchain system was deployed, combining incentive-compatible consensus (PoW), P2P networking, and an endogenous asset (BTC).

Since then, blockchain has evolved into a versatile infrastructure powering a broad spectrum of Decentralized Applications (DApps) – general-purpose smart-contract platforms. Its key properties – security, transparency, immutability, and accountability – make it particularly suited for applications where tamper-resistance and auditability are essential.

This shift is not merely technical, but institutional and social. By enabling open participation and removing central points of failure, blockchain systems support alternative models of governance, value exchange, and coordination that challenge traditional organizational structures. As a result, entire ecosystems of decentralized alternatives have emerged [216], redefining foundational aspects of how digital systems operate.

One of the most prominent domains enabled by this new paradigm is DeFi, which replaces traditional financial intermediaries with algorithmic protocols running on blockchains [216]. Users can interact with Decentralized Exchanges (DEXs) – such as Uniswap [7] – to trade tokens directly from their wallets, without intermediaries or centralized order books. These platforms rely on Automated Market Maker (AMM) algorithms [24] to determine pricing and provide liquidity.

Lending and borrowing protocols, such as AAVE [1] and Compound [161], enable users to supply liquidity and earn interest, or to borrow funds by locking up collateral. These platforms use smart contracts to enforce collateral ratios, interest rates, and liquidations without intermediaries, ensuring transparency and security.

Stablecoins, including (algorithmic and crypto-collateralized) DAI [285] and (fiat-backed) USDC [82], provide price-stable assets crucial for day-to-day transactions and risk hedging. By enabling programmable finance[7] on-chain, stablecoins facilitate a wide range of DeFi services such as payrolls, remittances, and algorithmic financial instruments [285].

Blockchain also supports new forms of decentralized governance, where decisions regarding protocol upgrades, fund allocation, and policy changes are made collectively through on-chain voting mechanisms [332]. In Decentralized Autonomous Organizations (DAOs) – which are self-governed communities structured around smart contracts – token holders can propose, deliberate, and vote on decisions in a transparent and tamper-proof environment [322]. Governance processes are encoded directly into the protocol, reducing reliance on centralized leadership and introducing new, bottom-up models of collective coordination [322].

Digital identity systems, often siloed and fragmented in traditional infrastructures, can be restructured on blockchain by using verifiable credentials and Self-Sovereign Identity (SSI) frameworks [264]. Users gain control over their data and can selectively disclose information to third parties. Reputation systems and attestations can also be maintained transparently on-chain, enabling trust mechanisms in anonymous or pseudonymous

[7] Programmable finance refers to the ability to encode financial logic – such as conditions for transfers, interest accrual, or automated compliance – directly into digital assets or smart contracts.

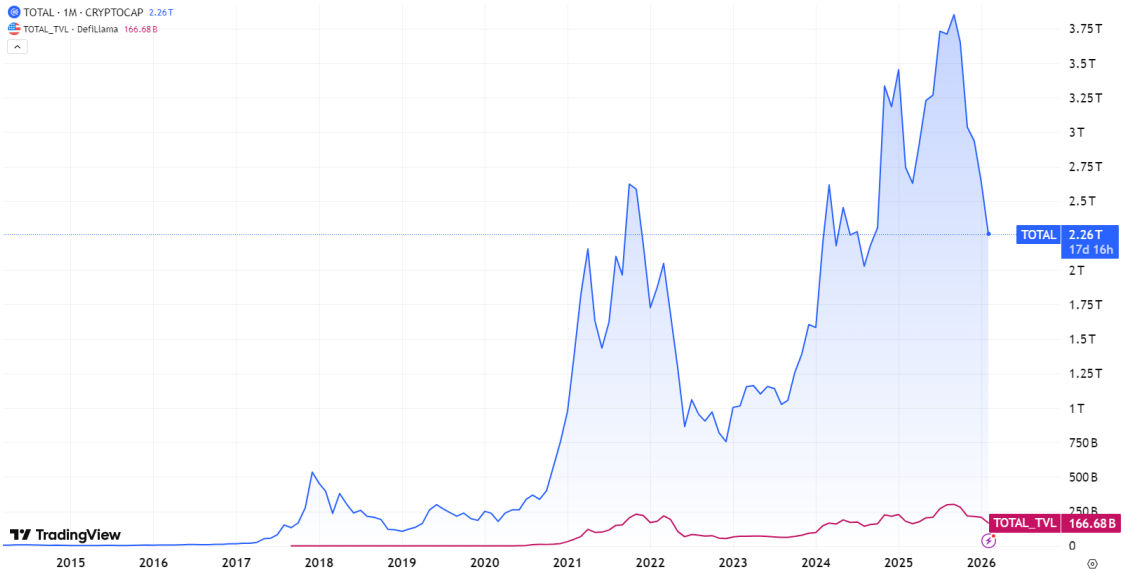


Figure (2.1) Historical evolution of cryptocurrency market capitalization and DeFi TVL, illustrating the divergence between speculative asset valuation and protocol-level engagement. Source: TradingView at https://www.tradingview.com/chart/

environments.

Beyond finance, blockchain is being adopted for content publishing, asset tokenization, and decentralized storage. Platforms like InterPlanetary File System (IPFS) [290] and Filecoin [265] distribute data storage across peers, while Non-Fungible Token (NFT)-based systems[8] enable ownership and provenance tracking for digital assets related to art, music, media, collectibles, and virtual goods [102]. Infrastructural services – such as computation (*e.g.,* Golem [152]), Domain Name System (DNS) (*e.g.,* Handshake [163]), or social platforms (*e.g.,* Steemit [202]) – are progressively shifting to decentralized, composable architectures.

As outlined in Chapter 1, the blockchain ecosystem has expanded rapidly. Figure 2.1 provides an overview of the evolution of both market capitalization and TVL in recent years, two widely used metrics in the blockchain space. While market capitalization represents the total value of a crypto-asset – calculated as price multiplied by circulating supply – and reflects investor sentiment and speculative interest, it does not necessarily indicate real usage. In contrast, TVL measures the capital actually committed within DeFi protocols, offering a clearer view of user participation and protocol adoption.

In essence, this new decentralized perspective enabled by blockchains redefines not only the mechanics of transaction validation but also the underlying assumptions of trust, authority, and coordination. It challenges established power structures by redistributing control, fostering censorship resistance, and embedding transparency into the core logic of digital systems. As these systems mature, their influence extends far beyond technology, shaping the future of institutional design and human cooperation.

[8] An NFT is a unique, indivisible digital token recorded on a blockchain, typically used to certify ownership and authenticity of digital or physical assets. Unlike fungible tokens (*e.g.,* cryptocurrencies), each NFT has distinct metadata and value.

2.1.3 Key Dimensions of Blockchain Systems

To navigate the contemporary blockchain landscape, we adopt a five-axis taxonomy that is widely used in research and engineering practice [336, 338]. These axes – permissioning, state and execution, consensus and finality, data structure and layering, governance and economics – separate concerns cleanly while making explicit the trade-offs that drive real-world designs.

Permissioning. Permissionless networks (*e.g.,* Bitcoin, Ethereum) allow open participation for nodes and users; Sybil-resistance is enforced by economic or resource assumptions rather than identity vetting. This openness maximizes neutrality and censorship resistance but complicates throughput and latency due to wide-area propagation and adversarial participation [235, 338]. By contrast, permissioned and consortium settings (*e.g.,* Hyperledger Fabric, Quorum) restrict validator identities and access policies to known organizations; narrower fault models enable faster deterministic finality and richer privacy policies at the cost of weaker openness and often lower decentralization [338].

State and execution model. Two dominant models are the Unspent Transaction Output (UTXO) abstraction and the account-based model. In UTXO systems (exemplified by Bitcoin), the ledger is a set of unspent transaction outputs; each transaction consumes prior outputs and creates new ones under spending conditions [235, 338]. Account-based systems (exemplified by Ethereum) maintain a global state mapping accounts to balances, nonces, and contract storage; transactions mutate this state through an execution environment – the Ethereum VM (EVM) – metered by gas [331]. While account semantics enable general-purpose smart contracts and composability, shared mutable state increases contention and complicates concurrent execution; modern clients mitigate this with transaction ordering rules, state access lists, and parallel schedulers where safe.

Consensus and finality. Consensus choices span resource-based (PoW), capital-based (PoS), identity-based (PoA), and classical BFT mechanisms. PoW achieves probabilistic finality through longest-(heaviest-) chain selection and difficulty adjustment; security hinges on honest majority of hashing power [235]. PoS replaces resource expenditure with stake. Identity-based PoA relies on small validator sets for low-latency confirmations, typically in consortium deployments. Classical BFT (*e.g.,* PBFT, Tendermint, HotStuff) offers deterministic finality under the standard fault threshold $f < n/3$, where n is the total number of participating replicas and f is the maximum number of tolerated Byzantine faults, trading scalability for strong safety guarantees; message complexity has improved from quadratic (PBFT) to streamlined pipelining (HotStuff) [64, 347]. Algorand's Binary Byzantine Agreement (BBA*) illustrates committee-based BFT with Verifiable Random Function (VRF) leader/committee selection to scale participation while preserving safety [76, 147].

Data structure and layers (modularity). Traditional ledgers are linear chains of hash-linked blocks; alternatives adopt DAG structures in which transactions or micro-blocks reference multiple predecessors to increase concurrency (*e.g.,* IOTA's Tangle) [263]. Orthogonal to the ledger data structure, modern ecosystems decompose responsibilities into layers – roles in the stack for consensus/settlement, data availability, and execution. In this thesis we introduce the following terminology. Layer 0 (L0) denotes multi-chain coordination substrates that provide interoperability and, in some designs, shared security (*e.g.,* Polkadot's relay chain finalizing parachains [65]; Cosmos zones communicating via IBC [95]). Layer 1 (L1) denotes the base ledger that provides consensus and finality over canonical state and often data availability (*e.g.,* Bitcoin, Ethereum). Layer 2 (L2) domains execute transactions off-chain (or in separate domains) and periodically post commitments plus data to L1 to inherit its security (*e.g.,* rollups such as Arbitrum/Optimism; Polygon PoS and Polygon zkEVM provide sidechain/rollup options).

Governance and economics. Token supply policies, fee markets, staking and slashing rules, treasury mechanisms, and on-chain governance processes (DAOs) jointly shape incentives, decentralization, and the security condition in practice [338]. Native tokens fund block production (subsidies/fees), align validator behavior (stake, slashing for safety/liveness faults in PoS/BFT), and determine resource pricing for computation and storage; governance frameworks coordinate protocol upgrades and parameter changes, spanning off-chain rough consensus to fully on-chain voting systems. Economic choices feed back into technical properties: fee market design affects transaction inclusion latency and Maximal Extractable Value (MEV) surface; staking distribution impacts Nakamoto-style decentralization metrics and fault tolerance.

Taken together, these axes offer an analytical framework for examining the rise and adoption of blockchain technologies across a wide range of applications and use cases through five interrelated dimensions: financial inclusion, massive asset proliferation, ecosystem expansion, global capital lock, and state-level adoption.

Financial inclusion. Blockchain enables access to financial services for individuals traditionally excluded from formal banking systems, particularly in underbanked or politically unstable regions [277]. Through P2P systems and mobile wallets, users can store value, send remittances, or access lending and savings tools without relying on centralized financial institutions. This vision – commonly framed as "banking the unbanked" [277] – has driven significant interest in blockchain applications in several regions such as Sub-Saharan Africa and Southeast Asia.

Massive proliferation. The cryptocurrency landscape has experienced exponential growth (see Chapter 1). To give readers an immediate sense of scale, Table 2.2 situates Bitcoin's market value alongside mega-cap technology firms in a cross-asset ranking [90]. As of December 2025, Bitcoin appears in the global top tier: it ranks 9th worldwide by

Table (2.2) Top 10 assets by market capitalization (USD). Source: CompaniesMarketCap, *Assets by Market Cap* (accessed Dec 2025); values are live and may fluctuate.

Rank	Name (Ticker)	Market cap (USD)
1	Gold (GOLD)	$29.414T
2	NVIDIA (NVDA)	$4.441T
3	Apple (AAPL)	$4.137T
4	Alphabet/Google (GOOG)	$3.888T
5	Microsoft (MSFT)	$3.591T
6	Silver (SILVER)	$3.302T
7	Amazon (AMZN)	$2.453T
8	Broadcom (AVGO)	$1.842T
9	**Bitcoin (BTC)**	**$1.825T**
10	Meta Platforms (META)	$1.697T

market capitalization, close to Amazon and Broadcom, and below the very largest assets (*e.g.,* gold, Nvidia, Apple, Alphabet, Microsoft, silver).

This explosion of tokens and digital assets traces back to the launch of Bitcoin and has since expanded across diverse use cases and economic models. These assets differ in technical properties and economic roles:

- *Native tokens* – such as ETH [331], Solana coin (SOL) [342], ADA [170] – power underlying blockchains and pay for transaction fees.
- *Utility tokens* – such as Basic Attention Token (BAT) [58] or Chainlink coin (LINK) [59] – grant access to platform-specific services or data feeds.
- *Governance tokens* – such as Maker (MKR) [193] and Uniswap (UNI) [7] – give holders voting rights over protocol upgrades and treasury management.
- *Stablecoins* – such as USDC [82], USDT [310] – and algorithmic variants – like DAI coin (DAI) [285] – aim to maintain stable value against fiat currencies and are widely used in DeFi and cross-border payments.
- *NFTs* represent unique digital assets and are used in digital art (*e.g.,* CryptoPunks [211], Bored Ape Yacht Club [199]), gaming (*e.g.,* Axie Infinity [106]), and intellectual property rights [321].

Expanding ecosystem. Blockchain's scope has expanded beyond cryptocurrencies and to a growing ecosystem of decentralized applications, becoming increasingly integrated with emerging technologies:

- In Artificial Intelligence (AI) – which refers to computational systems capable of performing tasks that typically require human intelligence, such as pattern recognition, decision-making, or language processing – blockchain provides traceability and auditability for training datasets and model outputs, as in Ocean Protocol [245] or Fetch.AI [134].

- In IoT, where a network of interconnected physical devices – such as sensors, wearables, and smart appliances – can authenticate and coordinate autonomously using lightweight blockchains (*e.g.,* Helium [167], IOTA [179]).

- In Edge Computing, which involves processing data closer to the source (*e.g.,* sensors or mobile devices), reducing latency and bandwidth usage. Here, blockchain contributes by ensuring data integrity, provenance, and verifiable execution at the edge, as explored in systems like Edge&Fog [249].

Global capital lock. The proliferation of DeFi protocols and tokenized ecosystems has led to substantial on-chain capital accumulation. A key metric in the evaluation of decentralized finance is the TVL, which quantifies the total value of assets deposited across crypto protocols, especially within DeFi platforms. It reflects the level of capital engaged in activities such as lending, staking, and liquidity provision. As of December 2025, global TVL stands at roughly 115 billions, led by Ethereum (71.5 B), followed by Solana (9.0 B), BSC (7.0 B), Tron (4.5 B), and Base (4.5 B) [105]. These decentralized economies operate continuously, without borders, and offer new paradigms for liquidity, collateral, and yield generation.

State-level adoption. Governments and institutions are increasingly exploring blockchain technologies to improve transparency, reduce bureaucratic friction, and enhance data integrity. Prominent examples include:

- Central Bank Digital Currency (CBDC): centralized, state-issued digital versions of fiat currencies (*e.g.,* EUR, USD, YEN) designed to safeguard monetary policy and enable programmable finance. While they may leverage blockchain infrastructure, CBDCs retain full control by the issuing central authority. Several pilot and production-stage projects exist worldwide. For example, China's e-CNY is already in advanced public trials [30]; the European Central Bank (ECB) is developing the Digital Euro [143]; the Bahamas launched the Sand Dollar, the first fully operational CBDC [328]; and Nigeria's eNaira has been deployed nationally [250].

- Land registry systems, such as those in Georgia [197] and Sweden [159], use blockchain to ensure immutable property records and reduce disputes.

- Supply chain tracking, deployed by companies and public entities, ensures product provenance and anti-counterfeiting (*e.g.,* IBM Food Trust [175], VeChain [289]).

- El Salvador has made Bitcoin legal tender [16], aiming to enhance financial inclusion and reduce remittance costs through public wallets and Bitcoin-based payment systems. In parallel, countries like Estonia are exploring blockchain to support transparent governance, secure civic data registries, and efficient public service delivery [302], positioning the technology as a foundational layer for digital state infrastructure.

In sum, blockchain's consequences are both technical and systemic, enabling new forms of economic coordination, governance, and global infrastructure. These shifts reinforce the need for rigorous evaluation frameworks to assess the performance, efficiency, and sustainability of blockchain-based systems across these increasingly diverse use case.

2.1.4 Representative Blockchains

Among the most representative blockchains, we present a set of widely referenced systems that will recur throughout the thesis. They span public and consortium deployments, distinct consensus mechanisms, and heterogeneous execution environments. Table 2.3 summarizes their core characteristics.

Algorand is a public L1 using PoS with VRF-based sortition to form ephemeral committees (BBA*), providing rapid confirmations with economic finality at inclusion [76,147]. Nodes communicate over a gossip overlay carried on TCP; the platform exposes HTTP(S) endpoints and WebSockets for event streams and confirmation, with validation/anti-duplication to suppress retransmissions [91]. Peers maintain bounded neighbor sets with per-IP/port limits and message de-duplication keyed by hashes/sequence numbers [91]. Applications target the Algorand VM (AVM) and are commonly authored in PyTeal [11].

Diem (formerly Libra) is a permissioned, payment-oriented platform coupling HotStuff-based BFT finality [38,347] with a resource-oriented programming model (Move/MoveVM). The stack builds on libp2p, a modular P2P framework providing authenticated transport and stream multiplexing; it exposes RPC for request/response and DirectSend for best-effort stream delivery between peers [20]. Peer admission follows identity/registry policies [20].

Ethereum is a public L1 hosting a large smart-contract ecosystem; post-Merge it adopts PoS with short-term probabilistic confirmations and checkpoint-based economic finality. Ethereum relies on devp2p[9]: node discovery via UDP-based Discovery (v4/v5) and secure peer sessions over TCP using RLPx (encrypted, framed) [118, 140]. Application-level capabilities (*e.g.,* the ETH wire protocol) are multiplexed over RLPx; implementations enforce message-size ceilings (order of 10 MB) and disconnect on oversized/malformed payloads [119], while maintaining peer-count/bandwidth quotas. In consortium-oriented Clique [308], a fixed validator set proposes blocks in a round-robin fashion at a configured interval (deterministic block production); execution remains EVM-compatible (*e.g.,* Geth). Clique reuses devp2p and the eth capability for block/transaction propagation; clients expose JSON-RPC for submission and state access [118, 140].

[9]Compared to libp2p, devp2p is Ethereum's execution-layer, Ethereum-specific P2P suite: it combines UDP-based peer discovery (Discv4/Discv5) with RLPx-encrypted TCP sessions and capability negotiation (*e.g.,* `eth/snap`) over the same peer connection. Post-Merge, Ethereum effectively operates two P2P networks: execution clients remain on devp2p, while consensus clients use a separate libp2p-based gossip network.

Table (2.3) Blockchains considered in this thesis: type, consensus/finality, execution environment, and DApp language.

Blockchain	Type	Consensus / Finality	VM / Runtime	DApp Language
Algorand	Public	BBA* (PoS); rapid economic finality [76, 147]	AVM	PyTeal [11]
Diem	Private	HotStuff (BFT); deterministic finality [38, 347]	MoveVM	Move
Ethereum (Clique)	Consortium	Clique (PoA); deterministic block production [308]	EVM (Geth)	Solidity
Quorum (IBFT)	Consortium	IBFT (BFT); deterministic finality [117, 280]	EVM (Geth)	Solidity
Solana	Public	TowerBFT+PoS; PoH-assisted ordering [292, 340, 342]	Sealevel	Rust/Anchor (Solang optional)

Quorum is an enterprise-focused, permissioned variant of Ethereum that augments privacy and governance features for consortium deployments [117]. Quorum inherits Ethereum's devp2p/RLPx stack and integrates Constellation/Tessera: the Transaction Manager distributes encrypted private payloads among authorized parties, while the Enclave safeguards keys and performs cryptographic operations [32]. With IBFT [280], validators exchange proposal/prepare/commit messages across rounds to achieve deterministic finality under standard Byzantine thresholds; control traffic rides on the same P2P transport with per-peer send buffers, round timeouts, and validator whitelisting [32, 280].

Solana is a high-throughput public L1 combining TowerBFT (with stake incentives) and Proof of History (PoH) – a verifiable delay-based time-stamping mechanism that produces a cryptographic sequence used as a shared clock to order events/transactions without requiring all-to-all time agreement – for fast, verifiable ordering [292, 340, 342]. The data plane is UDP-centric and uses Turbine, a fanout tree that shards and forwards blocks/entries across neighborhoods to reduce duplication and end-to-end latency; repair and vote protocols support retransmission and ledger healing [79]. Validators operate in peer neighborhoods that assist scaling; operational documentation discusses cluster sizes beyond a thousand validators [341]. Clients/validators may leverage modern transports such as QUIC for improved congestion control; RPC exposes commitment levels (processed/confirmed/finalized) and slot/block subscriptions to monitor chain progress [301]. Execution uses the Sealevel runtime for parallel transaction processing; programs are typically written in Rust (Anchor), with tooling such as Solang for Solidity-to-Solana transpilation [342].

2.2 Network Topologies

A network topology is the arrangement of nodes and links that determines who can communicate with whom and along which paths. Formally, it is a graph whose structure governs fundamental communication costs such as reachability, latency, redundancy, and robustness. Relevant structural features include degree distribution (the frequency of node degrees k; heavy tails indicate many low-degree nodes with a few hubs) [35, 243], diameter (the largest shortest-path distance between any two nodes; an upper bound on hop count), average path length (the mean of all shortest-path distances) [242, 243], clustering coefficient (triangle density; the probability that two neighbors of a node are linked) [327], assortativity (degree-degree correlation across edges; positive means hubs link to hubs, negative means hubs link to leaves) [241], betweenness centrality (the fraction of all-pairs shortest paths that traverse a node; a proxy for load and brokerage) [136], and edge/vertex connectivity (the minimum number of edges/vertices whose removal disconnects the graph; a measure of fault tolerance) [329].

Topologies are often described both by families (random, small-world, scale-free, regular meshes, trees, expander-like graphs, fat-tree) and by properties (conductance/-expansion, bisection bandwidth, path diversity), because it is these properties – rather than family names per se – that most directly predict dissemination time and resilience. Conductance/expansion capture how easily flow escapes any subset [81, 169]; bisection bandwidth is the minimum aggregate capacity required to split the network into two halves and proxies worst-case throughput under adversarial traffic [101, 200]; and path diversity (many edge/vertex-disjoint routes) reduces congestion sensitivity and improves failure tolerance [101, 329].

Two points guide the remainder of this section. First, dissemination time for a message depends jointly on the number of logical hops imposed by the topology and on per-hop costs such as network Round-Trip Time (RTT), queuing/verification, serialization; thus, both the graph structure and transport/processing matter [107, 188, 243]. Second, degree heterogeneity and clustering shape not only the mean but also the variance of propagation delays: hubs speed the first wave, yet can create back-pressure when saturated, and tightly clustered communities may delay escape unless enough shortcuts exist, making tails sensitive to the placement and health of high-betweenness nodes [35, 136, 327].

2.2.1 Topology Families and Structural Properties

When practitioners speak about topology families, they are pointing to recurring shapes that bring fairly stable communication traits. Unstructured meshes are the most basic example. In their idealized form, as in the Erdős–Rényi picture [116], the graph looks well-mixed: most nodes have roughly similar degree and no one is structurally special. Such graphs keep path lengths short (the diameter grows slowly with the network size), which explains why simple rumor spreading finishes quickly in theory and tends to behave well in practice when links are healthy [116, 188]. Unstructured meshes fit open, churn-heavy

environments where links come and go and where it would be costly to maintain strict structure; with modest degrees they tolerate random failures and give good average-case dissemination.

Real systems, however, rarely look perfectly random. Many display small-world features in the sense of Watts and Strogatz: neighbors of a node are often connected among themselves (high clustering), yet a few longer shortcut links collapse distances across the network [327]. This blend keeps local communication efficient while preserving fast global reach. The risk is that if shortcuts are too few or poorly placed, information can circulate inside communities before escaping, which shows up as inconsistent arrival times across regions. Small-world overlays are common when there is natural locality (*e.g.,* operator- or region-based peering) but also occasional long-range peers.

Another recurrent shape is scale-free: a heavy-tailed degree distribution where many nodes have few links and a handful act as hubs [34, 35]. Hubs are double-edged. They are excellent amplifiers at low load, pushing messages quickly into many parts of the graph; yet they also concentrate traffic, so if they saturate or are targeted, tails worsen and partitions become more plausible. This family often emerges spontaneously in permissionless settings where long-lived, well-provisioned nodes accumulate connections over time. Practical countermeures – caps on degree, keeping a mix of inbound and outbound peers – aim to reap fast spread without creating single points of congestion or eclipse.

Structured overlays sit at the other end of the spectrum. Here, nodes are arranged according to a rule or coordinate system that yields routing guarantees (*e.g.,* Chord's ring and Kademlia's XOR metric) [224, 306]. The appeal is predictability: short, bounded paths and balanced load by design. The cost is maintenance: when churn is high or identities are adversarial, maintaining structure can be heavy. These overlays shine when operators can enforce stable membership or when fast, reliable lookups are central to the application.

Between purely unstructured and strictly structured overlays there are regular meshes and expander-like designs with bounded degree and high conductance. Their hallmark is a uniformly small number of hops and many edge-disjoint routes, which makes worst-case congestion rarer and failure handling simpler. They are good fits for operator-controlled deployments where one can plan who connects to whom and keep degrees uniform.

Datacenter-inspired fabrics – such as fat-tree – offer an intuitive lesson even when not copied verbatim as overlays: large bisection bandwidth and abundant path diversity keep traffic from bottlenecking near the core [9, 83, 200]. In practice, borrowing these ideas means arranging peerings so that no single cut of the overlay can throttle an epoch's worth of messages, which stabilizes propagation under bursts.

Finally, hierarchical trees and tree-like broadcast are natural when one wants to minimize duplication on the fast path: a message is split and fanned out level by level, so each link carries only a fraction of the total. The trade-off is brittleness – one broken branch can starve an entire subtree – unless the tree is coupled with redundancy (*e.g.,* chunking plus repair) or backed by a looser mesh for fallback. Trees are attractive when reducing per-link load is paramount and when one can engineer repair mechanisms ahead of time.

A few properties explain most observed behavior in these families. Low diameter and high conductance shorten and smooth dissemination; high clustering improves local efficiency but, without enough bridges, can delay global convergence; heavy-tailed degrees speed the median but make tails sensitive to the health of a few hubs. In settings where membership and placement are controllable, regular or expander-like topologies deliver predictable latencies; in open networks with churn, unstructured meshes with diversity-aware peering often give the best robustness–simplicity trade-off. The right choice is thus less about a label and more about which properties – path length, redundancy, fault tolerance, and congestion resilience – matter most for the application at hand.

With these generalities in place, we move from topology fundamentals to blockchain overlays and the consequences for propagation, forks, and finality.

2.2.2 Overlay Networking and Gossip in Blockchains

Network topology is a first-order determinant of a blockchain's liveness and edge consistency: it governs how fast transactions and blocks disseminate, how often short forks occur, and how resilient the network is to churn and adversarial partitioning. Because most public blockchains operate as application-level overlays on the Internet underlay, the shape of the overlay – not merely raw bandwidth or latency – drives propagation time, bandwidth efficiency, and ultimately the usable throughput at a given security margin. In turn, propagation controls on-chain outcomes such as orphan/uncle (block) rates, leader timeouts, and time-to-finality [47, 93, 104]. Safety is a system property, yet liveness and performance are jointly constrained by the overlay's connectivity pattern (degree, diameter, bottlenecks), by the dissemination primitive (push/pull gossip, tree-based broadcast, reliable broadcast), and by traffic-engineering choices such as batching, reconciliation, and compression. Identical consensus mechanisms can therefore display very different tail latencies and orphaning under different overlays, even when the physical underlay is unchanged.

Network partitioning is not merely a theoretical concern: it can interact with protocol-level recovery mechanisms that are explicitly designed to regain finality under prolonged disruption. For instance, Ethereum PoS includes the inactivity leak, which activates after several epochs without finalization and progressively reduces the voting power of validators deemed unreachable, effectively reallocating weight to active participants. Under sustained partitions, this mechanism can lead to conflicting finalizations across network regions and can be exploited by Byzantine validators to reach unsafe voting power faster, highlighting a concrete safety-liveness tension induced by partitioned communication [255].

Blockchains do not broadcast into the Internet in principle; they speak through an overlay, a graph of long-lived connections between peers that sits on top of the IP layer. In practice, each node keeps a small set of transport links (typically TCP, increasingly QUIC/UDP) to selected neighbors and messages bounce across these links until they reach the rest of the network [47]. In permissionless settings, choosing neighbors is delicate: nodes must stay reachable, tolerate churn, and avoid being surrounded by an adversary

(eclipse). Bitcoin therefore combines a pseudo-random address manager (addrman) with peering limits that yield a partially random mesh; by default it maintains about eight outbound full-relay connections, plus block-relay-only links and many inbound peers, effectively setting the average degree and influencing the overlay's logical diameter [42–44]. Ethereum leans on Kademlia-style discovery: a Distributed Hash Table (DHT) that helps find fresh peers while preserving topological diversity [122, 224, 306].

Once connected, nodes exchange data mostly via gossip, an epidemic style of dissemination: when a node sees a new block or transaction, it forwards it to a subset of neighbors, who forward it onward, and so forth. The appeal is straightforward: on well-mixed graphs, gossip reaches almost everyone in a few waves and is naturally resilient to failures and churn [107, 188]. Modern implementations refine this basic idea. After the Merge, Ethereum adopted libp2p GossipSub v1.1: topic-based pub/sub with peer scoring and opportunistic grafting/pruning to keep latency low while discouraging spam and eclipse behavior [121, 314, 318]. Bitcoin followed a complementary path: reduce redundancy and shorten the block-relay critical path. BIP152 (Compact Blocks) exploits mempool overlap to shrink block payloads [93]. For long-haul links, dedicated fast-path relays use UDP and forward-error correction to approach speed-of-light propagation across continents [94].

Some protocols mold the overlay to their own data planes. Algorand filters and validates credentials and votes within BBA* rounds while throttling duplicates [76, 147]. The Avalanche family builds repeated neighbor sampling into its very convergence mechanism [309]. Solana's Turbine uses hierarchical fan-out: a block is split into erasure-coded chunks and fanned out level by level so each link carries only a fraction of the total [79, 297, 341, 342]. In permissioned BFT systems (Tendermint, HotStuff, Diem BFT), the transaction plane remains gossip-like, but consensus messages follow more structured paths – leader-to-quorum and quorum-to-all – to avoid quadratic blow-ups and stabilize pipelines [23, 38, 64, 347].

Two practical observations help interpret empirical behavior. First, overlays exhibit stretch relative to the underlay: two logical neighbors can be tens of milliseconds apart in RTT; reducing logical diameter is not enough unless peer selection also favors low-RTT, high-bandwidth paths while maintaining geographic diversity [71, 212]. Second, open networks tend to produce super-spreaders: high-degree nodes that accelerate the first wave of diffusion but, when saturated or poorly placed (*e.g.,* near an autonomous system bottleneck), amplify latency tails and increase run-to-run variability. Put simply, overlay shape does not replace consensus, but it determines how much real margin the protocol has to meet slot deadlines, avoid transient forks, and reach finality [104].

2.2.3 Propagation and Latency

Forks and timeouts are not mysterious bugs; they are what a blockchain looks like when the network is just a little too slow or too uneven. A useful intuitive model is to picture a relay race. Each new block (or vote) needs to reach enough nodes before the next

runner starts. If the baton is still in flight when the next runner goes, two runners may sprint on different tracks: in Nakamoto-style chains this shows up as short-lived forks and discarded orphan/uncle blocks [104, 145, 303]; in BFT systems it appears as view changes and delayed commits because some replicas did not see the right PREPARE/COMMIT messages in time [38, 64, 347].

Two ingredients drive how quickly a message spreads: how many hops it must traverse across the overlay and how costly each hop is in practice (network delay, queues, and the time to verify or serialize data). Short paths help little if every hop is slow; very fast links help little if the path snakes through many intermediaries. This is why topologies with small diameter and good path diversity tend to feel snappy, and why overlays that lean on a few busy hubs can feel fast on average but occasionally stall when those hubs are saturated.

The engineering playbook follows directly from this picture. First, reduce duplication without starving fan-out: techniques like Compact Blocks shrink what must be sent because most peers already have the transactions [93], set-reconciliation reduces the remaining mismatch [41], and Erlay separates block and transaction relay to avoid unnecessary chatter [238]. Second, shorten or stabilize paths: actively maintained pub/sub meshes (as in GossipSub) keep a healthy mix of neighbors so that messages have multiple short routes even under churn [121, 318]. Third, where geography dominates, fast-path relays help smooth the long intercontinental legs that otherwise inflate the last 10% arrival times [94]. Designs that shard the payload across links – such as Solana's Turbine – also keep per-link load bounded and limit how much any single hop can slow the whole wave [79].

Average delay is only half the story; the other half is dispersion. Two overlays with the same mean can behave very differently if one has fatter tails. In a thin-tail network, most nodes hear about a block within a tight window, so leaders and voters act in sync. In a fat-tail network, a few regions regularly lag, causing occasional timeouts, bursts of view changes, or spikes in uncle rate. Measurements in Bitcoin made this concrete, linking slower block relay to higher stale/orphan rates and motivating the steady move toward compression and dedicated relays [93, 94, 104]. In slot-based PoS systems like Ethereum, this is why mesh parameters and scoring rules are tuned so blocks and attestations reliably land inside the slot budget [121, 318]. In BFT mechanisms, the same tail effects stretch the critical path for PREPARE/COMMIT; HotStuff's linear communication and pacemaker reduce overhead but still rely on the overlay to prevent stragglers from dictating the pace [38, 64, 347].

Viewed this way, overlay design becomes a lever on consensus behavior. Moderately random, diversity-aware meshes tend to provide robust performance in open networks; actively maintained pub/sub helps keep effective fan-out high without storms; reconciliation and relay shortcuts trim the critical path; and when operators control placement, small-diameter, high-conductance fabrics (expanders, torus, fat-tree) make propagation more uniform and consensus pipelines more regular [9, 200].

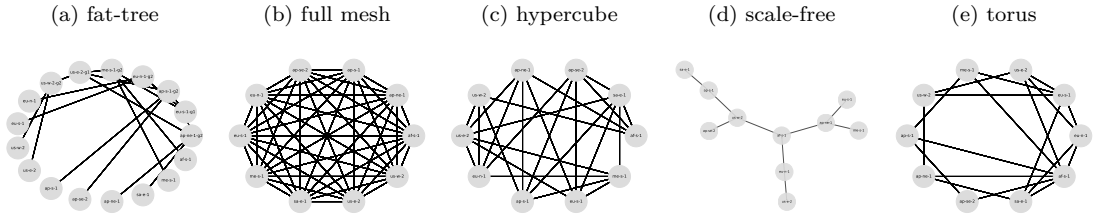


Figure (2.2) Real-world network topologies.

2.2.4 Common Deployments

Among the many shapes a network can take, five families are especially illustrative (Figure 2.2) because they expose different trade-offs among path length, redundancy, and load balance and cleanly map to common deployment intuitions – from tightly managed datacenter fabrics to wide-area overlays shaped by churn, geography, and heterogeneous node capacity.

Fat-tree (datacenter-style fabrics). Originating in high-performance networking, fat-tree aim for high bisection bandwidth and plentiful path diversity via multi-stage hierarchy [9,200]. They are a natural fit when operators control membership and placement – private or consortium deployments, validator clusters within a cloud region, or cross-rack aggregations. Crucially, fat-tree/Clos-style fabrics are a realistic abstraction for modern cloud datacenters: they capture the "many equal-cost paths" property that operators exploit (*e.g.,* via ECMP) to keep queueing and hot-spots predictable at high load [9]. In such settings, bounded logical diameter and many disjoint routes reduce both mean and tail of block/vote propagation, smoothing view changes in BFT pipelines and lowering transient fork rates at a given slot time. Pub/sub meshes (*e.g.,* GossipSub) and block-relay optimizations (*e.g.,* compact blocks) tend to perform predictably here because path lengths and queueing are well behaved.

Full mesh (complete logical connectivity). A full mesh is the conceptual upper bound on connectivity: one hop between any pair [8, 243, 295, 329]. In practice it approximates tightly coupled clusters – small validator committees, specialized relay backbones, or intra-region deployments – where quadratic state is acceptable. In wide-area deployments, full meshes are rarely feasible at scale, but they still model realistic "core cliques" (*e.g.,* a small set of regional gateways/relays or a committee layer) that intentionally keep all-to-all links to minimize control-plane delay. Hop count is minimal, so dissemination is fast and uniform, but link and CPU pressure per node grow with network size. For blockchains this favors scenarios with few, well-provisioned validators or where a relay tier interconnects key gateways to flatten the overlay for the rest of the network.

Hypercube (regular, bounded-degree meshes). Hypercubes combine logarithmic path length with bounded degree and many edge-disjoint routes [101, 320]. They suit environments that value predictable latency without high degrees: federations with stable sub-organizations or permissioned overlays that want uniform load and graceful degradation under failures. Beyond "physical" interconnects, the hypercube is also a useful WAN abstraction for structured overlays: in XOR-metric DHTs such as Kademlia, node identifiers live in a binary space where routing progresses by resolving prefix differences, yielding a virtual topology that is tightly related to hypercube-like structure and logarithmic reachability [224]. For consensus, the regular structure keeps gossip waves coherent and avoids hub congestion; leader-to-quorum patterns in BFT also benefit from multiple short, independent paths.

Scale-free (hub-amplified meshes). Heavy-tailed degree distributions emerge naturally in open systems where long-lived, well-provisioned nodes accumulate connections [34,35,75,77]. This is a useful lens for permissionless overlays and WAN-like settings: hubs accelerate the first propagation wave, improving typical block arrival times, but also concentrate load and risk larger tails if a hub saturates or is attacked. This matches the operational reality of public P2P systems, where heterogeneous capacity and uptime create "super-spreaders" that dominate early diffusion unless peering is explicitly regularized. Practical countermeasures – degree caps, autonomous system/geographic diversity in peer selection, and separating inbound from outbound connections – help preserve fast medians without creating eclipse or Denial of Service (DoS) bottlenecks. In PoS systems with slot budgets, mesh-scoring and opportunistic grafting (as in GossipSub) mitigate the downside by keeping effective fan-out high while discouraging fragile hub dependencies.

Torus (latency-symmetric grids). Torus provide uniform degrees and symmetric distances via wrap-around (cyclic) links [101,323]. They offer balanced load and predictable hop counts, making them attractive when fairness and latency symmetry across sites matter – *e.g.,* geo-distributed organizations that want each region to see similar block and vote timings. They also correspond to widely used virtual WAN overlays: CAN (Content-Addressable Network) embeds nodes in a d-dimensional coordinate space with wrap-around boundaries, *i.e.,* a torus, enabling greedy routing with controlled degree under churn. While perfect tori are uncommon in production, the pattern is a useful reference when comparing protocols under controlled, symmetry-preserving conditions [247]; reductions like chunked fan-out (as in tree-based broadcast) can be layered on top when per-link bandwidth must be tightly bounded.

These families are not labels for their own sake; they bundle properties that matter operationally. Table 2.4 summarizes a few closed-form structural counts used throughout the networking literature; they are included here as a compact cheat sheet for reasoning about degree and link growth without committing to any single deployment. Fat-tree and hypercubes deliver bounded diameter and many disjoint routes, which lowers both the mean and variance of broadcast completion times under bursty load. Full meshes minimize

Table (2.4) Canonical structural counts for reference topologies.

Topology (parameters)	**Average Degree**	**Links**
fat-tree (k ports, l levels)	k	$k^l/2$
full mesh (N nodes)	$N-1$	$N(N-1)/2$
hypercube (n dimensions)	n	$n \cdot 2^{n-1}$
scale-free (N nodes)	(Based on Power-Law)	
torus (n dims, N per dim)	$2n$	$n \cdot N^n$

hop count, which tightens the fast path but at the cost of per-node resource growth. Scale-free meshes speed up the first wave of diffusion, yet require explicit diversity and degree caps to avoid super-spreader bottlenecks. Torus enforce symmetry and predictable latency at moderate degrees, a good fit when one wants to compare protocols without confounding effects from hub congestion. Taken together, the five families span the main "deployment regimes" a blockchain may face: managed fabrics (fat-tree), small all-to-all committees (mesh), structured bounded-degree overlays (hypercube/torus), and heterogeneous public P2P networks (scale-free). These characteristics make the five families useful reference points for reasoning about overlay choices – and, later in this dissertation, for organizing controlled emulations – without committing to any particular deployment.

2.3 Typical Workloads

Blockchains are general-purpose replicated state machines. A workload is the exogenous stream of user actions that drives those machines: the arrival of transactions and smart-contract calls, their payloads, their contention on shared state, and the temporal structure of submissions. Alongside blockchain design and the overlay network, workloads co-determine throughput, latency/finality, orphan or fork rates, and energy consumption. Establishing portable, VM-agnostic workload definitions is therefore a prerequisite for credible cross-platform comparisons.

In what follows we first fix workload semantics (what is executed) and only then discuss and relate them to toolchains, so that later implementation choices do not back-drive definitions.

2.3.1 Evolution and Taxonomy of Workloads

Early empirical studies [97, 104, 145] modeled workloads almost primarily as streams of simple value transfers with constant-rate or near-Poisson arrivals to probe propagation, confirmation, and consensus limits on UTXO or account-based ledgers. With the advent of expressive smart-contract platforms – notably the EVM [331] and account-parallel runtimes such as Solana's Sealevel [343] – production traffic diversified to decentralized exchange interactions, liquidity events, auctions, lending, NFT mints, gaming updates, and adversarial spam floods. Execution models shape how identical business logic stresses

different platforms: single-threaded EVM bytecode with gas metering emphasizes serialization, packing efficiency, and fee-market dynamics [273, 331], whereas account-scoped parallelism enables concurrency conditioned on non-overlapping read/write sets [343]. A portable benchmark must therefore abstract application semantics without baking in a particular VM or runtime.

To characterize typical workloads without overfitting to any one chain, we adopt a taxonomy with orthogonal axes grounded in prior workload literature and in our benchmark lineage.

Temporal structure. This axis spans stationary streams, cyclostationary or diurnal patterns, burst-suppress/relax dynamics, heavy-tailed inter-arrivals, and trace-based replays of historical spikes; heavy tails and self-similarity are well documented in networked systems and service traffic and inform our bursty vs. constant regimes [60, 104, 330].

Source model. This dimension includes parametric synthetic generators with controllable arrival processes, trace replay from production or public datasets, and trace distillation that preserves marginal and joint distributions (*e.g.,* Zipfian key-touch, burst sizes) while anonymizing identifiers; such generators and replays are standard in YCSB, BlockBench, and Caliper and are inherited and extended in Diablo [92, 113, 154, 173].

Operation semantics. This axis covers a spectrum from transfer-only to smart-contract calls with varied read/write balance, event emissions, and payload sizes, including single-call idempotent actions and multi-step workflows; this axis is central to EVM fee/packing trade-offs and to account-parallel runtimes [331, 343]. It also captures DeFi interactions where AMM/CFMM invariants yield compact but highly contended state updates [24].

State-access pattern. The state-access axis distinguishes uniform from Zipfian/hotspot popularity, compact contended state (order books, CFMM pools) from diffuse access, and syntactic (shared-key) versus semantic (invariant-level) conflicts; Zipf-like popularity is a canonical assumption in storage/service benchmarks and underpins contention modeling [24, 60, 92].

Driver control. Driver configuration contrasts open-loop injection at target TPS from closed-loop, client-limited pacing, and fixed-rate from feedback/backpressure-driven regimes; this distinction is foundational in performance engineering and is reflected in mainstream toolchains, which fixes the injected process independently of committed outcomes [154, 173, 230].

Adversarial content. The content dimension separates benign mixes from stress/abuse floods, fee/mempool manipulation, and leader/view-change triggers; stress-style loads are common in blockchain scalability/security studies and inform overload scenarios [97, 145].

Determinism and reproducibility. This criterion entails seeded randomness for key selection and arrival jitter, fixed horizons, documented seeds, and replication across runs and infrastructures [113, 154].

This broader space subsumes the families used later and clarifies that static/constant loads, variable/bursty loads, trace-based replays of past events, and smart-contract-capable mixes are all first-class workload types treated explicitly in our methodology.

2.3.2 Canonical Families

Within the taxonomy above, a small number of recurring families provides compact abstractions that map well to real use-cases while remaining VM-agnostic. Payments represent long-running, nearly stationary retail flows that probe sustained capacity, mempool pressure, and fee-market stability [273]. Market/Exchange captures bursty, contended interactions on compact state (order placement, inventory checks) and also abstracts DeFi CFMM calls in which pool reserves and price quotes are updated atomically against an invariant [24]. High-frequency telemetry for games represents continuous, high-rate state updates that stress batching, block packing, runtime scheduling, and concurrency limits [113, 343]. Stress/abuse floods provide flat, high-intensity injections that expose backpressure, mempool eviction policies, commit-ratio collapse points, and leader/view-change behavior [97, 145].

These families are not exclusive: they intersect the axes above (temporal structure, source model, semantics, access patterns, control, adversarial content, reproducibility) and serve as reusable shapes of demand.

To make the families concrete, we outline a small portfolio of workloads that instantiate different corners of the taxonomy and that will also be considered as standardized test cases. Unless stated otherwise, the workloads below are taken from the Diablo benchmark suite [154], which introduces and implements the corresponding application logic (smart contracts, where applicable) and workload profiles [154]. In this dissertation, we integrate these Diablo workloads into our experimental harness; any deviations from the original suite are explicitly stated for each workload. The only newly contributed workload in this portfolio is PayPal, which is implemented on top of the same transfer-only transaction path used by the payment-style workloads.

Distributed Denial of Service (DDoS). A very high, transfer-only, flat injection over a short horizon (constant open-loop pacing), exercising the stress/abuse family with uniform state access; analytically useful to delineate collapse points and queuing/backpressure regimes [97, 145]. We adopt Diablo's transfer-only DDoS workload as implemented in the benchmark suite [154].

FIFA. A short-lived, extreme burst onto a compact counter/booking contract (hot-spot keys), representative of ticketing spikes [27]; it sits with high contention and transient-heavy arrivals. This workload is taken from Diablo's FIFA application and burst profile [154].

In this dissertation, we keep the application semantics and access hot-spots unchanged, but *increase the injection rate by approximately 10 times* relative to the default Diablo configuration to better stress peak-load behavior.

Google Apple Facebook Amazon Microsoft (GAFAM). A burst-then-settle market scenario with a fixed SKU set (buy/check actions), mapping to compact contended state and a cyclostationary temporal structure; it provides a clean lens on congestion/price-formation mechanics in fee markets [273]. We adopt Diablo's market/exchange workload GAFAM and its transaction mix and temporal structure as implemented in the benchmark suite [154].

Gaming. A continuous, multi-minute high-rate stream of small state updates, representative of telemetry-like workloads with locality; it highlights runtime scheduling and account-parallel execution limits [343]. We adopt Diablo's Gaming workload and the corresponding application logic as implemented in the benchmark suite [154].

PayPal. A constant-rate payment stream at low-to-mid intensity (stationary, diffuse access). PayPal is handled in this thesis as a transfer-only, steady-regime payment workload meant to complement VISA with a lower-intensity operating point. It builds on the same baseline transfer transaction path used by the payment-style workloads in Diablo (*i.e.,* no third-party smart-contract code), differing only in its calibrated submission rate and parameterization.

VISA. A higher-intensity card-rail proxy still below overload; both anchor the payments family and are useful for steady-regime analysis of latency and throughput under open-loop pacing [230]. We adopt Diablo's transfer-only VISA workload [154] but recalibrate its constant submission rate to match publicly reported VISA transaction volumes (used only as an order-of-magnitude calibration target) [154]. The workload semantics remain transfer-only and unchanged; only the rate parameter differs from the original Diablo configuration.

Together these examples span static/constant versus variable/bursty arrivals, transfer-only versus contract-capable mixes, compact versus diffuse state, and synthetic versus trace-inspired sources, without committing to any specific platform or execution tool. Importantly, the portfolio separates workload provenance (Diablo vs. newly contributed) from parameterization changes (rate recalibration or scaling), to avoid ambiguity about authorship and about what software artifacts are reused. This section intentionally remains at the level of semantics and taxonomy; execution frameworks and measurement methodology are orthogonal and introduced later on.

2.4 Benchmarking Tools

Given the explosive growth and heterogeneity of the blockchain landscape, choosing the most suitable protocol for a given application is far from straightforward. The current ecosystem encompasses (*i*) public, private, and consortium chains and (*ii*) a broad spectrum of consensus mechanisms and diverse profiles about performance, governance, and security. In light of this complexity, developers, researchers, and stakeholders must critically evaluate blockchain options based on the specific technical and economic requirements of their application domains. This evaluation spans the following dimensions:

- **Performance.** Typically assessed in terms of throughput (TPS) [108, 154], block latency (the time between transaction issuance and final confirmation [146, 156]), and scalability under increasing load. These metrics remain among the most frequently reported in blockchain comparisons [335].
- **Energy.** The energy profile of a blockchain reflects its sustainability and operational cost. This includes the total energy consumed per transaction and the overall footprint of the network [68, 261, 279].
- **Economics.** Quantitative indicators such as TVL, Gini coefficient (a measure of wealth concentration and inequality among participants) [40, 276], and velocity of money (the rate at which tokens circulate, indicating economic activity) [103] offer insights into the systemic economic behavior of a blockchain and its suitability for use cases requiring robust financial interactions.
- **Security.** Blockchain systems must guarantee transaction finality and withstand network faults. Attacks such as Sybil attacks (identity flooding) [62], 51% attacks (majority control) [344], and Eclipse attacks (peer isolation) [166] test the robustness of a system. For instance, PoW is highly resilient but energy-intensive [182, 261], while PoS may be susceptible to validator centralization if stake distribution is skewed [99].
- **Decentralization.** This is often discussed but poorly measured. It includes validator/node distribution, geographic dispersion, and diversity in control over network resources [207]. High hardware requirements or low participation incentives can lead to effective centralization even in permissionless systems [305].
- **Governance.** On-chain governance – where stakeholders vote directly on protocol upgrades – is implemented in platforms like Tezos [14] and Polkadot [65], enabling automated, fork-free protocol evolution. In contrast, off-chain governance, as seen in Bitcoin, depends on developer consensus and community coordination [235]. Governance design influences not only upgrade flexibility but also community trust and long-term resilience [69].
- **Interoperability.** As applications span multiple blockchains, protocols must enable secure cross-chain interaction and composability. Frameworks like Polkadot and

Cosmos facilitate interoperability via relay or hub-and-zone architectures [65, 95]. Cross-chain bridges such as Wormhole [259] support asset and data transfers, empowering complex multi-chain ecosystems.

The moral of the story is that there is no one-size-fits-all blockchains. The best protocol for a particular use case depends on the specific mix of technical requirements, economic goals, security guarantees, and deployment contexts. This complexity underscores the need for transparent, standardized, and multidimensional benchmarking frameworks – capable of comparing blockchain systems not just in idealized conditions but under realistic, dynamic, and application-relevant scenarios.

2.4.1 Technical Frameworks

Due to its growing adoption, evaluating the performance and scalability of blockchain systems has become fundamental. Benchmarking plays a key role in this process, supporting informed design and system tuning. A blockchain benchmark is a methodologically controlled process that executes a target system under specified conditions (workloads, network) to produce comparable measurements and to assess its suitability for a specific use case. It enables the verification of protocol performance claims and guides stakeholders in making informed decisions. Consequently, a variety of blockchain simulators and emulators have been developed to facilitate controlled, repeatable benchmarking experiments [15, 113, 137, 146, 196, 206, 252, 278]. They differ widely in scope and methodology, hence comparing them is complicated.

Several tools reduce scale and complexity by modeling only consensus or abstracting key behaviors. *Minichain* and *BlockLite-toward* focus only on the consensus algorithm due to resource constraints [324, 334]. Event-driven models such as *SimBlock* and *CBlockSim* compute block creation probabilistically and decouple block generation from message transmission [26]. We view simulators as reusable tools that replace full-system execution to study scaling/topology effects. We capture testbeds later (Table 6.1) as environment classes (*e.g.,* cloud, local cluster, geo-emulated cluster). Assumptions like always-full blocks (*BlockSim-m*) depart from real-world deployments with variable block occupancy [13]. Frameworks like *SIMBA* and *BlockPerf* do not execute real clients, limiting fidelity on critical data paths (mempools, signature verification), accuracy, and resource contention [131, 262].

A second family runs real clients with deployment/orchestration, sometimes exposing partial network control. *BlockBench* and *Diablo* provide workloads and scripts to bring up multi-node environments, but offer limited abstractions over the underlying testbed and network layer [113, 154]. *BCTMark* and *COCONUT* improve portability and parameterization (*e.g.,* YAML topologies, tunable client settings), yet typically lack rich topological diversity or strong network isolation [146, 278]. Domain or stack-specific initiatives include the *Core-Bitcoin Network Simulator* [15] and the *BBB*, *JABS*, and *DLPS* frameworks addressing gaps from earlier approaches [137, 252, 339]. In the Ethereum ecosystem, *HIVE* aids conformance and *Caliper* offers generic load generation; however, neither

provides a fully controlled end-to-end environment (*e.g.,* resource reservation, internal state introspection, per-link policies) [123,173]. Some – like *CIDDS* and *Bitcoin-Simulator* – model network characteristics but lack extensibility for diverse architectures [131, 252]. Existing tools rarely combine real-client execution, topology control, integrated energy measurement, and repeated-run variance analysis in one pipeline.

Table 2.5 provides an overview of the diversity and features of existing tools. Specifically, we report their publication year, programming language, supported network configurations, orchestration strategies, execution mode, portability, workload support, the presence of configurable network controls, whether its venue offered an artifact evaluation (AE) process, and the status of ACM and IEEE badging [4, 176]. For ACM and IEEE badges, we mark tools published before the introduction of the corresponding artifact and reproducibility schemes (2016 for ACM, 2019 for IEEE) with "–", indicating that badges were not yet available [55, 253]. The legend maps symbols to full, partial, none, or unknown support. Most tools lack ACM/IEEE badges for *reproducibility* (different team, same artifacts) and *replicability* (different team, independent artifacts) [4, 176]. We treat badges as auxiliary signals and focus on the operational capabilities needed for variance-centric evaluation (network control, topology enforcement, deterministic orchestration, repeated runs), marking missing documentation as "data unavailable". Building on observations from previous studies [108, 110], we view this systemic heterogeneity along comparability-critical dimensions as a further motivation to move away from new leaderboards or single-run point estimates and target instead experimental repeatability and performance predictability under controlled, geo-emulated conditions.

2.4.2 Energetic Approaches

With regard to energy-aware measurement approaches, the literature reveals a strong emphasis on solutions and methods for studying and analyzing energy consumption in blockchains [25,67,87,164]. These approaches primarily focus on post-deployment analysis. For example, the Cambridge Blockchain Network Sustainability Index (CBNSI) [67] 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 [154,278], 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 [279] leverage a highly effective benchmarking tool (BCTMark [278]) 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 Software-Defined Networking (SDN) scenarios. The authors of [164] propose a

Table (2.5) Overview of supported features across blockchain benchmarking tools. Legend: ◌ data unavailable; ✗ not reported; ◑ partially reported; ✓ fully reported; "–" badge scheme not yet in place (ACM/IEEE since 2016/2019). Abbreviations used: Centr. = Centralized, Distr. = Distributed, Simul. = Simulation, Emul. = Emulation, SC = Smart Contracts, TT = Transfer Transactions.

									ACM Badges			IEEE Badges		
Tool	**Year**	**Programming Language**	**Orchestration**	**Mode**	**Portability**	**Workload**	**Network Controls**	**Artif. Eval. Process**	**Artifacts Avail.**	**Results Reprod.**	**Results Replic.**	**Code/Data Avail.**	**Results Reprod.**	**Code/Data Reviewed**
Shadow-Bitcoin [232]	2015	Python	Centr.	Simul.	✗(Bitcoin)	TT	◌	◌	–	–	–	–	–	–
HIVE [123]	2016	Go	Distr.	Simul.	✗(Ethereum)	◌	◌	◌	✗	✗	✗	–	–	–
Simcoin [281]	2016	Python	Centr.	Simul.	✗(Bitcoin)	TT	◌	◌	✗	✗	✗	–	–	–
Bitcoin-Simulator [145]	2016	C++	Centr.	Simul.	✓(Bitcoin-like)	TT	◑	◌	✗	✗	✗	–	–	–
BlockBench [113]	2017	C++,Go	◌	Simul.	✓(Private)	SC	◌	✓	✗	✗	✗	–	–	–
CIDDS [196]	2018	Python	◌	Simul.	✓(DAG-based)	◌	◑	◌	✗	✗	✗	–	–	–
Caliper [173]	2018	Java	Distr.	Emul.	✓(Ethereum,Hyperledger)	SC	◑	◌	✗	✗	✗	–	–	–
Minichain [334]	2019	Python	Centr.	Emul.	✓	◌	◑	◌	✗	✗	✗	✗	✗	✗
BlockLite [324]	2019	Java	Centr.	Emul.	✓(PoW-public)	◌	◌	◌	✗	✗	✗	✗	✗	✗
BlockSim-f [130]	2019	Python	Centr.	Simul.	✓	◌	◑	◌	✗	✗	✗	✗	✗	✗
SimBlock [26]	2019	Java	Centr.	Simul.	✓	TT	◑	◌	✗	✗	✗	✗	✗	✗
BlockSim-m [13]	2020	Python	Centr.	Simul.	✓	TT	◑	◌	✗	✗	✗	✗	✗	✗
Core-Bit-Netw-Simul. [15]	2020	Python	Centr.	Simul.	✓	◌	◑	◌	✗	✗	✗	✗	✗	✗
BBB [252]	2020	Python	Centr.	Emul.	✓(Private)	◌	◑	◌	✗	✗	✗	✗	✗	✗
SIMBA [131]	2020	Python	Centr.	Simul.	✓	◌	◑	◌	✗	✗	✗	✗	✗	✗
BCTMark [278]	2020	Python	Distr.	Emul.	✓	TT	◑	◌	✗	✗	✗	✗	✗	✗
BlockPerf [262]	2021	C/C++,Python	Distr.	Both	✗(Bitcoin)	TT	✗	◌	✗	✗	✗	✗	✗	✗
DLPS [137]	2021	Python	Distr.	Emul.	✓	TT	◑	◌	✗	✗	✗	✗	✗	✗
Gromit [237]	2022	Python	Distr.	Emul.	✓	SC	◑	◌	✗	✗	✗	✗	✗	✗
CBlockSim [215]	2022	◑	✗	✗	✗	✗	✓	✗	✗					
Diablo [154]	2022	Go,Python,Perl	Distr.	Emul.	✓	TT,SC	✗	✓	✓	✗	✗	✗	✗	✗
JABS [339]	2023	Java	Centr.	Simul.	✓	TT	◑	◌	✗	✗	✗	✗	✗	✗
TangleSim [206]	2023	Go,Python	◌	Simul.	✓(DAG-based)	◌	✗	◌	✗	✗	✗	✗	✗	✗
COCONUT [146]	2023	Jinja,Java	Distr.	Emul.	✓	TT,SC	◌	✓	✓	✗	✗	✗	✗	✗
BBSF [268]	2023	C++,Go	◌	◌	◌	◌	✗	✗	✗	✗	✗	✗	✗	✗
GFBE [214]	2024	◌	Distr.	Emul.	✓	SC	✗	◌	✗	✗	✗	✗	✗	✗
STABL [156]	2024	◌	Distr.	Emul.	✓	TT,SC	◑	◌	✗	✗	✗	✗	✗	✗
Hammer [319]	2024	◌	◌	◌	◌	◌	✗	✗	✗	✗	✗	✗	✗	✗
RBlockSim [260]	2025	◌	◌	◌	◌	◌	◑	✓	✓	✓	✗	✗	✗	✗
Lilith [108]	**2025**	**Bash,Python**	**Distr.**	**Emul.**	✓	**TT,SC**	✓	✓	✓	✗	✗	✗	✗	✗

Field-Programmable Gate Array (FPGA)-based testbed for estimating Bitcoin's energy consumption. The research in [25] 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 [288] and Blockchain Security over SDN (BSS) [37] lack an evaluation of energy consumption. This gap could potentially introduce security challenges within the architecture. The authors of [346] 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 [87], 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 [74] and Stellar [225] 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 [279] 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 [80] using Linux TC [57] for network modeling, the framework is unable to fully emulate or replicate a complete network topology.

2.4.3 Economic Indicators

Besides technical metrics (TPS, latency, energy), there are economic indicators that capture ecosystem balance and health such as TVL, wealth concentration via Gini/Nakamoto-style measures [40, 276], participation and circulation proxies (active addresses, velocity) [31, 103, 213, 233], and dormancy-based metrics (Bitcoin Days Destroyed, average dormancy) [19,139,294]. In practice, economic assessment relies on complementary families of indices and methods, each highlighting a distinct facet of the on-chain economy:

- **Capital lock-in and credit capacity.** TVL is a proxy for the amount of capital committed to protocols and thus for collateralization and credit creation potential within DeFi ecosystems. It contextualizes throughput-oriented benchmarks with a measure of financial depth.

- **Distributional concentration.** Inequality measures (*e.g.,* Gini) and security-

oriented concentration metrics (*e.g.,* Nakamoto-style indices based on the minimum set controlling a target share of stake, mining, or governance power) quantify how ownership and control are distributed among participants [40, 276]. These indicators are typically computed over address/entity distributions and can be tracked over time to study concentration dynamics.

- **Participation and network usage.** Activity metrics such as active addresses, transaction counts, and retained vs. returning users capture user engagement and the breadth of economic participation [31, 213, 233, 337]. Graph-based rankings and flow-based centrality may be used to weight participants by their transactional role [337].
- **Circulation and monetary velocity.** Velocity-style measures operationalize how quickly units of value change hands, linking transactional activity to the effective utilization of supply [103]. Variants include turnover ratios computed over rolling windows, supply-segmented velocities (*e.g.,* excluding illiquid tranches), and protocol-specific adjustments.
- **Dormancy and coin-age dynamics.** Metrics based on holding time – *e.g., Bitcoin Days Destroyed* and *average dormancy* – weight flows by the age of moved coins to reveal reactivation of idle balances and shifts between saving and spending regimes [19, 139, 294]. These indicators help distinguish bursts of speculative churn from structurally renewed circulation.
- **Production-oriented efficiency.** Building on notions from economics, technical and scale efficiency separate input utilization from returns to scale, enabling comparisons of output per unit input and identification of (dis)economies of scale in protocol operations [36, 115]. Such analyses can be paired with resource-cost lenses, including energy expenditure per unit of economic output [68].
- **Market microstructure and price dynamics.** Price-based indices add a complementary view on volatility regimes, liquidity conditions, and trend efficiency, informing how market states co-vary with on-chain activity and participation [187].
- **Resource utilization and slack.** Concepts such as unused capacity can be adapted to blockchain settings to reason about underutilized throughput, idle liquidity, or latent collateral that could support more economic activity without additional resource input [223, 283].

2.5 Open Challenges

Despite ongoing efforts, blockchain benchmarking remains a young and fragmented area. The community has built a spectrum of simulators, emulators, and orchestration frameworks, yet credible comparisons among blockchains and deployments are still hard to achieve. The difficulties are not merely technical; they arise from the interaction between

system design, execution environments, network dynamics, and economic behavior. Below we illustrate the main open challenges, highlighting where current practice falls short and what capabilities are needed going forward.

2.5.1 Protocol Claims and Evidence-Based Validation

Many blockchain systems publicly advertise ambitious performance and scalability claims that are often not representative of real-world deployment scenarios. These claims typically assume ideal network conditions, overlook the impact of network dynamics, and underestimate the consequences of validator centralization, security vulnerabilities, and fundamental trade-offs. As a result, the promised throughput and latency numbers rarely translate to production environments. Examples are:

- **EOSIO.** Claimed throughput of over 4,000 TPS using a Delegated Proof of Stake (DPoS) consensus with low latency and high scalability. However, empirical studies conducted on public deployments demonstrate significantly lower sustained throughput – often in the low hundreds of TPS – and limited decentralization, as the network is controlled by a small number of block producers [208, 209].

- **Solana.** Advertised to support up to 200,000 TPS with low transaction fees and fast finality. In practice, Solana has experienced multiple network outages in 2021 and 2022 triggered by congestion from spam transactions and bot activity, revealing the fragility of the network under stress. Moreover, the high hardware requirements for validators have led to centralization pressures, reducing the effective decentralization of the network. Independent researchers report a peak of approximately 60,000 TPS with a finality time of around 12 seconds under controlled testing conditions [154].

Benchmarks should explicitly state the assumptions behind claims and reproduce them under transparent, instrumented setups with matched workloads and network policies; otherwise, comparisons risk rewarding tuning for synthetic cases rather than robust performance.

2.5.2 Standardization and Comparability

There is neither an agreed-upon benchmark suite, metric taxonomy, or reporting template that captures the core dimensions of blockchain operation – throughput and latency distributions, energy per transaction, failure behavior and recovery, and economic activity signals – under well-defined workloads and network topologies, nor one that integrates a heterogeneous mix of economic indicators into a coherent framework. The field is still relatively young and lacks standardization in both methodology and tooling [203]. Existing benchmarking tools are often fragmented, blockchain specific, and not designed for extensibility or cross-platform portability: some abstract away client execution and data structures, others run full binaries but expose limited control over the network layer or resource reservation [26, 113, 123, 131, 137, 146, 154, 173, 252, 262, 278, 324, 334, 339]. Most

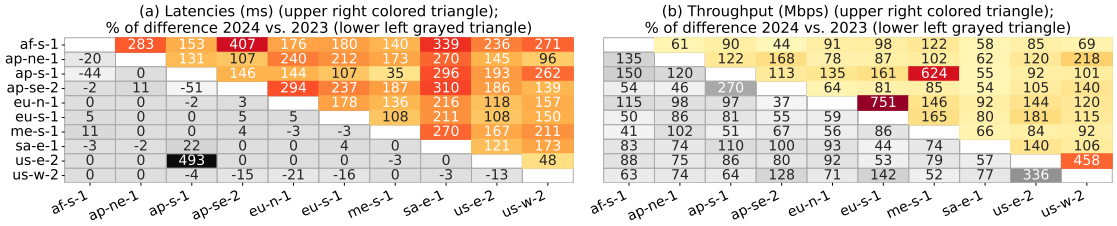


Figure (2.3) Latency (ms) and throughput (Mbps) heatmaps for 2024 measurements (upper colored triangles); % difference 2024 vs. 2023 (lower gray triangles). AWS regions: *af-s-1* (Cape Town), *ap-ne-1* (Tokyo), *ap-s-1* (Mumbai), *ap-se-2* (Sydney), *eu-n-1* (Stockholm), *eu-s-1* (Milan), *me-s-1* (Bahrain), *sa-e-1* (Sao Paulo), *us-e-2* (Ohio), and *us-w-2* (Oregon).

studies prioritize isolated performance metrics over performance predictability, leaving execution variability and experimental repeatability underexamined [108, 154].

2.5.3 Variability and Cost of Uncontrolled Deployment Environments

Blockchain benchmarks primarily faces the inherent complexities of the execution environment. On the one hand, network emulators and VMs can reproduce network conditions on a single local machine (*e.g.,* Netlab [256]); however, such setups remain inadequate for blockchain benchmarking. Blockchain networks typically involve hundreds or thousands of concurrent nodes, exhibiting complex interactions, diverse latency profiles, and resource contention patterns that local environments cannot realistically capture. Furthermore, attempting to emulate large-scale distributed behavior on a single machine often results in oversimplification, ultimately collapsing into a mere simulation. On the other hand, cloud infrastructures are common in blockchain benchmarking for scalability and geolocation support. Yet they introduce substantial variability and cost.

First, cloud networks are dynamic and opaque. While cloud platforms offer scalability and high-performance nodes, their infrastructures are continuously evolving and fluctuating network conditions introduce significant variability. We analyzed cross-region AWS RTT traces (Apr. 2023–Jan. 2025) of which we report in Figure 2.4 the temporal standard deviation of daily-mean RTTs, confirming sizable and time-varying latency dispersion in public clouds, consistent with prior work [201]. We focused on 10 AWS regions across different continents: *af-s-1* (Cape Town), *ap-ne-1* (Tokyo), *ap-s-1* (Mumbai), *ap-se-2* (Sydney), *eu-n-1* (Stockholm), *eu-s-1* (Milan), *me-s-1* (Bahrain), *sa-e-1* (Sao Paulo), *us-e-2* (Ohio), and *us-w-2* (Oregon). Figure 2.4 shows how latency standard deviation (in milliseconds) evolves over time, revealing clear temporal instability in cloud networking. For instance, the Milan (*eu-s-1*) to Bahrain (*me-s-1*) latency deviation ranged from 20 ms (April 2023) to 5 ms (July 2023), rising again in later months. While often reflecting service improvements, such fluctuations undermine the repeatability of experimental results.

Second, cloud providers offer limited control over low-level network parameters. Users cannot configure packet loss, link delays, or custom topologies without workarounds. Network structures evolve behind the scenes and resource contention can arise from other

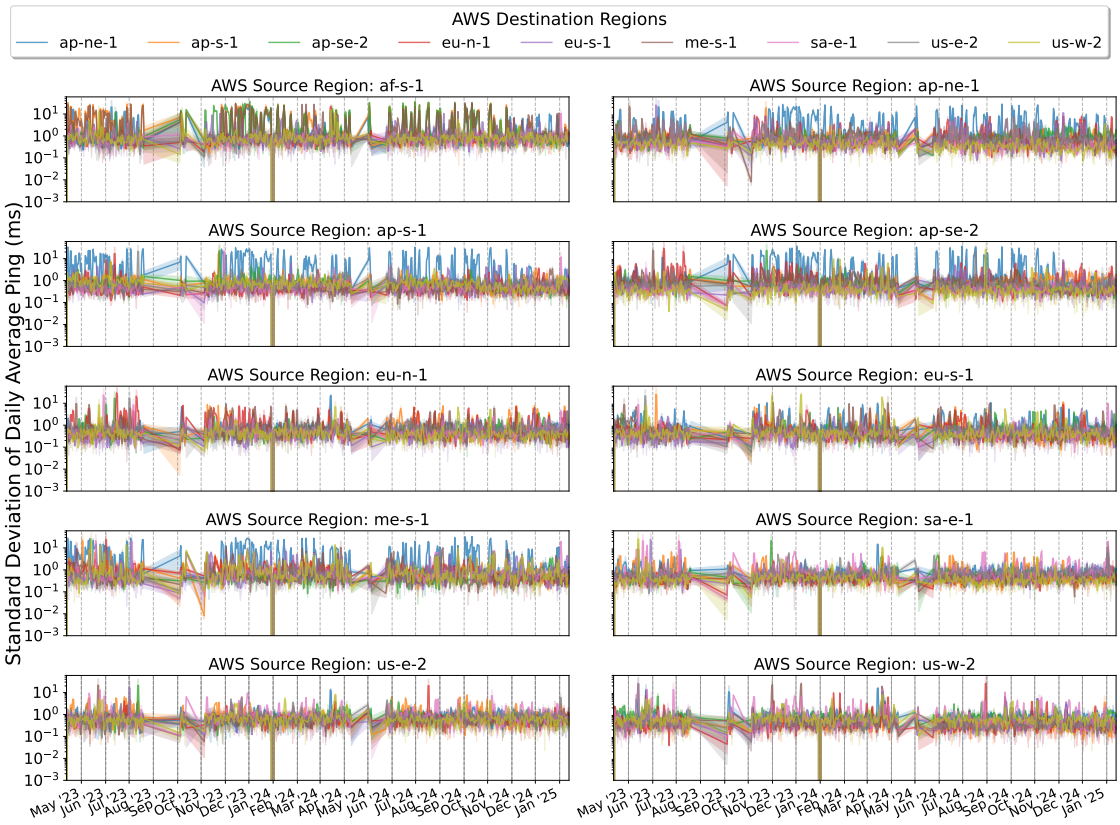


Figure (2.4) Long-term standard deviation of daily-mean RTTs (Apr. '23–Jan. '25) across 10 AWS regions (see Figure 2.3). Each panel shows one source region against all destination regions; measurements use one Amazon Elastic Compute Cloud (EC2) virtual machine per region and cover all region pairs.

tenants.

Third, cloud-based experimentation is expensive. Large-scale experiments necessitate deploying a significant number of instances (physical nodes or VMs) to accurately replicate blockchain networks, sometimes consisting of thousands of nodes [254]. Our cost evaluation study in Table 2.6 reports high costs for running several nodes continuously for a week, *e.g.,* operating 40 nodes incur costs of 2,800 USD, excluding extra charges for network services and data transfer. This demonstrates the high cost associated with cloud-based measurements at scale.

These factors underscore the need for controlled environments – especially full-stack emulation with well-defined, instrumented, and repeatable infrastructure and topology.

2.5.4 Network Variability and Controllability

A particularly underexplored dimension is the network. Blockchains operate in inherently dynamic network environments, subject to a wide variety of unpredictable conditions that significantly impact performance and reliability, including but not limited to: node failures and crashes, packet loss and retransmissions, network congestion and fluctuating latency, link interruptions and network partitions. In blockchain systems, performance is influenced

Table (2.6) Weekly cloud cost comparison for three scenarios: (i) 10 nodes, 30/60 vCPUs, 64/120 GB RAM; (ii) 200 nodes, 4 vCPUs, 8 GB RAM; (iii) 200 nodes, 8 vCPUs, 16 GB RAM.

Cloud provider (instance types)	**(i)**	**(ii)**	**(iii)**
Alibaba (i: ecs.c5.8xlarge/ecs.hfc6.10xlarge; ii: ecs.c5.xlarge; iii: ecs.c5.2xlarge.)	1,900 USD	7,500 USD	11,500 USD
AWS (i: c5.9xlarge; ii: c5.xlarge; iii: c5.2xlarge.)	2,000 USD	8,000 USD	14,000 USD
Azure (i: f32sv2-standard/f48sv2-standard; ii: f4sv2-standard; iii: f8sv2-standard.)	2,200 USD	8,500 USD	12,800 USD
Google Cloud Platform (i: c2-standard-30/c2-standard-60; ii: c2-standard-4; iii: c2-standard-8.)	2,500 USD	9,000 USD	13,500 USD

not only by the consensus mechanism – coordinating agreement under failures – but also by topological factors – including the number of nodes, their interconnection layout, and the latency distribution across links. Performance and energy hinge on overlay connectivity and dissemination. Small changes in degree, diameter, and RTT distributions can alter block propagation time, fork rates, leader timeouts, and tail latency; they can also shift the balance between compute and communication costs [25,108,110,158]. Yet only a minority of frameworks provide programmable topologies with per-link latency/bandwidth/jitter/loss and fixed routing/queuing. These factors contribute to strong variability in operating conditions, which challenge the applicability of idealized benchmarks and complicate consistent performance evaluation.

2.5.5 Adversarial Dynamics

Real networks exhibit node churn, packet loss, congestion, and partitions; validators may fail or misbehave; traffic patterns may be adversarial (spam, bot surges). Benchmarks often assume benign steady-state conditions and omit structured stress testing. Yet public history shows that congestion and bot activity can degrade liveness or finality in otherwise high-throughput designs, revealing brittle failure modes that clean-room setups miss. Incorporating fault injections (crash/restart, link drops, partitions), churn processes, and adversarial traffic into standard scenarios remains a gap.

2.5.6 Resource Management and Measurement Fidelity

Accurate benchmarking requires modeling CPU contention, I/O pressure, cryptographic verification pipelines, and storage behavior – together with network scheduling. Frameworks that do not execute real clients, or that lack resource reservation and introspection, struggle to capture these interactions [123, 131, 173, 262]. Even when real binaries are used, the absence of deterministic orchestration (boot order/timing, pinned images/configurations, CPU/memory affinities) reduces reproducibility and complicates root-cause analysis [162, 201].

2.5.7 Missing a Unified Index

In traditional economies, metrics like the *Gini* coefficient [40, 276] – which quantifies wealth distribution – and *Pareto* efficiency [217] – which describes an optimal allocation of resources where no individual's well-being can improve without negatively affecting another

– are used to identify economic inefficiencies and potential resource underutilization. Furthermore, it is interesting that resource allocation analyses often incorporate concepts such as the cost of unused capacity [283].

As for the research gap in relation to economic efficiency within blockchain and cryptocurrency ecosystems, we identify numerous studies that examine supply distribution, active user participation, and idle resources within these ecosystems.

Supply distribution impacts fairness, decentralization, and network stability [185]. Consensus mechanisms and other aspects impact cryptocurrency distribution, where wealth concentration can affect security and exchange rates [194, 276]. Actually, cryptocurrencies exhibit high inequality patterns similar to traditional economies (*e.g.,* DOGE [220] Gini coefficient – 0.82 – is similar to US one – 0.84 [276]).

User participation affects liquidity and network value. In [337], an improved version of the PageRank algorithm – a Google technology for ranking web pages based on their importance – is used to evaluate Bitcoin user participation by taking into account both the consistency and the variability of transaction patterns. Key metrics like active addresses, transaction volume, and circulation frequency strongly correlate with price trends and economic activity [31, 213, 233].

Idle assets signal inefficiencies in resource use [19, 139, 294]. A key metric here is "Bitcoin Days Destroyed", which measures transaction volume while accounting for how long bitcoins have remained unused – highlighting the economic impact of previously dormant coins becoming active again [294]. Another useful metric is average dormancy, which tracks the duration that bitcoins remain inactive before reuse thus providing insights into circulation patterns, although it does not directly measure monetary velocity or account for the full money supply and price levels [19, 139].

Together, these elements highlight the need for an integrated framework to evaluate uniformly economic efficiency in blockchain systems [223].

Addressing these problems is essential for advancing blockchain research and enabling the design of more robust, efficient, and trustworthy blockchain systems. Credible blockchain benchmarking must bring methodological discipline and multi-dimensional measurement into a single, controllable, and reproducible process. Only then can we relate protocol design and deployment choices to their real effects on performance, energy, and economic aspects.

Chapter 3

Lilith: A New Blockchain Benchmarking Framework

This chapter presents Lilith as the methodological bridge between the benchmarking challenges identified in Chapter 2 and a network-controlled solution for repeatable blockchain evaluation. It first motivates the framework and clarifies its role in the dissertation methodology (§3.1), then introduces the two main building blocks on which it relies: the Diablo benchmark suite (§3.2) and the Kollaps network emulator (§3.3). The chapter next describes how Lilith generates and instantiates logical overlay topologies from measured network data (§3.4), the built-in workload profiles it bundles (§3.5), and the blockchain backends considered in the study (§3.6). It subsequently details the benchmark execution workflow (§3.7), the metrics gathering pipeline (§3.8), and the testing configurations adopted throughout the evaluation (§3.9). Overall, Lilith provides the experimental backbone for the comparative studies developed in the forthcoming chapters, including the configuration, measurement, and reporting choices that support repeatable and dispersion-aware analysis across heterogeneous blockchain systems.

3.1 Lilith Overview

The motivation behind this dissertation arises from a fundamental yet insufficiently addressed observation: evaluations of blockchain technologies often fail to reflect the conditions under which these systems are designed to operate. Performance benchmarks are often run in idealized lab settings; energy models overlook deployment heterogeneity; economic analyses rarely account for structural asymmetries; and experimental repeatability and prediction are more aspirational than achieved. In contrast, this work adopts a grounded methodology that actively introduces and controls variability, while emphasizing transparency and experimental repeatability. In doing so, this dissertation challenges conventional benchmarking practices and offers a methodological approach that exemplifies a multidimensional framework for evaluating blockchain efficiency under more representative conditions. The findings expose latent system behaviors that often remain hidden in standard setups and suggest new directions for both future research and protocol design.

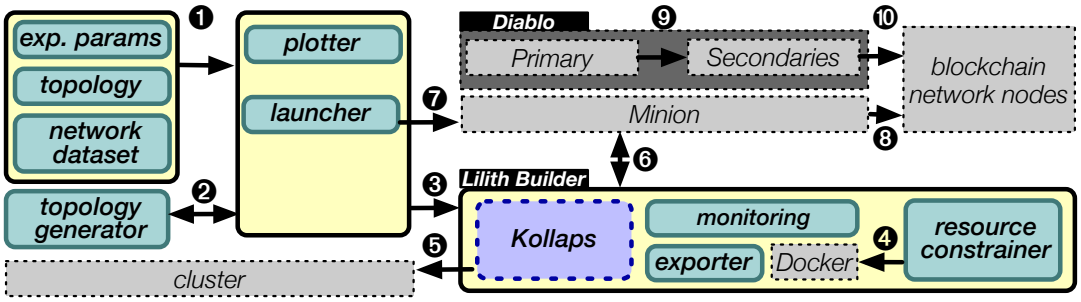


Figure (3.1) Architecture and execution flow of a Lilith experiment (dashed contour lines represent existing components).

The previous chapter (see in particular §2.5) highlighted how current blockchain benchmarking practices remain fragmented: results often rely on point estimates rather than dispersion-aware reporting; workloads and overlays are coupled in ad-hoc ways that hinder comparability; and, most critically, the network is treated as background noise rather than a first-class experimental input. In this setting, even identical protocol stacks can exhibit markedly different tail latencies and recovery behaviors when evaluated under different, uncontrolled overlays. The consequence is a literature rich in numbers but poor in like-for-like comparisons and limited in experimental repeatability.

This chapter introduces Lilith, our proposal to turn those gaps into design requirements for a practical blockchain benchmarking framework. The central idea is simple: if performance and liveness hinge on dissemination paths and transport conditions, then a benchmark must control the overlay and make its parameters explicit. Lilith implements this idea by combining a programmable, topology-aware network substrate with a blockchain workload runner, so that experiments specify not only what transactions are issued and when, but also who hears which message under what link conditions. Rather than seeking Internet-scale fidelity, the framework targets controlled variability: per-run conditions are fixed and inspectable; runs are scripted to reduce incidental nondeterminism; and reports foreground distributions and run-to-run offsets, not just averages.

Lilith is system-agnostic at the orchestration and emulation layers, supports both clusters and large-scale emulation, and integrates new backends through thin adapters without changing the orchestrator. Its novelty is to benchmark real blockchain clients under controlled emulation, making topology and link constraints explicit and reproducible. This enables topology-aware analysis of performance, energy, and run-to-run variability under fixed configurations without relying on opaque cloud networking. It integrates, extends, and coordinates Diablo's workflow and Kollaps's network emulation capabilities. It features a topology generator that creates *ad-hoc* network topologies, which are then fed into the Kollaps module for precise emulation. Additionally, Lilith can enforce arbitrary resource constraints, similar to selecting a specific instance in the cloud. Figure 3.1 illustrates the execution flow of Lilith.

In the integrated setting of Lilith, *Diablo* [154] provides the experiment scaffolding and cross-chain workload execution, while *Kollaps* [17] offers an emulator for overlay

topologies with configurable latency, bandwidth, jitter, and packet loss, among others. Their combination lets us treat the network as an input knob and package full experimental artifacts – scripts, seeds, topology specifications, and analysis notebooks – aligned with community expectations for reproducibility and artifact badging [4, 176]. In turn, this enables a clearer mapping from research questions to experimental conditions: we can ask how overlay structure shapes throughput and tail latency under the same consensus, whether dispersion tightens when conditions are controlled, and how results observed under emulator control should be contextualized against the variance of multi-tenant clouds. The entire experiment lifecycle is automated by using containerized services to improve reproducibility and traceability, in the spirit of general-purpose evaluation frameworks such as Fex [248], but specialized here to blockchain workloads and geo-emulated networking.

Lilith is not tied to a single physical deployment model. It can run on a local cluster, on cloud-provisioned hosts, or on hybrid infrastructures, provided that the execution units used by the benchmark – containers, VMs, or native processes – can be orchestrated and monitored. In a cloud setting, Lilith interacts with the infrastructure at two levels. First, the cloud provider supplies the physical hosts, regions, and instance types on which blockchain nodes, workload generators, and monitoring services are deployed. Second, Lilith and Kollaps impose an explicit experimental network on top of that substrate, by configuring the overlay topology and its link properties, such as latency, bandwidth, jitter, and packet loss. In this sense, Lilith does not assume that the cloud network is naturally homogeneous. Rather, it uniformizes the effective network conditions experienced by the blockchain nodes by fixing the experimental topology, resource limits, software versions, deployment order, and workload profile. This distinction is important for public-cloud experiments. Multi-region clouds expose realistic wide-area effects, but they also introduce uncontrolled variability due to multi-tenancy, routing changes, heterogeneous instance generations, and time-varying background traffic. Lilith can use cloud measurements as input datasets or run directly on cloud instances while enforcing a reproducible overlay through Kollaps. The result is not a claim that the underlying cloud network is identical across runs; instead, the goal is to make the benchmarked communication layer explicit, measurable, and repeatable enough to support controlled comparison across blockchains.

This chapter serves as a conceptual bridge from the challenges of Chapter 2 to a concrete solution path. We do not present an exhaustive system description here; instead, we motivate why Lilith is necessary and what role it plays in the dissertation's methodology. The subsequent sections detail the architecture and execution model, formalize the notion of benchmark profiles, the metric and reporting schema, and the artifact layout used throughout our case studies. The guiding premise remains constant: by making the overlay explicit and reproducible, benchmarking becomes more comparable across heterogeneous blockchains and more honest about uncertainty – turning variability from a confounder into an object of study.

3.2 The Diablo Benchmark Suite

We selected Diablo [154] as a building block since it is the one matching most of the features in Table 2.5 and supports the different blockchain types discussed in §2.1.4. An experiment coordinator called *primary* manages a distributed workload generation mechanism between *secondary* nodes interacting with the specific blockchain via a dedicated client interface, ensuring synchronized evaluations. Unlike many other tools that require an existing blockchain infrastructure, along with the addresses of its nodes, the latest version of Diablo introduces a set of Perl scripts called *Minion* [155], which automate the process of building the infrastructure. Diablo can inject realistic workloads, *e.g.,* smart contracts and transfer transactions, with varying volumes and complexities, and supports multi-region AWS deployments.

3.3 The Kollaps Network Emulator

To assess the impact of network properties on blockchain systems we leveraged Kollaps [17], a decentralized network emulator for large-scale applications that models end-to-end properties (*e.g.,* latency, bandwidth, and packet loss). Its fully distributed model ensures scalability while supporting dynamic changes to the topology, *e.g.,* link or node removals, service disruptions, *etc.* The routing paths of user-level data flows on the emulated topologies can be dynamically determined – as in the link-state protocol Open Shortest Path First (OSPF) that calculates routes on the shortest path tree – based on criteria such as latency or hop count. Kollaps supports native processes as well as VMs and can directly integrate with container orchestrators like *Docker Swarm* and *Kubernetes*. Its network modeling features make it ideal for, though not limited to, cluster environments.

3.4 Network Topology Generation

The first step (Figure 3.1-❶) is to define the experiment parameters, including (*i*) the network dataset, (*ii*) the target topology (Figure 3.1-❷), and (*iii*) additional experiment parameters, *e.g.,* the number of blockchain nodes or the characteristics of the workload.

The network dataset consists of experimental measurements of the network properties related to nodes and regions. By specifying the network dataset structure (a CSV file with columns `src-region`, `dst-region`, `latency`, `throughput`, `hops`), users can conduct experiments and replicate network scenarios captured from various sources. This provides a method to replicate existing deployments such as a cloud environment. Additionally, users should provide the target topology shape that defines how nodes are connected. When the input dataset is collected from a cloud provider, the CSV file represents a measured view of the physical or virtualized underlay, for example region-to-region latency, available throughput, and hop-related information. Lilith then maps these measurements onto the selected logical overlay and lets Kollaps enforce the corresponding link properties during the benchmark. Conversely, when the goal is to compare systems

under a synthetic but controlled cloud-like scenario, the same CSV schema can encode a target profile rather than a passive measurement. This makes cloud benchmarking repeatable at the level that matters for the experiment: the blockchain nodes observe the same configured communication constraints, even if the physical cloud substrate remains externally managed and potentially variable.

In this dissertation, the term network topology primarily refers to the logical communication topology imposed among blockchain nodes by LILITH, *i.e.,* the application-level overlay through which transactions, blocks, and consensus messages are propagated. This topology should not be conflated with the physical Internet or datacenter underlay. The physical underlay may exhibit heavy-tailed or scale-free-like properties at some aggregation levels, especially in wide-area settings, but blockchain protocols do not communicate over a single fixed physical graph. Instead, they construct peer-to-peer overlays, relay layers, validator committees, or permissioned connectivity patterns on top of the underlying infrastructure. Therefore, the selected topologies are not intended to claim that the physical network takes five different shapes. Rather, they let us isolate how different overlay properties – degree distribution, diameter, path redundancy, and hub dependence – affect performance and energy under the same measured latency and bandwidth conditions.

Topologies alter propagation diameter, path diversity, and hub contention, so small timing jitter can shift queueing/backpressure and timeout/commit schedules, shaping tails and failures even under fixed network conditions. In our experiments, we evaluate diverse logical network topologies – fat-tree, full mesh, hypercube, scale-free, and torus – as proxies for common communication patterns observed in blockchain deployments. Dynamic or hybrid overlays may change exact rankings, but not the structural dimensions compared here. Scale-free is included because open wide-area systems may naturally produce heterogeneous degree distributions, where a few well-connected peers or relays act as hubs. The other topologies are included because blockchain deployments are not limited to open public networks: permissioned systems, cloud-hosted validator sets, relay backbones, and committee-based protocols may impose more structured, denser, or more regular communication patterns. Under our emulator, each topology is exercised with geo-latency and bandwidth conditions derived from measured network data, allowing us to test structured fan-out (*e.g.,* fat-tree/torus), dense committee communication (full mesh), and heterogeneous peer connectivity (scale-free) in a controlled manner. LILITH supports the five topologies in Figure 2.2 with nodes as validators, end users as Diablo entities, and gateways as cloud regions connected by the defined topology. Additionally, it utilizes the 10 AWS regions listed in Figure 2.3. Table 3.1(a) summarizes qualitative overlay properties for interpretation. We evaluate all five topologies to avoid cherry-picking and regard topology as a dispersion modulator.

A **fat-tree** topology, often used in data centers, consists of multiple gateways organized in layers. For our emulated network of 10 AWS regions, we allocate 20% (2 gateways per region) at the first level and 50% (5 regions) at the second level. Gateways for the first and second levels are selected based on the lowest latencies in the dataset, while links between gateways at these levels are assigned randomly to limit dynamic routing paths

and promote information flow throughout the topology. Blockchain nodes connect to the gateways at the second level.

In a **full mesh** topology each node is directly connected to every other node, reflecting the connectivity of cloud-based deployments. In such topologies, high-latency links simulate long intercontinental connections. This topology is easy to construct, with all gateways interconnected, simplifying the scan of the network dataset to capture region/gateway pairs and latency information.

A **hypercube** topology connects nodes via binary-based adjacency, forming a multi-dimensional structure common in parallel and IoT systems. We encode each of the 10 AWS regions as 4-bit binary vectors representing numbers between 0 and 9 and generate links by toggling one bit at a time, accepting only valid region indices.

In a **scale-free** topology few nodes have significantly more connections than others [77]. For our 10-AWS-region setting, we map each region to one gateway/node in the topology and then generate the inter-region links through the preferential attachment algorithm [77], so that nodes with higher degrees are more likely to be selected as new connections. Thus, the 10 regions are encoded as the 10 nodes of the scale-free graph, while measured RTTs are assigned to the corresponding inter-region links. This topology resembles Internet-like (WAN) networks [77].

A **torus** topology connects nodes in a wrap-around grid, enabling efficient data transfer, and is often used in supercomputing [323]. We construct a 2D torus (2 rows and 5 columns) for 10 AWS regions, placing one gateway per slot. Starting with the lowest-latency region pair in the first column, we iteratively add columns by selecting new region pairs linked to the most recently added gateways, prioritizing low latency. Each gateway connects to its previous column and adjacent row gateway.

Table 3.1 summarizes their structural properties, as well as aggregated statistics over the imposed throughput and latency properties. Specifically, we report: *degree*, the average number of neighbors per gateway; *links*, the total number of links; and *average latency* and *average throughput*, computed from the respective network datasets used to define each topology.

By incorporating latency and throughput into the structure of the network dataset, evaluators can construct customized topologies that prioritize specific performance goals, such as low latency over high throughput. Organizing the dataset accordingly and selecting regions based on these metrics enable more targeted and meaningful performance assessments. Furthermore, with Kollaps's routing selection mechanism, we can determine whether to optimize for the best latency or the fewest hops in the link selection.

3.5 Built-in Workloads

Lilith ships with a set of ready-to-use configurations that cover both the demand side (workloads) and the supply side (blockchain backends), including setups that reproduce configurations from prior work [154]. The goal is to enable immediate, comparable experiments without bespoke setup, while still allowing users to tailor profiles to their

own questions. §2.3 already introduced the workload families and their semantics (payments, market/exchange bursts, gaming telemetry, and stress/abuse). Here we focus on what Lilith bundles as executable configurations, with explicit provenance and default parameters for reproducibility.

Unless stated otherwise, the workloads DDoS, FIFA, GAFAM, Gaming, and VISA are integrated from the Diablo benchmark suite [154]. In Lilith, we preserve their application semantics (transaction mix and contract logic, when applicable) and expose them through our declarative configuration interface. This thesis contributes one additional bundled workload (PayPal) and introduces two explicit parameter updates with respect to Diablo: (*i*) VISA is recalibrated to publicly reported throughput figures (order-of-magnitude calibration) and (*ii*) FIFA is executed at an approximately 10 times higher injection rate to probe peak-load behavior. They span (*i*) payment-style transfer workloads at moderate and high transaction rates (PayPal and VISA), (*ii*) bursty and contention-heavy smart-contract workloads derived from application traces (FIFA, GAFAM, and Gaming), and (*iii*) stress-oriented overload conditions (DDoS). Accordingly, the injected TPS values are controlled benchmarking inputs rather than claims that all profiles mirror current production traffic. We keep offered-load intensity identical across systems to expose both stable and near-saturation regimes. Overload outcomes are treated as informative saturation/near-collapse outcomes (variance/failures).

The workload interface is not limited to single transfer transactions or isolated single-contract invocations. In principle, Lilith can support more realistic application-level patterns, including nested smart-contract requests, whenever the target blockchain and its client adapter expose the required deployment and invocation primitives. A nested smart-contract workload can be represented as a parameterized call graph, where one entry transaction triggers a sequence of contract-to-contract calls, cross-module invocations, or state-dependent execution paths. Such a profile would let the benchmark control additional dimensions, including call depth, fan-out, read/write set overlap, state hotspots, gas or fee limits, and the probability of conditional branches. In the experiments of this dissertation, we focus on the bundled workload portfolio – transfer-only workloads and compact smart-contract workloads – because they provide reproducible, cross-system stress profiles and keep the comparison tractable across heterogeneous execution environments. Nested smart-contract requests are therefore not part of the reported measurements, but they are compatible with the framework's design: they would require extending the workload generator and the blockchain-specific adapter, rather than changing the network-emulation or orchestration pipeline.

DDoS. (Diablo workload, unchanged defaults.) A transfer-only, high, flat injection over a short horizon (constant open-loop pacing), used to stress robustness under sustained overload regimes [97, 145, 154]. The default bundled profile injects 10,000 TPS for 120 s (Table 3.1b).

FIFA. (Diablo workload, modified rate.) FIFA models a ticketing-style flash crowd inspired by the FIFA 1998 World Cup website workload [27]. It targets a compact, highly contended counter-like smart contract (hot-spot keys). Compared to the default Diablo configuration [154], Lilith runs FIFA at an approximately 10 times higher injection rate (45,000 TPS for 100 s by default) while keeping the contract logic and access hot-spots unchanged.

GAFAM. (Diablo workload, unchanged defaults.) GAFAM represents a bursty market/exchange scenario over a compact state, with buy/check actions defined on a fixed set of stock keeping units (SKUs), derived from trace-inspired dynamics (an initial short-lived peak followed by a low steady regime) [154]. In Lilith, we bundle the same GAFAM application/workload profile as implemented in Diablo [154]; the default configuration peaks at 20,000 TPS and then stabilizes around 100 TPS (Table 3.1b).

Gaming. (Diablo workload, unchanged defaults.) Gaming replays a high-frequency update trace inspired by an online battle arena game (*Dota2*) [154, 304]. We bundle the same Diablo Gaming profile [154]: 276 s at an almost constant 13,000 injected TPS by default.

PayPal. (Novel contribution, new bundled workload.) PayPal is a steady-regime, transfer-only payment stream at low-to-mid intensity (stationary, diffuse access). Its profile is derived from the 193 TPS average in [229]. For simplicity, it is implemented as a constant-rate transfer-only profile (200 TPS for 300 s by default), reusing only the generic transaction-submission path available in Lilith (*i.e.,* no third-party smart-contract code).

VISA. (Diablo workload, recalibrated rate.) VISA is a higher-intensity transfer-only payment proxy still below overload, useful to probe steady-regime latency/throughput under open-loop pacing. The workload logic itself is integrated from Diablo [154]. Starting from the 1,770 TPS figure reported in [154], we recalibrate the injection rate to 1,800 injected TPS for 300 s (Table 3.1b).

Extensibility is supported by allowing new workloads to be added through the same declarative interface used by the bundled profiles (rate functions or trace replay, payload encoders, and optional contract stubs), without changing the orchestrator.

3.6 Blockchains under Test

The default backend set covers heterogeneous blockchain designs in consensus, execution, and networking so that comparisons are not bound to a single family. We include a public PoS chain with fast finality (Algorand), a permissioned BFT design (Diem with HotStuff), two EVM-based stacks with distinct finality models (Ethereum Clique and Quorum IBFT), and a high-throughput PoS+BFT system with a parallel runtime (Solana). To ensure consistency, we used the blockchain configurations from [154], with each blockchain

Table (3.1) Selected topologies (a), workloads (b), and blockchains (c).

Topology	Degree	Links	Avg. Lat. (ms)	Avg. TPut (Mbps)	Degree profile	Graph diameter	Path diversity	Centralization
fat-tree (k=4, l=2)	k	$k^l/2$	102	712	structured	low	high	low
full mesh (N nodes)	$N-1$	$N(N-1)/2$	194	600	uniform	very low	high	low
hypercube (n=4)	n	$n \cdot 2^{(n-1)}$	208	560	uniform	low	medium	low
scale-free (N nodes)	(Based on Power Law)		218	432	hub-heavy	medium	medium	high
torus (N=5, n=2)	$2n$	$n \cdot N^n$	197	728	uniform	high	low	low

(a) Topologies integrated in Lilith.

Workload	Type	Scenario	Duration (s)	Injected (TPS)
DDoS	Transfer transaction	Constant rate	120	10,000
FIFA	Smart contract	High sending rate	100	45,000
GAFAM	Smart contract	Burst	180	20,000 down to 100
Gaming	Smart contract	Intensive	276	13,000
PayPal	Transfer transaction	Constant rate	300	200
VISA	Transfer transaction	Constant rate	300	1,800

(b) Workloads integrated in Lilith.

Blockchain	Consensus	VM	DApp	Block Finality (s)	Claimed TPut (TPS)
Algorand	BBA* [76, 147]	AVM	PyTeal [11]	3.3 [12]	7.5K [12]
Diem	HotStuff [347]	MoveVM	Move	100 [258]	60–1K [350]
Ethereum	Clique [308]	geth	Solidity	10–20 [221]	10–15 [282]
Quorum	IBFT [280]	geth	Solidity	2–15 [228]	0.7K–2.5K [98]
Solana	TowerBFT [340]	Sealevel	Solang	12 [154]	65K [84]

(c) Blockchains integrated in Lilith.

version specified by the respective repository commits. We pin commits for stability and comparability with prior work [108, 110], avoiding confounding version drift with dispersion; the methodology is version-agnostic and can be rerun on newer releases. Their high-level traits are reported in Table 3.1 and detailed in §2.1; here we summarize the rationale for their inclusion.

Algorand [76, 147] uses a pure PoS consensus algorithm, selecting nodes via sortition to propose blocks. Transactions are finalized immediately upon inclusion, minimizing fork risks. The platform provides a blocking API for transaction confirmation and uses WebSockets over HTTP (TCP) for node communication, leveraging a gossip protocol [91]. Nodes validate messages to avoid duplicates, with default settings allowing up to 15 connections per IP and 2,400 incoming connections per port. During our experiments, we focused on detecting transaction commits by polling the blockchain only after blocks were added, which significantly boosted performance. Tests were conducted using Algorand at commit `116c06e`.

Diem [38] uses an adapted version of HotStuff [347] for deterministic finality with low communication overhead. Its nodes limit memory pools to 100 transactions per signer, addressed in tests by submitting transactions from 2,000 accounts. Our testing was based on Diem's testnet branch at commit `4b3bd1e`. Though no longer active, Diem provides valuable insights into permissioned infrastructures.

Ethereum [331] is a public blockchain platform for decentralized applications (DApps) and smart contracts. In our experiments, the Clique PoA variant [308] was used, with blocks added at 1-second intervals by validators in a round-robin manner. Dynamic fee

adjustments were configured to handle gas variability introduced by the London update (August 2021). Our benchmarks employ the Go implementation of the Ethereum protocol, specifically at commit `72c2c0a`.

Quorum [117] is a permissioned, enterprise Ethereum client that preserves Ethereum JSON-RPC while adding permissioning and optional private transactions via an external transaction manager. Following previous studies [39], we configured it with IBFT [280] at commit `919800f` to address message delays and vulnerabilities in PoA systems [308].

Solana [342] faces forks like Ethereum and requires 30 confirmations to finalize transactions [301]. Blocks are added every 400 milliseconds, with a simplified data structure and EdDSA signature replacing ECDSA. Solana's API supports commitment levels and block monitoring, with our evaluation periodically fetching block hashes within typical DApp time constraints. We used commit `0d36961`.

For each backend, Lilith bundles minimal adapters (RPC/ABI, transaction encoders, confirmation/finality probes) and default node/client configurations aligned with prior work [154]. This makes Lilith system-agnostic at the orchestration and emulation layers: new blockchains can be integrated by implementing the thin adapter interface (submit, poll/confirm, encode/decode, deploy), without modifying the orchestrator or the workload engines.

3.7 Benchmark Execution

Lilith manages the installation of Docker daemons and Kollaps on the cluster nodes, deploys the necessary experiment images, and sets up a network to distribute the experiment among the machines (Figure 3.1-❸). The initialization phases can also enforce resource limits (Figure 3.1-❹) to mimic specific computing or networking constraints (leveraging `cgroup`'s container properties). The architecture integrates with Docker containers, though a similar approach can be used with other execution units (*e.g.,* VMs, native processes) with additional engineering efforts. Once all steps are completed, the services can be initiated (Figure 3.1-❺).

The Lilith builder interacts (Figure 3.1-❻) with Diablo's Minion so that the launcher (Figure 3.1-❼) triggers Diablo's mechanics to start its benchmark suite. Diablo's Minion [155] component is in charge of installing and bootstrapping the blockchain network (Figure 3.1-❽). This process includes installing both Diablo and the blockchains dependencies on the designated machines. The primary node coordinates the experiment (Figure 3.1-❾): it orchestrates the execution of the various secondary nodes. To ensure a ready blockchain environment where validators are aware of the current blockchain state and clients can actively send transaction requests, the primary coordinates the blockchain configuration by creating accounts and the genesis block, which is then distributed to all blockchain nodes.

A given workload effectively starts when the secondaries inject the workload transactions into the blockchain network (Figure 3.1-❿). Once finished, they collect and transmit

the results back to the primary. The Lilith architecture includes a modular monitoring component, which is in charge of collecting network usage with the `vnstat` tool [312]. Also, an exporter module enables further refinements of both benchmark results and execution trace during post-processing. The exporter gathers network monitoring data, validator logs from the blockchain nodes, and benchmark execution records from the primary node. A script plots the results as shown in the next chapters, allowing performance and network metrics to be checked.

State initialization and seeding. In the current Lilith workflow, blockchain networks are executed in a fresh-start regime inherited from Diablo/Minion: hosts are reset between experiments and each run starts from a clean chain data directory and a newly generated genesis configuration. Before traffic injection, the bootstrap phase performs only minimal state seeding required by the benchmark portfolio, namely resource generation such as funded account creation and, for smart-contract workloads, deployment of the benchmark contracts (and their initial parameters) so that secondaries can immediately invoke them. This matches Diablo's execution model, where the primary performs resource generation, including contract deployment, before distributing executable work to secondaries.

Importantly, Lilith does not preload large historical snapshots (*e.g.,* mainnet-like states) by default. Therefore, in this dissertation the measurements primarily characterize protocol behavior under controlled overlays and workloads with a small, benchmark-initialized state, rather than long-term storage maintenance under multi-year chain growth.

Implications of large state. In production, validators maintain a large and evolving state in a key-value store (*e.g.,* LevelDB in Geth and Ethereum-derived clients such as GoQuorum, as well as RocksDB in Diem and Solana), which can exceed memory and shift bottlenecks toward cache behavior and disk I/O. Such effects can significantly impact throughput and latency compared to an empty or small-state deployment. Extending Lilith to benchmark this regime would require explicit snapshot/state preloading mechanisms per client; this is orthogonal to the overlay emulation pipeline and is left as future work.

3.8 Metrics Gathering

Lilith measurement pipeline combines client-side traces, node/validator logs, overlay counters, host-level resource monitors, and (when available) energy probes. Time is aligned by the orchestrator so that per-source timestamps can be merged without post-hoc drift correction.

We report five primary signals. First, the *commit ratio* quantifies the fraction of submitted transactions that are durably committed (committed/submitted), making admission failures visible under overload or churn. Second, *throughput* counts committed TPS over sliding windows, with both sustained and peak values. Third, *latency to confirmation* (often called block finality in prior work [154]) measures end-to-end time

from issuance to first confirmation at the chain's finality layer. Fourth, *network load* captures per-interface throughput in Mbps and effective overlay utilization. Fifth, energy probes based on Intel RAPL provide host-level processor/package energy indicators and cumulative consumption (kWh), normalized as energy per committed transaction for cross-run comparability. These values are a controlled proxy for computational energy, not a full whole-system estimate.

Besides these primaries, Lilith collects host metrics (CPU, memory, disk I/O) to assist diagnosis. All raw signals are exported as CSV/JSON with an explicit schema and are bundled with plotting notebooks to reproduce figures as in the next chapters. To preserve comparability, we mandate dispersion-aware reporting (run-to-run offsets under identical seeds) rather than point estimates alone.

3.9 Testing Configurations

Lilith can run experiments on different testbeds (*e.g.,* single machine, clusters, cloud platforms). By using Kollaps, Lilith deterministically emulates WAN topologies (including externally defined ones) with per-link controls (*e.g.,* latency/bandwidth) and latency-aware routing. We deploy clients in Docker Swarm with host networking, fixed cgroup caps (8 vCPUs, 16 GB of RAM), and host-mounted volumes for state/logs; the container stack is identical across runs, so it does not confound within-configuration dispersion [133]. Unless noted, nodes run inside Docker with pinned images and CPU affinities; an analogous setup with virtual machines or native processes is feasible but not required. The setup comprises:

- **Deployment & isolation.** All nodes run on one on-premises cluster; services are containerized with fixed CPU/RAM (pinned) to avoid cross-run interference.
- **Network emulation.** Kollaps enforces per-link latency/capacity (and time evolution) on overlay edges for the selected topology; validators are not physically geo-distributed.
- **Run protocol.** For each configuration, we run 10 independent trials; each resets state (restart + fresh ledger), reapplies emulation, replays the trace, then tears down.
- **Measurement window.** Metrics use a fixed post-warm-up steady-state window whose bounds are detailed in the related scripts.

We vary three classes of factors:

- *Topology and placement.* We instantiate the families in Table 3.1(a) and assign validators, seed nodes, and clients to regions/gateways according to the target graph; per-link latency, bandwidth, jitter, and packet loss are taken from the selected network dataset and enforced by Kollaps.

- *System scale and resources.* We tune the number of validators per region and the client fan-out, then we bound host resources (CPU cores, memory limits, and per-container I/O) to emulate different instance classes.
- *Demand and resilience.* We execute the bundled workloads in open- or closed-loop mode and inject controlled perturbations – packet drops, bandwidth caps, latency spikes, link and node failures, and temporary partitions – to observe admission, backlog growth, and recovery.

Fault injections are scripted with absolute times or event triggers (*e.g.,* "after backlog exceeds threshold") and their effects are reflected in both performance and overlay traces. This design allows us to answer like-for-like questions – *e.g.,* how a change in overlay diameter affects tail latency under the same workload and resource envelope – without confounding cloud variance.

Chapter 4

Performance Efficiency: Impact of Network Topologies

This chapter, which builds upon the article [108] that received the ACM Artifact Evaluation Badge v1.1 and the Best Student Paper Award, uses the Lilith benchmarking framework to present an extensive evaluation of five blockchain systems under diverse network topologies and workload types. We start by recalling the issue of performance evaluation (§4.1) and defining our evaluation setup (§4.2). The results show that key performance metrics such as throughput and latency are highly sensitive to network structure, exposing critical bottlenecks and performance variabilities often overlooked in previous benchmarking efforts (§4.3). We finally discuss outcomes and limitations of our approach (§4.4).

4.1 Performance Evaluation of Blockchains

As discussed in Chapter 2, systematic and reproducible benchmarking of blockchains remains an open problem. An essential aspect of benchmarking blockchains is to control the communication topology experienced by blockchain nodes, namely the application-level overlay together with the latency, bandwidth, loss, and routing properties inherited from the physical underlay [17, 205]. This distinction is important because the same physical infrastructure can host different blockchain overlays and the same overlay can behave differently when mapped onto different wide-area latency and bandwidth conditions. Despite the significant impact of network dynamics (*e.g.,* changes in link properties, node failures, *etc.*), few studies have examined their influence on blockchain performance, with notable examples including the effect of network latency on key parameters (*e.g.,* block size, frequency, or propagation [104, 240]) as well as node failures and network contraction (*e.g.,* China's 2021 ban on Bitcoin mining [267]) and their consequences on the network [272].

Our research aims to validate cluster as a cost-effective, reproducible environment for the study of the impact of network topologies on blockchain performance. Aside from prior work on hypercube topology benefits for Bitcoin [313], to the best of our knowledge our work is the first in-depth experimental study across multiple real-world blockchains.

In this chapter we measure the performance of five industry-grade blockchain systems – Algorand [76,147], Diem [38], Ethereum [331], Quorum [117], and Solana [342] – across five network topologies – fat-tree, full mesh, hypercube, scale-free, and torus – under different workloads – smart contract requests and transfer transactions. Although blockchains often adopt topologies suited to their design, such as fat-tree or hypercube for private ones like Diem and scale-free for public systems like Ethereum, this study explores blockchain topology reconfigurations to improve efficiency without altering their public or private nature. Here are our key contributions:

- We release a 12-month-long network trace of cloud performance monitoring (available at `https://zenodo.org/records/11457020`) that captures variability and challenging network conditions observed in cloud-based deployments.
- We release several ready-to-use network topologies and traces, enabling practitioners to experiment with distributed systems beyond our blockchain measurement study.
- The main observations are: (*i*) Algorand and Diem show consistent performance across various network topologies; (*ii*) Ethereum has the lowest TPS rate but remains resilient to network issues, *i.e.,* packet loss, link congestion, node crash, and increasing latency; (*iii*) full mesh and hypercube topologies improve performance across all tested blockchains; (*iv*) torus topology excels under heavy workloads; (*v*) increasing the number of nodes reduces commit rate and raises block latency, while higher network bandwidth has no effect on latency; (*vi*) Quorum is the blockchain more affected by network dynamics.

For our measurements we use Lilith, the benchmarking framework previously presented in Chapter 3. The goal is to achieve reproducibility by having control over the network topology.

4.2 Evaluation Setup

This section presents our experimental setup by detailing the experiment configurations. Specifically, we analyze the impact of five distinct topologies on five blockchains running six different workloads. We focus on four key performance metrics integrated in Lilith: (*i*) *commit ratio*, *i.e.,* the ratio of submitted to committed transactions; (*ii*) *throughput*, measured in terms of TPS; (*iii*) *block latency*, *i.e.,* the average time required to finalize transactions, also referred to as *block finality* [154]; and (*iv*) *network load*, measured in Mbps.

Our evaluation intends to provide insights into the following specific research questions:

RQ1 How do topologies affect blockchain performance and which is the optimal topology for each blockchain?

RQ2 What is the impact of network perturbations on performance, such as packet loss, congestion, node failures, and increased latency, and which blockchain is most affected?

4.2.1 Experimental Assets

Recalling the definitions introduced in Chapter 2 and refined in Chapter 3, our experiments combine five canonical network topologies – fat-tree, full mesh, hypercube, scale-free, and 2D torus – with six workloads that span transfer-only and smart-contract activity (DDoS 10,000 injected TPS; FIFA 45,000; GAFAM burst 20,000→100; Gaming 13,000; PayPal 200; VISA 1,800). We exercise five representative blockchains – Algorand (BBA*/AVM), Diem (HotStuff/MoveVM), Ethereum in its Clique PoA variant (geth/Solidity), Quorum with IBFT (geth/Solidity), and Solana (TowerBFT/Sealevel) – using the configurations and client-side submission/confirmation discipline established in [154]. For comparability and reproducibility, we pin repository commits for each client (see §3.6) and reuse the same API-level confirmation policy per chain.

4.2.2 Testing Configurations

Our cluster is composed of seven Dell PowerEdge R630 server machines, each equipped with two 16-core/32-thread Intel Xeon E5-2683v4 clocked at 2.10 GHz CPU and 128 GB of RAM, connected by a Dell S6010-ON 40 GbE switch. The nodes run Linux Ubuntu 22.04 LTS, kernel v5.15.0-107-generic.

We evaluate the impact of topologies by using different configurations, taking into account (*i*) the number of nodes per region, (*ii*) varying link latencies, and (*iii*) the bandwidth capacity for each blockchain node. We also study how the blockchains are affected by faults happening in the network, *i.e.,* emulating packet drop, congestion, node failures, and increased latency, to observe failures in the reception/transmission of messages between nodes during the workload execution. We present our results in the next section.

4.3 Measurement Results

We study the impact of topologies on the various blockchains by focusing on the three key performance metrics established at the beginning of §4.2. To ensure statistical validity, results are averaged over ten independent runs; an exception is made for dynamic experiments (see §4.3.4), for which – due to their length and time constraints – we average over three runs.

4.3.1 Small Deployment

We begin by deploying one node per region, for a total of 10 nodes. The results are depicted, for each topology, on the left side of Figure 4.1. For less demanding workloads (GAFAM and PayPal), Algorand and Diem achieve a commit ratio greater than 80%, which is close to 100% for PayPal, with Diem slightly outperforming the former. Although the results are consistent across different topologies, we observe that Algorand experiences a 50% drop in TPS (to 78 TPS) on a scale-free topology with the GAFAM workload.

Quorum achieves a commit rate above 90% for the PayPal workload in hypercube (Figure 4.1e) and torus (Figure 4.1i) topologies, due to low congestion in the torus and the high connectivity of hypercube, which ease transaction retrieval. Solana exceeds 75% in full mesh (Figure 4.1c) and hypercube (Figure 4.1e), whereas congestion in the remaining topologies reduces the rate to below 50%.

Under heavy workloads Algorand performs better with a rate below 8% for transfer transactions (DDoS, see Figure 4.1c) and below 3% for smart contract transactions (Gaming and FIFA, see Figure 4.1a). This is explained by Algorand's high capacity transaction pool, which can handle up to 75,000 transactions [10].

As noted in prior works [154], Quorum fails to show commits for the most demanding workloads. We ascribe this to the leader bottleneck in the BFT consensus used in Quorum, which can saturate memory pools or network queues under high workloads.

Ethereum maintains a commit rate below 7% (see Figure 4.1a). It achieves higher block latency ($\geq$148 ms) for PayPal and VISA (less demanding transfer workloads) compared to latency below 90 ms for high-demanding workloads (DDoS, FIFA, Gaming). This is mainly due to the period between its consecutive blocks regardless of the network bandwidth.

Algorand and Diem obtain the best results with the PayPal workload independent from the topology. Interestingly, VISA is too demanding for all blockchains, with a commit rate always below 32%. Algorand achieves the best results with the DDoS workload (up to 8% commit rate over 10,000 injected TPS). Regarding smart contract executions, we observe the best results with Algorand and Diem (achieving 100 TPS for FIFA, up to 150 TPS for GAFAM, and around 200 TPS for Gaming). Quorum and Solana obtain up to 85 TPS for non-intensive smart contracts. Finally, benchmarks show poor results for all intensive smart contract workloads (FIFA and Gaming).

4.3.2 Scaling the Network

We now focus on larger networks, deploying four nodes per region for a total of 40 nodes, limited by our cluster resources to avoid overload. The results for each topology are shown on the right side of Figure 4.1. As expected, a higher number of nodes per region results in more links contending for the region gateway capacity. Even with a larger set of available nodes, Quorum fails to handle high workloads. This is expected since the consensus protocol used by Quorum has a quadratic cost with the number of nodes. Solana commits only for the GAFAM workload for all the topologies, while PayPal and VISA just with torus (see Figure 4.1i), failing for all others. We explain this behavior with network congestion: Solana may drop newly submitted transactions if the outstanding rebroadcast queue of an RPC node exceeds 10,000 transactions [298], preventing them from being forwarded to the leader. Additionally, validators can become unresponsive during periods of high load [219, 299].

Ethereum achieves a low TPS regardless of the number of nodes in the network. Algorand is the least affected by network size or by the additional network congestion.

Notably, credential messages (*e.g.,* used to authenticate participants within Algorand) are significantly smaller in size (between 100 and 200 bytes) compared to block proposals (5 MB). Their smaller size allows them to propagate more swiftly through the network. This faster propagation helps peer nodes prioritize block proposals more effectively, thereby alleviating network congestion.

4.3.3 Network Load

We study how the topology affects network traffic among the blockchain nodes. Figure 4.2 depicts these results. We notice that workloads with transfer transactions are lighter compared to smart-contract-based ones. For instance, by looking at benchmarks with one node per region (see Figure 4.2 left side), the PayPal workload consumes up to 270 Mbps on hypercube topology, while FIFA consumes up to 2,948 Mbps on the same topology. Moving towards four nodes per region (see Figure 4.2 right side), we notice a general increase of network data along the GAFAM workload. In this case, the PayPal workload consumes up to 470 Mbps across all the tested topologies, while FIFA consumes up to 4,405 Mbps on every topology. With a configuration of one node per region Solana shows the highest network throughput, followed by Algorand and then Diem. But with four nodes per region Algorand shows a higher network throughput, followed by Solana and Quorum. On the other hand, Ethereum typically has the lowest bandwidth utilization, followed by Quorum with one node per region and Diem with four nodes per region.

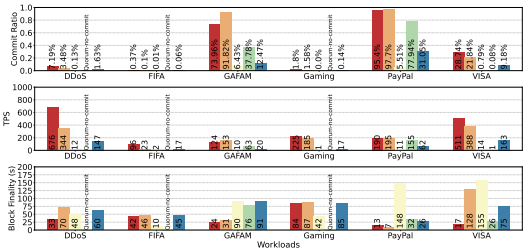
4.3.4 Network Dynamics

We now examine the effects of network degradation – packet loss, congestion, node failures, and increasing latency – on blockchain performance. These experiments used a full mesh topology and one node per region. For the packet loss, congestion, and node failure experiments, we employed the PayPal workload (200 injected TPS for 300 seconds) to test on a simple and sustained workload. Dynamic events were triggered 60 seconds into the workload, with normal conditions restored after another 60 seconds.

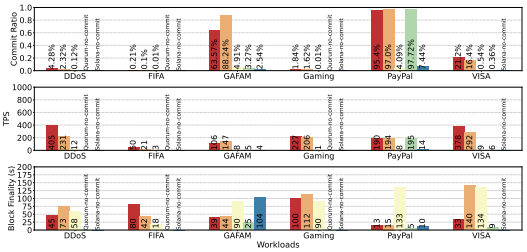
In the packet loss experiments (see Figure 4.3a), we applied discrete packet drop percentages (0%, 10%, 20%, 30%) to 1/3 of the links, selected at random according to a uniform distribution. As expected, the performance of all systems degraded substantially with higher packet loss. Algorand and Solana remained robust at 10% loss. Interestingly, Quorum's throughput sometimes improved, likely due to random link selection allowing more efficient transaction distribution or isolating non-congested nodes.

During the bandwidth congestion experiments (see Figure 4.3b), we selected 10%, 20%, and 30% of the links at random according to a uniform distribution and reduced their bandwidth to 20% of their original capacity. Blockchains showed consistent performance under increased congestion, with Solana being the most resilient.

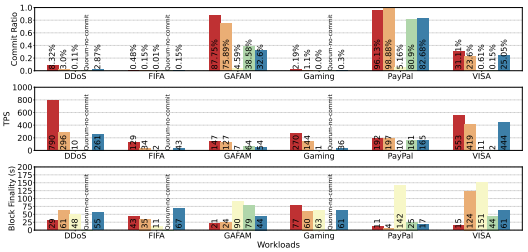
Node crash experiments (with 10%, 20%, and 30% of total nodes crashed) revealed that Algorand struggled significantly with 30% crashes (see Figure 4.3c), while Ethereum, despite lower performance, was the most resilient. Diem, Quorum, and Solana exhibited



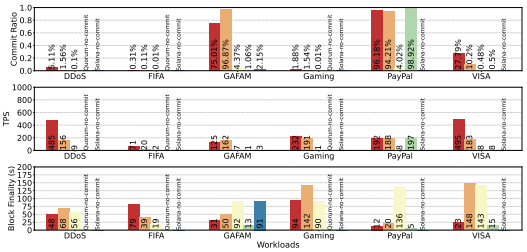
(a) Fat-tree, one node/region.



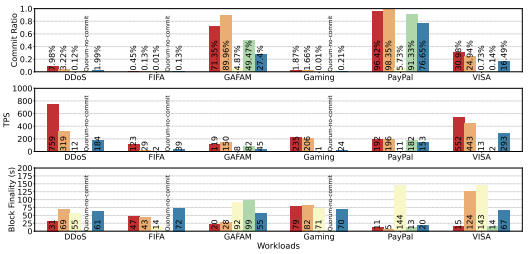
(b) Fat-tree, four nodes/region.



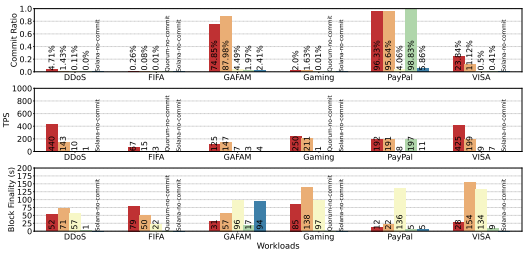
(c) Full mesh, one node/region.



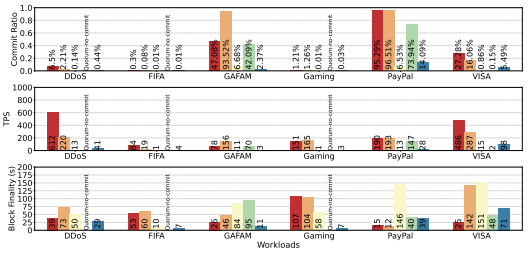
(d) Full mesh, four nodes/region.



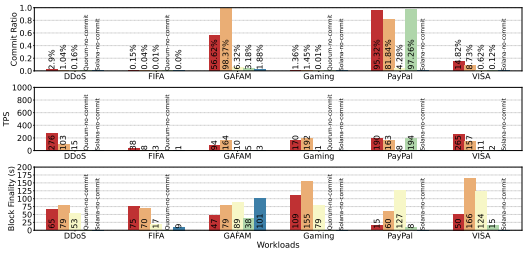
(e) Hypercube, one node/region.



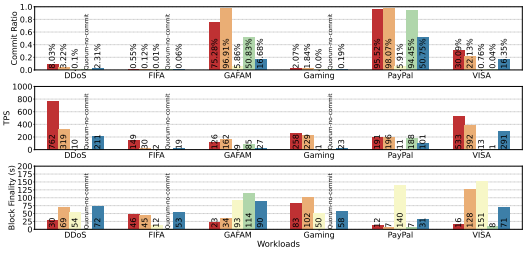
(f) Hypercube, four nodes/region.



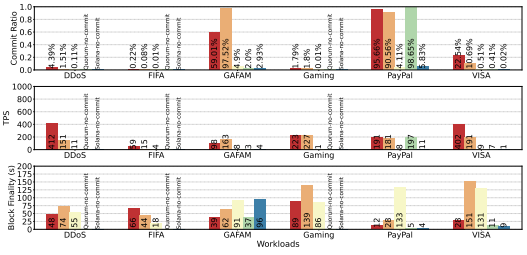
(g) Scale-free, one node/region.



(h) Scale-free, four nodes/region.



(i) Torus, one node/region.



(j) Torus, four nodes/region.

Figure (4.1) Blockchain performance across various workloads using the 2023 AWS dataset (**Algorand**, **Diem**, **Ethereum**, **Quorum**, **Solana**).

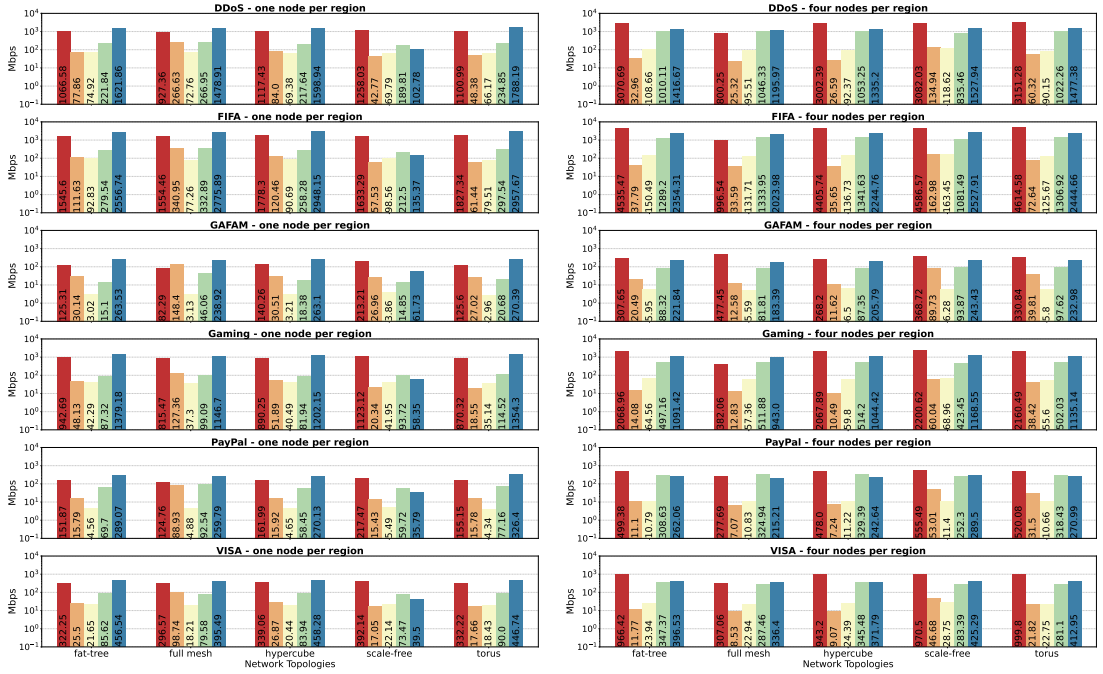


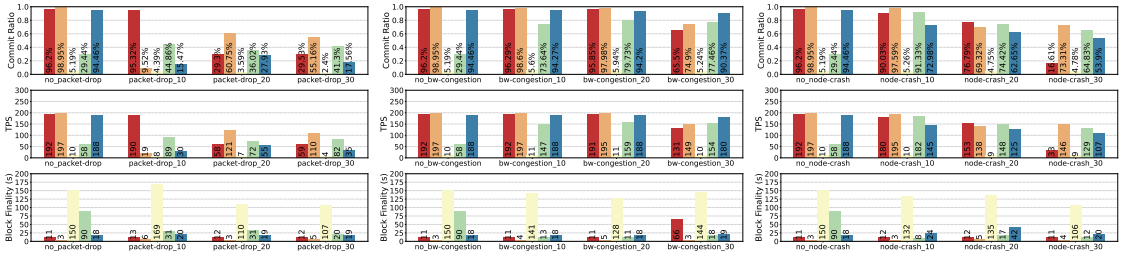
Figure (4.2) Network load (Mbps) during workload execution (**Algorand**, **Diem**, **Ethereum**, **Quorum**, **Solana**).

predictable, gradual performance drops as crashes increased.

For the increased latency experiments (see Figures 4.4 and 4.5), we extracted minimum and maximum latency values for region pairs from the network dataset, ranging approximately from 30 ms to 500 ms. We employed adapted and extended workloads: for the PayPal workload, we conducted a 20-minute experiment where latencies began to increase at the 10-minute mark (N1), gradually peaking over 200 seconds (N2) and maintaining that level for an additional 60 seconds (N3) before returning to baseline for the last 6 minutes; in the GAFAM workload, we performed a 5-minute experiment where latencies rose after the first minute (S1), peaked after another minute (S2), and remained at the maximum until the third minute (S3), gradually reverting to initial levels thereafter. These scenarios were repeated starting with a baseline factor of 1×, followed by incremental factors of 10×, simulating real-world conditions such as peak transaction periods or network congestion during significant market changes [157]. As can be seen from Figures 4.4 and 4.5, blockchains utilizing traditional consensus mechanisms, like Quorum, exhibited longer recovery times as latencies increase. Algorand also demonstrated repercussions with the PayPal workload due to rising latency, while Diem, Ethereum, and Solana showed more robust behavior as latencies increase.

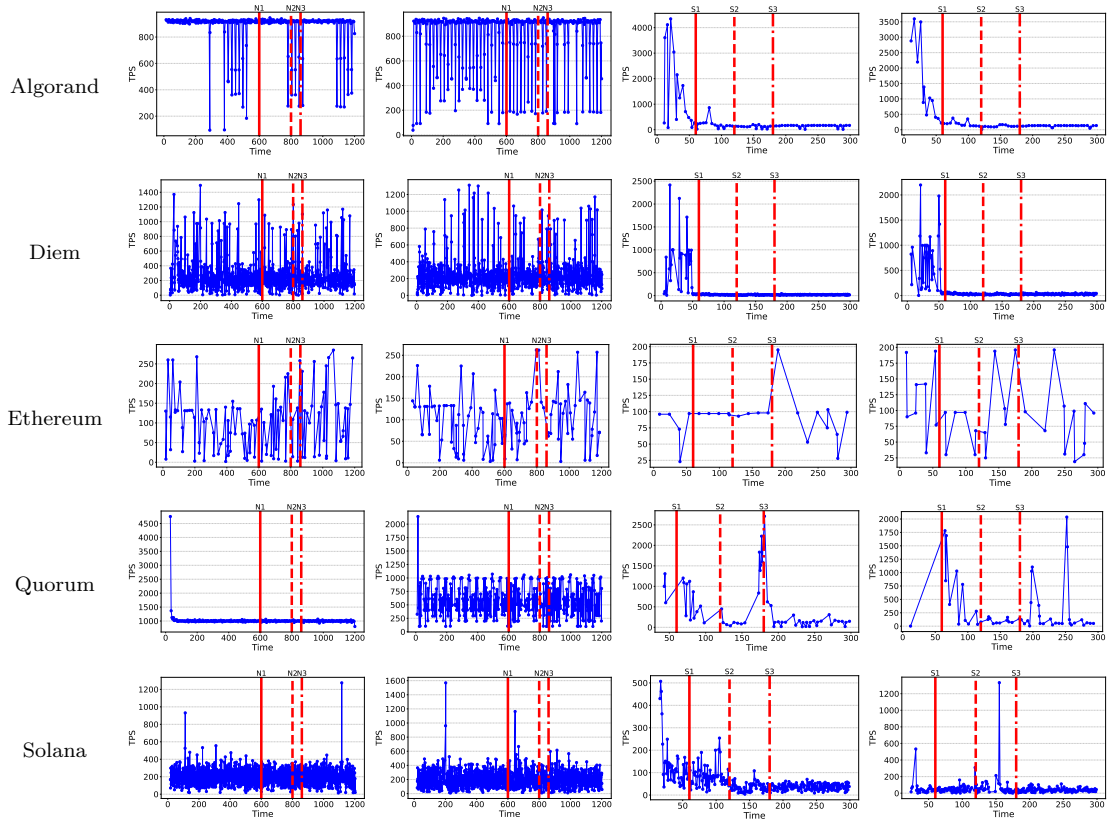
4.4 Discussion and Limitations

Among the blockchains presented in §4.2 and used in our experiments of §4.3, Algorand and Diem are the most performant ones, exhibiting more consistent results (*e.g.*, higher



(a) Packet drop. (b) Link bandwidth congestion. (c) Node crash.

Figure (4.3) Network dynamics, one node per region, full mesh topology (**Algorand**, **Diem**, **Ethereum**, **Quorum**, **Solana**).



Latencies 1× Latencies 10×

Figure (4.4) Benchmark with increasing latencies (PayPal workload). Red lines mark latency variation events (see §4.3.4).

Latencies 1× Latencies 10×

Figure (4.5) Benchmark with increasing latencies (GAFAM workload). Red lines mark latency variation events (see §4.3.4).

TPS) across different topologies. Note that the results from Algorand are remarkable, as it is a public blockchain with a decentralized consensus mechanism.

As the number of nodes per region/gateway in the network increases, the blockchains exhibit a reduced commit rate while their average block latency augments. This is expected since most blockchains (and in particular the ones under test) are known to scale poorly with the number of nodes due to the inefficiencies of the underlying consensus protocol [239].

In the light of our experimental results, we are now able to answer the research questions posed in §4.2:

RQ1 Torus generally allows for better results with more demanding workloads. We achieve the best overall results, across the five tested blockchains, atop full mesh and hypercube, due to their high degree and link capacity.

RQ2 Our results demonstrate that Diem and Solana are the most affected by packet loss, followed by Quorum and Algorand. Classical consensus mechanism, as in Quorum, are less reactive to increasing latencies. Despite consistently exhibiting lower TPS compared to the others, Ethereum is the least affected by these dynamic network events.

These results should not be interpreted as a one-to-one mapping between blockchain type and physical network topology. Public blockchains often run over open peer-to-peer overlays deployed on top of a heterogeneous Internet underlay, and such overlays may exhibit scale-free-like or heavy-tailed connectivity patterns. Permissioned or consortium systems, instead, may deliberately impose more structured overlays because membership, validator placement, and relay infrastructure are easier to control. The contribution of our experiments is therefore comparative rather than descriptive: by keeping the physical testbed and workload fixed while changing the logical overlay, we quantify how different overlay properties – density, diameter, path diversity, and hub concentration – affect throughput, latency, and robustness. In this sense, topology is treated as a controllable design dimension rather than as a claim that real deployments always instantiate one specific graph family.

Limitations. Lilith turns out to be resource-intensive for large-scale blockchain simulations, especially with thousands of nodes. For instance, Solana requires 8 GB of RAM and 16 threads per node, limiting scalability. Our experiments were limited to a 40-node network, sufficient to capture behaviors representative of real-world blockchains. While larger networks could offer deeper insights, some studies show that performance indices like consensus efficiency, throughput, and latency stabilize with relatively few nodes [154, 198]. Our setup, though smaller than public blockchains, enables controlled experiments on performance trends. Manual adjustments addressed deployment variations, limiting Lilith flexibility. A full mesh topology ensured comparability; decentralized topologies will be explored by using Lilith. Besides networking, factors like consensus and block size also affect performance and will be studied.

Reproducibility. We release our dataset at https://doi.org/10.5281/zenodo.11409100 with variance data showing Algorand and Diem as stable and Quorum and Solana more sensitive. Findings inform resilient designs for permissioned systems.

Chapter 5

Energy Efficiency: Impact of Network Topologies

This chapter, which is based on the article [110], investigates how network topology and workload composition affect the energy efficiency of blockchain protocols. We start by recalling the issue of energy consumption (§5.1) and defining our evaluation methodology (§5.2). By capturing resource utilization under controlled conditions, we discover non-obvious energy-performance trade-offs, showing that certain topologies, while performant, are less energy-efficient and offering new insights for the design of energy-aware blockchain infrastructures (§5.3). We finally discuss outcomes and limitations of our approach (§5.4).

5.1 Energy Consumption of Blockchains

To achieve decentralization, a self-governing and computationally intensive validation process executed by participating nodes, the consensus protocol [335], is necessary. 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 [182]. 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 [244] 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) [316]. 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 [67]).

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 [154] – 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 [313]. We investigate their impact on the energy consumption of five public/private blockchains: Algorand [76, 147], Diem [38], Ethereum Clique [331], Quorum IBFT [117], and Solana [342]. 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 among 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 experiments, we extended Lilith, the blockchain benchmarking framework presented in Chapter 3, by including 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. Rather than redefining workloads (introduced in Chapter 2), here we select a subset: we use PayPal and VISA as steady transfer workloads and GAFAM as a burst-then-settle smart-contract workload (see Chapter 2 and §3.5 for full definitions and provenance). We include neither extreme stress tests such as DDoS and FIFA, nor high-frequency telemetry (Gaming), because the goal here

is to study energy and networking effects in representative operating regimes rather than overload collapse behavior.

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 [182]. Despite the millions of participants racing to solve the puzzle, only one is ultimately successful, leaving the others' computational efforts effectively wasted [182]. 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 [87].

While Bitcoin's PoW consumes between 200 and 950 kWh per transaction, Ethereum, before transitioning to PoS, required approximately 75 kWh per transaction [87], highlighting the need for more energy-efficient consensus mechanisms. These figures are not used as direct baselines for our controlled experiments, because they refer to production networks with Internet-scale participation, heterogeneous hardware, and background activity that are absent from our testbed. They nevertheless provide an order-of-magnitude context for interpreting the much smaller values measured under controlled, non-mining experimental conditions. 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% [182].

By integrating RAPL energy counters in Lilith, 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 [17], and obtain energy measurements without the need for external or manual instrumentation.

5.2 Evaluation Methodology

This section describes the experimental evaluation with details on the emulated environment as well as experiment configurations.

5.2.1 Framework

We conducted our experiments by using Lilith, the blockchain benchmark suite presented in Chapter 3. To collect energy data, we leveraged Lilith's extensibility to integrate RAPL counters [190]. 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 kWh, we first divide the values by $1,000,000$ to obtain joules and then use the conversion factor $1\text{J} = 2.7778 \times 10^{-7}\,\text{kWh}$.

5.2.2 Experimental Settings

Unless stated otherwise, this chapter reuses the same experimental setup introduced in Chapter 3, including the cluster infrastructure and OS stack, the pinned blockchain client versions and configurations, the five network topologies (fat-tree, full mesh, hypercube, scale-free, and 2D torus), and the client-side submission/confirmation discipline.

5.3 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 ten times. The average energy consumption across these runs is reported in Figures 5.1 to 5.6. In Figures 5.1 to 5.5 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 TPS 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 Figures 5.1 to 5.5).

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 to 0.5 kWh per node with 40 nodes. In Figure 5.6, we focus on the same topology for all blockchains. We examine the average energy consumption per transaction, comparing it across all blockchains for the three workloads and each topology.

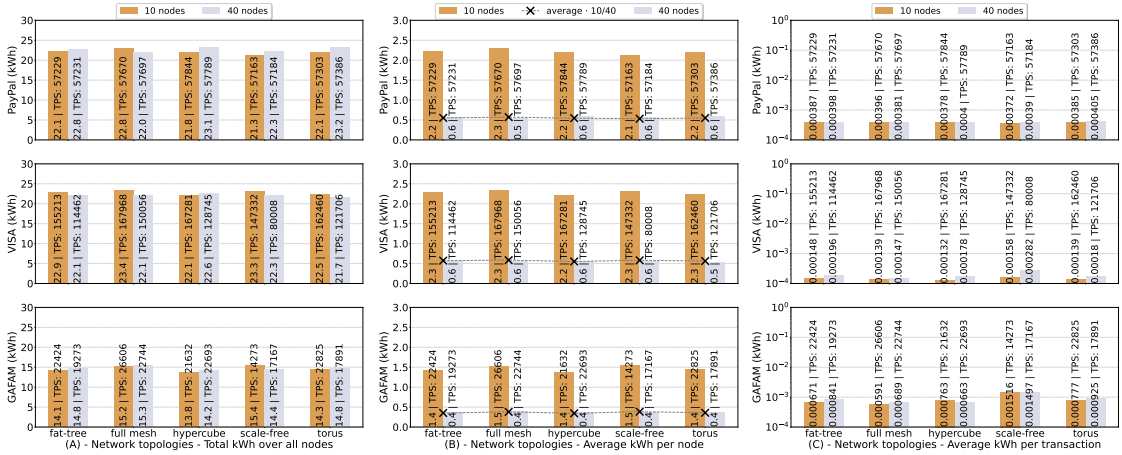


Figure (5.1) Algorand energy consumption (kWh): total over all nodes (A), average per node (B), average per transaction (C).

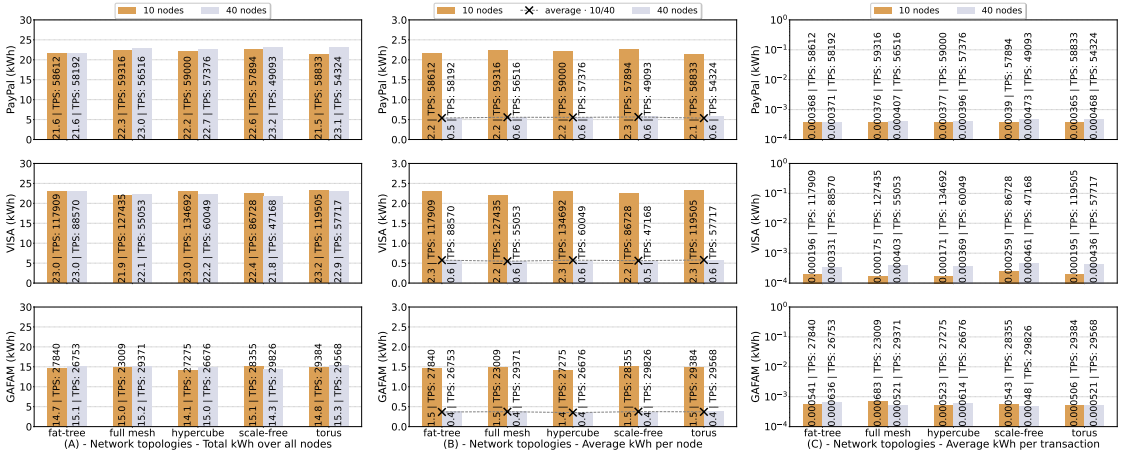


Figure (5.2) Diem energy consumption (kWh): total over all nodes (A), average per node (B), average per transaction (C).

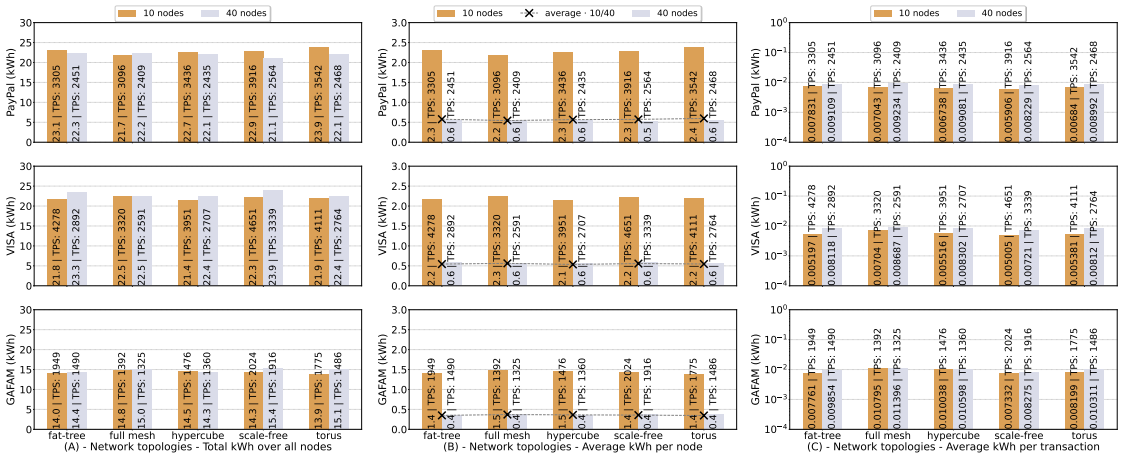


Figure (5.3) Ethereum Clique energy consumption (kWh): total over all nodes (A), average per node (B), average per transaction (C).

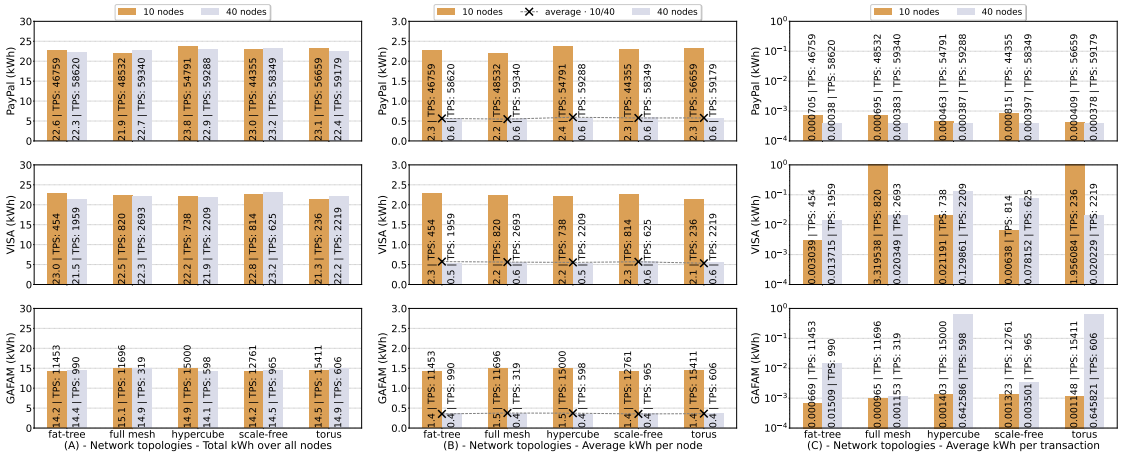


Figure (5.4) Quorum IBFT energy consumption (kWh): total over all nodes (A), average per node (B), average per transaction (C).

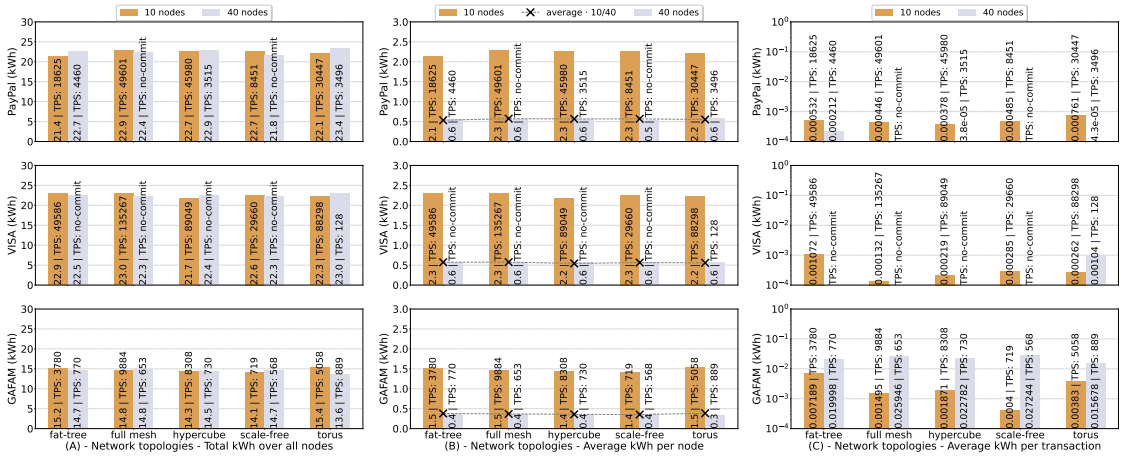


Figure (5.5) Solana energy consumption (kWh): total over all nodes (A), average per node (B), average per transaction (C).

5.3.1 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 Figure 5.1 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 Figure 5.2 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 Figure 5.3 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 Figure 5.4 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

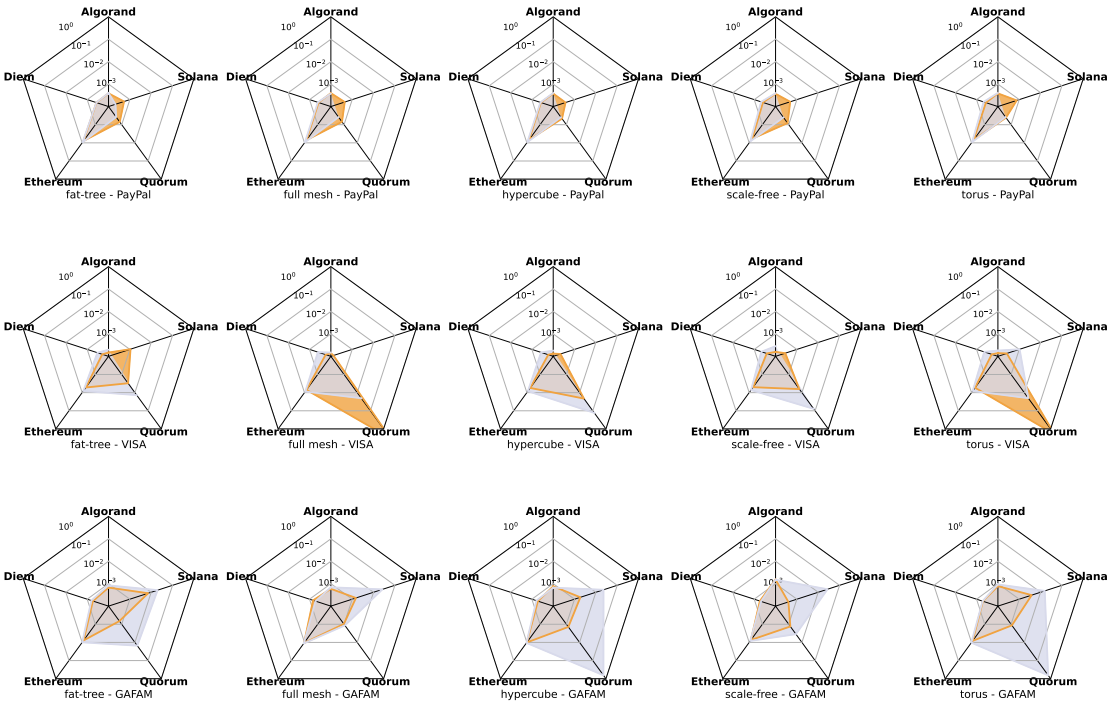


Figure (5.6) Comparison of average energy consumption (kWh) per transaction across blockchains, stratified by topology (10 nodes, 40 nodes).

with intensive transaction processing workloads like VISA, showing a +15% increase in energy consumption. Finally, Solana (see Figure 5.5 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.

5.3.2 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 Figures 5.1 to 5.5), 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, Figure 5.6 allows us to individually analyze each topology across all blockchains. Protocols like

Algorand and Diem exemplify this efficiency, maintaining low energy consumption per transaction (in the 10^{-4} order) under topologies such as hypercube and full mesh (see Figures 5.1 and 5.2 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 a 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.

5.3.3 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.

5.4 Discussion and Limitations

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 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.

Limitations. We conducted experiments on a 40-node blockchain network, offering insights into small-scale benchmarking [198]. While we did not explore large-scale dynamics, prior research [154] supports the representativeness of a 40-node setup.

We collected data from all ten runs, allowing variance and confidence interval calcula-

tions, 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 [325]. 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.

Our measurements should be interpreted as controlled, testbed-level energy measurements rather than as direct estimates of the energy footprint of the corresponding public mainnets. We compare them with real-world figures reported in the literature only as contextual anchors and order-of-magnitude sanity checks. The much lower values observed in our controlled experiments are expected because our setup excludes Internet-wide node populations, heterogeneous hardware, non-experimental background activity, and global mining races. Therefore, this comparison is not an apples-to-apples validation against production networks. The contribution of this chapter is comparative: under the same hardware, workload, instrumentation, and emulated topology, it quantifies how topology changes the relative energy efficiency of the tested blockchain implementations.

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 (§3.6), aligning with Diablo [154], 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 chapter.

Chapter 6

Experimental Repeatability and Performance Predictability: A Network-Controlled Approach

This chapter, which relies on the work [111], presents a multi-run, network-controlled measurement campaign and a public dataset that quantify run-to-run variability and performance predictability across blockchains, topologies, workloads, and node-set sizes. We start by recalling the issue of repeatability and predictability (§6.1) and defining our methodology (§6.2). Using dispersion metrics (e.g., worst-case deviation) together with analysis of variance and intraclass correlation, we disentangle configuration effects from exogenous noise and identify both stable and volatile blockchain-topology pairs under geographically emulated network conditions (§6.3). The study highlights how uncontrolled infrastructures such as public clouds can hide or distort these effects and advocates controlled benchmarking frameworks like Lilith *in order to support thorough, repeatable comparative evaluations (§6.4).*

6.1 Repeatability and Predictability

This chapter targets two critical aspects of experimental rigor in blockchain benchmarking: *experimental repeatability*, that is the ability to reproduce outcomes under identical experimental conditions [4], and *performance predictability*, the ability of the system to guarantee consistent performance (*e.g.,* latency, throughput) that are not easily degraded by adverse conditions, in line with [307].

This raises several issues. Given a fixed blockchain-topology-workload configuration, how stable are results across independent runs? And how much of the variance stems from network topology and system design as opposed to infrastructure noise? To achieve a dependable performance evaluation, conclusions must be robust across repeated controlled runs, not just single point estimates.

We address these gaps with a variance-centric benchmarking methodology that foregrounds execution variability and treat intra-configuration dispersion – *i.e.,* how outcomes spread across independent runs under the same configuration – as a primary evaluation

target rather than incidental noise. By variance-centric we mean that we quantify both typical dispersion around the mean and behavior in the tails: we pair robust dispersion indices like interquartile range (IQR) and normalized standard deviation (Std) with worst-case deviation (WCD) indices that capture how far individual runs drift from their configuration mean, in absolute units and as a percentage. We complement these per-configuration views with factorial ANOVA and intraclass correlation coefficient (ICC), which decompose variance across experimental factors and quantify run-to-run consistency.

To concretize this vision, we build on recent benchmarking frameworks [17,108,154] to construct a controlled, geo-emulated testbed and obtain a public dataset of ten independent runs per configuration. We focus on a single cluster to study experimental repeatability under deterministic conditions. This is a prerequisite baseline for later cross-infrastructure reproducibility and replicability studies, which we leave as future work. Wide-area effects are preserved and reproduced logically at the network layer via deterministic geo-emulation of latency, link throughput, and topology dynamics derived from measured WAN traces (Figure 2.4), decoupling networking causes from infrastructure noise. In this way, all validators physically run on the same controlled cluster (with a validator being a blockchain node). We implement this environment with Lilith (Chapter 3), which automates deployment and workload generation while providing repeatable, controlled network emulation on a fixed computing substrate.

Our experiments span five industry-grade blockchains (Algorand [147], Diem [38], Ethereum Clique [308], Quorum IBFT [117], and Solana [342]), five network topologies (fat-tree, full mesh, hypercube, scale-free, torus), and three workloads (GAFAM, PayPal, VISA). These dimensions cover realistic heterogeneity in consensus, architecture, and use cases and reproduce configurations studied in prior performance- and energy-centric work [108,110,154], including legacy systems (*e.g.,* Diem). For each blockchain-topology-workload configuration, we collect ten independent runs under identical software versions and deterministic network conditions, together with per-run metrics. The resulting dataset (available at `https://doi.org/10.5281/zenodo.17681717`) is a new, variance-centric release. For each configuration we characterize dispersion via IQR and Std as well as tail behavior via WCD, defined as the maximum deviation of any run from the configuration mean and reported both in absolute units (TPS, Lat, kWh) and as a percentage [315]. Factorial ANOVA [183] decomposes variance across blockchains, topologies, workloads, and validator-set size, while ICC [192] quantifies how much of the observed variability is systematic as opposed to pure noise. While we reuse the same systems/topologies/workloads of recent geo-emulated studies [108,110] for comparability, we ask a different question: not which configuration is fastest on average, but how predictable results are under controlled geo-emulated WAN conditions. Accordingly, we report dispersion and tail risk anchored to the configuration mean (absolute units and %), separate failures from successful runs, and attribute systematic drivers via ANOVA/ICC. Our findings include:

- Throughput and latency exhibit substantial run-to-run variability: typical dispersion

(IQR%, normalized Std%) stays moderate for Algorand, Diem, and Ethereum Clique, yet single-run deviations can reach 1,200%, with absolute ranges up to 684.80 TPS and 247.86 s. Energy is steadier and workload-driven, with ranges within 14–19 kWh, in line with prior energy measurements [110].

- Factorial ANOVA shows that the blockchain and its interaction with the workload explain most throughput and latency variance, while topology and validator-set size have a smaller impact; high ICC values indicate that most dispersion is systematic and rooted in design choices rather than uncontrolled infrastructure noise.

- Topology, workload, and validator-set size mainly modulate variability: torus tends to concentrate the most stable configurations, PayPal and VISA workloads amplify swings while GAFAM workload sits in between. Scaling from 10 to 40 validators can reduce some extremes, yet increase relative sensitivity in already fragile blockchains.

In addition to consolidating what has been presented in Chapters 4 and 5, this work makes the following contributions: (*i*) a four-level predictability taxonomy (P0–P3); (*ii*) a variance-first evaluation methodology for geo-emulated blockchain experiments that reports typical dispersion (IQR%, Std%) and tail risk (WCD/WCD%) anchored to per-configuration means and explicit failure regimes; (*iii*) a public dataset of 10 independent runs per configuration under pinned commits and deterministic network conditions (vs. 3 runs in recent geo-emulated studies [108, 110]); (*iv*) a dataset-wide statistical attribution (ANOVA/ICC) that ranks dominant dispersion drivers and quantifies run-to-run consistency.

6.2 Methodology

Benchmarking distributed systems such as blockchains requires minimizing external variability to ensure meaningful performance comparisons, particularly when assessing the impact of network topology or workload dynamics. To this end, all experiments were conducted by using Lilith so as to ensure deterministic, repeatable runs under programmable network conditions.

6.2.1 P0–P3 Labels, Predictability, and Network Isolation

We use P0–P3 as a maturity-reporting label for evidence quality, grounded in repeatability/reproducibility/replicability notions (see end of §2.4.1) and whether dispersion is measured under controlled repetitions: P0 means uncontrolled observations (*e.g.,* mainnet traces) [251]; P1 means controlled but mean/point-estimate-centric testbeds [108,110,154]; P2 means controlled studies quantifying run-to-run dispersion (this work); P3 means validated models predicting performance distributions ex ante. This ladder is only context for variance-aware reporting, not a novelty claim. This perspective echoes earlier work on dependability benchmarking and measurement theory, which argued that meaningful claims require controlled fault and load models together with well-defined metrics [56,317].

Table (6.1) Representative blockchain performance studies and their position in the P0–P3 ladder. Legend: ✓ variance reporting; ◑ some distributional analysis (*e.g.,* percentiles) but not under controlled repeated setups; ◌ data unavailable; ✗ single-run or averages only. $B \times T \times W \times S$ = Blockchains × Topologies × Workloads × Scales (validator-set sizes).

Work	Year	Environment	Variance focus	Level	$B \times T \times W \times S$	Runs
[113]	2017	Local testbed, multi-platform	✗	P1	$3 \times 1 \times 6 \times 7$	5
[173]	2018	Cloud/local, cross-platform	✗	P1	$3 \times 1 \times 3 \times 1$	1
[278]	2020	Energy-centric testbed	◑	P1	◌	◌
[268]	2023	Single-machine multi-node testbed	✗	P1	$2 \times 1 \times 2 \times 3$	3
[146]	2023	Latency benchmarking	◑	P1	$7 \times 1 \times 6 \times 4$	◌
[251]	2023	Ethereum mainnet	✓	P0	$1 \times 1 \times 1 \times 1$	1
[218]	2023	Bitcoin traces + queueing model	✓	P0	$1 \times 1 \times 1 \times 1$	1
[154]	2023	Cloud, cross-platform	✗	P1	$6 \times 1 \times 5 \times 2$	◌
[156]	2024	Tail-latency analysis on testbed	◑	P1/P2	$5 \times 1 \times 1 \times 1$	◌
[319]	2024	Cloud testbed	✗	P1	$4 \times 1 \times 1 \times 1$	3
[260]	2025	Parallel/distributed simulator	✗	P1	$1 \times 1 \times 2 \times 4$	5
[108]	2025	Geo-emulated testbed	✗	P1	$5 \times 5 \times 5 \times 2$	3
[110]	2025	Geo-emulated testbed	✗	P1	$5 \times 5 \times 3 \times 2$	3
This work	**2026**	**Geo-emulated testbed**	✓	**P2**	$\mathbf{5 \times 5 \times 3 \times 2}$	**10**

Table 6.1 maps representative performance studies, not tools per se, within this ladder, distinguishing between control over the execution environment (mainnet vs. testbed, where we treat testbeds as environment classes such as cloud, local clusters, and emulation-based platforms) and whether variability is explicitly measured and reported. Mainnet-oriented studies [218, 251] explicitly analyze distributions of confirmation times and related signals, but operate in evolving, uncontrolled environments and therefore remain at P0. Testbed-based tools such as BlockBench, Caliper, and Diablo move to P1 by providing controlled execution environments, yet typically report single-run averages or coarse percentiles without systematic multi-run analysis. Recent geo-emulated studies [108, 110] further strengthen control by enforcing deterministic wide-area topologies and network conditions, but still emphasize point estimates and remain silent about run-to-run dispersion.

Bridging this gap from P0/P1 to P2 requires explicit control over the execution conditions that drive variability, in particular the network layer and the orchestration of nodes and workloads. Blockchain performance arises from system logic, execution engines, and network dynamics. In practice, performance predictability depends on strong network isolation [162, 201] and requires: *(i)* deterministic orchestration of nodes and services (*i.e.,* fixed boot order and timing, pinned images and configurations, CPU and memory affinities); *(ii)* explicit network topology and link properties (*i.e.,* per-link latency, bandwidth, packet loss; fixed routing and queueing); *(iii)* controlled workload injection (*i.e.,* predefined open- or closed-loop modes, fixed inter-arrivals or traces, warm-up and steady-state windows); *(iv)* minimization of external variability via automation and emulation (*i.e.,* scripted builds, identical environments, network emulation, background noise disabled).

Most existing tools only partially satisfy these requirements (Table 2.5), which helps explain why the state of the art largely stops at P0/P1. Prior works using Lilith reported

mainly mean-centric outcomes [108, 110] with 3 repetitions per configuration. We extend them with a 10-run, variance-centric P2 design focused on predictability (see §3.1). We do not pursue a cross-system leaderboard; cross-system comparisons serve only to disentangle the roles of network topology and system design in driving variability.

6.2.2 Controlled Testbed and Experimental Design

All experiments run on an on-premises cluster of 7 dual-socket servers, each with two 16-core Intel Xeon CPUs (32 hardware threads in total) and 128 GB of RAM, interconnected via a non-blocking 40 GbE switch and running Ubuntu 22.04 LTS with kernel 5.15. This setup avoids the unpredictability of shared or virtualized public clouds while giving full control over the logical network: network conditions derived from geo-latency traces and time-varying network conditions are enforced by the emulator across nodes that physically reside in a single cluster. The cluster thus serves as a stable execution substrate, while geo-distribution is realized logically at the network layer, reducing cost, enabling deterministic orchestration (deployment timing, CPU isolation, topology enforcement), and providing a clean baseline for experimental repeatability analysis.

We evaluate the five network topologies of §3.4 and the five blockchains of §3.6 under the three workload configurations of §3.5 and two validator-set sizes (10, 40) on a single cluster, then apply variance-centric statistical metrics. Blockchains are run with specific Git commits – the exact source snapshot – and literature-based configs [108, 110, 154]; seeded, pre-scheduled topologies/workloads ensure experimental repeatability.

6.2.3 Statistical Treatment of Variability

Our experimental campaign executes multiple independent runs for each configuration $c = (B, T, W, S)$, where B is the blockchain, T the network topology, W the workload, and S the validator-set size (10 or 40 validators). For each run we record throughput (TPS), block latency, and total cluster energy (kWh). Let $y_{c,r}^{(k)}$ be the value of metric $k \in \{\text{TPS}, \text{Lat}, \text{En}\}$ for configuration c at run r, $r = 1, \ldots, n_c$ with $n_c = 10$. To lighten notation we omit (k) and write $y_{c,r}$, with the understanding that all quantities are computed per metric. For each configuration c we compute the sample mean m and sample variance σ^2:

$$m_c = \frac{1}{n_c} \sum_{r=1}^{n_c} y_{c,r} \qquad (6.1) \qquad\qquad \sigma_c^2 = \frac{1}{n_c} \sum_{r=1}^{n_c} (y_{c,r} - m_c)^2 \qquad (6.2)$$

with standard deviation $\sigma_c = \sqrt{\sigma_c^2}$, the minimum and maximum values:

$$y_c^{\min} = \min_r \; y_{c,r}, \qquad y_c^{\max} = \max_r \; y_{c,r} \qquad (6.3)$$

and the empirical 25th and 75th percentiles $Q_{1,c}$ and $Q_{3,c}$ of $\{y_{c,r}\}_r$. The interquartile range (IQR) is given by:

$$\text{IQR}_c = Q_{3,c} - Q_{1,c} \qquad (6.4)$$

capturing the width of the central 50% of the runs and offering robustness to occasional

extremes [315]. To compare spreads across configurations with different scales, we normalize both IQR and standard deviation by the mean:

$$\mathrm{IQR}\%_c = 100 \cdot \frac{\mathrm{IQR}_c}{m_c} \qquad (6.5) \qquad\qquad \mathrm{Std}\%_c = 100 \cdot \frac{\sigma_c}{m_c} \qquad (6.6)$$

so that $\mathrm{IQR}\%_c$ and $\mathrm{Std}\%_c$ provide complementary views of typical dispersion around the mean (robust quantile-based vs. moment-based).

To characterize tail behavior we also measure how far the most unstable runs can drift from the configuration mean. We first define the downward and upward deviations from m_c as $\Delta_c^{\downarrow} = m_c - y_c^{\min}$ and $\Delta_c^{\uparrow} = y_c^{\max} - m_c$, respectively. The worst-case deviation (WCD) is then the largest of these two one-sided excursions:

$$\mathrm{WCD}_c = \max\{\Delta_c^{\downarrow},\ \Delta_c^{\uparrow}\} = \max\{|y_c^{\min} - m_c|,\ |y_c^{\max} - m_c|\} \qquad (6.7)$$

with normalized form:

$$\mathrm{WCD}\%_c = 100 \cdot \frac{\mathrm{WCD}_c}{m_c} \qquad (6.8)$$

which measures the worst single-run deviation as a percentage of the configuration mean. For interpretability in the original units we also track the absolute range:

$$\Delta_{\mathrm{abs},c} = y_c^{\max} - y_c^{\min} = \Delta_c^{\downarrow} + \Delta_c^{\uparrow} \qquad (6.9)$$

which quantifies the full peak-to-peak excursion across runs (TPS, seconds, kWh). We retain extreme outcomes in WCD/WCD%: they are part of the distribution and expose tail risk under a fixed configuration. In summary, we use five variability metrics per configuration: two to capture typical dispersion around the mean ($\mathrm{IQR}\%_c$ and $\mathrm{Std}\%_c$) and three to capture worst-case behavior in the tails (WCD_c, $\mathrm{WCD}\%_c$, and $\Delta_{\mathrm{abs},c}$), with the former driving our "typical variability" analysis and the latter our worst-case and fragility analysis.

While these indices quantify how much variability each configuration exhibits, they do not explain *where* variability originates. Even under deterministic orchestration, intrinsic protocol/implementation nondeterminism and threshold effects (mempool/backpressure, batching, timeouts) can amplify small jitters into divergent commit paths. To attribute variance to design choices, we perform a factorial ANOVA on per-run metrics [183]. Let y_i denote an observation from run i with associated factors blockchain B, topology T, workload W, and size S. For each metric we fit the *fixed-effects model*:

$$y_i = \mu + \alpha_B + \beta_T + \gamma_W + \delta_S + (\alpha\beta)_{B\times T} + (\alpha\gamma)_{B\times W} + \varepsilon \qquad (6.10)$$

where μ is the overall mean (*i.e.,* the grand mean across all observations y_i), α_B, β_T, γ_W, and δ_S are the main effects of B, T, W, and S, and $(\alpha\beta)_{B\times T}$ and $(\alpha\gamma)_{B\times W}$ are two-way interactions capturing how topology and workload sensitivity depend on the system design. Higher-order interactions and residual noise are absorbed by ε.

Finally, we quantify the experimental repeatability of runs within the same configuration via the intraclass correlation coefficient (ICC), a standard run-to-run consistency

index in repeated-measures studies [192]. Treating each configuration $c = (B, T, W, S)$ as a group in the statistical ANOVA sense, we consider the one-way random-effects model:

$$y_{c,r} = \mu + A_c + \varepsilon_{c,r} \tag{6.11}$$

where A_c is the random effect of configuration c with variance $\sigma^2_{\text{between}}$ and $\varepsilon_{c,r}$ captures run-to-run noise with variance σ^2_{within}. Following the ICC(1,1) convention of Koo and Li [192], we estimate these variances from a one-way ANOVA across configurations and define:

$$\text{ICC} = \frac{\sigma^2_{\text{between}}}{\sigma^2_{\text{between}} + \sigma^2_{\text{within}}} \tag{6.12}$$

Intuitively, ICC is the fraction of total variance explained by differences between configurations rather than by run-to-run noise; values below 0.5 indicate poor experimental repeatability, 0.5–0.75 moderate, 0.75–0.9 good, and above 0.9 excellent [192].

Our factorial ANOVA (Eq. 6.10) is a descriptive variance decomposition (shares/-effect sizes in Table 6.4), not a p-value-driven hypothesis test. The design is balanced with (n_c=10) independent runs under deterministic orchestration and fixed network conditions; under such designs ANOVA is robust to moderate residual non-normality/heteroscedasticity and we rely on ranked factor/interaction dominance. For ICC, we use the one-way random-effects ICC(1,1) (Eq. 6.11–6.12) as a run-to-run consistency index: variance due to configuration differences vs. within-configuration run noise [192].

6.3 Experimental Results

We now evaluate experimental repeatability and performance predictability under fully controlled, geo-emulated conditions. The results below come from executed experiments on our on-premises geo-emulated cluster. For each configuration, we instantiate the selected topology, dataset, workload, and resource constraints; deploy and bootstrap the backend through Minion; inject the workload; collect performance, network, and energy measurements; and export logs for post-processing and repeated-run analysis. This workflow is repeated independently for every blockchain/topology/workload configuration. All results are labelled P2 in our P0–P3 evidence ladder: for each configuration $c = (B, T, W, S)$, we execute 10 independent runs under identical software and network conditions. Ten runs are a practical compromise between statistical depth and campaign breadth. Given the large factorial space and the cost of each geo-emulated execution, this budget supports comparative P2-level repeatability analysis and variance attribution, focusing on intra-testbed variability, while finer distributional and tail characterization as well as cross-infrastructure reproducibility and replicability is left to future work.

Each workload is run on every blockchain (§3.6), across all topologies (§3.4) and two validator-set sizes (10 and 40 validators, capturing how a 4-times increase in validator count affects performance at validator scale rather than Internet-wide overlays), with

Table (6.2) Overall blockchain-level variability. For each metric we report IQR% and Std%. Percentages are with respect to the per-configuration mean; absolute magnitudes are in Figures 6.1–6.3 (left). The most stable blockchains (minimum values) are highlighted in , the least stable ones in . Highlights select best/worst stability over all T, W, S per blockchain; generating configurations appear in the per-factor breakdown (Figures 6.1–6.3, Tables 6.5–6.7).

	TPS		Lat		kWh	
Blockchain	IQR%	Std%	IQR%	Std%	IQR%	Std%
Algorand	6.11	6.73	5.91	11.73	14.75	12.91
Diem	5.44	6.02	10.61	10.53	15.96	12.11
Ethereum Clique	11.68	11.61	7.40	7.14	15.86	13.93
Quorum IBFT	44.59	56.60	28.91	121.47	18.00	13.68
Solana	34.71	59.58	17.47	44.51	16.21	12.99

homogeneous nodes (8 vCPUs, 16 GB RAM). This diversity of workloads, connectivity patterns, and blockchains lets us observe how experimental repeatability evolves under flat vs. bursty traffic and across network structures. We track three metrics: (*i*) throughput (TPS), measuring the number of transactions successfully processed per second; (*ii*) block latency (seconds), *i.e.,* the time between the submission of a transaction and its inclusion in a committed block; (*iii*) total energy consumption (kWh) among nodes. We include energy as a provisioning metric to contrast stability across metrics, since it is steadier than TPS/latency under the same controlled conditions.

Tables 6.2, 6.5, 6.6, and 6.7 summarize typical dispersion around configuration means via IQR% and Std% (§6.2.3), refining the view along blockchain, topology, workload, and validator-set size. Best and worst values are highlighted in and . Table 6.4 reports factorial ANOVA and ICC, decomposing variance across design factors and quantifying run-to-run consistency. We also report the fraction of unexplained variability, the share of variance attributed to ε. Table 6.3 and Figures 6.1–6.3 expose worst-case run behavior. For each configuration, we analyze distributions by anchoring deviations to the per-configuration mean m and reporting them in absolute units (ΔTPS/ΔLat/ΔkWh; left) and as % of m (right), since the same % can imply very different absolute impacts (*e.g.,* 30% is $\pm$3 TPS for Clique vs. $\pm$30 TPS for Algorand). Per-configuration means are in the released dataset. Right-hand axes are clipped at $\pm$100%, with out-of-range values explicitly labeled (*e.g.,* "+1,200%"). We distinguish two failure modes: (*i*) *NO* (non-operational) for deployments or workload executions that fail entirely; (*ii*) mean $= 0$ when nodes receive the workload but fail to commit blocks, typically due to system limitations or network congestion. *NO* denotes a failure regime in all 10 runs and should be read as non-operational, not "high variance"; variability metrics are reported only for 10/10 successful configurations.

Table (6.3) Worst-case run-to-run blockchain swings. For each metric we report: the maximum observed range (Δ_{abs}), the largest absolute drop below the mean ($\Delta^{\downarrow}$), the largest absolute increase above the mean ($\Delta^{\uparrow}$), and their percentage counterparts ($\Delta^{\downarrow}\%$, $\Delta^{\uparrow}\%$) relative to the configuration mean. The least stable values (WCD and WCD%, depending on the column) are highlighted in . All WCD metrics are computed over 10/10 successful runs; failure regimes (*NO* or m=0) are excluded.

	TPS					Lat (s)					En (kWh)				
Blockchain	Δ_{abs}	$\Delta^{\downarrow}$	$\Delta^{\uparrow}$	$\Delta^{\downarrow}\%$	$\Delta^{\uparrow}\%$	Δ_{abs}	$\Delta^{\downarrow}$	$\Delta^{\uparrow}$	$\Delta^{\downarrow}\%$	$\Delta^{\uparrow}\%$	Δ_{abs}	$\Delta^{\downarrow}$	$\Delta^{\uparrow}$	$\Delta^{\downarrow}\%$	$\Delta^{\uparrow}\%$
Algorand	169.40	96.03	94.28	85.57	84.45	42.03	13.69	29.01	36.16	91.18	19.00	6.65	12.86	31.03	54.99
Diem	396.50	232.03	164.47	79.32	56.22	90.90	67.62	39.67	72.29	108.56	13.95	8.48	7.67	39.24	51.73
Ethereum Clique	11.90	7.22	8.22	65.52	76.18	99.48	29.91	79.51	20.66	53.55	16.85	8.23	9.63	37.13	44.17
Quorum IBFT	171.90	135.50	103.65	94.43	1094.52	247.86	79.35	202.99	85.03	928.83	14.18	7.66	9.16	36.79	41.83
Solana	684.80	444.45	320.40	70.29	1200.00	242.40	101.43	171.16	39.46	1200.00	14.22	6.19	8.13	43.06	36.46

6.3.1 Typical Dispersion around the Mean

Table 6.2 shows that throughput and latency are markedly more volatile than energy, even when we restrict our attention to the central part of the distribution. For TPS, Diem and Algorand are the most stable (IQR% around 5–6%), Ethereum Clique sits in the middle ($\approx$ 12%), and Solana and Quorum IBFT show much larger spreads (above 30% and 40%). Latency follows the same ordering, with Algorand, Diem, and Ethereum Clique below 11% IQR% and Quorum IBFT and Solana markedly higher. Energy is comparatively steadier across all chains: IQR% is in a narrow 15–18% band and Std% around 12–14%. Table 6.2 reports dispersion (IQR%/Std%) around configuration means, not mean TPS/latency; absolute magnitudes are in the Δ (unit) panels of Figures 6.1–6.3.

Tables 6.5–6.7 refine this picture along topology, workload, and validator-set size. Topology primarily acts as a selector of better and worse regimes rather than a dominant variance source: within each blockchain, specific topologies systematically host minima and maxima of IQR% and Std%. As discussed in §3.4, topology amplifies timing jitter and can separate stable/unstable regimes within a blockchain-workload pair (Tables 6.3–6.5), yet it contributes less to aggregate variance than the blockchain and its workload interaction (Table 6.4). Across blockchains, torus and full mesh frequently concentrate the most stable configurations, especially for Diem and Ethereum Clique, while fat-tree, hypercube, and scale-free more often expose larger spreads for Solana and Quorum IBFT (Table 6.5). Workload choice is similarly impactful: low-rate, steady workloads (PayPal) are typically benign; bursty GAFAM traces increase dispersion; VISA, with high sustained load, most often amplifies queuing and variability, particularly on Solana and Quorum IBFT (Table 6.6). Overall, typical spread is largely driven by the blockchain-workload pair, with topology and validator-set size acting as secondary modulators.

6.3.2 Worst-Case Behavior

While IQR% and Std% capture typical behavior, operators also need to understand operational tails. Table 6.3 collapses worst-case swings per blockchain. For TPS, Solana exhibits the most dramatic excursions, with Δ_{abs} above 680 TPS and $\Delta^{\uparrow}\%$ reaching $1,200\%$ in at least one configuration; Quorum IBFT reaches similar extreme relative spikes

Table (6.4) Variance decomposition and run-to-run reliability. Each entry reports the percentage of variance explained by the corresponding factor or interaction in the factorial ANOVA; ε denotes the residual term (higher-order interactions and unexplained noise). The last column reports the intraclass correlation coefficient (ICC) for per-configuration runs. B: Blockchains; T: Topologies; W: Workloads; S Scales (validator-set sizes).

Metric	B	T	W	S	$B{\times}T$	$B{\times}W$	ε	ICC
TPS	42.7	1.0	10.4	3.1	1.1	22.2	19.5	0.911
Lat	42.9	0.4	6.1	0.3	0.7	22.0	27.6	0.804
En	0.0	0.0	64.3	0.0	0.1	0.1	35.5	0.632

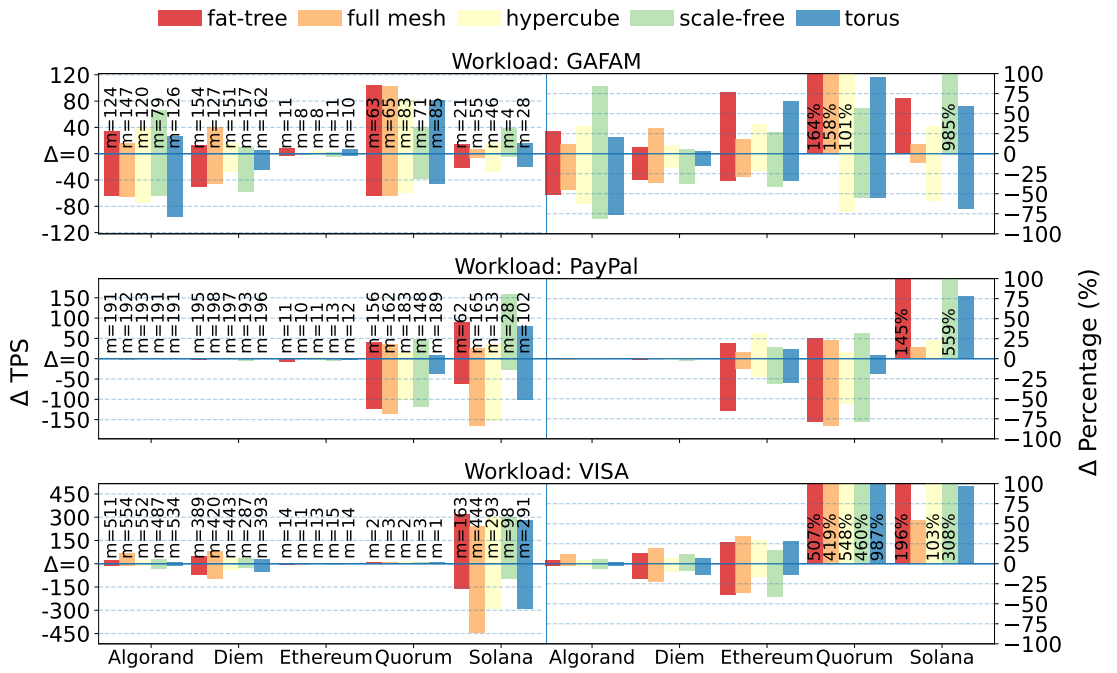
despite smaller absolute ranges. Diem and Algorand occupy an intermediate regime, while Ethereum Clique is the most predictable in absolute terms (worst-case range 11.90 TPS).

Latency tails are even more revealing: both Quorum IBFT and Solana reach ranges above 240 s and relative spikes close to or exceeding 900–1, 200%, indicating that isolated runs can experience order-of-magnitude slowdowns compared to their configuration averages. Algorand and Ethereum Clique, instead, show smaller ranges and lower relative swings. Energy again remains the least extreme metric: even when ranges are nontrivial, percentage deviations are much smaller than those seen for TPS and latency. The run-level plots in Figures 6.1–6.3 confirm that the tallest bars and 1200% labels are concentrated in Solana and Quorum IBFT under high-load workloads and unfavorable topologies, whereas Algorand, Diem, and Ethereum Clique form tighter bands around their means. We do not discard extreme WCD% as outliers: when runs are successful, these large deviations are repeatable in specific blockchain-topology-workload regimes and quantify tail risk that mean-centric reporting can hide. We treat failures as a separate regime (NO, $m{=}0$), distinct from operational configurations that exhibit extreme yet valid WCD% tails.

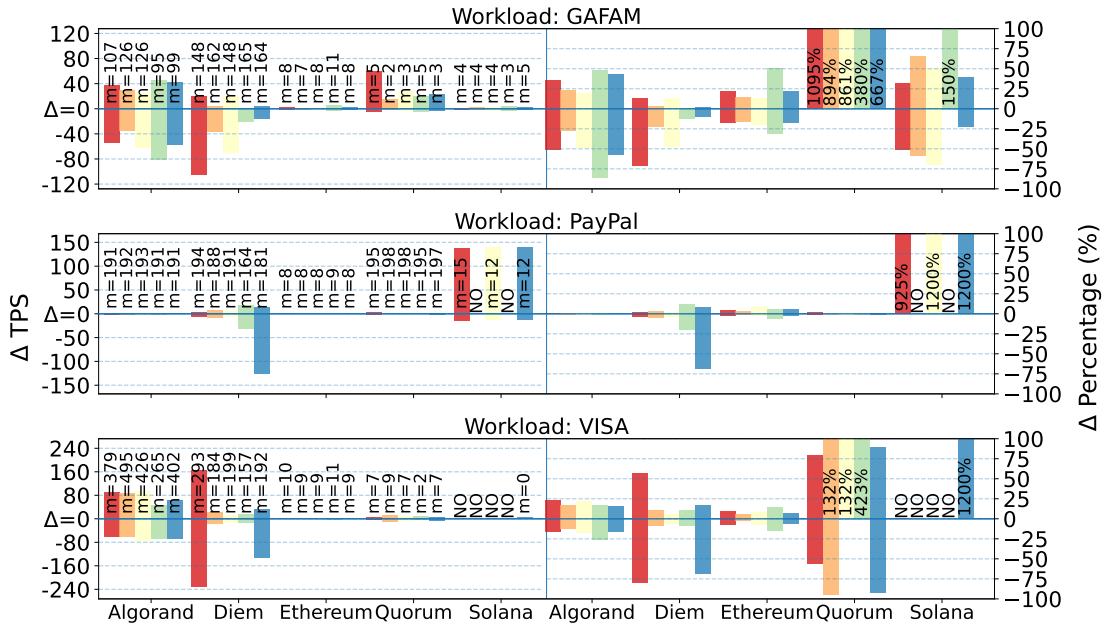
6.3.3 Variability Sources

We separate controlled factors (hardware, versions, orchestration, topology, deterministic network) from intrinsic ones: run-to-run dispersion can still arise from protocol/implementation nondeterminism and threshold effects (queueing, batching, timeouts), amplified by topology and load. Table 6.4 clarifies the sources of variability. For throughput, the blockchain factor B alone explains 42.7% of the variance, with its interaction with workload $B{\times}W$ adding 22.2%; workload W itself contributes 10.4%, while topology T and size S explain only a few percent each. Latency exhibits a similar decomposition, with B and $B{\times}W$ together accounting for almost two thirds of the variance. Energy is qualitatively different: W alone explains more than 64% of the variability, while B and T are negligible, confirming that energy is primarily determined by workload rather than topology or scale. The residual term ε absorbs the remaining variance, upper-bounding measurement noise and higher-order interactions.

The ICC values 0.911 for TPS, 0.804 for latency, and 0.632 for energy indicate that runs within the same configuration are generally consistent and that most dispersion is systematic across configurations rather than pure noise. Combined with the typical

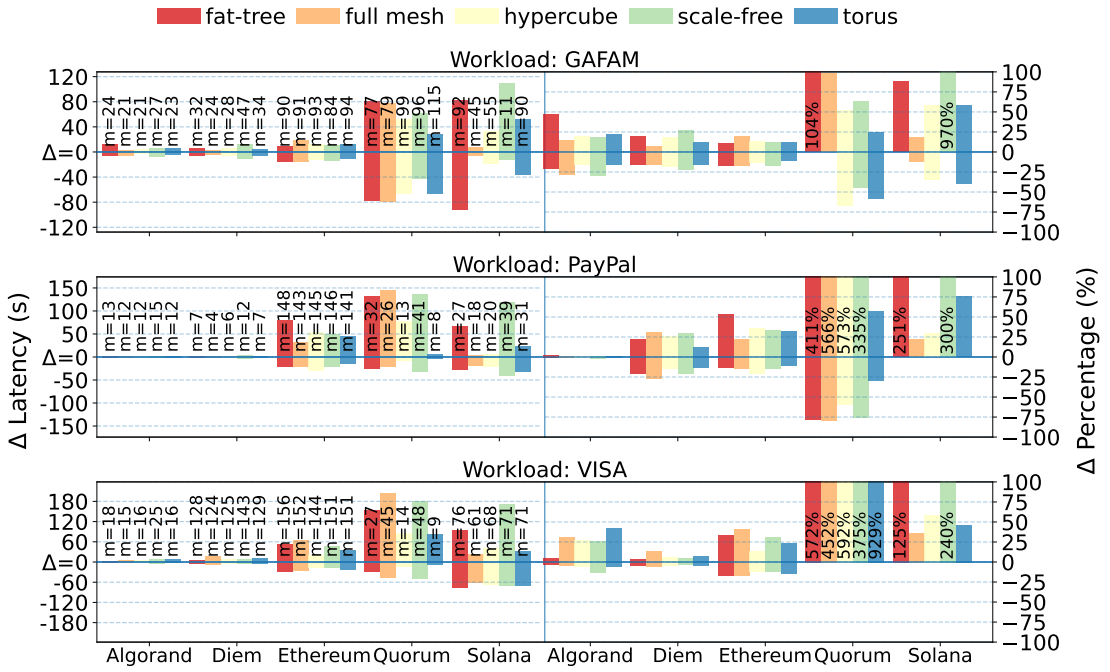


(a) 10 nodes – GAFAM, PayPal & VISA – Throughput deviation

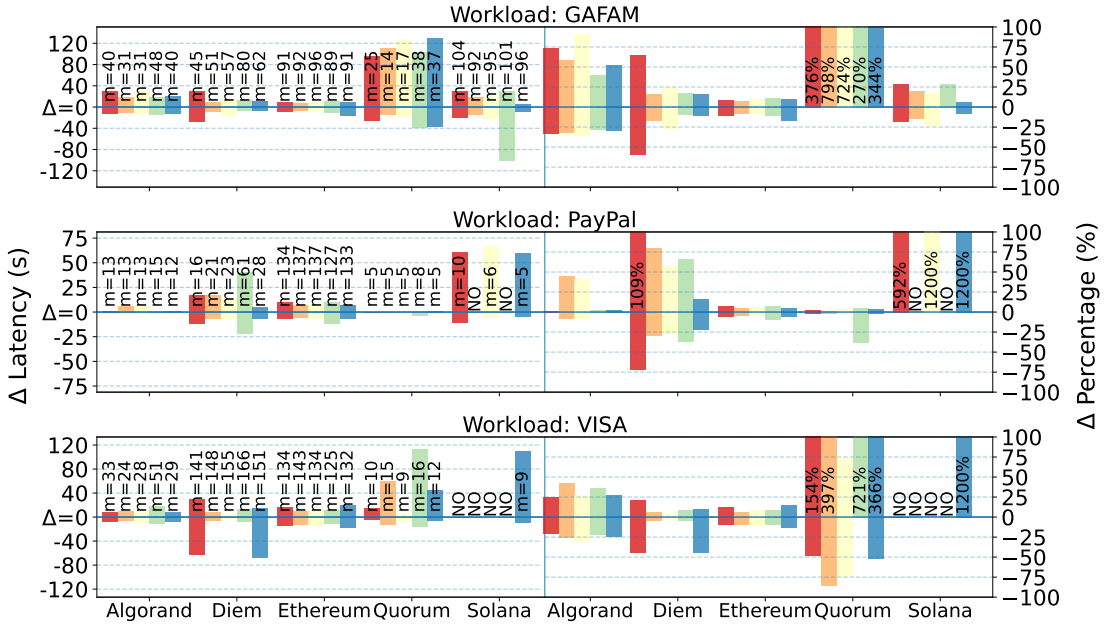


(b) 40 nodes – GAFAM, PayPal & VISA – Throughput deviation

Figure (6.1) Throughput deviation across workloads. For each panel and configuration, deviations (Δ) are computed against the mean m over the 10 runs: left = absolute swing (TPS); right: relative swing (% from mean). Right axis clipped at $\pm 100\%$, with out-of-scale values explicitly labeled. Panels show (a) 10 nodes and (b) 40 nodes. Failure regimes are explicitly labeled: *NO* denotes 10/10 non-operational runs, while m=0 denotes 10/10 runs where the workload was delivered but no blocks were committed.



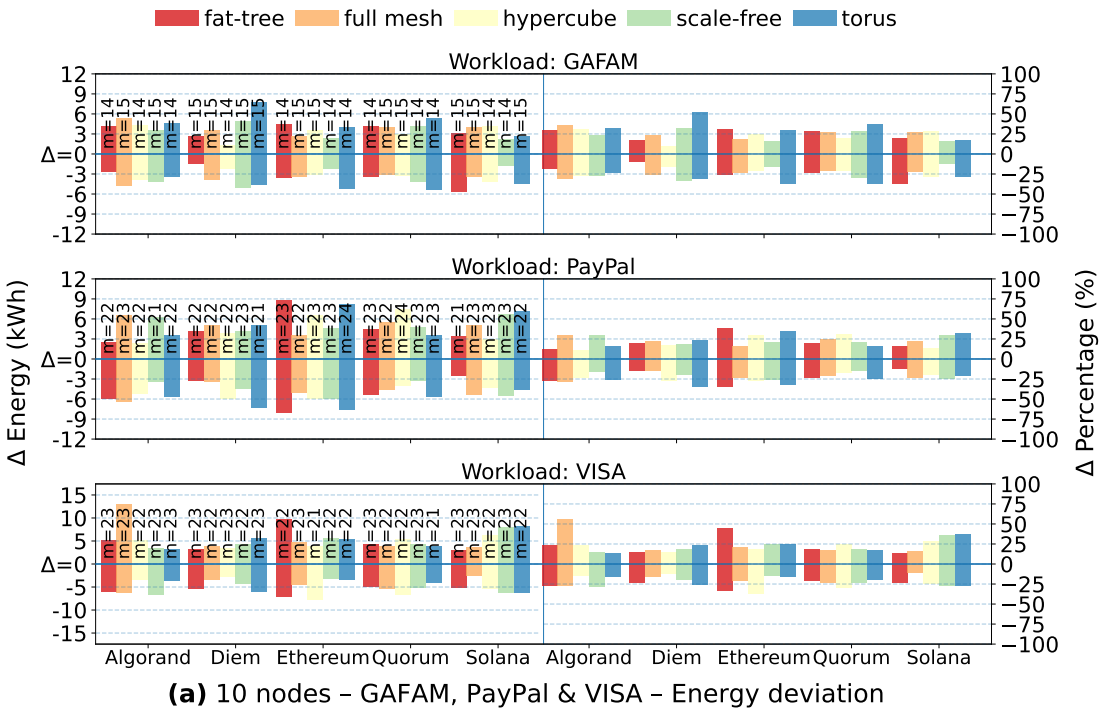
(a) 10 nodes – GAFAM, PayPal & VISA – Latency deviation



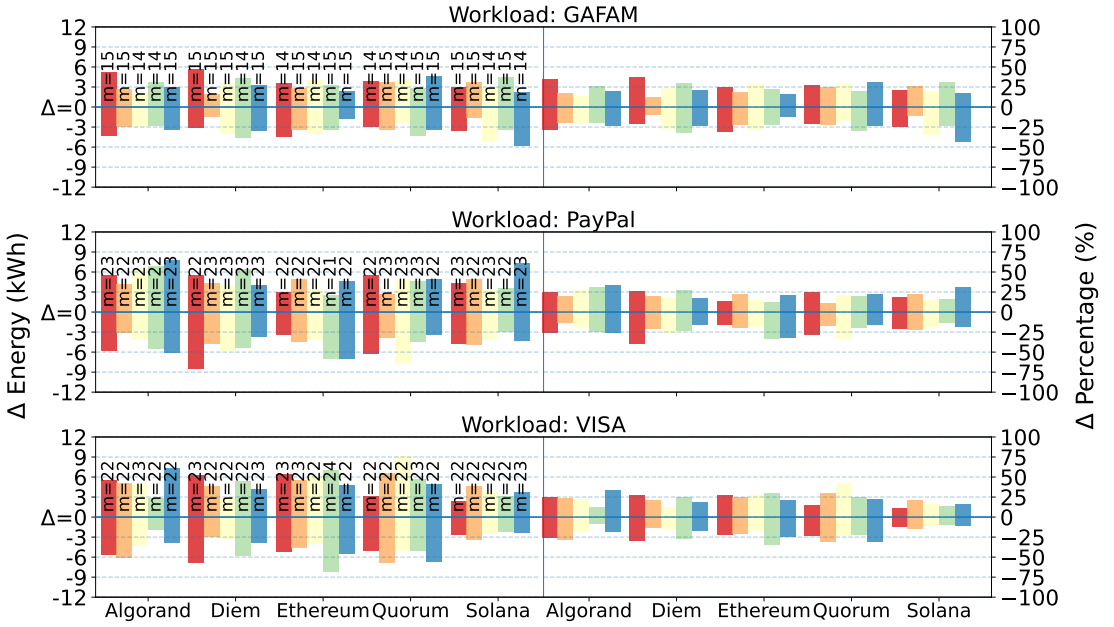
(b) 40 nodes – GAFAM, PayPal & VISA – Latency deviation

Figure (6.2) Latency deviation (same conventions as Figure 6.1).

and worst-case indices, this suggests that heavy tails are driven by specific blockchain-workload combinations: for a fixed configuration, runs usually cluster tightly around the mean, but certain regimes repeatedly trigger large swings that are relevant for service-level guarantees. Overall, blockchain type is the main driver of performance variability; workload amplifies/attenuates that variability depending on the platform, with topology remaining a secondary but operationally relevant factor.



(a) 10 nodes – GAFAM, PayPal & VISA – Energy deviation



(b) 40 nodes – GAFAM, PayPal & VISA – Energy deviation

Figure (6.3) Energy deviation (same conventions as Figure 6.1).

6.4 Discussion and Limitations

In this section we interpret the variability patterns through blockchain and network design, extracting practice-oriented guidance for system designers and benchmark authors. Where attribution is uncertain, we mark it as informed, data-driven inference. From a dependability viewpoint, dispersion and failures are first-class outcomes that can

Table (6.5) Experimental repeatability for each blockchain-topology pair (same conventions as Table 6.2).

Blockchain	**Topology**	IQR% TPS	Std% TPS	IQR% Lat	Std% Lat	IQR% En	Std% En
Algorand	fat-tree	11.34	7.61	5.78	7.71	16.01	13.83
Algorand	full mesh	3.43	6.54	4.79	14.32	20.51	14.92
Algorand	hypercube	8.48	7.99	8.10	13.98	14.79	12.37
Algorand	scale-free	7.66	7.65	12.93	11.59	12.42	12.69
Algorand	torus	8.07	5.55	3.01	10.41	14.58	13.46
Diem	fat-tree	11.01	11.00	19.19	13.83	16.12	12.89
Diem	full mesh	6.70	6.71	10.91	8.47	12.02	10.49
Diem	hypercube	5.41	4.25	12.22	11.26	13.38	10.87
Diem	scale-free	5.54	5.22	9.33	9.62	18.44	14.62
Diem	torus	2.52	5.87	9.37	7.54	19.41	14.88
Ethereum Clique	fat-tree	11.68	15.12	7.09	7.21	23.13	17.17
Ethereum Clique	full mesh	7.68	8.07	8.40	6.82	14.58	12.21
Ethereum Clique	hypercube	10.77	10.47	6.04	6.03	13.79	14.54
Ethereum Clique	scale-free	15.79	14.42	8.24	8.28	13.22	12.51
Ethereum Clique	torus	13.23	11.61	10.73	8.07	21.30	14.72
Quorum IBFT	fat-tree	49.72	70.88	47.60	118.83	18.00	13.40
Quorum IBFT	full mesh	44.57	67.09	17.16	148.37	15.31	13.44
Quorum IBFT	hypercube	28.75	64.47	32.52	95.11	19.33	14.79
Quorum IBFT	scale-free	91.74	101.71	99.89	138.98	14.38	13.36
Quorum IBFT	torus	11.28	47.22	12.95	60.35	18.99	13.77
Solana	fat-tree	108.53	103.71	27.87	73.70	17.84	12.71
Solana	full mesh	12.13	35.88	12.50	20.25	16.28	11.89
Solana	hypercube	28.30	43.04	21.15	41.55	13.82	10.96
Solana	scale-free	100.98	141.68	85.15	129.74	16.98	12.34
Solana	torus	39.88	58.30	7.75	42.64	16.34	14.98

undermine service guarantees; thus we treat performance as a distribution and report stability and tails rather than one-shot means, even in controlled setups.

6.4.1 Positioning the Dataset and Study Scope

As shown in Table 6.1, earlier work targeted P1 in the predictability taxonomy (mean-centric performance or energy with respect to topology) and used only 3 repetitions per configuration. Here we move to P2 and treat dispersion as a first-class output: each configuration is run 10 times, enabling both typical-dispersion metrics (IQR%, Std%) and worst-case-deviation metrics (Δ_{abs}, $\Delta^{\downarrow}/\Delta^{\uparrow}$, $\Delta^{\downarrow}\%$, $\Delta^{\uparrow}\%$), as well as ANOVA and ICC over the full dataset. Mean trends match prior mean-centric reports [108, 110] under comparable configurations, by design. We show that dispersion, tails, and failures can overturn interpretation: "best on average" may still be brittle, as exposed by WCD/WCD% (Table 6.3) and run-level deviations (Figures 6.1–6.3). The statistical treatment in §6.2.3 and the results in §6.3 support a coherent picture. First, throughput and latency exhibit substantially higher dispersion than energy: typical spread (IQR%, Std%) is moderate for several systems, but can become large under adverse blockchain-topology-workload combinations, whereas energy is confined to a narrower band and is primarily shaped by workload. Second, factorial ANOVA attributes most TPS and latency variance to the blockchain factor and its interaction with the workload, with topology and validator-set size acting as secondary modulators on average. Third, ICC values above 0.8 for TPS and latency indicate that runs of the same configuration are internally consistent: most

Table (6.6) Experimental repeatability for each blockchain-workload pair (same conventions as Table 6.2).

Blockchain	**Workload**	IQR% TPS	Std% TPS	IQR% Lat	Std% Lat	IQR% En	Std% En
Algorand	GAFAM	24.36	30.70	15.37	19.48	19.98	14.31
Algorand	PayPal	0.50	0.37	0.76	0.72	11.64	12.19
Algorand	VISA	7.66	6.73	11.33	12.59	14.67	13.18
Diem	GAFAM	9.03	11.65	11.50	10.20	18.44	15.53
Diem	PayPal	1.68	1.25	15.81	16.40	17.02	12.08
Diem	VISA	9.30	6.39	3.06	3.07	13.31	10.58
Ethereum Clique	GAFAM	17.63	16.79	7.13	6.69	15.86	13.30
Ethereum Clique	PayPal	5.24	4.89	5.70	7.33	13.63	12.64
Ethereum Clique	VISA	9.50	9.58	9.04	8.16	21.79	14.46
Quorum IBFT	GAFAM	53.70	125.66	61.68	114.29	16.39	14.13
Quorum IBFT	PayPal	0.84	4.75	4.51	15.73	18.56	13.04
Quorum IBFT	VISA	67.57	155.40	33.46	172.59	17.01	14.06
Solana	GAFAM	34.71	35.34	13.49	20.91	17.34	15.30
Solana	PayPal	3.48	136.95	16.48	131.36	15.46	12.32
Solana	VISA	67.08	81.69	22.98	60.59	15.24	10.51

Table (6.7) Experimental repeatability for each blockchain with 10- and 40-node scaling (same conventions as Table 6.2).

	TPS				Lat (s)				En (kWh)			
Blockchain	IQR% 10n	Std% 10n	IQR% 40n	Std% 40n	IQR% 10n	Std% 10n	IQR% 40n	Std% 40n	IQR% 10n	Std% 10n	IQR% 40n	Std% 40n
Algorand	2.08	2.32	13.94	11.02	3.07	8.34	19.55	17.64	13.46	13.73	17.01	12.64
Diem	6.02	5.71	5.06	6.34	9.46	7.34	16.11	13.87	13.53	10.70	18.71	14.36
Ethereum Clique	15.49	15.62	5.18	4.65	10.12	10.40	5.73	4.88	14.82	14.68	19.27	13.67
Quorum IBFT	55.61	56.20	43.84	56.99	29.83	149.69	27.22	95.68	18.09	13.89	16.11	12.96
Solana	65.04	59.68	16.22	59.48	27.63	47.48	3.53	32.18	16.60	14.58	15.67	12.45

variability is systematic across configurations rather than dominated by uncontrolled noise. The worst-case metrics in Table 6.3 then quantify how far individual runs can deviate from their configuration averages in absolute and relative terms. Even with deterministic networking and fixed software/hardware, run-to-run dispersion can arise from intrinsic nondeterminism in (*i*) protocols (leader/committee rotation, timeout-driven phase changes), (*ii*) implementation/runtime (asynchronous scheduling, batching thresholds, lock contention, garbage-collection pauses), and (*iii*) workload-induced queueing/backpressure near saturation; topology can amplify it via diameter/path diversity and contention. Thus extreme WCD% captures legitimate tails near congestion/failure boundaries (not removable Internet noise) and can be inflated when the mean is small.

6.4.2 System/Topology-Level Variance Interpretation

Following Laprie's dependability terminology [195], a root-cause analysis (RCA) traces deviations at the service boundary back along the fault-error-failure chain to specific internal faults. We do not claim such a full RCA; we only correlate the observed typical and worst-case variability with system and topology design choices to qualitatively position the studied blockchains. Topology can multiply intrinsic nondeterminism: large-diameter/low-diversity overlays turn small jitter into propagation skew, while dense or hub-heavy overlays reduce depth but increase contention/queueing. Thus fragilities may be exposed or masked by the blockchain-workload interaction (Table 6.4), with the most fragile topology being not known a priori. Recurring tail channels include fat-tree (multipath

vs. burst/queue synchronization), full mesh (all-to-all contention, especially multi-round BFT), hypercube (multi-hop jitter compounding across rounds/timeouts), scale-free (hub hotspots, backpressure, uneven queueing), and torus (long paths/low fan-out stressing dissemination and amplifying bursts). We observe consistent signatures (Figures 6.1–6.3, Table 6.3): (*i*) latency upward tails from multi-round commit/round changes; (*ii*) high peak TPS with heavy tails from pipelining+backpressure near thresholds; (*iii*) moderate typical dispersion but asymmetric extremes from committee churn/stragglers and path-sensitive propagation/verification; (*iv*) low dispersion from bounded pacing and conservative admission/block sizing; (*v*) frequent *NO* or m=0 indicating near-collapse, non-operational regimes.

Ethereum Clique consistently falls in a "stable to moderately stable" regime under our emulated overlays. Its PoA-style commit path with fixed block interval, conservative gas targets, and devp2p gossip [191, 308] keeps coordination overhead relatively modest. This matches low IQR% and Std% for both TPS and latency (Table 6.2) and comparatively small worst-case swings in Table 6.3 (limited Δ_{abs}, $\Delta^{\downarrow}/\Delta^{\uparrow}$), making Clique the most predictable system overall in our campaign.

Diem shows mid-to-low dispersion, consistent with a HotStuff-family commit path driven by a pacemaker that bounds progress [38, 347]. Typical spread around the mean remains limited for TPS and latency and worst-case deviations ($\Delta^{\downarrow}/\Delta^{\uparrow}$, $\Delta^{\downarrow}\%$, $\Delta^{\uparrow}\%$) are generally moderate. Instability mainly appears when quorum paths lengthen or bursts increase coordination work rather than in the baseline regime, which aligns with the ANOVA shares and with Diem's design as a tightly controlled, permissioned platform.

Algorand's committee-based BBA* with VRF-based selection [147] yields IQR% and Std% for TPS and latency comparable to Diem, indicating reasonably tight typical behavior. However, committee churn and stragglers along unfavorable paths can produce asymmetric tails: the worst-case metrics show non-negligible $\Delta^{\downarrow}$ and $\Delta^{\uparrow}$ excursions and higher energy dispersion, reflecting the extra verification and committee work. This suggests a profile where day-to-day behavior is stable but extreme runs can still drift significantly when the committee interacts poorly with overlay and workload.

Quorum IBFT is markedly sensitive to delay skew and head-of-line blocking, particularly on dense overlays [117]. A multi-round BFT path with proposer rotation and round changes can accumulate stalled rounds under skewed RTTs or backpressure. This is reflected in both typical dispersion (large IQR% and Std% for TPS and especially latency) and in the worst-case metrics, where Quorum IBFT attains some of the largest Δ_{abs} and $\Delta^{\uparrow}$ values for latency, with $\Delta^{\uparrow}\%$ approaching the highest levels in the dataset. These patterns are compatible with an "unpredictable under stress" profile: acceptable typical behavior, but long and volatile tails when the network or workload pushes the system close to saturation.

Solana's aggressive pipelining, precomputed leader schedule, and Turbine-style fan-out [296, 297, 342] enable high peaks but expose the system to pronounced backpressure and propagation gaps when overlays deepen or workloads approach saturation. Leader-based scheduling plus tree-structured dissemination can amplify local slowdowns into a

large variance. This is visible in both the typical metrics (high TPS IQR% and Std%) and the worst-case ones, where Solana records the largest Δ_{abs} for throughput and some of the highest $\Delta^{\uparrow}\%$ values (including the 1,200% outliers). The resulting profile is "high-performance but variance-prone": excellent peak throughput, but substantial sensitivity to topology and workload stress.

Across blockchains, dispersion is shaped by a small set of interacting design levers: commit-path pacing (round structure, leader or committee churn, timeout adaptation) [117, 147, 308, 347]; admission and slot sizing relative to propagation-and-verify conditions; overlay and dissemination strategy (diameter, fan-out, randomness vs. trees) [297]; workload saturation and backpressure; implementation/runtime choices such as batching, verification, and buffering [222]. The ANOVA results align with this view: the blockchain factor aggregates many of these levers, while workload and $B \times W$ capture how close each system operates to saturation; topology and size mainly act as multipliers that either expose or mask fragilities, especially in the tails highlighted by the worst-case metrics.

6.4.3 Implications for Practice and Benchmark Design

In addition to comparative results, Lilith acts as a decision-support framework for multiple stakeholders. For protocol developers, it enables controlled sensitivity analyses of blockchain behavior under topology changes, resource constraints, and adverse network conditions, clarifying trade-offs among throughput, latency, robustness, and energy efficiency.

For system designers, our results point to a small set of concrete levers. Admission control and block/gas sizing should be delay-aware, adjusting to observed propagation and backlog (as in target-utilization mechanisms like EIP-1559 [66]) so that both typical dispersion (IQR%, Std%) and worst-case swings ($\Delta^{\uparrow}/\Delta^{\downarrow}$, $\Delta^{\uparrow}\%$, $\Delta^{\downarrow}\%$) stay bounded. Pacemakers, leader schedules, and committees should adapt to RTT variance and congestion [347], while overlays should favor randomized or multipath gossip over brittle trees [297] to avoid single-branch slowdowns. Finally, signature aggregation and parallel verification [222], combined with operating away from saturation, help keep both average behavior and tails under control.

For network operators and consortium deployers, topology should be treated as an explicit deployment parameter: connectivity-rich topologies may favor peak performance, while structured alternatives such as fat-tree can improve energy efficiency in several regimes. Lilith therefore supports what-if analyses for topology selection, validator placement, and reconfiguration.

For benchmark designers and researchers, Lilith integrates topology control, workload injection, monitoring, and repeated execution in one pipeline, enabling reproduction of prior studies, fairer assessment of performance claims, and clearer separation between network-substrate and protocol-level effects. Experimental repeatability should be a prerequisite for performance claims. Means alone are insufficient: aggregate key performance indicators should be accompanied by dispersion around the mean (IQR%, Std%) and

worst-case deviations (per-configuration Δ_{abs}, $\Delta^{\downarrow}/\Delta^{\uparrow}$, $\Delta^{\downarrow}\%$, $\Delta^{\uparrow}\%$). Experiments should target level P2 with enough repeated configurations to support per-configuration indices, ANOVA, and ICC. Topologies should span degree/diameter/path-diversity extremes and include at least one delay-sensitive stress workload (*e.g.,* VISA) to expose variance-prone regimes. Head-to-head differences smaller than intra-configuration dispersion should be treated as inconclusive and emphasis should be placed on trends that persist across topologies and workloads rather than on single favorable configurations.

6.4.4 Experimental Limitations

This evaluation has explicit scope and boundaries. We fix a single controlled cluster as computing substrate and emulate geo-distribution at the network layer. This isolates experimental repeatability but does not claim cross-infrastructure reproducibility/replicability. The cluster scale is limited to tens of nodes; larger deployments may surface additional effects (*e.g.,* long-tail congestion or queuing). Our node counts (10-40) target the size of validator committees/quorums common in permissioned deployments (*e.g.,* IBFT production networks commonly running with 4-8 validators [174]) and in per-epoch committees of some public chains (*e.g.,* EOSIO-based networks with 21 elected block producers per round [209]). Our claims therefore speak to the execution unit that actually runs consensus, not to the global population of nodes. Claims are therefore scoped to committee-scale deployments (10-40 nodes) under identical software and emulated network conditions; no extrapolation is made to Internet-scale overlays with hundreds or thousands of nodes, heterogeneous validator populations, or time-varying/adversarial networks.

The five topologies are archetypes chosen to span degree/diameter/path-diversity extremes: full mesh as an idealized upper bound (typical of small committees), fat-tree approximating leaf-spine cloud fabrics, scale-free capturing heterogeneous overlays, and torus/hypercube as regular stressors. We do not claim these mirror any single production layout; they are controlled extremes to probe sensitivity. Containerization details are in §6.2; since the container stack is fixed across runs (host networking, host-mounted volumes), it does not confound within-configuration dispersion [133]. Ten runs per configuration improve robustness but may still under-sample tails; *e.g.,* a 1% failure probability has $1 - (0.99)^{10} \approx 9.6\%$ chance of appearing at least once. Space limits preclude cumulative distribution function (CDF) and confidence interval (CI) plots; we therefore report robust dispersion and tail metrics (IQR%, Std%, WCD/WCD%) and release per-run data to support downstream CDF, CI, and Kolmogorov-Smirnov analyses. With n=10, statistical power is limited for small effects and higher-order interactions; exhaustive distribution-level analyses across all configurations would be bulky and would also require multiple-comparison control. We therefore keep the thesis variance-first and leave confidence-interval or quantile bands, together with representative distribution-level analyses, to future work.

While Lilith (via Kollaps) can deterministically emulate network impairments (latency,

loss, partitions/topology changes) [108], we assume benign validators running reference implementations and do not inject protocol-level Byzantine actions (*e.g.,* equivocation/invalid votes); thus we do not quantify malicious behavior. Network-level adversarial patterns are feasible within Lilith's control layer, whereas protocol-specific Byzantine fault injection requires system-specific hooks and is left to future work. As blockchain software evolves and ledger state grows, future work should run long-lived controlled experiments that track version drift and state size to assess their impact on predictability.

These constraints do not weaken the main message: even in favorable and controlled setups, achieving experimental repeatability is challenging. Performance-evaluation conclusions should be drawn from trends that persist across topologies and runs, after pinning network parameters and software versions; head-to-head differences smaller than intra-setup variability (P2 configuration) should be treated as inconclusive. Extending the parameter space (more nodes, richer workloads, adversarial networks, Byzantine faults, longitudinal runs) and coordinating experiments across multiple clusters with consistent network characteristics are natural next steps toward reproducible and predictive blockchain performance evaluation.

Chapter 7

Economic Efficiency: An Entropy-Based Approach

This chapter, which is based on the article [109], introduces the Entropy Balance index (EB-index), a novel metric for quantifying economic efficiency in blockchain ecosystems. After introducing the issue of economic efficiency (§7.1) and surveying the existing literature (§7.2), we define the EB-index by aggregating multiple heterogeneous indicators – such as transaction volume, address activity, and wealth distribution – into a single entropy-derived score (§7.3). When applied to real data from six major cryptocurrencies, it highlights systemic differences in economic dynamics and resource concentration (§7.4).

7.1 Economic Efficiency

With the expansion of applications of cryptocurrencies and blockchains, attention has increasingly turned to assessing their properties, especially in terms of efficiency. Despite the various possible interpretations of this word – such as performance efficiency based on transactions per second [108, 144], energy consumption in blockchain networks [68, 110], or price trends in crypto markets [187] – its use in the realm of cryptocurrencies introduces new dimensions. This is due to the unique features associated with blockchain technology, characterized by decentralized governance, peer-to-peer interactions, and a heterogeneous structure and range of use cases for digital assets. For instance, to the best of our knowledge, no study has thoroughly explored the *economic efficiency of cryptocurrencies* by considering key factors such as supply distribution, user participation, and exchange activity; all of these are dimensions in which wealth concentration [185, 276], low engagement [213, 337], and asset dormancy [19, 139, 294] can undermine network economic vitality. Table 7.1 shows these issues with regard to various cryptocurrencies up to April 2025. Notably, Bitcoin has a high asset dormancy (up to 99.56%), while Ripple exhibits a low engagement (4.4%). Moreover, the lack of a clear conceptual framework for analyzing economic efficiency in this context underscores the need to define it through measurable links between network behavior and core economic indicators.

This study examines the concept of economic efficiency for the class of cryptocurrencies

Table (7.1) Wealth concentration, engagement, and asset dormancy of major cryptoassets from July 2010 up to April 2025. For wealth concentration we use a Gini coefficient adapted to average balances over class intervals. Engagement is defined as the share of addresses with positive balance over the total. Asset dormancy is calculated as one minus the ratio of active supply to circulating supply. Source: *Coin Metrics* [86].

Asset	**Wealth Concetration** (Gini coeff.)		**Engagement** (%)		**Asset Dormancy** (%)	
	Min	Max	Min	Max	Min	Max
Bitcoin (BTC) [235]	0.31	0.97	0.33	24.73	84.69	99.56
Ethereum coin (ETH) [331]	0.84	0.89	0.41	30.49	63.03	98.32
Ripple coin (XRP) [74]	0.94	0.98	0.08	4.40	36.74	99.79
USD Coin (USDC) [82]	0.93	0.99	0.86	90.34	5.21	89.29
Dogecoin (DOGE) [220]	0.82	0.98	0.70	90.78	64.78	99.54
Cardano coin (ADA) [170]	0.46	0.97	0.07	29.07	78.03	98.25

that use public distributed ledgers to record transfers, offer accessible data, and leverage blockchain for transparency, traceability, and consistency [210]. It is worth noting that similar analyses in traditional economies are hindered by limited access to raw data – controlled by centralized authorities – and inconsistencies in reporting across sources, such as variable monetary aggregates calculated by central banks [126]. Since data accessibility is ensured within blockchains, the problem to tackle is the lack of a unifying index expressing economic efficiency that enables a comprehensive comparison of cryptocurrencies instead of proceeding parameter by parameter.

We propose a theoretical framework that uses *Shannon entropy* [287] – a foundational concept in *information theory*, originally introduced by Claude Shannon in 1948 – to define the Entropy Balance index (EB-index), an economic index capable of aggregating a set of parameters describing economic qualities of a cryptocurrency. Our method follows these guiding principles: (*i*) rest on a solid theoretical foundation, (*ii*) remain adaptable to changes in parameters, (*iii*) reward cryptocurrencies exhibiting well-balanced values for the chosen parameters, and (*iv*) support parameter weighting without compromising the theoretical foundation.

Claude Shannon's seminal paper "*A Mathematical Theory of Communication*" [287] introduced a rigorous, axiomatic framework for quantifying information in communication systems. Entropy has subsequently found extensive use in diverse fields – from linguistics and neurobiology to machine learning and medical diagnostics – underscoring its flexible role as an index of exchanged information, the degree of fragmentation of a set, or the balance in the distribution of resources in both natural and engineered systems [28, 70, 73, 112, 129, 132, 148, 171, 184, 326]. Its role as a measure of economic (in-)equality was already formalized by Theil [311], who proposed entropy-based and divergence-based indices as alternatives to the Gini index, highlighting their decomposability across population subgroups. There are some applications also in the context of blockchain, for example to quantify the degrees of decentralization [150, 333], to measure the stability in blockchain consensus dynamics [21], and to express portfolio diversification [270].

To illustrate the robustness of our approach, we apply the entropy framework by using two distinct sets of economic qualities, *Set1* and *Set2*, each representing structural and behavioral aspects of cryptocurrency economies. *Set1* focuses on a minimal combination of basic financial activity metrics, while *Set2* adopts a richer, more granular perspective on network dynamics and internal economic organization. These attributes guide the selection of corresponding on-chain parameters, which we retrieve from the *Coin Metrics*® platform [86], ensuring an empirical and data-driven rooting. The entropy based on these two sets of features is then employed to evaluate six major cryptocurrencies with high market capitalization, spanning various economic roles and use cases (payments, smart contracts, stablecoins, and memecoins): Bitcoin (BTC) [235], Ethereum coin (ETH) [331], Ripple coin (XRP) [74], USD Coin (USDC) [82], Dogecoin (DOGE) [220], Cardano coin (ADA) [170].

This study paves the way to a foundational and flexible methodology for evaluating economic efficiency in cryptoassets and offering insights to investors, policymakers, and researchers navigating the evolving landscape of blockchain-based economies. As part of this work, we release our refined dataset (available at `https://doi.org/10.5281/zenodo.15221823`) for result reproducibility as well as independent analysis execution.

7.2 Efficiency of Cryptocurrencies: A Literature Review

The literature review reveals the absence of a universally accepted definition of efficiency in this setting. Among the various interpretations, some focus on energy consumption and expenditure [68], while others emphasize market price trends by analyzing fluctuations in cryptocurrency prices and their implications for economic stability and efficiency [187]. Still others assess efficiency based on the production objectives of the considered economy, including the presence or absence of technological components [36]. In [115], the discussion centers on technical and scale efficiency, which differ in scope. The former measures how effectively resources are allocated to maximize output, so as to ensure the optimal utilization of inputs, while the latter examines the relationship between input growth and output expansion, in order to identify whether economies of scale are being achieved [115]. Additionally, in [115], efficiency is evaluated both qualitatively – by comparing actual delivery times with planned schedules – and quantitatively – by analyzing the ratio of actual versus expected outputs relative to expenditures.

7.3 The Quest for an Aggregated Economic Efficiency Index

A possible way to address the divergent interpretations of economic efficiency recalled in §7.2 is to define this concept in cryptocurrencies by linking transfer dynamics with basic economic parameters, thus enabling a more structured and comprehensive analysis. In particular, it is necessary to aggregate parameters in a coherent way, with an index able to satisfy the guiding principles mentioned in §7.1. We propose using Shannon entropy to define a theoretical framework that measures economic efficiency in cryptocurrency

ecosystems through the Entropy Balance index (EB-index), aligning with established standards for reliability [226, 227, 246] and leveraging the transparency of blockchain data. Starting from a probability distribution associated with a set of events or parameters chosen to represent the qualities that determine the economic efficiency of a cryptocurrency, we employ Shannon entropy to measure how good the balance is within this set. The EB-index captures the complex internal structure of cryptoassets without relying on arbitrary single metrics. By aggregating heterogeneous dimensions – such as user activity, transactions, and supply – it reveals patterns that univariate analyses may miss. For instance, two assets may look similarly "successful" under a headline indicator (*e.g.,* market capitalization or transferred value), yet differ radically in their internal economic organization: one may exhibit broad participation and high active supply, while the other is sustained by a narrow set of heavy holders with low engagement and high dormancy. A balance-aware aggregation is designed to discriminate precisely these cases by penalizing profiles where economic activity concentrates in one (or few) dimensions while the others remain weak. We make this intuition concrete in §7.4, where the entropy-based construction is instantiated on real on-chain series (via Coin Metrics) and the resulting trajectories highlight how different parameter sets (*Set1* vs. *Set2*) can lead to materially different efficiency rankings and interpretations.

In this section we recall the basics of the entropy measure (§7.3.1) and then we discuss a number of economic parameters among which to select the ones to be used in the entropy formula (§7.3.2).

7.3.1 The Entropy Measure

Let $q_1, q_2, \ldots, q_k$ be a set of k *quality parameters*, with $l_i \leq q_i \leq u_i$, where l_i and u_i represent, respectively, the lower and upper bounds of the interval of variability for the corresponding parameter. We assume that each q_i can be oriented either as benefit-type (higher values indicate higher economic efficiency) or cost-type (lower values indicate higher economic efficiency). Before aggregation, we transform cost-type parameters so that all normalized scores follow a common direction, *i.e.,* larger normalized values always represent better economic efficiency. The idea behind our economic efficiency index for a cryptocurrency is to combine these parameters in a way that rewards cryptocurrencies exhibiting a well-balanced set of parameters with good performance. We interpret the resulting score as a proxy for the system's overall economic "health" and longer-run development potential, rather than as a mechanistic or causal driver of future growth. This interpretation follows the broader use of composite indicators in economics and finance: multidimensional summary measures are routinely constructed to capture latent conditions (*e.g.,* activity, robustness, efficiency) and are often used as leading indicators of subsequent outcomes, such as turning points and medium-term dynamics in economic activity. Similarly, in corporate finance, composite scores derived from multiple fundamentals are empirically linked to later outcomes (*e.g.,* distress and survival), illustrating how cross-sectional "health" summaries can carry predictive content. In the cryptocurrency domain,

a growing empirical literature finds that fundamentals related to adoption and network activity are value-relevant and can help predict returns, supporting the use of multi-parameter indicators as forward-looking proxies when interpreted cautiously. Importantly, we do not claim causality: the index provides a principled aggregation of contemporaneous conditions; any link with subsequent "development" should be understood as an empirical association to be tested (*e.g.,* with lagged regressions on future adoption, liquidity, or valuation proxies). This composition must allow for changes in the number of parameters without affecting the nature or compromising the coherence of the index.

The first step is to normalize each q_i within the interval $[0, 1]$, thereby obtaining the set $R = \{r_1, r_2, \ldots, r_k\}$ with $0 \leq r_i \leq 1$. This normalization is necessary because we cannot compare quantities with different scales, *e.g.,* one parameter varying within $[0, 1]$ and another varying within $[0, u]$, where u is an unbounded real number.

The economic efficiency index should intuitively reach its maximum value when all r_i attain the upper bound of the interval $[0, 1]$, *i.e.,* when they are all equal to 1. When $r_i = 1$, we can assume the maximum economic efficiency for the single parameter i, as opposed to an economic inefficiency of 0. Conversely, as r_i approaches 0, it is customary to assume that inefficiency grows to infinity. From this perspective, we need to introduce an analytic function[1] $\mathcal{I}$ satisfying the following constraints:

$$\begin{array}{ll} \text{when } r_i = 1 & \text{we have } \mathcal{I}(r_i) = 0 \\ \text{when } r_i \to 0 & \text{we have } \mathcal{I}(r_i) \to +\infty \end{array} \tag{7.1}$$

It is well known that there exist infinitely many functions satisfying these constraints; among all the possible ones, we choose:

$$\mathcal{I}(r_i) = -\log_b r_i \tag{7.2}$$

This choice immediately leads to the *Shannon entropy* defined by using base-2 logarithm [287]:

$$H_2(P) = -\sum_{i=1}^{k} p_i \log_2 \ p_i \tag{7.3}$$

after the normalization:

$$p_i = \frac{r_i}{\sum_{i=1}^{k} r_i} \tag{7.4}$$

so as to derive a probability distribution (p.d.) $P = \{p_1, p_2, ..., p_k\}$ from the set $R = \{r_1, r_2, ..., r_k\}$ of parameters in the interval $[0, 1]$. Recall that:

$$0 \leq H_2(P) \leq \log_2 k \quad \text{where } H_2(P) = \begin{cases} 0 & \text{iff } P \text{ is degenerate} \\ \log_2 \ k & \text{iff } P \text{ is uniform} \end{cases} \tag{7.5}$$

P degenerate means it is in the form $0, \ldots, 1, \ldots, 0$, while P uniform corresponds to

[1] An analytic function is a function that (i) is locally representable by a convergent power series, hence it can be expressed as a sum of terms based on powers of the variable, and (ii) is differentiable at every point in its domain.

$p_i = 1/k \; \forall i$ [96, 287]. By taking the logarithms to the base k, we obtain a normalization of the entropy:

$$0 \leq H(P) \leq 1 \tag{7.6}$$

The motivation for choosing the Shannon entropy lies in the fact that it can be proven, through a theorem, that Shannon entropy is the *unique* function, among infinitely many possible ones, that satisfies a specific set of postulates outlining the natural properties an information measure should reasonably possess [6, 100, 287].

The distinguishing postulate is the so-called *branching property*, expressed as follows in the case of k events:

$$H(p_1, p_2, \ldots, p_k) = H(p_1 + p_2, p_3, \ldots, p_k) + (p_1 + p_2) H\left(\frac{p_1}{p_1 + p_2}, \frac{p_2}{p_1 + p_2}\right) \tag{7.7}$$

that describes how entropy behaves when a p.d. is broken down into successive steps. The relation expresses the average information loss incurred when two events are grouped together and made indistinguishable; this loss is given by the entropy of the two events, weighted by the sum of their probabilities. Furthermore, it should be noted that this branching property can be generalized to a partition $\{ \mathcal{J}_i \mid 1 \leq i \leq m \}$, $m < k$, of $\{1, 2, \ldots, k\}$ with p.d. $S = \{s_1, s_2, \ldots, s_m\}$:

$$H(p_1, p_2, \ldots, p_k) = H(s_1, s_2, \ldots, s_m) + \sum_{i=1}^{m} s_i H^{(i)} \tag{7.8}$$

where $H^{(i)} = -\sum_{j \in \mathcal{J}_i} \left(\frac{p_j}{s_i}\right) \log \left(\frac{p_j}{s_i}\right)$ is the entropy associated with $\mathcal{J}_i$ and $\sum_{j \in \mathcal{J}_i} \frac{p_j}{s_i} = 1$ so that p_j / s_i is effectively a p.d. In this last case, the relation expresses the average information loss incurred when some subsets of events are grouped together and made indistinguishable; this loss is given by the sum of the entropies related to the subsets, weighted by their probabilities [6]. These equations provide the theoretical foundation underpinning entropy and our approach.

Another important advantage of the Shannon entropy is that it is possible to associate a set of *weights* or *utilities* with the p.d. P, say $\mathcal{W} = \{w_1, w_2, \ldots, w_k\}, w_i \geq 0$, without losing the branching property. This leads to the definition of the *Beliş-Guiaşu weighted entropy* [39]:

$$\mathcal{H}(P; \mathcal{W}) = -\sum_{i=1}^{k} w_i \cdot p_i \log_2 \; p_i \tag{7.9}$$

It is simple to check that the branching property still holds as soon as we correctly define the weight of the composition of two events e_1 and e_2 as a weighted sum of the utilities of the single events:

$$\mathit{Weight}(e_1, e_2) = w_{12} = \frac{p_1}{p_1 + p_2} w_1 + \frac{p_2}{p_1 + p_2} w_2 \tag{7.10}$$

The branching property can then be expressed in the following way ($p_{12} = p_1 + p_2$):

$$\mathcal{H}(p_1, \ldots, p_k; w_1, \ldots, w_k) = \mathcal{H}(p_{12}, p_3, \ldots, p_k; w_{12}, w_3, \ldots, w_k) + p_{12} \cdot \mathcal{H}\left(\frac{p_1}{p_{12}}, \frac{p_2}{p_{12}}; w_1, w_2\right) \tag{7.11}$$

The presence of the weights w_i in the weighted entropy (7.9) causes the function to be zero when $w_i = 0$, for all i. Moreover, it is also reduced to zero when the useful events are impossible or the possible events are useless. The last case occurs also when P is degenerate.

7.3.2 Economic Parameters

Among the most commonly used methods to measure complex socio-economic phenomena – such as Gross Domestic Product (GDP) or Human Poverty Index (HPI) – composite indices play a central role. They are widely used due to their adherence to established theoretical and functional requirements that ensure their reliability [227, 246]. According to [226, 227, 246], guidelines and steps to summarize a set of economic indicators and construct a composite index are the following:

1. *Phenomenon definition*, *i.e.,* the theoretical framework to achieve a clear conceptualization of what is being measured and establish selection criteria for determining whether an indicator should be included.

2. *Data modeling*, *i.e.,* the selection of relevant, timely, and accessible data sources, considering their correlation to minimize redundancy.

3. *Data processing*, composed by (*i*) normalization, which ensures comparability between indicators, (*ii*) weighting, which assigns priority to indicators based on their relevance, and (*iii*) aggregation, which combines the normalized indicators into a unified framework.

As discussed in §7.3.1, our entropy-based approach adheres to all these phases, providing a solid foundation for evaluations. In the following, we illustrate how each phase contributes to the construction of our framework for assessing economic efficiency.

Phenomenon definition. The economic efficiency analysis we aim to conduct goes beyond purely economic parameters measured in some currency. It also considers additional aspects, such as the intensity and frequency of transfers, as well as user participation. To establish a solid foundation for this study on economic efficiency, we introduce in Table 7.2 several basic cryptocurrency parameters. Throughout this chapter, the unit of observation is the address rather than a real-world user/entity, consistent with the granularity of the public on-chain series provided by Coin Metrics. We therefore do not assume a one-to-one mapping between users and addresses: a single user may control many addresses (especially in UTXO systems due to address rotation and change outputs), while custodial services and exchanges can aggregate the activity of many users into

Table (7.2) Basic parameters.

Attribute	Acronym	Definition
Digital Asset Supply	S	The total amount of a digital asset (coin or token) available within the tokenomics of a cryptocurrency.
Current Supply	S_{current}	The number of coins or tokens that have been minted or mined to date.
Transfer Set	T	The set of all transfers t executed within the network during a given period.
Transfer	t	A single on-chain value movement. In account-based ledgers, a transfer involves one sender and one receiver address; in UTXO ledgers, a transaction may consume multiple inputs and create multiple outputs (including change), thus inducing multiple address-level transfers.
Sender of a Transfer	sender(t)	The sending address (account-based) or the set of input addresses spending UTXOs (UTXO-based).
Receiver of a Transfer	receiver(t)	The receiving address (account-based) or the set of output addresses created by a UTXO transaction (UTXO-based).
Transfer Amount	amount(t)	The value in coins or tokens associated with a transfer t.
Address Set	A	The set of all addresses a.
Funded Address Set	A_{funded}	The set of addresses holding at least 1 USD during a given period.
Active Address Sent Set	A_{active}	The set of unique addresses, counted only once, that are involved in sending transfers over a given period. It refers to all activities leading to a change in the ledger, excluding the null address used for issuance purposes.
Address	a	A pseudonymous identifier used to send/receive digital assets (not a unique real-world user). An entity may control multiple addresses and custodians/exchanges may pool many users into few addresses.
Address Balance	balance(a)	The value in coins or tokens associated with the address a.
Address Sent	sent(a, t)	The value in coins or tokens sent from address a during a transfer t.

a small set of hot/cold addresses. As a consequence, address-count-based indicators (*e.g.,* engagement, participation) should be interpreted as proxies for adoption rather than exact user counts, which can be influenced by privacy practices and by Sybil behavior, since pseudonymous identifiers can be created at negligible cost. For UTXO-based networks, a single entity can also be associated with many distinct UTXOs across multiple transactions/addresses, which further reinforces the need to interpret "user activity" metrics at address level. We report these limitations explicitly and leave entity-level robustness checks (*e.g.,* wallet/entity clustering) as future work; such clustering requires additional assumptions and can be unreliable in the presence of mixing protocols.

We use these basic parameters to clearly define and formalize the set of key economic quality attributes that characterize these digital economic systems. In particular, we identify *primitive quality* (PQ) attributes, which are defined by a single parameter, and *derived quality* (DQ) attributes, which result from a mathematical combination of multiple PQs. These are presented in Table 7.3 along with their definitions, formulas, units, and range intervals. When applicable, we explicitly indicate the use of the native cryptocurrency unit (marked as "Ntv") rather than USD.

Data modeling. Having properly defined the quality attributes, we can now identify the on-chain parameters to be measured for each quality, based on a selection criterion that reflects their relevance to the corresponding attribute. To achieve this, we can exploit blockchain data intelligence platforms that provide several on-chain data in user-friendly

Table (7.3) Key economic quality attributes.

Economic Attribute	Acronym	Definition	Formula	Unit	Interval
PQ1 - Transferred Value	T_{value}	The total Ntv value exchanged within the system over a given period, excluding issuance account transfers, which record asset creation (*e.g.*, Bitcoin coinbase transactions).	$T_{\text{value}} = \sum_{t \in T} \text{amount}(t)$	Ntv	$[0, +\infty)$
PQ2 - Transfer Count	T_{count}	The total number of transfers t executed within the network during a given period.	$T_{\text{count}} = \lvert T \rvert$	Count	$[0, +\infty)$
PQ3 - Number of Funded Addresses	NA_{funded}	The number of addresses continuously holding at least 1 USD during a given period.	$NA_{\text{funded}} = \lvert A_{\text{funded}} \rvert$	Count	$[0, +\infty)$
PQ4 - Number of Active Addresses Sent	NA_{active}	The total number of unique addresses, counted only once, that are involved in sending transfers over a given period.	$NA_{\text{active}} = \lvert A_{\text{active}} \rvert$	Count	$[0, +\infty)$
PQ5 - Active Supply	S_{active}	The amount of coins/tokens that have been moved at least once within a given time period, excluding double counting of the same units being recycled.		Ntv	$[0, +\infty)$
PQ6 - Daily Digital Asset to USD Price Rate	USD_{price}	The price in USD per native unit of the coin or token at the close of the day.		USD	$(0, +\infty)$
PQ7 - Market Capitalization	Cap	The total value of a cryptocurrency in USD, calculated by multiplying its current price by the total circulating supply of coins.	$Cap = S \cdot USD_{\text{price}}$	USD	$(0, +\infty)$
DQ1 - Participation	$A_{\text{participation}}$	The proportion of active addresses relative to the total addresses holding at least 1 USD. A higher ratio suggests a more engaged user base and a healthy level of participation within the cryptocurrency ecosystem.	$A_{\text{participation}} = NA_{\text{active}} / NA_{\text{funded}}$	Count	$[0, 1]$
DQ2 - Mean Transfer Size	MTS	The mean size of a transfer, measured in Ntv. It is calculated by dividing the total value transferred by the number of transfers between distinct addresses during a given period.	$MTS = T_{\text{value}} / T_{\text{count}}$	Ntv	$[0, +\infty)$
DQ3 - Mean Transfers per Active Address	MTA_{active}	The mean number of transfers per active address. It is calculated by dividing the total number of transfers by the number of active addresses during a given period. This metric provides insights into the intensity of usage per user.	$MTA_{\text{active}} = T_{\text{count}} / NA_{\text{active}}$	Count	$[0, +\infty)$
DQ4 - Active Supply Turnover Rate	TR	The ratio between the total value transferred and the active supply over a given period. A higher TR indicates more frequent economic activity and greater liquidity.	$TR = T_{\text{value}} / S_{\text{active}}$	Count	$[0, +\infty)$
DQ5 - Active Supply Ratio	ASR	The proportion of the current supply that is actively participating in transactions. A higher ASR signifies that a larger portion of the available cryptocurrency is being used rather than held passively.	$ASR = S_{\text{active}} / S_{\text{current}}$	Count	$[0, 1]$
DQ6 - Wealth Distribution	WD	The degree to which wealth is distributed across the network participants, where: I is a set of balance intervals each of which represents a range of account balances; $NA_{\text{funded}}(i)$ is the number of accounts whose balance falls within interval i; $S_{\text{current}}(i)$ is the total supply in interval i; $\text{avgbal}(i) = S_{\text{current}}(i) / NA_{\text{funded}}(i)$ is the average balance per address in interval i.	$WD = \frac{\sum_{k \in I} \sum_{h \in I} NA_{\text{funded}}(k) \cdot NA_{\text{funded}}(h) \cdot \lvert \text{avgbal}(k) - \text{avgbal}(h) \rvert}{2 \cdot NA_{\text{funded}} \cdot \sum_{i \in I} S_{\text{current}}(i)}$	Count	$[0, 1]$
DQ7 - Mean Transfer per Market Cap	$MTMC$	The mean size of a transfer over a specific period relative to the cryptocurrency's market capitalization.	$MTMC = (T_{\text{value}} \cdot USD_{\text{price}}) / Cap$	USD	$[0, +\infty)$

Table (7.4) Blockchain data intelligence platforms comparison.

Name	Assets and Protocols	API Access	Metrics
Bitquery [51]	40+	3-months as researcher	Blocks and transactions related
Glassnode [149]	600+	Limited free plan	400+
Blockdaemon [54]	23+	Limited free plan	Balances, blocks, fee estimator, transactions
Blockchair [53]	14	1-year 100K requests as student	Raw stats metrics per blockchain
Coin Metrics [86]	200+	2-year as researcher	~ 300

formats. Table 7.4 compares some of these platforms with respect to the available metrics, the number of accessible assets and protocols, and the type of API access.

We rely on *Coin Metrics* [86] as its indicators categorization aids us in identifying both blockchain and cryptocurrency components from economic and technological perspectives. Additionally, it offers a wide range of protocols and provides researchers with the opportunity to access premium data for free up to 2 years (upon request). In Table 7.5

Table (7.5) *Coin Metrics* parameters selection.

PQ	*Coin Metrics* Parameter	Description	Unit
PQ1	Xfer'd Val	The sum of Ntv transferred (*i.e.,* the aggregate "size" of all transfers) in a given period.	Ntv
PQ2	Xfer Cnt	The count of transfers in a given period, including all user-initiated actions recorded on the chain, excluding protocol-mandated changes like coinbase transactions or new issuance.	Count
PQ3	Addresses Count with Balance $\geq$ 1 USD	The total count of unique addresses holding at least 1 USD by the end of a given period.	Count
PQ4	Active Addresses (Sent)	The total count of unique sending addresses active in the network in a given period, excluding duplicates from previous activity.	Count
PQ5	1-Day Active Supply	The sum of unique native units that transacted at least once within a single day. Native units that transacted more than once are only counted once.	Ntv
PQ6	USD Denominated Closing Price	The price of the asset denominated in USD.	USD
PQ7	Market Cap	The value of the current supply in USD. Also referred to as network value or market capitalization.	USD
-	Current Supply	The sum of all native units ever created and currently visible on the ledger (*i.e.,* issued) in a given period.	Ntv
-	Val in Addrs w/ Bal $\geq$ X Ntv	The total of native units held in addresses with a balance of X Ntv or more at the end of a given period, excluding non-native tokens.	Ntv

we present the on-chain parameters from *Coin Metrics* that best represent the previously defined quality attributes, enabling a comprehensive study focused on economic efficiency within cryptocurrency networks [31, 223]. We use *Coin Metrics* to download this data with a daily granularity.

Data processing. The data processing stage is essential to ensure that economic efficiency is grounded in solid theoretical principles. To meet the methodological requirements of normalization, weighting, and aggregation, we adopt the rigorous framework of Shannon entropy [287].

After downloading the relevant metrics from *Coin Metrics*, we align all datasets to a common time reference, using the earliest available timestamp (2010-07-18) for PQ6 as the baseline (USD Denominated Closing Price). Next, we convert into USD all metrics originally expressed in native units. This applies to the following metrics from Table 7.5: `Xfer'd_Val` (PQ1), `1-Day_Active_Supply` (PQ5), `Current_Supply`, `Val_in_Addrs_w/Bal`$\geq$`X_Ntv`. Particular attention should be paid to the calculation of DQ6 (Wealth Distribution), which considers balance class intervals rather than individual address balances as in the original Gini formula. These intervals range from 0.001 units of the native currency up to 100K. Furthermore, we operate at the address-level granularity provided by Coin Metrics and do not attempt to de-anonymize or map addresses to unique entities. Accordingly, all address-based quantities are interpreted as proxies for user-level behavior; assessing sensitivity under entity-level clustering (and under alternative heuristics) is left as future work.

We then normalize data because normalization ensures that each metric contributes equally, independent of scale differences (*e.g.,* market cap or address count). While metrics may reflect protocol-specific factors, the entropy-based approach remains agnostic to context, focusing on internal distributions. For fairer comparisons, assets should ideally

Table (7.6) Selected cryptocurrencies and their financial and technical characteristics as of April 2025. Source: *Coin Metrics*.

Asset	Category	*Coin Metrics* Capitalization	Max Supply	Avg. Network Size	Underlying Blockchain	Consensus Mechanism
Bitcoin (BTC) [235]	Payment	$1,653.3 billions	21 millions Bitcoin (BTC)	21.886 [49]	Bitcoin	PoW
Ethereum coin (ETH) [331]	Smart Contract	$217.6 billions	No fixed supply	13.347 [48]	Ethereum	PoS
Ripple coin (XRP) [74]	Payment	$214 billions	100 billions Ripple coin (XRP)	150+ [269]	Ripple coin (XRP) Ledger	Consensus Protocol
USD Coin (USDC) [82]	Stablecoin	$40 billions	No fixed supply	Depends on the blockchain	Operates on multiple blockchains	Depends on the blockchain
Dogecoin (DOGE) [220]	Memecoin/Payment	$25.1 billions	No fixed supply	315 [291]	Dogecoin	PoW
Cardano coin (ADA) [170]	Payment/Smart Contract	$23.3 billions	45 billions Cardano coin (ADA)	N.A.	Cardano	PoS

belong to the same category. For each metric and asset (except PQ7), we use *min-max normalization* [226, 227, 246] based on the individual asset's range, taken within the entire lifetime of the cryptocurrency. For PQ7 (Market Capitalization), we apply a global normalization by using the *min* and *max* values across all assets to preserve comparability. To handle DQ6 (WD), where lower values indicate greater efficiency, we use the specific transformation $1 - norm\_val$ to reflect this inverse relationship. This adjustment ensures that all metrics behave consistently – higher normalized values always reflect greater economic efficiency – while remaining within the $[0, 1]$ range.

Finally, we compute the EB-index derived from the chosen parameters for an illustrative example by using both the standard formulation (7.3) and its weighted variant (7.9). Our analysis, encompassing all data and resulting output, is conducted on a daily granularity. The corresponding results are presented in §7.4.

7.4 An Illustrative Example

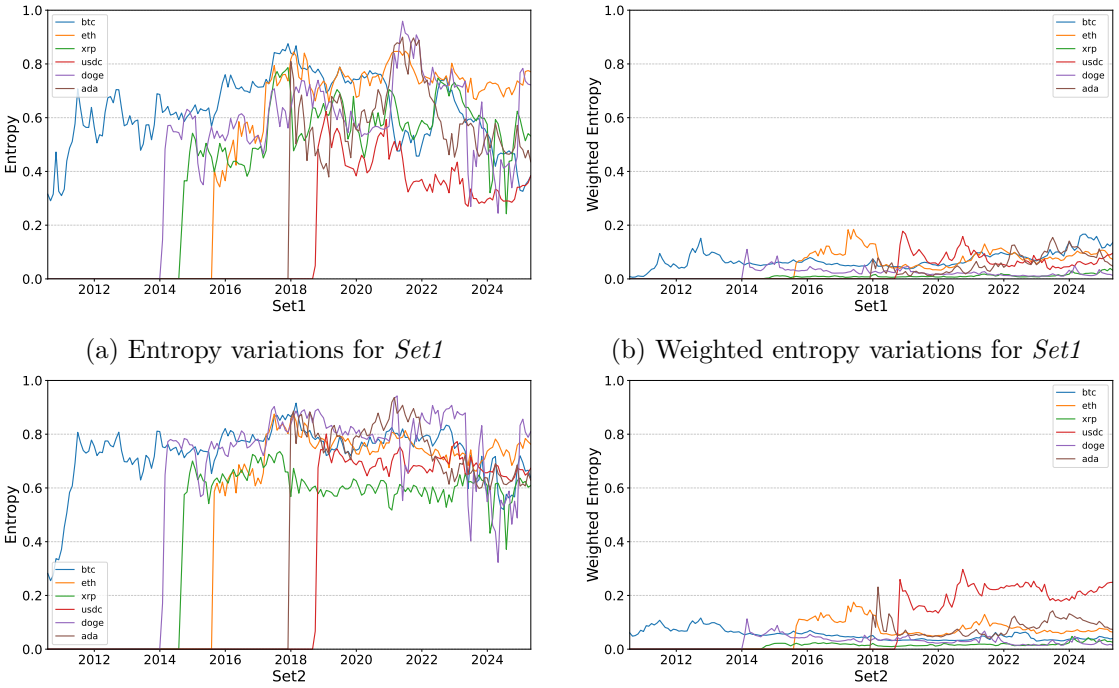
In this section, we present a case study to illustrate the application of our EB-index to evaluate economic efficiency in cryptocurrencies. Our intention is not to provide any conclusive assessment of the performance or quality of the selected cryptocurrencies, but rather to demonstrate the practical implementation of the proposed theoretical framework within a real-world financial context.

We conduct a comparative analysis of the entropies obtained by the six major cryptocurrencies cited in §7.1 – BTC, ETH, XRP, USDC, DOGE, and ADA – over the period from 2010 to 2025; their main financial and technical characteristics, such as capitalization, max supply, underlying blockchain, and consensus mechanism, are briefly summarized in Table 7.6. The choice of the aforementioned cryptocurrencies also takes into account the fact that not all cryptocurrencies have the desired parameters available on *Coin Metrics*.

The analysis is based on two distinct sets of derived quality attributes, constructed from selected parameters available on *Coin Metrics* (see §7.3.2): *Set1* and *Set2*, each designed to capture different layers of cryptocurrency economic behavior. *Set1* is composed of a minimal combination of fundamental financial activity metrics – such as market capitalization – primarily reflecting broader macroeconomic sentiment. In contrast, *Set2* offers a more granular view, incorporating a richer selection of attributes that describe structural network dynamics and internal economic organization, thereby capturing slower-moving, user-driven behavioral trends. The composition of these two sets is detailed in Table 7.7.

Table (7.7) Derived quality attributes chosen for the two sets *Set1* and *Set2*.

Set	Qualities	Description
Set1		It combines user activity metrics with market financial indicators.
	DQ1	Participation ($A_{\text{participation}}$)
	DQ3	Mean Transfers per Active Address (MTA_{active})
	DQ7	Mean Transfer per Market Cap ($MTMC$)
	PQ7	Market Capitalization (Cap)
Set2		It comprises six dynamic qualities, with a focus on user engagement, transaction characteristics, and the distribution of wealth within the network.
	DQ1	Participation ($A_{\text{participation}}$)
	DQ2	Mean Transfer Size (MTS)
	DQ3	Mean Transfers per Active Address (MTA_{active})
	DQ4	Active Supply Turnover Rate (TR)
	DQ5	Active Supply Ratio (ASR)
	DQ6	Wealth Distribution (WD)



(a) Entropy variations for *Set1*

(b) Weighted entropy variations for *Set1*

(c) Entropy variations for *Set2*

(d) Weighted entropy variations for *Set2*

Figure (7.1) Comparison between unweighted and weighted entropy for (a-b) *Set1* (DQ1, DQ3, DQ7, PQ7) and (c-d) *Set2* (DQ1, DQ2, DQ3, DQ4, DQ5, DQ6). For the weighted entropy the chosen weights are $w_i = r_i$.

7.4.1 EB-index Based on Unweighted Entropy: Good Balance

The entropy variations for the chosen cryptocurrencies, computed by using the standard formula (7.3), are illustrated in Figure 7.1.

The comparison across different attribute sets highlights how the selection of metrics

can substantially affect the observed performance and behavior of cryptoassets.

In *Set1*, which combines user activity with financial metrics, most assets exhibit a gradual growth with peak values between 0.85 and 0.9 – particularly Dogecoin (DOGE), Cardano coin (ADA) and Ethereum coin (ETH), which display the highest levels of entropy for the entire period. This suggests a peak in homogeneity and well-balanced development during that period. After 2020, ETH stabilizes and becomes the most consistent asset across the entire set, maintaining values steadily above 0.6 – a sign of structural resilience. Bitcoin (BTC), on the other hand, undergoes a temporary recovery between 2022 and 2023, followed by another sharp decline. The dynamics of the entropy values are generally well pronounced (0.25–0.9), with a lot of sharp peaks and bottoms, highlighting that the index is capable of capturing key aspects associated with variations in the economic parameters.

Set2, built on dynamic, interaction-based features such as transaction size, turnover rate, and wealth distribution, reveals higher and less dispersed values, with only the period of time from 2023 to mid 2024 characterized by some steep declines in value.

Ethereum exhibits the highest entropy, reflecting a well-balanced and resilient ecosystem, especially during market stress. As represented in Figure 7.2, in *Set1* a drop in DQ3 offset by rising Market Cap preserved balance, while in *Set2* opposing shifts in DQ5 and DQ6 had a compensatory effect.

It is worth noting that some assets can perform differently when using different sets of parameters. For example Ripple coin (XRP) ranks as the least efficient asset in *Set2*, with values averaging near 0.6 for the majority of the period, reflecting poorer distribution and transactional engagement. On the contrary, it can be considered average compared to the others in *Set1*. This underlines again the importance of carefully selecting parameters based on the characteristics of the cryptocurrency one aims to highlight.

While some global events – such as market crashes, regulatory changes, or liquidity shocks – may coincide with entropy shifts, their causal impact on the distributional structure of the parameters is not addressed in this work. Our primary goal remains the introduction of a well-founded economic index, flexible in both composition and number of parameters, capable of expressing the balanced integration of economic qualities as an indicator of systemic efficiency – irrespective of exogenous causes. It remains the responsibility of the researcher to choose the set of parameters and economic qualities that best suit her/his research.

7.4.2 EB-index Based on Weighted Entropy: Performance or Importance

The proposed EB-index assumes equal relevance for all selected economic qualities. However, this approach has a limitation: it rewards only the good balance among qualities, regardless of their absolute value. As an example, consider two cryptocurrencies: the first one with all normalized qualities reaching the maximum value $r_i = 1, \forall i$; the second one with $r_i = 0.5, \forall i$. From (7.4) it is clear that both the two corresponding p.d.'s

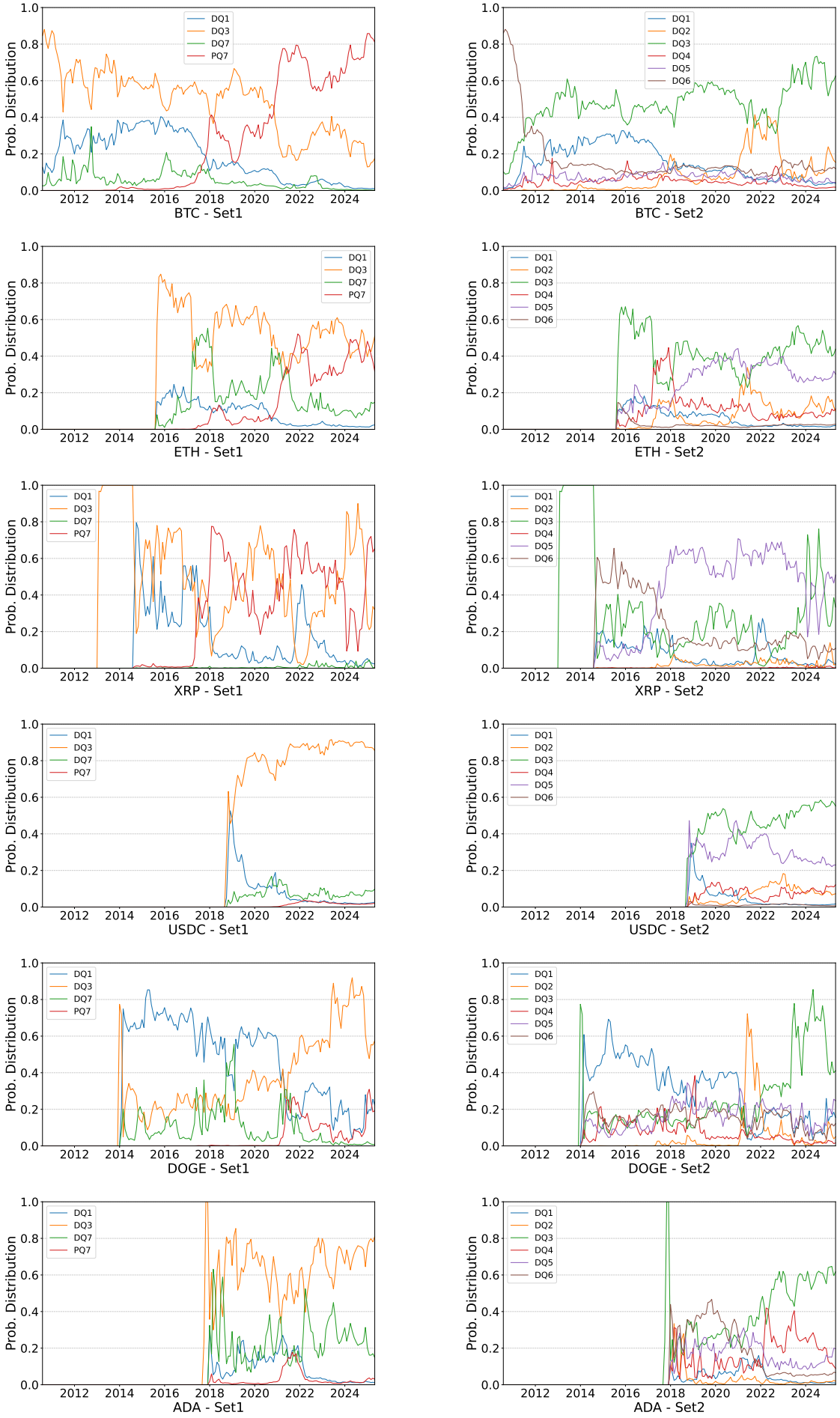


Figure (7.2) Probability distribution of the selected attributes within each set, computed for each asset.

are uniform and correspond to the maximum value of normalized entropy, *i.e.,* $H(P) = 1$ with k the base of the logarithm. But the first cryptocurrency has all absolute values greater than that of the second one. To overcome this limitation we can use the weighted entropy (7.9) and set $w_i = r_i$, so each quality contributes proportionally to its normalized absolute value. In this way, the entropy metric becomes sensitive not only to a good balance of the qualities, but also to the absolute performance of the system, thus better capturing the economic efficiency of the evaluated blockchain protocol. Note, however, that when we use the term "absolute", we are always comparing the value of the given quality to its own highest value over the historical series, within the context of the same cryptocurrency.

In Figure 7.1, the unweighted entropy calculation is shown alongside its weighted counterpart with $w_i = r_i$ for each quality q_i. This highlights a significant shift in the interpretation of the results, particularly for *Set2*, where interestingly the USD Coin (USDC) stablecoin emerges as the most balanced and highly entropic representation. Moreover, it is worth noting that the entropy score drops significantly below 0.3, indicating that a full balance among the selected economic features has not yet been achieved within the sets.

Another possibility offered by the weighted entropy is that of increasing the importance of some qualities to the detriment of others, by assigning $w_i = u_i$, where u_i is the *utility* (upper bound) of the corresponding quality. Note that in any case we can consider the last two weighting principles in a joint manner by setting $w_i = r_i \cdot u_i$, without losing the formal properties satisfied by the Shannon entropy.

Chapter 8

Conclusions

This chapter summarizes the dissertation's main results on blockchain efficiency, enabled by Lilith *and the EB-index as documented by artifact availability, in terms of performance, energy, experimental repeatability plus performance predictability, and economic balance (§8.1). It then outlines future work toward scalable, topology-aware, and sustainable benchmarking (§8.2).*

8.1 Summary of Results

This dissertation set out to compare and understand blockchains along four complementary axes – performance efficiency, energy efficiency, experimental repeatability plus performance predictability, and economic balance – under controlled yet realistic conditions.

To this end, we developed and used Lilith, a benchmarking framework described in Chapter 3 that (*i*) exposes network topology as a first-class parameter, (*ii*) orchestrates workloads deterministically, (*iii*) supports geo-distributed latency emulation derived from public cloud measurements, and (*iv*) enforces strict software versioning and rigorous experimental controls. In Chapters 4, 5, and 6, Lilith enabled a systematic exploration that connects architectural choices (consensus, deployment, and overlay structure) to observable outcomes (throughput, latency, energy per transaction, and stability of those measures).

In addition, in Chapter 7 we formalized the EB-index, an entropy-driven metric that aggregates multiple on-chain economic signals (*e.g.,* user activity, token distribution, and valuation proxies) into a single interpretable score that rewards balanced, system-wide improvements and remains coherent even as the set of tracked parameters evolves (via weighted or unweighted formulations).

We now recall our findings in more detail along with the related artifacts.

Performance (Chapter 4). We evaluated blockchain performance under diverse network topologies and AWS-informed geo-distributions, addressing research questions RQ1 and RQ2. Our results show that topology is not merely an environmental factor but a primary driver of end-to-end behavior: it shapes path lengths, queuing, and concurrency, hence it directly affects throughput and confirmation latency. The on-premise,

controlled setup made the experiments reproducible while remaining realistic, thanks to the integration of empirically observed inter-region network characteristics. Beyond answering RQ1 and RQ2, Lilith proved effective at managing complex infrastructures, reproducing cloud-like configurations, and exercising dynamic network setups in a principled manner.

Energy (Chapter 5). At a high level, energy efficiency emerges from the interaction among network topology, workload profile, and protocol design rather than from any single component. Denser overlays that shorten end-to-end paths tend to reduce energy per transaction at moderate load, while heavy or burst-driven traffic expose coordination costs and engineering limits across systems. Differences among blockchains reflect design choices – consensus mechanisms, degree of centralization, and implementation maturity – so no universal ranking holds across contexts. The most stable takeaway is prescriptive: align deployment and topology with the expected workload to reduce avoidable waste, as mismatches amplify inefficiencies. Finally, energy results should be read with their operating envelope (topology, load, software version) rather than as context-free point estimates, especially as public overlays evolve over time.

Experimental repeatability and performance predictability (Chapter 6). Using a tightly controlled, geo-emulated, multi-run dataset, we examined experimental repeatability and performance predictability in blockchain performance evaluation under nominally identical conditions. A variance-centric analysis based on 10 runs per configuration, combining typical-dispersion metrics (*e.g.,* IQR% and Std%), worst-case deviation indicators, factorial ANOVA, and intraclass correlation, shows that throughput and latency can still exhibit substantial and heterogeneous variability even when software and network settings are fixed, whereas energy remains comparatively steadier and is driven mostly by workload. Most of the observed variance in throughput and latency is explained by the blockchain–workload pair, while topology and validator-set size mainly act as secondary modulators that expose or attenuate existing fragilities. Balanced topologies such as torus, and in some cases full mesh, tend to host more stable configurations; VISA-like workloads consistently stress systems; and scaling from 10 to 40 nodes can reduce some absolute extremes while increasing relative sensitivity for certain blockchains. In the campaign, Ethereum Clique emerges as the most predictable system, Algorand and Diem occupy an intermediate regime, and Solana and Quorum IBFT show the largest tails. Lilith's contribution is therefore not to claim absolute reproducibility, but to make variability explicit, measurable, and attributable, so that blockchain performance claims are framed within quantified dispersion envelopes rather than reduced to simple point estimates.

Economic balance (Chapter 7). We introduced and applied the EB-index, an entropy-based metric – grounded in the axiomatic framework of Shannon entropy – designed to capture systemic balance across multiple economic parameters of a crypto-asset. The index rewards coherent, well-distributed progress across attributes and is robust to changes in the

number of tracked parameters. Empirical analyses over multiple attribute sets and time windows revealed that structural and valuation-driven components leave distinct signatures in the index, enabling a compact view of a system's internal economic organization. At the same time, the study clarified important practicalities: (*i*) the index's quality depends on the granularity and availability of on-chain data; (*ii*) certain proxies (*e.g.,* Gini via balance classes or Active Addresses Sent conflating economic and non-economic state updates such as approvals and contract interactions) can introduce noise; and (*iii*) the equal-weight assumption, while transparent, may misrepresent parameter salience in specific contexts. To address (*iii*), we discussed the Beliş-Guiaşu weighted-entropy formulation and outlined paths to data-driven or expert-elicited weights – all while preserving the index's axiomatic coherence. Overall, the EB-index offers a flexible lens for assessing structural balance without prescribing a universal metric set; efficiency is context-dependent and sensitivity analyses are an essential companion for deployment.

Availability and reproducibility. To support verification and follow-on work, we released all artifacts used in this dissertation:

- LILITH code and experiments: https://doi.org/10.5281/zenodo.11409100.
- Cloud latency dataset (16 months, 34 AWS regions, JSON schema on Zenodo): https://doi.org/10.5281/zenodo.11457019.
- EB-index datasets: https://doi.org/10.5281/zenodo.15221823.
- Repeatability dataset and artifacts: https://doi.org/10.5281/zenodo.17681717.

These releases include topology files, workload configurations, and step-by-step instructions to reproduce the figures and tables in this thesis.

8.2 Future Work

While the dissertation advanced the state of the art in controlled benchmarking and offered new economic instrumentation, several avenues remain open.

Scale, heterogeneity, and duration. Our experiments targeted up to 40 nodes on homogeneous clusters. Scaling to hundreds of nodes across multiple clusters and heterogeneous machines is a natural extension. This will illuminate cross-hardware effects (CPU micro-architectures, storage, network interface cards) and emergent behaviors under higher fan-out and deeper pipelines. Long-lived campaigns (days to weeks) should track version drift, state growth, and schema upgrades, connecting software evolution to stability regressions and shifts in energy/performance envelopes.

Richer network dynamics and adversarial settings. LILITH already supports dynamic network changes. Future studies should incorporate real fault traces, time-varying latency/jitter/loss, node churn, and adversarial scenarios (*e.g.,* network partitions,

equivocation, censorship at edges). Such campaigns would quantify robustness and recovery costs across topologies, including the interplay between overlay selection and Byzantine-tolerant mechanisms.

Broader protocol and workload coverage. Adding platforms such as Avalanche and expanding workload families (stateful contracts with storage-heavy patterns, MEV-like bursts, DeFi-specific order flows, L2 rollup traffic) will stress additional parts of the stack. Synthetic-to-trace pipelines that replay real incidents (fee spikes, congestion waves) can bridge the gap between lab and field.

Energy measurement granularity and modeling. Machine-level energy was a pragmatic choice for breadth and comparability. Future work should pursue container- or process-level energy attribution (where feasible) and couple power telemetry with CPU, memory, and I/O profiling to explain phase behavior. Reporting energy consumption in kWh remains appropriate for portability, but mapping to regional electricity mixes would enable lifecycle carbon assessments. With larger clusters, statistical reporting (variance, confidence intervals) can be shown alongside medians without overwhelming readability (*e.g.,* via violin plots or uncertainty bands).

Continuous, topology-aware benchmarking. Instituting continuous integration for performance/energy – with fixed topologies and signed workload manifests – would surface stability regressions earlier. Standardized reporting templates, metadata schemas (software versions, commit hashes, topology configurations), and badgeable checklists (open artifacts, seeds, manifests) can help normalize expectations across studies and facilitate fairer comparisons.

Toward benchmark standardization. Lilith is not proposed in this dissertation as an officially certified or standardized benchmark. However, its design deliberately follows principles that would make such a process possible: explicit workload profiles, reproducible deployment artifacts, fixed software versions, declared network conditions, documented metrics, and public datasets. A natural future direction is therefore to align Lilith's benchmark profiles and reporting schema with ongoing performance-assessment and standardization efforts for distributed ledger technologies. Relevant directions include DLT-specific performance-assessment recommendations, such as ITU-T F.751.6 and ITU-T F.751.11, blockchain/DLT standardization activities such as ISO/TC 307, and general benchmark-governance models inspired by organizations such as SPEC, TPC, or STAC. This would upgrade Lilith from a research benchmarking framework toward a community-validated benchmark suite with standardized workloads, measurement procedures, and reporting requirements.

Advancing the EB-index. Three directions stand out:

1. Weighted formulations: adopt Beliş-Guiaşu weighted entropy with domain-informed or data-driven weights (*e.g.,* via expert elicitation, feature importance, or causal relevance) while preserving axiomatic guarantees.

2. Causal inference: move beyond correlation by modeling how exogenous shocks (policy, listings, outages) propagate through on-chain parameters and testing hypotheses about stability and recovery.

3. Sensitivity and scope: conduct formal sensitivity analyses of metric sets and windows and refine proxies that conflate economic and non-economic activity (*e.g.,* separate actor classes, use transfer-typed disaggregation). Furthermore, extending coverage to more assets, segments, and market regimes would improve external validity.

Overlay–underlay co-design. A natural extension of this work is to model the interaction between dynamic blockchain overlays and the physical underlay more explicitly. In the present dissertation, the overlay is fixed within each experiment so that its impact can be isolated and compared across systems. In real deployments, however, peer selection, relay usage, validator churn, and routing changes may continuously reshape the effective communication graph. Future versions of Lilith could therefore combine measured physical underlay traces with dynamic overlay-rewiring policies, making it possible to study when a scale-free-like underlay dominates the observed behavior and when application-level overlay design can mitigate, amplify, or bypass those physical constraints.

From measurement to guidance. Finally, combining the four axes – performance, energy, experimental repeatability/performance predictability, and economic balance – into decision aids for operators and designers is an actionable path to: recommend topology-aware deployments; forecast energy costs under expected traffic; flag instability envelopes after software upgrades; and track EB-index trajectories as health indicators. Such tools would close the loop from measurement to operations, supporting both public and permissioned ecosystems.

Bibliography

[1] AAVE Dev Team. AAVE protocol overview, 2020. URL: https://aave.com/docs.

[2] K. Abbasi, A. Alam, N. A. Brohi, I. A. Brohi, and S. Nasim. P2P lending Fintechs and SMEs' access to finance. *Economics Letters*, 204:1–3, 2021. doi:10.1016/j.econlet.2021.109890.

[3] E. Aben. Latency into your network – as seen from RIPE Atlas, 2021. URL: https://labs.ripe.net/author/emileaben/latency-into-your-network-as-seen-from-ripe-atlas/.

[4] ACM. Artifact review and badging – current, 2025. URL: https://www.acm.org/publications/policies/artifact-review-and-badging-current.

[5] B. Acun, P. Miller, and L. V. Kalé. Variation among processors under turbo boost in HPC systems. In *Proc. of the 30th Int. Conf. on Supercomputing (ICS 2016)*, pages 6:1–6:12. ACM Press, 2016. doi:10.1145/2925426.2926289.

[6] J. Aczél and Z. Daróczy. *On measures of information and their characterizations*, volume 115 of *Mathematics in Science and Engineering*. Academic Press, 1975. URL: https://books.google.com/books?id=wd0TMQAACAAJ.

[7] H. Adams, M. Salem, N. Zinsmeister, S. Reynolds, A. Adams, M. Toda, A. Henshaw, E. Williams, and D. Robinson. Uniswap v4 Whitepaper, 2024. URL: https://github.com/Uniswap/v4-core/blob/main/docs/whitepaper/whitepaper-v4.pdf.

[8] I. F. Akyildiz, X. Wang, and W. Wang. Wireless mesh networks: A survey. *Computer Networks*, 47(4):445–487, 2005. doi:10.1016/j.comnet.2004.12.001.

[9] M. Al-Fares, A. Loukissas, and A. Vahdat. A scalable, commodity data center network architecture. *ACM SIGCOMM Computer Communication Review*, 38(4):63–74, 2008. doi:10.1145/1402946.1402967.

[10] Algorand. Node configuration settings, 2024. URL: https://developer.algorand.org/docs/run-a-node/reference/config/.

[11] Algorand. PyTeal Documentation, 2024. URL: https://pyteal.readthedocs.io/en/latest/.

[12] Algorand. Why Algorand, 2024. URL: https://developer.algorand.org/docs/get-started/basics/why_algorand/.

[13] M. Alharby and A. van Moorsel. BlockSim: An extensible simulation tool for blockchain systems. *Frontiers in Blockchain*, 3:28:1–28:16, 2020. doi:10.3389/fbloc.2020.00028.

[14] V. Allombert, M. Bourgoin, and J. Tesson. Introduction to the Tezos blockchain. In *Proc. of the 17th Int. Conf. on High Performance Computing & Simulation (HPCS 2019)*, pages 1–10. IEEE Computer Society, 2019. doi:10.1109/HPCS48598.2019.9188227.

[15] L. Alsahan, N. Lasla, and M. Abdallah. Local Bitcoin Network Simulator for performance evaluation using lightweight virtualization. In *Proc. of the 3rd IEEE Int. Conf. on Informatics, IoT, and Enabling Technologies (ICIoT 2020)*, pages 355–360. IEEE Computer Society, 2020. doi:10.1109/ICIoT48696.2020.9089630.

[16] F. Alvarez, D. Argente, and D. van Patten. Are cryptocurrencies currencies? Bitcoin as legal tender in El Salvador. *Science*, 382(6677):1–50, 2023. doi:10.1126/science.add2844.

[17] S. Amaro, M. Matos, and V. Schiavoni. Kollaps: Decentralized and efficient network emulation for large-scale systems. *IEEE/ACM Trans. on Networking*, 33(1):35–50, 2025. doi:10.1109/TNET.2024.3478050.

[18] Amazon Web Services. AWS global infrastructure – regions and availability zones, 2025. URL: https://docs.aws.amazon.com/global-infrastructure/latest/regions/aws-availability-zones.html.

[19] M. Ambrosia, J. Dorrell, and T. Stockwell. Is active Bitcoin supply decreasing? An empirical analysis. *Journal of Economics and Finance*, 48(4):1166–1186, 2024. doi:10.1007/s12197-024-09691-w.

[20] Z. Amsden, R. Arora, S. Bano, M. Baudet, S. Blackshear, and A. Bothra. The Libra Blockchain, 2019. URL: https://mitsloan.mit.edu/shared/ods/documents?PublicationDocumentID=5859.

[21] A. G. Anagnostakis and E. Glavas. Entropy and stability in blockchain consensus dynamics. *Information*, 16(2):1–18, 2025. doi:10.3390/info16020138.

[22] E. Anceaume, A. Del Pozzo, T. Rieutord, and S. Tucci-Piergiovanni. On finality in blockchains. In *Proc. of the 25th Int. Conf. on Principles of Distributed Systems (OPODIS 2021)*, volume 217 of *Leibniz Int. Proc. in Informatics*, pages 1–19. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022. doi:10.4230/LIPICS.OPODIS.2021.6.

[23] E. Androulaki, A. Barger, V. Bortnikov, C. Cachin, K. Christidis, A. De Caro, D. Enyeart, C. Ferris, G. Laventman, Y. Manevich, S. Muralidharan, C. Murthy, B. Nguyen, M. Sethi, G. Singh, K. Smith, A. Sorniotti, C. Stathakopoulou, M. Vukolić, S. W. Cocco, and J. Yellick. Hyperledger Fabric: A distributed operating system for permissioned blockchains. In *Proc. of the 13th EuroSys Conf. (EuroSys 2018)*, pages 1–15. ACM Press, 2018. doi:10.1145/3190508.3190538.

[24] G. Angeris and T. Chitra. Improved price oracles: Constant function market makers. Technical report, 2020. arXiv:2003.10001.

[25] 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*, 15(6):1–37, 2022. doi:10.3390/a15060193.

[26] Y. Aoki, K. Otsuki, T. Kaneko, R. Banno, and K. Shudo. SimBlock: A blockchain network simulator. In *Proc. of the 38th IEEE INFOCOM Workshops (INFOCOM-WKSHPS 2019)*, pages 325–329. IEEE Computer Society, 2019. doi:10.1109/INFCOMW.2019.8845253.

[27] M. Arlitt and T. Jin. A workload characterization study of the 1998 World Cup Web site. *IEEE Network*, 14(3):30–37, 2000. doi:10.1109/65.844498.

[28] F. Attneave. *Applications of information theory to psychology: A summary of basic concepts, methods, and results*. Holt, 1959. URL: https://books.google.it/books?id=VnB9AAAAMAAJ.

[29] R. Auer, B. Haslhofer, S. Kitzler, P. Saggese, and F. Victor. The technology of Decentralized Finance (DeFi). BIS Working Papers 1066, Bank for International Settlements, 2023. URL: https://ideas.repec.org/p/bis/biswps/1066.html.

[30] H. C. Bai, L. W. Cong, M. Luo, and P. Xie. Adoption of central bank digital currencies: Initial evidence from China. *Journal of Corporate Finance*, 91:1–29, 2025. doi:10.1016/j.jcorpfin.2025.102735.

[31] T. Bakhtiar, X. Luo, and I. Adelopo. Network effects and store-of-value features in the cryptocurrency market. *Technology in Society*, 74:1–12, 2023. doi:10.1016/j.techsoc.2023.102320.

[32] A. Baliga, I. Subhod, P. Kamat, and S. Chatterjee. Performance evaluation of the Quorum blockchain platform. Technical report, 2018. arXiv:1809.03421.

[33] Bank for International Settlements. The next-generation monetary and financial system. In *Annual Economic Report 2025*. 2025. URL: https://www.bis.org/publ/arpdf/ar2025e3.htm.

[34] A.-L. Barabási. The network takeover. *Nature Physics*, 8(1):14–16, 2012. doi:10.1038/nphys2188.

[35] A.-L. Barabási and R. Albert. Emergence of scaling in random networks. *Science*, 286(5439):509–512, 1999. doi:10.1126/science.286.5439.509.

[36] M. Bardoscia, G. Livan, and M. Marsili. Statistical mechanics of complex economies. *Journal of Statistical Mechanics: Theory and Experiment*, 2017(4):1–30, 2017. doi:10.1088/1742-5468/aa6688.

[37] S. R. Basnet and S. Shakya. BSS: Blockchain security over software defined network. In *Proc. of the 3rd Int. Conf. on Computing, Communication and Automation (ICCCA 2017)*, pages 720–725. IEEE Computer Society, 2017. doi:10.1109/CCAA.2017.8229910.

[38] 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. Technical report, The Libra Association, 2019. URL: https://developers.diem.com/papers/diem-consensus-state-machine-replication-in-the-diem-blockchain/2019-06-28.pdf.

[39] M. Beliş and S. Guiaşu. A quantitative-qualitative measure of information in cybernetic systems. *IEEE Trans. on Information Theory*, 14(4):593–594, 1968. doi:10.1109/TIT.1968.1054185.

[40] N. Bhintade. Understanding wealth distribution with Gini and Nakamoto coefficients, 2023. URL: https://bitquery.io/blog/wealth-distribution-in-token-economy.

[41] G. Bissias, B. N. Levine, N. Narula, and G. Andresen. Graphene: Efficient set reconciliation for block propagation. In *Proc. of the 33rd ACM Conf. on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM 2019)*, pages 303–317. ACM Press, 2019. doi:10.1145/3341302.3342082.

[42] Bitcoin Core Developers. Block-relay-only connections, 2019. URL: https://github.com/bitcoin/bitcoin/blob/master/doc/release-notes/release-notes-0.19.0.1.md.

[43] Bitcoin Core Developers. Bitcoin developer guide: P2P network, 2024. URL: https://github.com/bitcoin-dot-org/developer.bitcoin.org/blob/master/devguide/p2p_network.rst.

[44] Bitcoin Core Developers. Reduce traffic, 2024. URL: https://github.com/bitcoin/bitcoin/blob/master/doc/reduce-traffic.md.

[45] Bitcoin Wiki contributors. Protocol rules, 2020. URL: https://en.bitcoin.it/wiki/Protocol_rules.

[46] BitcoinBlockHalf.com. Bitcoin block reward halving countdown, 2025. URL: https://www.bitcoinblockhalf.com/.

[47] Bitcoin.org Project. Bitcoin developer guide: P2P network, 2020. URL: https://developer.bitcoin.org/reference/p2p_networking.html.

[48] Bitfly explorer. Clients - ethernodes.org - The Ethereum network and Node explorer, 2025. URL: https://www.ethernodes.org/.

[49] Bitnodes, 2025. URL: https://bitnodes.io/.

[50] Bitnodes. Global Bitcoin nodes by country, 2025. URL: https://bitnodes.io/nodes/all/countries/1d/.

[51] Bitquery. Blockchain API and crypto data products, 2025. URL: https://bitquery.io/.

[52] Block Logic, LLC. Solana mainnet data centers, 2025. URL: https://www.validators.app/data-centers?locale=en&network=mainnet.

[53] Blockchair. Blockchair, 2024. URL: https://blockchair.com/it.

[54] Blockdaemon. Empowering blockchain infrastructure with wallet, node management & staking solutions, 2024. URL: https://www.blockdaemon.com/.

[55] R. F. Boisvert. Incentivizing reproducibility. *Communications of the ACM*, 59(10):5–5, 2016. doi:10.1145/2994031.

[56] A. Bondavalli, A. Ceccarelli, L. Falai, and M. Vadursi. Foundations of measurement theory applied to the evaluation of dependability attributes. In *Proc. of the 37th IEEE/IFIP Int. Conf. on Dependable Systems and Networks (DSN 2007)*, pages 522–533. IEEE Computer Society, 2007. doi:10.1109/DSN.2007.52.

[57] S. T. Borda and J. Ermont. An evaluation of software-based TSN traffic shapers using Linux tc. In *Proc. of the 18th IEEE Int. Conf. on Factory Communication Systems (WFCS 2022)*, pages 1–4. IEEE Computer Society, 2022. doi:10.1109/WFCS53837.2022.9779163.

[58] Brave Software. Basic Attention Token whitepaper, 2017. URL: https://basicattentiontoken.org/static-assets/documents/BasicAttentionTokenWhitePaper-4.pdf.

[59] L. Breidenbach, C. Cachin, B. Chan, A. Coventry, S. Ellis, A. Juels, F. Koushanfar, A. Miller, B. Magauran, and D. Moroz. Chainlink 2.0: Next steps in the evolution of decentralized oracle networks, 2021. URL: https://research.chain.link/whitepaper-v2.pdf.

[60] L. Breslau, P. Cao, L. Fan, G. Phillips, and S. Shenker. Web caching and Zipf-like distributions: Evidence and implications. In *Proc. of the 18th IEEE Conf. on Computer and Communications Societies (INFOCOM 1999)*, pages 126–134. IEEE Computer Society, 1999. doi:10.1109/INFCOM.1999.749260.

[61] K. J. Brooks. Bitcoin hits record high. Here's what's driving up the price, 2024. URL: https://www.cbsnews.com/news/bitcoin-price-stock-cryptocurrency-etf-approval/.

[62] J. Brown-Cohen, A. Narayanan, A. Psomas, and S. M. Weinberg. Formal barriers to longest-chain Proof-of-Stake protocols. In *Proc. of the 20th ACM Conf. on Economics and Computation (EC 2019)*, pages 459–473. ACM Press, 2019. doi:10.1145/3328526.3329567.

[63] M. K. Brunnermeier. Deciphering the liquidity and credit crunch 2007-2008. *Journal of Economic Perspectives*, 23(1):77–100, 2009. doi:10.1257/jep.23.1.77.

[64] E. Buchman, J. Kwon, and Z. Milosevic. The latest gossip on BFT consensus (tendermint). Technical report, 2018. arXiv:1807.04938.

[65] J. Burdges, A. Cevallos, P. Czaban, R. Habermeier, S. Hosseini, F. Lama, H. K. Alper, X. Luo, F. Shirazi, A. Stewart, and G. Wood. Overview of Polkadot and its design considerations. Technical report, 2020. arXiv:2005.13456.

[66] V. Buterin, E. Conner, R. Dudley, M. Slipper, I. Norden, and A. Bakhta. EIP-1559: Fee market change for ETH 1.0 chain, 2019. URL: https://eips.ethereum.org/EIPS/eip-1559.

[67] Cambridge Center for Alternative Finance. Cambridge Blockchain Network Sustainability Index (CBNSI). URL: https://ccaf.io/cbnsi/cbeci.

[68] Cambridge Center for Alternative Finance. Cambridge Bitcoin Electricity Consumption Index (CBECI), 2024. URL: https://ccaf.io/cbnsi/cbeci/methodology.

[69] S. Cao, T. Miller, M. Foth, W. Powell, X. Boyen, and C. Turner-Morris. Integrating on-chain and off-chain governance for supply chain transparency and integrity. Technical report, 2021. arXiv:2111.08455.

[70] A. Casagrande, F. Fabris, and R. Girometti. Beyond kappa: An informational index for diagnostic agreement in dichotomous and multivalue ordered-categorical ratings. *Medical and Biological Engineering and Computing*, 58:3089–3099, 2020. doi:10.1007/s11517-020-02261-2.

[71] M. Castro, P. Druschel, A. Ganesh, A. Rowstron, and D. S. Wallach. Secure routing for structured peer-to-peer overlay networks. In *Proc. of the 5th Symp. on Operating Systems Design and Implementation (OSDI 2002)*, pages 299–314. USENIX Association, 2002. URL: https://www.usenix.org/conference/osdi-02/secure-routing-structured-peer-peer-overlay-networks.

[72] M. Castro and B. Liskov. Practical Byzantine fault tolerance. In *Proc. of the 3rd Symp. on Operating Systems Design and Implementation (OSDI 1999)*, pages 173–186. USENIX Association, 1999. URL: https://dl.acm.org/citation.cfm?id=296824.

[73] P. Chanda, E. Costa, J. Hu, S. Sukumar, J. Van Hemert, and R. Walia. Information theory in computational biology: Where we stand today. *Entropy*, 22(6):1–34, 2020. doi:10.3390/e22060627.

[74] B. Chase and E. MacBrough. Analysis of the XRP ledger consensus protocol. Technical report, 2018. arXiv:1802.07242.

[75] F. Chen, Z. Chen, X. Wang, and Z. Yuan. The average path length of scale-free networks. *Communications in Nonlinear Science and Numerical Simulation*, 13(7):1405–1410, 2008. doi:10.1016/j.cnsns.2006.12.003.

[76] J. Chen and S. Micali. Algorand: A secure and efficient distributed ledger. *Theoretical Computer Science*, 777:155–183, 2019. doi:10.1016/j.tcs.2019.02.001.

[77] Q. Chen and D. Shi. The modeling of scale-free networks. *Physica A: Statistical Mechanics and its Applications*, 335(1):240–248, 2004. doi:10.1016/j.physa.2003.12.014.

[78] H. S. Chena, J. T. Jarrell, K. A. Carpenter, D. S. Cohen, and X. Huang. Blockchain in healthcare: A patient-centered model. *Biomedical Journal of Scientific & Technical Research*, 20(3):15017–15022, 2019. doi:10.26717/BJSTR.2019.20.003448.

[79] R. Chern. Turbine: Block propagation on Solana, 2024. URL: https://www.helius.dev/blog/turbine-block-propagation-on-solana.

[80] 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 Trans. on Parallel and Distributed Systems*, 33(6):1464–1477, 2022. doi:10.1109/TPDS.2021.3111159.

[81] F. R. K. Chung. *Spectral Graph Theory*, volume 92 of *CBMS Regional Conf. Series in Mathematics*. American Mathematical Society, 1997. URL: https://bookstore.ams.org/cbms-92.

[82] Circle Internet Group. The home of USDC, by the issuer of USDC, 2025. URL: https://www.usdc.com/.

[83] C. Clos. A study of non-blocking switching networks. *Bell System Technical Journal*, 32(2):406–424, 1953. doi:10.1002/j.1538-7305.1953.tb01433.x.

[84] CoinCu News. Solana TPS: How many TPS can Solana handle?, 2024. URL: https://coincu.com/knowledge/solana-tps-how-many-tps-can-solana-handle/.

[85] CoinGecko. CoinGecko: Cryptocurrency prices, charts, and market capitalizations, 2025. URL: https://www.coingecko.com.

[86] CoinMetrics. Network data pro overview, 2025. URL: https://docs.coinmetrics.io/network-data/network-data-overview.

[87] R. Cole and L. Cheng. Modeling the energy consumption of blockchain consensus algorithms. In *Proc. of the IEEE Int. Conf. on Internet of Things (iThings 2018) and IEEE Green Computing and Communications (GreenCom 2018) and IEEE Cyber, Physical and Social Computing (CPSCom 2018) and IEEE Smart Data (SmartData 2018)*, pages 1691–1696. IEEE Computer Society, 2018. doi:10.1109/Cybermatics_2018.2018.00282.

[88] Committee on Payments and Market Infrastructures. Distributed ledger technology in payment, clearing and settlement: An analytical framework. Technical report, Bank for International Settlements, 2017. URL: https://www.bis.org/cpmi/publ/d157.pdf.

[89] Committee on Payments and Market Infrastructures. Considerations for the use of stablecoin arrangements in cross-border payments. Technical report, Bank for International Settlements, 2023. URL: https://www.bis.org/cpmi/publ/d220.htm.

[90] CompaniesMarketCap. Assets ranked by market cap, 2025. URL: https://companiesmarketcap.com/assets-by-market-cap/.

[91] M. Conti, A. Gangwal, and M. Todero. Blockchain trilemma solver Algorand has dilemma over undecidable messages. In *Proc. of the 14th Int. Conf. on Availability, Reliability and Security (ARES 2019)*, pages 1–8. ACM Press, 2019. doi:10.1145/3339252.3339255.

[92] B. F. Cooper, A. Silberstein, E. Tam, R. Ramakrishnan, and R. Sears. Benchmarking cloud serving systems with YCSB. In *Proc. of the 1st ACM Symp. on Cloud Computing (SoCC 2010)*, pages 143–154. ACM Press, 2010. doi:10.1145/1807128.1807152.

[93] M. Corallo. BIP 152: Compact Block Relay, 2016. URL: https://github.com/bitcoin/bips/blob/master/bip-0152.mediawiki.

[94] M. Corallo. FIBRE: Fast Internet Bitcoin Relay Engine, 2016. URL: https://bitcoinfibre.org/.

[95] Cosmos Developers. Cosmos: A network of distributed ledgers, 2020. URL: https://github.com/cosmos/cosmos/blob/master/WHITEPAPER.md.

[96] T. M. Cover and J. A. Thomas. *Elements of information theory*. John Wiley & Sons, 1999. doi:10.1002/047174882X.

[97] K. Croman, C. Decker, I. Eyal, A. E. Gencer, A. Juels, A. Kosba, A. Miller, P. Saxena, E. Shi, E. G. Sirer, D. Song, and R. Wattenhofer. On scaling decentralized blockchains. In *Proc. of the 20th Int. Conf. on Financial Cryptography and Data Security (FC 2016)*, volume 9604 of *Lecture Notes in Computer Science*, pages 106–125. Springer, 2016. doi:10.1007/978-3-662-53357-4_8.

[98] Cryptoexchange.com. What is quorum blockchain?, 2024. URL: https://cryptoexchange.com/learning/what-is-quorum.

[99] CryptoQuant. Ethereum: Total Value Staked, 2025. URL: https://cryptoquant.com/asset/eth/chart/eth2/total-value-staked?window=DAY&sma=0&ema=0&priceScale=log&metricScale=linear&chartStyle=line.

[100] I. Csiszár. Axiomatic characterizations of information measures. *Entropy*, 10(3):261–273, 2008. doi:10.3390/e10030261.

[101] W. J. Dally and B. Towles. *Principles and Practices of Interconnection Networks*. Morgan Kaufmann, 2004. URL: https://shop.elsevier.com/books/principles-and-practices-of-interconnection-networks/dally/978-0-12-200751-4.

[102] D. Das, P. Bose, N. Ruaro, C. Kruegel, and G. Vigna. Understanding security issues in the NFT ecosystem. In *Proc. of the 29th ACM SIGSAC Conf. on Computer and Communications Security (CCS 2022)*, pages 667–681. ACM Press, 2022. doi:10.1145/3548606.3559342.

[103] F. M. De Collibus, C. Campajola, and C. J. Tessone. The microvelocity of money in Ethereum. *EPJ Data Science*, 14:11, 2025. doi:10.1140/epjds/s13688-024-00518-6.

[104] C. Decker and R. Wattenhofer. Information propagation in the Bitcoin network. In *Proc. of the 13th IEEE Int. Conf. on Peer-to-Peer Computing (P2P 2013)*, pages 1–10. IEEE Computer Society, 2013. doi:10.1109/P2P.2013.6688704.

[105] DeFiLlama. All chains – TVL ranking, 2025. URL: https://defillama.com/chains.

[106] A. J. Delic and P. H. Delfabbro. Profiling the potential risks and benefits of emerging "Play to Earn" games: A qualitative analysis of players' experiences with Axie Infinity. *Int. Journal of Mental Health and Addiction*, 22(1):634–647, 2024. doi:10.1007/s11469-022-00894-y.

[107] A. Demers, D. Greene, C. Hauser, W. Irish, J. Larson, S. Shenker, H. Sturgis, D. Swinehart, and D. Terry. Epidemic algorithms for replicated database maintenance. In *Proc. of the 6th ACM Symp. on Principles of Distributed Computing (PODC 1987)*, pages 1–12. ACM Press, 1987. doi:10.1145/41840.41841.

[108] V. P. Di Perna, M. Bernardo, F. Fabris, S. Amaro, M. Matos, and V. Schiavoni. Impact of network topologies on blockchain performance. In *Proc. of the 19th ACM Int. Conf. on Distributed and Event-Based Systems (DEBS 2025)*, pages 122–133. ACM Press, 2025. doi:10.1145/3701717.3730540. URL: https://zenodo.org/records/11409100.

[109] V. P. Di Perna, M. Foderaro, F. Fabris, and M. Bernardo. An entropy-based approach to evaluating the economic efficiency of cryptocurrencies. In *Proc. of the 7th Distributed Ledger Technology Workshop (DLT 2025)*, volume 4105 of *CEUR Workshop Proceedings*, pages 9:1–9:16. CEUR-WS.org, 2025. URL: https://ceur-ws.org/Vol-4105/paper09.pdf.

[110] V. P. Di Perna, V. Schiavoni, F. Fabris, and M. Bernardo. Blockchain energy consumption: Unveiling the impact of network topologies. In *Proc. of the 7th IEEE Int. Conf. on Blockchain and Cryptocurrency (ICBC 2025)*, pages 67–76. IEEE Computer Society, 2025. doi:10.1109/ICBC64466.2025.11114569.

[111] V. P. Di Perna, V. Schiavoni, F. Fabris, M. Bernardo, and M. Matos. Repeatability and performance predictability in network-controlled blockchain benchmarking. In *submitted for publication*, 2026. URL : https://doi.org/10.5281/zenodo.17681717.

[112] A. G. Dimitrov, A. A. Lazar, and J. D. Victor. Information theory in neuroscience. *Journal of Computational Neuroscience*, 30(1):1–5, 2011. doi:10.1007/s10827-011-0314-3.

[113] T. T. A. Dinh, J. Wang, G. Chen, R. Liu, B. C. Ooi, and K. L. Tan. BLOCKBENCH: A framework for analyzing private blockchains. In *Proc. of the 43rd ACM Int. Conf. on Management of Data (SIGMOD 2017)*, pages 1085–1100. ACM Press, 2017. doi:10.1145/3035918.3064033.

[114] C. Dwork, N. Lynch, and L. Stockmeyer. Consensus in the presence of partial synchrony. *Journal of the ACM*, 35(2):288–323, 1988. doi:10.1145/42282.42283.

[115] Economic Commission for Latin America and the Caribbean (ECLAC). Methods of measuring the economy, efficiency and effectiveness of public expenditure, 2015. URL: https://www.cepal.org/sites/default/files/project/files/annex_7_methods_of_measuring_economy_efficiency_and_effectivenes.pdf.

[116] P. Erdős and A. Rényi. On the evolution of random graphs. *Publications of the Mathematical Institute of the Hungarian Academy of Sciences*, 5:17–61, 1960. URL: https://snap.stanford.edu/class/cs224w-readings/erdos60random.pdf.

[117] T. Espel, L. Katz, and G. Robin. Proposal for protocol on a Quorum blockchain with zero knowledge. *Cryptology ePrint Archive*, 2017:1–22, 2017. URL: https://eprint.iacr.org/2017/1093.

[118] Ethereum. Devp2p/rlpx.md at master - ethereum/devp2p, 2014. URL: https://github.com/ethereum/devp2p/blob/master/rlpx.md.

[119] Ethereum. Devp2p/caps/eth.md at master - ethereum/devp2p, 2024. URL: https://github.com/ethereum/devp2p/blob/master/caps/eth.md.

[120] Ethereum Community. Ethereum network stats (ethstats), 2016. URL: https://github.com/cubedro/eth-netstats.

[121] Ethereum Foundation. Ethereum consensus layer: Phase 0 P2P interface, 2024. URL: https://ethereum.github.io/consensus-specs/specs/phase0/p2p-interface/.

[122] Ethereum Foundation. Ethereum devp2p discovery protocol (Kademlia-like), 2024. URL: https://github.com/ethereum/devp2p.

[123] Ethereum Foundation. Ethereum/hive: Ethereum end-to-end test harness, 2024. URL: https://github.com/ethereum/hive.

[124] Etherscan. Ethereum node tracker, 2025. URL: https://etherscan.io/nodetracker.

[125] EthOS Dev. The Beacon chain: Ethereum 2.0 explainer, 2022. URL: https://ethos.dev/beacon-chain.

[126] European Central Bank. Monetary aggregates, 2024. URL: https://www.ecb.europa.eu/stats/money_credit_banking/monetary_aggregates/html/index.en.html.

[127] European Investment Bank. EIB issues its first ever digital bond on a public blockchain, 2021. URL: https://www.eib.org/en/press/all/2021-141-european-investment-bank-eib-issues-its-first-ever-digital-bond-on-a-public-blockchain.

[128] I. Eyal and E. G. Sirer. Majority is not enough: Bitcoin mining is vulnerable. *Communications of the ACM*, 61(7):95–102, 2018. doi:10.1145/3212998.

[129] F. Fabris. Shannon information theory and molecular biology. *Journal of Interdisciplinary Mathematics*, 12(1):41–87, 2009. doi:10.1080/09720502.2009.10700611.

[130] C. Faria and M. Correia. BlockSim: Blockchain Simulator. In *Proc. of the 2nd IEEE Int. Conf. on Blockchain (Blockchain 2019)*, pages 439–446. IEEE Computer Society, 2019. doi:10.1109/Blockchain.2019.00067.

[131] S. M. Fattahi, A. Makanju, and A. Milani Fard. SIMBA: An efficient simulator for blockchain applications. In *Proc. of the 50th IEEE/IFIP Int. Conf. on Dependable Systems and Networks (DSN-S 2020)*, pages 51–52. IEEE Computer Society, 2020. doi:10.1109/DSN-S50200.2020.00028.

[132] M. Feixas, A. Bardera, J. Rigau, Q. Xu, and M. Sbert. *Information theory tools for image processing*. Springer, 2014. doi:10.2200/S00560ED1V01Y201312CGR015.

[133] W. Felter, A. Ferreira, R. Rajamony, and J. Rubio. An updated performance comparison of virtual machines and Linux containers. In *Proc. of the 15th IEEE Int. Symposium on Performance Analysis of Systems and Software (ISPASS 15)*, pages 171–172. IEEE, 2015. doi:10.1109/ISPASS.2015.7095802.

[134] Fetch.AI. Fetch.AI: Autonomous economic agents on a decentralized network, 2021. URL: https://fetch.ai/.

[135] L. D. Fink. Larry Fink's 2025 annual chairman's letter to investors, 2025. URL: https://www.blackrock.com/corporate/literature/presentation/larry-fink-annual-chairmans-letter.pdf.

[136] L. C. Freeman. A set of measures of centrality based on betweenness. *Sociometry*, 40(1):35–41, 1977. doi:10.2307/3033543.

[137] G. Fridgen, J. Sedlmeir, P. Ross, A. Luckow, J. Lockl, and D. Miehle. The DLPS: A new framework for benchmarking blockchains. In *Proc. of the 54th Hawaii Int. Conf. on System Sciences (HICSS 2021)*, pages 6855–6864, 2021. doi:10.24251/HICSS.2021.822.

[138] Y. Fu, M. Jing, J. Zhou, P. Wu, Y. Wang, L. Zhang, and C. Hu. Quantifying the blockchain trilemma: A comparative analysis of Algorand, Ethereum 2.0, and beyond. In *Proc. of the 2nd IEEE Int. Conf. on Metaverse Computing, Networking, and Applications (MetaCom 2024)*, pages 97–104. IEEE Computer Society, 2024. doi:10.1109/MetaCom62920.2024.00028.

[139] N. Gandal, J. Hamrick, T. Moore, and M. Vasek. The rise and fall of cryptocurrency coins and tokens. *Decisions in Economics and Finance*, 44:981–1014, 2021. doi:10.1007/s10203-021-00329-8.

[140] Y. Gao, J. Shi, X. Wang, Q. Tan, C. Zhao, and Z. Yin. Topology measurement and analysis on Ethereum P2P network. In *Proc. of the 24th IEEE Symp. on Computers and Communications (ISCC 2019)*, pages 1–7. IEEE Computer Society, 2019. doi:10.1109/ISCC47284.2019.8969695.

[141] A. E. Gencer, S. Basu, I. Eyal, R. van Renesse, and E. G. Sirer. Decentralization in Bitcoin and Ethereum networks. In *Proc. of the 22nd Int. Conf. on Financial Cryptography and Data Security (FC 2018)*, volume 10957 of *Lecture Notes in Computer Science*, pages 439–457. Springer, 2018. doi:10.1007/978-3-662-58387-6_24.

[142] G. Gensler. Statement on the approval of spot Bitcoin exchange-traded products, 2024. URL: https://www.sec.gov/newsroom/speeches-statements/gensler-statement-spot-bitcoin-011023.

[143] D. Georgarakos, G. Kenny, L. Laeven, and J. Meyer. Consumer attitudes towards a central bank digital currency. *SSRN*, 5176496:1–83, 2025. doi:10.2139/ssrn.5176496.

[144] E. Georgiadis. How many transactions per second can Bitcoin really handle? Theoretically. *Cryptology ePrint Arch.*, pages 1–416, 2019. URL: https://eprint.iacr.org/2019/416.

[145] 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 *Proc. of the 23rd ACM SIGSAC Conf. on Computer and Communications Security (CCS 2016)*, pages 3–16. ACM Press, 2016. doi:10.1145/2976749.2978341.

[146] F. C. Geyer, H.-A. Jacobsen, R. Mayer, and P. Mandl. An end-to-end performance comparison of seven permissioned blockchain systems. In *Proc. of the 24th Int. Middleware Conf. (Middleware 2023)*, pages 71–84. ACM Press, 2023. doi:10.1145/3590140.3629106.

[147] Y. Gilad, R. Hemo, S. Micali, G. Vlachos, and N. Zeldovich. Algorand: Scaling Byzantine agreements for cryptocurrencies. In *Proc. of the 26th Symp. on Operating Systems Principles (SOSP 2017)*, pages 51–68. ACM Press, 2017. doi:10.1145/3132747.3132757.

[148] R. Girometti and F. Fabris. Informational analysis: A Shannon theoretic approach to measure the performance of a diagnostic test. *Medical and Biological Engineering and Computing*, 53:899–910, 2015. doi:10.1007/s11517-015-1294-7.

[149] Glassnode. On-chain market intelligence, 2025. URL: https://glassnode.com/.

[150] S. P. Gochhayat, S. Shetty, R. Mukkamala, P. Foytik, G. A. Kamhoua, and L. Njilla. Measuring decentrality in blockchain based systems. *IEEE Access*, 8:178372–178390, 2020. doi:10.1109/ACCESS.2020.3026577.

[151] O. Goldreich. *Foundations of Cryptography: Volume 2 – Basic Applications*. Cambridge University Press, 2004. URL: https://www.wisdom.weizmann.ac.il/~oded/foc-vol2.html.

[152] Golem Team. Golem overview, 2025. URL: https://docs.golem.network/docs/golem/overview.

[153] G. Gorton. The development of opacity in U.S. banking, 2014. URL: http://hdl.handle.net/20.500.13051/8211.

[154] V. Gramoli, R. Guerraoui, A. Lebedev, C. Natoli, and G. Voron. Diablo: A benchmark suite for blockchains. In *Proc. of the 18th European Conf. on Computer Systems (EuroSys 2023)*, pages 540–556. ACM Press, 2023. doi:10.1145/3552326.3567482.

[155] V. Gramoli, R. Guerraoui, A. Lebedev, C. Natoli, and G. Voron. Diablo Blockchain Benchmark Suite, 2023. URL: https://diablobench.github.io/.

[156] V. Gramoli, R. Guerraoui, A. Lebedev, and G. Voron. Evaluating blockchain fault tolerance with Stabl. In *Proc. of the 55th IEEE/IFIP Int. Conf. on Dependable Systems and Networks (DSN-S 2025)*, pages 182–183. IEEE Computer Society, 2025. doi:10.1109/DSN-S65789.2025.00062.

[157] M. Guennewig. Blockchain congestion facilitates currency competition. Technical report, 2024. URL: https://ideas.repec.org/p/bon/boncrc/crctr224_2024_549.html.

[158] S. Gupta, S. Rahnama, J. Hellings, and M. Sadoghi. ResilientDB: Global scale resilient blockchain fabric. *Proc. of the VLDB Endowment*, 13(6):868–883, 2020. doi:10.14778/3380750.3380757.

[159] E. Haaramo. Sweden trials blockchain for land registry management. *Computer Weekly*, 2017. URL: https://www.computerweekly.com/news/450421958/Sweden-trials-blockchain-for-land-registry-management.

[160] S. Haber and W. S. Stornetta. How to time-stamp a digital document. *Journal of Cryptology*, 3(2):99–111, January 1991. doi:10.1007/BF00196791.

[161] R. Hahn. Compound III documentation, 2020. URL: https://docs.compound.finance/.

[162] N. Handigol, B. Heller, V. Jeyakumar, B. Lantz, and N. McKeown. Reproducible network experiments using container-based emulation. In *Proc. of the 8th Int. Conf. on Emerging Networking Experiments and Technologies (CoNEXT 2012)*, pages 253–264. ACM Press, 2012. doi:10.1145/2413176.2413206.

[163] Handshake Team. Handshake developer documentation, 2025. URL: https://hsd-dev.org/.

[164] V. T. Hayashi, F. V. de Almeida, and A. E. 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*, pages 90–97. SBC, 2021. doi:10.5753/sbseg_estendido.2021.17344.

[165] Z. He, J. Li, and Z. Wu. Don't trust, verify: The case of slashing from a popular Ethereum explorer. In *Companion Proc. of the 32nd ACM Web Conf. 2023 (WWW 2023)*, pages 1078–1084. ACM Press, 2023. doi:10.1145/3543873.3587555.

[166] E. Heilman, A. Kendler, A. Zohar, and S. Goldberg. Eclipse attacks on Bitcoin's peer-to-peer network. In *Proc. of the 24th USENIX Conf. on Security Symp. (SEC 2015)*, pages 129–144. USENIX Association, 2015. URL: https://www.usenix.org/conference/usenixsecurity15/technical-sessions/presentation/heilman.

[167] Helium Foundation. Helium network: A decentralized wireless infrastructure, 2022. URL: https://www.helium.com/.

[168] Hong Kong Monetary Authority. HKSAR government's digital green bonds offering, 2024. URL: https://www.hkma.gov.hk/eng/news-and-media/press-releases/2024/02/20240207-6/.

[169] S. Hoory, N. Linial, and A. Wigderson. Expander graphs and their applications. *Bulletin of the American Mathematical Society*, 43(4):439–561, 2006. doi:10.1090/S0273-0979-06-01126-8.

[170] C. Hoskinson. Cardano introduction, 2025. URL: https://docs.cardano.org/about-cardano/introduction/.

[171] B. G. Hu. Information theory and its relation to machine learning. In *Proc. of the 9th Chinese Intelligent Automation Conf. (CIAC 2015)*, volume 336 of *Lecture Notes in Electrical Engineering*, pages 1–11. Springer, 2015. doi:10.1007/978-3-662-46469-4_1.

[172] H. Huang. An emulator enabling to test blockchain sharding protocols. Technical report, 2023. arXiv:2311.03612.

[173] Hyperledger. Hyperledger Caliper — hyperledger.github.io, 2018. URL: https://hyperledger.github.io/caliper/.

[174] Hyperledger Besu. Create a private network using IBFT 2.0, 2025. URL: https://besu.hyperledger.org/private-networks/tutorials/ibft.

[175] IBM. IBM Food Trust delivers traceability, quality assurance to major olive oil brands with blockchain, 2020. URL: https://newsroom.ibm.com/2020-11-11-IBM-Food-Trust-Delivers-Traceability-Quality-Assurance-to-Major-Olive-Oil-Brands-with-Blockchain.

[176] IEEE. Overview of IEEE Xplore Digital Library content - Reproducibility Badges, 2025. URL: https://ieeexplore.ieee.org/Xplorehelp/overview-of-ieee-xplore/about-content#reproducibility-badges.

[177] N. Inkster. Evolution of the chinese Internet: Freedom and control. *Adelphi Series*, 55(456):19–50, 2015. doi:10.1080/19445571.2015.1181441.

[178] International Organization for Standardization. Blockchain and distributed ledger technologies – vocabulary, 2024. URL: https://www.iso.org/standard/82208.html.

[179] IOTA. IOTA documentation, 2025. URL: https://docs.iota.org/.

[180] F. Irresberger and R. Yang. Coin concentration of Proof-of-Stake blockchains. *Economics Letters*, 229:1–5, 2023. doi:10.1016/j.econlet.2023.111219.

[181] M. M. Islam, M. M. Merlec, and H. P. In. A comparative analysis of Proof-of-Authority consensus algorithms: Aura vs Clique. In *Proc. of the 19th IEEE Int. Conf. on Services Computing (SCC 2022)*, pages 327–332. IEEE Computer Society, 2022. doi:10.1109/SCC55611.2022.00054.

[182] 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. *Int. Journal of Advanced Computer Science and Applications*, 13(4):590–598, 2022. doi:10.14569/IJACSA.2022.0130469.

[183] M. Janczyk and R. Pfister. *Factorial Analysis of Variance (ANOVA)*. Springer, 2023. doi:10.1007/978-3-662-66786-6_9.

[184] E. T. Jaynes. Information theory and statistical mechanics. *Physical review*, 106(4):620–630, 1957. doi:10.1103/PhysRev.106.620.

[185] M. Juodis, E. Filatovas, and R. Paulavičius. Overview and empirical analysis of wealth decentralization in blockchain networks. *ICT Express*, 10(2):380–386, 2024. doi:10.1016/j.icte.2024.02.002.

[186] R. Karanjai, L. Xu, L. Chen, N. Diallo, and W. Shi. Decentralized FaaS over multi-clouds with blockchain based management for supporting emerging applications. In *Proc. of the 39th ACM SIGAPP Symp. on Applied Computing (SAC 2024)*, pages 122–130. ACM Press, 2024. doi:10.1145/3605098.3636029.

[187] J. Karasiński. The weak-form efficiency of cryptocurrencies. *Research Papers in Economics and Finance*, 7(1):31–47, 2023. doi:10.18559/ref.2023.1.198.

[188] R. M. Karp, C. Schindelhauer, S. Shenker, and B. Vöcking. Randomized rumor spreading. In *Proc. of the 41st Symp. on Foundations of Computer Science (FOCS 2000)*, pages 565–574. IEEE Computer Society, 2000. doi:10.1109/SFCS.2000.892324.

[189] C. Kenny. The Equifax data breach and the resulting legal recourse. *Brooklyn Journal of Corporate, Financial & Commercial Law*, 13:215–238, 2018. URL: https://brooklynworks.brooklaw.edu/bjcfcl/vol13/iss1/10/.

[190] 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. on Modelling and Performance Evaluation of Computing Systems*, 3(2):1–26, 2018. doi:10.1145/3177754.

[191] L. Kiffer, A. Salman, D. Levin, A. Mislove, and C. Nita-Rotaru. Under the hood of the Ethereum gossip protocol. In *Proc. of the 25th Int. Conf. on Financial Cryptography and Data Security (FC 2021)*, volume 12675 of *Lecture Notes in Computer Science*, pages 437–456. Springer, 2021. doi:10.1007/978-3-662-64331-0_23.

[192] T. K. Koo and M. Y. Li. A guideline of selecting and reporting intraclass correlation coefficients for reliability research. *Journal of Chiropractic Medicine*, 15(2):155–163, 2016. doi:10.1016/j.jcm.2016.02.012.

[193] R. Kozhan and G. Viswanath-Natraj. Fundamentals of the MakerDAO governance token. In *Proc. of the 3rd Int. Conf. on Blockchain Economics, Security and Protocols (Tokenomics 2021)*, volume 97 of *Open Access Series in Informatics*, pages 1–5. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022. doi:10.4230/OASIcs.Tokenomics.2021.11.

[194] B. Kusmierz and R. Overko. How centralized is decentralized? Comparison of wealth distribution in coins and tokens. In *Proc. of the 3rd IEEE Int. Conf. on Omni-layer Intelligent Systems (COINS 2022)*, pages 1–6. IEEE Computer Society, 2022. doi:10.1109/COINS54846.2022.9854972.

[195] J.-C. Laprie. Dependability: Basic concepts and terminology. In *Dependability: Basic Concepts and Terminology*, volume 5 of *Dependable Computing and Fault-Tolerant Systems*, pages 3–245. Springer, 1992. doi:10.1007/978-3-7091-9170-5_1.

[196] M. R. A. Lathif, P. Nasirifard, and H.-A. Jacobsen. CIDDS: A configurable and distributed DAG-based distributed ledger simulation framework. In *Proc. of the 19th Int. Middleware Conf. (Middleware 2018 – Posters)*, pages 7–8. ACM Press, 2018. doi:10.1145/3284014.3284018.

[197] N. Lazuashvili. *Integration of the blockchain technology into the Land Registration system. A case study of Georgia.* Ph.D. Thesis, 2019. doi:10.13140/RG.2.2.35689.13920/1.

[198] A. Lebedev and V. Gramoli. On the relevance of blockchain evaluations on bare metal. In *Proc. of the 7th Int. Symp. on Distributed Ledger Technology (SDLT 2023)*, volume 1975 of *Communications in Computer and Information Science*, pages 22–38. Springer, 2024. doi:10.1007/978-981-97-0006-6_2.

[199] E. Lee. The Bored Ape business model: Decentralized collaboration via blockchain and NFTs. *SSRN*, 3963881:1–7, 2021. doi:10.2139/ssrn.3963881.

[200] C. E. Leiserson. Fat-trees: Universal networks for hardware-efficient supercomputing. *IEEE Trans. on Computers*, C-34(10):892–901, 1985. doi:10.1109/TC.1985.6312192.

[201] P. Leitner and J. Cito. Patterns in the Chaos – A study of performance variation and predictability in public IaaS clouds. *ACM Trans. on Internet Technology*, 16(3):1–23, 2016. doi:10.1145/2885497.

[202] C. Li and B. Palanisamy. Incentivized blockchain-based social media platforms: A case study of Steemit. In *Proc. of the 10th ACM Conf. on Web Science (WebSci 2019)*, pages 145–154. ACM Press, 2019. doi:10.1145/3292522.3326041.

[203] J. Li and X. Tang. Roadmap of blockchain standardization. In *Blockchain Application Guide: Methodology and Practice*, pages 193–204. Springer, 2022. doi:10.1007/978-981-19-5260-9_12.

[204] Z. Li. Performance overhead comparison between hypervisor and container-based virtualization, 2017. URL: https://pure.qub.ac.uk/files/468106104/1708.01388.pdf.

[205] Z. Li and P. Mohapaira. The impact of topology on overlay routing service. In *Proc. of the 23rd IEEE Conf. on Computer Communications (INFOCOM 2004)*, volume 1, pages 1–418. IEEE Computer Society, 2004. doi:10.1109/INFCOM.2004.1354513.

[206] B. Y. Lin, D. Dziubałtowska, P. Macek, A. Penzkofer, and S. Müller. TangleSim: An agent-based, modular simulator for DAG-based distributed ledger technologies. In *Proc. of the 5th IEEE Int. Conf. on Blockchain and Cryptocurrency (ICBC 2023)*, pages 1–5. IEEE Computer Society, 2023. doi:10.1109/ICBC56567.2023.10174950.

[207] Q. Lin, C. Li, X. Zhao, and X. Chen. Measuring decentralization in Bitcoin and Ethereum using multiple metrics and granularities. In *Proc. of the 37th IEEE Int. Conf. on Data Engineering Workshops (ICDEW 2021)*, pages 80–87. IEEE Computer Society, 2021. doi:10.1109/ICDEW53142.2021.00022.

[208] H. Liu, Y. Mao, and X. Li. An empirical analysis of EOS blockchain: Architecture, contract, and security. Technical report, 2025. arXiv:2505.15051.

[209] J. Liu, W. Zheng, D. Lu, J. Wu, and Z. Zheng. From decentralization to oligopoly: A data-driven analysis of decentralization evolution and voting behaviors on EOSIO. *IEEE Trans. on Computational Social Systems*, 10(5):2752–2763, 2023. doi:10.1109/TCSS.2022.3191350.

[210] X. F. Liu, C. G. Akcora, Z. Y. Zhang, and J. G. Liu. Editorial: Cryptocurrency transaction analysis from a network perspective. *Frontiers in Physics*, 10:1–2, 2022. doi:10.3389/fphy.2022.876983.

[211] F. L. Loi, K. Tang, and Y. You. Cultural price premium: Evidence from CryptoPunks. *SSRN*, 4202168:1–61, 2023. doi:10.2139/ssrn.4202168.

[212] E. K. Lua, J. Crowcroft, M. Pias, R. Sharma, and S. Lim. A survey and comparison of peer-to-peer overlay network schemes. *IEEE Communications Surveys & Tutorials*, 7(3):72–93, 2005. doi:10.1109/COMST.2005.1610546.

[213] Y. Lyu. Active addresses of Ethereum. *SSRN*, 3879975:1–23, 2021. doi:10.2139/ssrn.3879975.

[214] L. Ma, X. Liu, Y. Li, C. Zhang, G. Shi, and K. Li. GFBE: A generalized and fine-grained blockchain evaluation framework. *IEEE Trans. on Computers*, 73(3):942–955, 2024. doi:10.1109/TC.2024.3349654.

[215] X. Ma, H. Wu, D. Xu, and K. Wolter. CBlockSim: A modular high-performance blockchain simulator. In *Proc. of the 4th IEEE Int. Conf. on Blockchain and Cryptocurrency (ICBC 22)*, pages 1–5. IEEE, 2022. doi:10.1109/ICBC54727.2022.9805504.

[216] I. Mabruri, C. Cerqueda, C. Jack, A. Lui, Y. Wu, W. Wu, V. P. Di Perna, K. Bear, and B. Z. Zhang. Dynamic taxonomy: A bridge from DeFi to TradFi. *SSRN*, 5034378:1–26, 2024. doi:10.2139/ssrn.5034378.

[217] V. Makarskyy. Cryptoeconomic theory: Pareto efficiency, 2018. URL: https://medium.com/blockchannel/cryptoeconomic-theory-pareto-efficiency-89d34664f9d#:~:text=An%20outcome%20is%20called%20Pareto,without%20making%20anyone%20worse%20off.

[218] I. Malakhov, A. Marin, and S. Rossi. Analysis of the confirmation time in proof-of-work blockchains. *Future Generation Computer Systems*, 147:275–291, 2023. doi:10.1016/j.future.2023.04.016.

[219] S. Malwa. Solana rolls out update to tackle network congestion, 2024. URL: https://www.coindesk.com/tech/2024/04/15/solana-rolls-out-update-to-tackle-network-congestion/.

[220] B. Markus and J. Palmer. Dogechain whitepaper v.1.1, 2022. URL: https://dogechain.dog/DogechainWP.pdf.

[221] I. Mashaly. Geth-POA Tutorial – README: Block Time, 2019. URL: https://github.com/ibrahimmashaly/geth-poa-tutorial/blob/1257b0d397edd7bc5a508a36c37c0d8df898ff9a/README.md#block-time.

[222] M. Matos, V. Schiavoni, P. Felber, R. Oliveira, and E. Rivière. Lightweight, efficient, robust epidemic dissemination. *Journal of Parallel and Distributed Computing*, 73(7):987–999, 2013. doi:10.1016/j.jpdc.2013.01.018.

[223] C. Mattsson. Networks of monetary flow at native resolution. Technical report, 2019. arXiv:1910.05596.

[224] P. Maymounkov and D. Mazieres. Kademlia: A peer-to-peer information system based on the XOR metric. In *Proc. of the 1st Peer-to-Peer Systems Int. Workshop (IPTPS 2002)*, volume 2429 of *Lecture Notes in Computer Science*, pages 53–65. Springer, 2002. doi:10.1007/3-540-45748-8_5.

[225] D. Mazieres. The Stellar consensus protocol: A federated model for Internet-level consensus. *Stellar Development Foundation*, 32:1–45, 2015. URL: https://www.stellar.org/papers/stellar-consensus-protocol.pdf.

[226] M. Mazziotta and A. Pareto. A non-compensatory approach for the measurement of the quality of life. In *Quality of Life in Italy: Research and Reflections*, volume 48

of *Social Indicators Research Series*, pages 27–40. Springer, 2012. doi:10.1007/978-94-007-3898-0_3.

[227] M. Mazziotta and A. Pareto. Methods for constructing composite indices: One for all or all for one? *Rivista Italiana di Economia, Demografia e Statistica*, 67(2):67–80, 2013. URL: https://www.istat.it/en/files/2013/12/Rivista2013_Mazziotta_Pareto.pdf.

[228] M. Mazzoni, A. Corradi, and V. Di Nicola. Performance evaluation of permissioned blockchains for financial applications: The ConsenSys Quorum case study. *Blockchain: Research and Applications*, 3(1):1–11, 2022. doi:10.1016/j.bcra.2021.100026.

[229] D. Mechkaroska, V. Dimitrova, and A. Popovska-Mitrovikj. Analysis of the possibilities for improvement of blockchain technology. In *Proc. of the 26th Telecommunications Forum (TELFOR 2018)*, pages 1–4, 2018. doi:10.1109/TELFOR.2018.8612034.

[230] D. A. Menascé and V. A. F. Almeida. *Capacity Planning for Web Services: Metrics, Models, and Methods*. Prentice Hall, 2002. doi:10.5555/560806.

[231] R. C. Merkle. A digital signature based on a conventional encryption function. In *Advances in Cryptology (CRYPTO 1987)*, volume 293 of *Lecture Notes in Computer Science*, pages 369–378. Springer, 1988. doi:10.1007/3-540-48184-2_32.

[232] A. Miller and R. Jansen. Shadow-Bitcoin: Scalable simulation via direct execution of multi-threaded applications. In *Proc. of the 8th USENIX Conf. on Cyber Security Experimentation and Test (CSET 2015)*, pages 1–8. USENIX Association, 2015. URL: https://www.usenix.org/conference/cset15/workshop-program/presentation/miller.

[233] R. Moncada, E. Ferro, M. Fiaschetti, and F. Medda. Blockchain tokens, price volatility, and active user base: An empirical analysis based on tokenomics. *Int. Journal of Financial Studies*, 12(4):1–30, 2024. doi:10.3390/ijfs12040107.

[234] Monetary Authority of Singapore. MAS expands industry collaboration to scale asset tokenisation for financial services (project guardian). Technical report, MAS, 2024. URL: https://www.sgpc.gov.sg/api/file/getfile/MAS%20Media%20Release%20-%20MAS%20Expands%20Industry%20Collaboration%20to%20Scale%20Asset%20Tokenisation%20for%20Financial%20Services.pdf?path=%2Fsgpcmedia%2Fmedia_releases%2Fmas%2Fpress_release%2FP-20240627-1%2Fattachment%2FMAS+Media+Release+-+MAS+Expands+Industry+Collaboration+to+Scale+Asset+Tokenisation+for+Financial+Services.pdf.

[235] S. Nakamoto. Bitcoin: A peer-to-peer electronic cash system. 2009. URL: http://www.bitcoin.org/bitcoin.pdf.

[236] A. Narain, T. Mancini-Griffoli, and A. Jobst. Blockchain consensus mechanisms: A primer for supervisors. IMF Fintech Notes 2022/001, International Monetary Fund, 2022. URL: https://www.imf.org/en/Publications/fintech-notes/Issues/2022/01/25/Blockchain-Consensus-Mechanisms-511769.

[237] B. Nasrulin, M. De Vos, G. Ishmaev, and J. Pouwelse. Gromit: Benchmarking the performance and scalability of blockchain systems. In *Proc. of the 4th IEEE Int. Conf. on Decentralized Applications and Infrastructures (DAPPS 2022)*, pages 56–63. IEEE Computer Society, 2022. doi:10.1109/DAPPS55202.2022.00015.

[238] G. Naumenko, G. Maxwell, P. Wuille, S. Fedorova, and I. Beschastnikh. Erlay: Efficient transaction relay for Bitcoin. In *Proc. of the 26th ACM SIGSAC Conf. on Computer and Communications Security (CCS 2019)*, pages 817–831. ACM Press, 2019. doi:10.1145/3319535.3363213.

[239] R. Neiheiser, M. Matos, and L. Rodrigues. Kauri: Scalable BFT consensus with pipelined tree-based dissemination and aggregation. In *Proc. of the 28th ACM SIGOPS Symp. on Operating Systems Principles (SOSP 2021)*, pages 35–48. ACM Press, 2021. doi:10.1145/3477132.3483584.

[240] T. Neudecker. Characterization of the Bitcoin Peer-to-Peer network (2015-2018). Technical report, Karlsruher Institut für Technologie (KIT), 2019. doi:10.5445/IR/1000091933.

[241] M. E. J. Newman. Assortative mixing in networks. *Physical Review Letters*, 89(20):208701, 2002. doi:10.1103/PhysRevLett.89.208701.

[242] M. E. J. Newman. The structure and function of complex networks. *SIAM Review*, 45(2):167–256, 2003. doi:10.1137/S003614450342480.

[243] M. E. J. Newman. *Networks: An Introduction*. Oxford University Press, 2010. doi:10.1093/acprof:oso/9780199206650.001.0001.

[244] 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*, 7:85727–85745, 2019. doi:10.1109/ACCESS.2019.2925010.

[245] Ocean Protocol Foundation Ltd. Ocean protocol: Tools for the Web3 data economy. Technical whitepaper, 2022. URL: https://oceanprotocol.com/tech-whitepaper.pdf.

[246] OECD, European Union, and European Commission, Joint Research Centre. *Handbook on Constructing Composite Indicators: Methodology and User Guide*. OECD Publishing, 2008. doi:10.1787/9789264043466-en.

[247] J. M. Oglio. *Torus, TRAIL, and SmartShards: Topological approaches to fortify against byzantine faults*. Ph.D. Thesis, Kent State University, 2025. URL: http://rave.ohiolink.edu/etdc/view?acc_num=kent1743842899708713.

[248] O. Oleksenko, D. Kuvaiskii, P. Bhatotia, and C. Fetzer. Fex: A software systems evaluator. In *Proc. of the 47th IEEE/IFIP Int. Conf. on Dependable Systems and Networks (DSN 2017)*, pages 543–550. IEEE Computer Society, 2017. doi:10.1109/DSN.2017.25.

[249] A. Ometov, O. L. Molua, M. Komarov, and J. Nurmi. A survey of security in cloud, edge, and fog computing. *Sensors*, 22(3):1–27, 2022. doi:10.3390/S22030927.

[250] P. K. Ozili. eNaira Central Bank Digital Currency (CBDC) for financial inclusion in Nigeria. *SSRN*, 4237168:1–25, 2023. doi:10.2139/ssrn.4237168.

[251] M. Pacheco, G. Oliva, G. K. Rajbahadur, and A. Hassan. Is my transaction done yet? An empirical study of transaction processing times in the Ethereum blockchain platform. *ACM Trans. on Software Engineering and Methodology*, 32(3), 2023. doi:10.1145/3549542.

[252] H. Pan, X. Duan, Y. Wu, L. Tseng, M. Aloqaily, and A. Boukerche. BBB: A lightweight approach to evaluate private blockchains in clouds. In *Proc. of the 21st IEEE Global Communications Conf. (GLOBECOM 2020)*, pages 1–6. IEEE Computer Society, 2020. doi:10.1109/GLOBECOM42002.2020.9322354.

[253] M. Parashar. Editor's note: IEEE Trans. on Parallel and Distributed Systems (TPDS) reproducibility initiative. *IEEE Trans. on Parallel and Distributed Systems*, 30(8):1690, 2019. doi:10.1109/TPDS.2019.2922052.

[254] S. Park, S. Im, Y. Seol, and J. Paek. Nodes in the Bitcoin network: Comparative measurement study and survey. *IEEE Access*, 7:57009–57022, 2019. doi:10.1109/ACCESS.2019.2914098.

[255] U. Pavloff, Y. Amoussou-Guenou, and S. Tucci-Piergiovanni. Byzantine attacks exploiting penalties in Ethereum PoS. In *Proc. of the 54th IEEE/IFIP Int. Conf. on Dependable Systems and Networks (DSN 2024)*, pages 53–65. IEEE Computer Society, 2024. doi:10.1109/DSN58291.2024.00020.

[256] I. Pepelnjak. Netlab: A virtual networking labbing tool, 2024. URL: https://netlab.tools/.

[257] Y. Pi, F. Qian, Y. Zhang, and Z. M. M. Li. Latency imbalance among Internet load-balanced paths: A cloud-centric view. *Proc. of the ACM on Measurement and Analysis of Computing Systems*, 4(2):32:1–32:29, 2020. doi:10.1145/3392150.

[258] G. A. Pierro and R. Tonelli. A study on Diem distributed ledger technology. In *Proc. of the 4th Distributed Ledger Technology Workshop (DLT 2022)*, volume

3166 of *CEUR Workshop Proceedings*, pages 33–47. CEUR-WS.org, 2022. URL: https://ceur-ws.org/Vol-3166/paper03.pdf.

[259] B. Pillai, J. Tharani, and V. Muthukkumarasamy. Wormhole cross-chain bridge transactions flow: An exploratory study. In *Proc. of the 3rd IEEE Int. Workshop on Cryptocurrency Exchanges (CryptoEx 2025)*, 2025. URL: https://www.researchgate.net/publication/392369155_Wormhole_Cross-Chain_Bridge_Transactions_Flow_An_Exploratory_Study.

[260] A. Pimpini and A. Pellegrini. RBlockSim: Parallel and distributed simulation for blockchain benchmarking. In *Proc. of the 39th ACM SIGSIM Conf. on Principles of Advanced Discrete Simulation (PADS 2025)*, pages 176–185. ACM Press, 2025. doi:10.1145/3726301.3728421.

[261] M. Platt, J. Sedlmeir, D. Platt, J. Xu, P. Tasca, N. Vadgama, and J. I. Ibañez. The energy footprint of blockchain consensus mechanisms beyond Proof-of-Work. In *Proc. of the 21st IEEE Int. Conf. on Software Quality, Reliability and Security (QRS 2021)*, pages 1135–1144. IEEE Computer Society, 2021. doi:10.1109/QRS-C55045.2021.00168.

[262] J. Polge, S. Ghatpande, S. Kubler, J. Robert, and Y. Le Traon. BlockPerf: A hybrid blockchain emulator/simulator framework. *IEEE Access*, 9:107858–107872, 2021. doi:10.1109/ACCESS.2021.3101044.

[263] S. Popov. The Tangle, 2018. URL: https://www.cryptocompare.com/media/38553943/iota.pdf.

[264] A. Preukschat and D. Reed. *Self-Sovereign Identity: Decentralized Digital Identity and Verifiable Credentials*. Manning Publications, 2021. URL: https://www.manning.com/books/self-sovereign-identity.

[265] Protocol Labs Research. Filecoin: A decentralized storage network, 2025. URL: https://research.protocol.ai/publications/filecoin-a-decentralized-storage-network/.

[266] X. Pu, L. Liu, Y. Mei, S. Sivathanu, Y. Koh, C. Pu, and Y. Cao. Who is your neighbor: Net I/O performance interference in virtualized clouds. *IEEE Trans. on Services Computing*, 6(3):314–329, 2013. doi:10.1109/TSC.2012.2.

[267] A. Qin and E. Livni. China cracks down harder on cryptocurrency with new ban, 2021. URL: https://www.nytimes.com/2021/09/24/business/china-cryptocurrency-bitcoin.html#:~:text=China%20intensified%20its%20crackdown%20on,for%20newly%20created%20crypto%20tokens.

[268] K. Ren, J. F. B. Van Buskirk, Z. Y. Ang, S. Hou, N. R. Cable, M. Monares, H. F. Korth, and D. Loghin. BBSF: Blockchain benchmarking standardized framework.

In *Proc. of the 1st Workshop on Verifiable Database Systems (VDBS 2023)*, pages 10–18. ACM Press, 2023. doi:10.1145/3595647.3595649.

[269] Ripple. Faq, 2025. URL: https://xrpl.org/about/faq.

[270] N. Rodriguez-Rodriguez and O. Miramontes. Shannon entropy: An econophysical approach to cryptocurrency portfolios. *Entropy*, 24(11):1–15, 2022. doi:10.3390/e24111583.

[271] P. Rogaway and T. Shrimpton. Cryptographic hash-function basics: Definitions, implications, and separations for preimage resistance, second-preimage resistance, and collision resistance. In *Proc. of the 11th Int. Workshop on Fast Software Encryption (FSE 2004)*, volume 3017 of *Lecture Notes in Computer Science*, pages 371–388. Springer, 2004. doi:10.1007/978-3-540-25937-4_24.

[272] E. Rohrer, J. Malliaris, and F. Tschorsch. Discharged payment channels: Quantifying the Lightning network resilience to topology-based attacks. In *Proc. of the 4th IEEE European Symp. on Security and Privacy Workshops (EuroS&P-W 2019)*, pages 347–356. IEEE Computer Society, 2019. doi:10.1109/EuroSPW.2019.00045.

[273] T. Roughgarden. Transaction fee mechanism design for the Ethereum blockchain: An economic analysis of EIP-1559. Technical report, Columbia University, 2021. URL: https://timroughgarden.org/papers/eip1559.pdf.

[274] D. Ryan and V. Buterin. EIP-2982: Serenity Phase 0, 2020. URL: https://eips.ethereum.org/EIPS/eip-2982.

[275] S. Saberi, M. Kouhizadeh, J. Sarkis, and L. Shen. Blockchain technology and its relationships to sustainable supply chain management. *Int. Journal of Production Research*, 57(7):2117–2135, 2019. doi:10.1080/00207543.2018.1533261.

[276] A. R. Sai, J. Buckley, and A. Le Gear. Characterizing wealth inequality in cryptocurrencies. *Frontiers in Blockchain*, 4:1–20, 2021. doi:10.3389/fbloc.2021.730122.

[277] A. Saif and Q. Hu. Powered by blockchain technology, DeFi (decentralized finance) strives to increase financial inclusion of the unbanked by reshaping the world financial system. *Modern Economy*, 12:1–16, 2021. doi:10.4236/me.2021.121001.

[278] D. Saingre, T. Ledoux, and J.-M. Menaud. BCTMark: A framework for benchmarking blockchain technologies. In *Proc. of the 17th IEEE/ACS Int. Conf. on Computer Systems and Applications (AICCSA 2020)*, pages 1–8. IEEE Computer Society, 2020. doi:10.1109/AICCSA50499.2020.9316536.

[279] D. Saingre, T. Ledoux, and J.-M. Menaud. Measuring performances and footprint of blockchains with BCTMark: A case study on Ethereum smart contracts energy consumption. *Cluster Computing*, 25(4):2819–2837, 2022. doi:10.1007/s10586-021-03441-x.

[280] R. Saltini. IBFT liveness analysis. In *Proc. of the 1st IEEE Int. Conf. on Blockchain (Blockchain 2019)*, pages 245–252. IEEE Computer Society, 2019. doi:10.1109/Blockchain.2019.00039.

[281] SBA Research. Simcoin: Blockchain Simulation Framework with Docker and Python, 2017. URL: https://github.com/sbaresearch/simcoin.

[282] M. Schäffer, M. di Angelo, and G. Salzer. Performance and scalability of private Ethereum blockchains. In *Proc. of the Blockchain and Central and Eastern Europe Forum (BPM 2019)*, volume 361 of *Lecture Notes in Business Information Processing*, pages 103–118. Springer, 2019. doi:10.1007/978-3-030-30429-4_8.

[283] Z. Sebestyén and V. Juhász. The impact of the cost of unused capacity on production planning of flexible manufacturing systems. *Periodica Polytechnica Social and Management Sciences*, 11(2):185–200, 2003. URL: https://pp.bme.hu/so/article/view/1688.

[284] K. Sekniqi, D. Laine, S. Buttolph, and E. G. Sirer. Avalanche platform whitepaper, 2020. URL: https://cdn.prod.website-files.com/5d80307810123f5ffbb34d6e/6008d7bbf8b10d1eb01e7e16_Avalanche%20Platform%20Whitepaper.pdf.

[285] K. Shah, D. Lathiya, N. Lukhi, K. Parmar, and H. Sanghvi. A systematic review of decentralized finance protocols. *Int. Journal of Intelligent Networks*, 4:171–181, 2023. doi:10.1016/j.ijin.2023.07.002.

[286] Z. Shahbazi and Y.-C. Byun. Integration of blockchain, IoT and machine learning for multistage quality control and enhancing security in smart manufacturing. *Sensors*, 21(4):1–18, 2021. doi:10.3390/s21041467.

[287] C. E. Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27(3):379–423, 1948. doi:10.1002/j.1538-7305.1948.tb01338.x.

[288] 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*, 55(9):78–85, 2017. doi:10.1109/MCOM.2017.1700041.

[289] Z. She. VeChain: A renovation of supply chain management – a look into its organization, current activity, and prospect. In *Proc. of the 1st Int. Conf. on Educational Informatization, E-commerce and Information System (ICEIEIS 2022)*, pages 29–30, 2022. URL: https://webofproceedings.org/proceedings_series/ESSP/ICEIEIS%202022/G68.pdf.

[290] R. Shi, R. Cheng, B. Han, Y. Cheng, and S. Chen. A closer look into IPFS: Accessibility, content, and performance. *Proc. of the ACM on Measurement and Analysis of Computing Systems*, 8(2):1–31, 2024. doi:10.1145/3656015.

[291] T. Sikder. Why is Dogecoin down? 69% node crash due to network vulnerability, 2024. URL: https://www.financemagnates.com/trending/why-is-dogecoin-down-69-node-crash-due-to-network-vulnerability/.

[292] J. Sliwinski, Q. Kniep, R. Wattenhofer, and F. Schaich. Halting the Solana blockchain with epsilon stake. In *Proc. of the 25th Int. Conf. on Distributed Computing and Networking (ICDCN 2024)*, pages 45–54, 2024. doi:10.1145/3631461.3631553.

[293] C. Smith and J. Rennison. How Washington plans to defend the dollar. *Financial Times*, 2024. URL: https://www.ft.com/content/bfafb8f7-bd1c-48bb-85f4-8ba25475c0a3.

[294] R. D. Smith. Bitcoin average dormancy: A measure of turnover and trading activity. *Ledger*, 3:91–99, 2018. doi:10.5195/ledger.2018.99.

[295] Y. Sokolik and O. Rottenstreich. Age-aware fairness in blockchain transaction ordering. In *Proc. of the 28th IEEE/ACM Int. Symp. on Quality of Service (IWQoS 2020)*, pages 1–9. IEEE Computer Society, 2020. doi:10.1109/IWQoS49365.2020.9212952.

[296] Solana. getLeaderSchedule RPC method. URL: https://solana.com/docs/rpc/http/getleaderschedule.

[297] Solana. Turbine block propagation. URL: https://docs.solana.com/cluster/turbine-block-propagation.

[298] Solana. Retrying transactions, 2024. URL: https://solana.com/docs/advanced/retry.

[299] Solana. Transaction confirmation and expiration, 2024. URL: https://solana.com/docs/advanced/confirmation.

[300] Solana Beach. Solana Beach Explorer, 2025. URL: https://solanabeach.io/.

[301] Solana-Labs, 2022. URL: https://github.com/solana-labs/solana/blob/master/programs/vote/src/vote_state/mod.rs#L34.

[302] M. Solvak and A. Lauringson. A case study of the public sector digital ecosystem in Estonia. *Computer*, 57(5):44–49, 2024. doi:10.1109/MC.2024.3375866.

[303] Y. Sompolinsky and A. Zohar. Secure high-rate transaction processing in Bitcoin. In *Proc. of the 19th Int. Conf. on Financial Cryptography and Data Security (FC 2015)*, volume 8975 of *Lecture Notes in Computer Science*, pages 507–527. Springer, 2015. doi:10.1007/978-3-662-47854-7_32.

[304] Steam. Dota 2 on steam, 2013. URL: https://store.steampowered.com/app/570/Dota_2/.

[305] J. Stinner. On the economics of Bitcoin mining: A theoretical framework and simulation evidence. In *Proc. of the 43rd Int. Conf. on Information Systems (ICIS 2022)*, pages 2065–2081. AIS, 2022. URL: https://aisel.aisnet.org/icis2022/blockchain/blockchain/7.

[306] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan. Chord: A scalable peer-to-peer lookup service for Internet applications. In *Proc. of the 15th ACM Conf. on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM 2001)*, pages 149–160. ACM Press, 2001. doi:10.1145/383059.383071.

[307] W. Sun, Z. Xu, W. Ni, and L. Chen. InTime: Towards performance predictability in Byzantine fault tolerant Proof-of-Stake consensus. *Proc. of the ACM on Management of Data*, 3(1):1–27, 2025. doi:10.1145/3709740.

[308] P. Szilágyi. EIP-225: Clique proof-of-authority consensus protocol, 2017. URL: https://eips.ethereum.org/EIPS/eip-225.

[309] Team Rocket. Scalable and probabilistic leaderless BFT consensus through metastability, 2019. arXiv:1906.08936.

[310] Tether, 2025. URL: https://assets.ctfassets.net/vyse88cgwfbl/5UWgHMvz071t2Cq5yTw5vi/c9798ea8db99311bf90ebe0810938b01/TetherWhitePaper.pdf.

[311] H. Theil. *Economics and information theory*. North-Holland Publishing Company, 1967. URL: https://search.worldcat.org/title/167102.

[312] T. Toivola. VnStat – a network traffic monitor for Linux and BSD, 2014. URL: https://humdi.net/vnstat/.

[313] B. Toshniwal and K. Kataoka. Comparative performance analysis of underlying network topologies for blockchain. In *Proc. of the 35th Int. Conf. on Information Networking (ICOIN 2021)*, pages 367–372, 2021. doi:10.1109/ICOIN50884.2021.9333978.

[314] Trail of Bits. GossipSub v1.1 security audit, 2020. URL: https://github.com/libp2p/specs/tree/master/pubsub/gossipsub.

[315] J. W. Tukey. *Exploratory Data Analysis*. Addison-Wesley, 1977. URL: https://books.google.com/books?id=UT9dAAAAIAAJ.

[316] R. Vaishnavi, C. Athira, C. Pradeep, Sankalpkumar, Eashwara Prasanna, and V. Srikanth. Energy efficiency in blockchain social networks. *Int. Journal of Research Publication and Reviews*, 5(3):819–825, 2024. doi:10.55248/gengpi.5.0324.0632.

[317] M. Vieira and H. Madeira. Benchmarking the dependability of different OLTP systems. In *Proc. of the 33th IEEE/IFIP Int. Conf. on Dependable Systems and Networks (DSN 2003)*, pages 305–310. IEEE Computer Society, 2003. doi:10.1109/DSN.2003.1209940.

[318] D. Vyzovitis, Y. Napora, D. McCormick, D. Dias, and Y. Psaras. Gossipsub-v1.1 evaluation report, 2020. URL: https://research.protocol.ai/publications/gossipsub-v1.1-evaluation-report/.

[319] G. Wang, Y. Zhang, C. Ying, X. Lit, and G. Yu. Hammer: A general blockchain evaluation framework. In *Proc. of the 44th IEEE Int. Conf. on Distributed Computing Systems (ICDCS 2024)*, pages 391–402. IEEE Computer Society, 2024. doi:10.1109/ICDCS60910.2024.00044.

[320] H. Wang, H. Li, Q. Ye, P. Lu, Y. Yang, P. H. J. Chong, X. Chu, Q. Lv, and A. Smahi. A physical topology for optimizing partition tolerance in consortium blockchains to reach CAP guarantee bound. *Trans. on Emerging Telecommunications Technologies*, 34(9):e4820, 2023. doi:10.1002/ett.4820.

[321] R. Wang, J.-A. Lee, and J. Liu. Unwinding NFTs in the shadow of IP law. *American Business Law Journal*, 61(1):31–55, 2024. doi:10.1111/ablj.12237.

[322] S. Wang, W. Ding, J. Li, Y. Yuan, L. Ouyang, and F.-Y. Wang. Decentralized autonomous organizations: Concept, model, and applications. *IEEE Trans. on Computational Social Systems*, 6(5):870–878, 2019. doi:10.1109/TCSS.2019.2938190.

[323] T. Wang, Z. Su, Y. Xia, B. Qin, and M. Hamdi. NovaCube: A low latency Torus-based network architecture for data centers. In *Proc. of the 15th IEEE Global Communications Conf. (GLOBECOM 2014)*, pages 2252–2257. IEEE Computer Society, 2014. doi:10.1109/GLOCOM.2014.7037143.

[324] X. Wang, A. Al-Mamun, F. Yan, and D. Zhao. Toward accurate and efficient emulation of public blockchains in the cloud. In *Proc. of the 12th Int. Conf. on Cloud Computing (CLOUD 2019)*, volume 11513 of *Lecture Notes in Computer Science*, pages 67–82. Springer, 2019. doi:10.1007/978-3-030-23502-4_6.

[325] M. Warade, K. Lee, C. Ranaweera, and J.-G. Schneider. Monitoring the energy consumption of docker containers. In *Proc. of the 47th IEEE Computers, Software, and Applications Conf. (COMPSAC 2023)*, pages 1703–1710. IEEE Computer Society, 2023. doi:10.1109/COMPSAC57700.2023.00263.

[326] J. Warner. Linguistics and information theory: Analytic advantages. *Journal of the American Society for Information Science and Technology*, 58(2):275–285, 2007. doi:10.1002/asi.20488.

[327] D. J. Watts and S. H. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393(6684):440–442, 1998. doi:10.1038/30918.

[328] K. Wenker. Retail Central Bank Digital Currencies (CBDC), disintermediation and financial privacy: The case of the Bahamian Sand Dollar. *FinTech*, 1(4):345–361, 2022. doi:10.3390/fintech1040026.

[329] D. B. West. *Introduction to Graph Theory*. Prentice Hall, 2001. URL: https://dwest.web.illinois.edu/igt/.

[330] W. Willinger, M. S. Taqqu, R. Sherman, and D. V. Wilson. Self-similarity through high-variability: Statistical analysis of Ethernet LAN traffic at the source level. *IEEE/ACM Trans. on Networking*, 5(1):71–86, 1997. doi:10.1109/90.554723.

[331] G. Wood. Ethereum: A secure decentralised generalised transaction ledger. *Ethereum project yellow paper*, 151:1–32, 2014. URL: https://ethereum.github.io/yellowpaper/paper.pdf.

[332] A. J. Wright. The rise of decentralized autonomous organizations: Opportunities and challenges. *Stanford Journal of Blockchain Law and Policy*, 4(2):152–176, 2021. URL: https://stanford-jblp.pubpub.org/pub/rise-of-daos.

[333] K. Wu, B. Peng, H. Xie, and Z. Huang. An information entropy method to quantify the degrees of decentralization for blockchain systems. In *Proc. of the 9th IEEE Int. Conf. on Electronics Information and Emergency Communication (ICEIEC 2019)*, pages 1–6. IEEE Computer Society, 2019. doi:10.1109/ICEIEC.2019.8784631.

[334] X. Wu, J. Yan, and D. Jin. Virtual-time-accelerated emulation for blockchain network and application evaluation. In *Proc. of the 27th ACM SIGSIM Conf. on Principles of Advanced Discrete Simulation (PADS 2019)*, pages 149–160. ACM Press, 2019. doi:10.1145/3316480.3322889.

[335] J. Xu, C. Wang, and X. Jia. A survey of blockchain consensus protocols. *ACM Computing Surveys*, 55(13s):1–35, 2023. doi:10.1145/3579845.

[336] X. Xu, I. Weber, M. Staples, L. Zhu, J. Bosch, L. Bass, C. Pautasso, and P. Rimba. A taxonomy of blockchain-based systems for architecture design. In *Proc. of the 14th IEEE Int. Conf. on Software Architecture (ICSA 2017)*, pages 243–252. IEEE Computer Society, 2017. doi:10.1109/ICSA.2017.33.

[337] T. Xue, Y. Yuan, and C. Wang. An approach for evaluating user participation in Bitcoin. In *Proc. of the 3rd IEEE Int. Conf. on Data Science in Cyberspace (DSC 2018)*, pages 858–864. IEEE Computer Society, 2018. doi:10.1109/DSC.2018.00138.

[338] D. Yaga, P. Mell, N. Roby, and K. Scarfone. Blockchain technology overview. Technical Report NISTIR 8202, NIST, 2018. doi:10.6028/NIST.IR.8202.

[339] H. Yajam, E. Ebadi, and M. A. Akhaee. JABS: A blockchain simulator for researching consensus algorithms. *IEEE Trans. on Network Science and Engineering*, 11(1):3–13, 2024. doi:10.1109/TNSE.2023.3282916.

[340] A. Yakovenko. Tower BFT: Solana's high performance implementation of PBFT, 2020. URL: https://medium.com/solana-labs/tower-bft-solanas-high-performance-implementation-of-pbft-464725911e79.

[341] A. Yakovenko. Turbine-Solana's block propagation protocol solves the scalability trilemma, 2020. URL: https://medium.com/solana-labs/turbine-solanas-block-propagation-protocol-solves-the-scalability-trilemma-2ddba46a51db.

[342] A. Yakovenko. Solana: A new architecture for a high performance blockchain v0.8.13, 2021. URL: https://solana.com/solana-whitepaper.pdf.

[343] A. Yakovenko and Solana Labs. Sealevel: Parallel processing of smart contracts, 2019. URL: https://solana.com/en/news/sealevel---parallel-processing-thousands-of-smart-contracts.

[344] M. M. Yakubu, M. F. B. Hassan, K. U. Danyaro, A. Z Junejo, M. Siraj, S. Yahaya, S. Adamu, and K. Abdulsalam. A systematic literature review on blockchain consensus mechanisms' security: Applications and open challenges. *Computer Systems Science and Engineering*, 48(6):1437–1481, 2024. doi:10.32604/csse.2024.054556.

[345] T. Yan, S. Li, B. Kraner, L. Zhang, and C. J. Tessone. Analyzing reward dynamics and decentralization in Ethereum 2.0: An advanced data engineering workflow and comprehensive datasets for Proof-of-Stake incentives. Technical report, 2024. arXiv:2402.11170.

[346] 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 Trans. on Services Computing*, 13(4):625–638, 2020. doi:10.1109/TSC.2020.2966970.

[347] M. Yin, D. Malkhi, M. K. Reiter, G. G. Gueta, and I. Abraham. HotStuff: BFT consensus with linearity and responsiveness. In *Proc. of the 38th ACM Symp. on Principles of Distributed Computing (PODC 2019)*, pages 347–356. ACM Press, 2019. doi:10.1145/3293611.3331591.

[348] S. Yousefi and B. Mohamadpour Tosarkani. An analytical approach for evaluating the impact of blockchain technology on sustainable supply chain performance. *Int. Journal of Production Economics*, 246:1–22, 2022. doi:10.1016/j.ijpe.2022.108429.

[349] J. Zakrzewski. Towards verification of Ethereum smart contracts: A formalization of core of Solidity. In *Proc. of the 10th Int. Conf. on Verified Software, Theories, Tools, and Experiments (VSTTE 2018)*, volume 11294 of *Lecture Notes in Computer Science*, pages 229–247. Springer, 2018. doi:10.1007/978-3-030-03592-1_13.

[350] J. Zhang, J. Gao, Z. Wu, W. Yan, Q. Wo, Q. Li, and Z. Chen. Performance analysis of the Libra blockchain: An experimental study. In *Proc. of the 2nd Int.*

Conf. on Hot Information-Centric Networking (HotICN 2019), pages 77–83, 2019. doi:10.1109/HotICN48464.2019.9063213.

[351] G. Zyskind, O. Nathan, and A. S. Pentland. Decentralizing privacy: Using blockchain to protect personal data. In *Proc. of the 36th IEEE Security and Privacy Workshops (SPW 2015)*, pages 180–184. IEEE Computer Society, 2015. doi:10.1109/SPW.2015.27.