

Review

# A Survey on Network Optimization Techniques for Blockchain Systems

Robert Antwi <sup>1,\*</sup>, James Dzisi Gadze <sup>1,†</sup>, Eric Tutu Tchao <sup>2</sup>, Axel Sikora <sup>3,\*</sup>, Henry Nunoo-Mensah <sup>2</sup>, Andrew Selasi Agbemenu <sup>2</sup>, Kwame Opunie-Boachie Obour Agyekum <sup>1</sup>, Justice Owusu Agyemang <sup>1</sup>, Dominik Welte <sup>3</sup> and Eliel Keelson <sup>2</sup>

<sup>1</sup> Department of Telecommunication Engineering, Kwame Nkrumah University of Science and Technology (KNUST), Kumasi AK-448-3929, Ghana; jdgadze.coe@knust.edu.gh (J.D.G.); kooagyekum@knust.edu.gh (K.O.-B.O.A.); joagyemang@knust.edu.gh (J.O.A.)

<sup>2</sup> Department of Computer Engineering, Kwame Nkrumah University of Science and Technology (KNUST), Kumasi AK-448-3929, Ghana; ettchao.coe@knust.edu.gh (E.T.T.); hnunoo-mensah@knust.edu.gh (H.N.-M.); asagbemenu@knust.edu.gh (A.S.A.); ekeelson@knust.edu.gh (E.K.)

<sup>3</sup> Institute of Reliable Embedded Systems and Communication Electronics (ivESK), Offenburg University of Applied Sciences, 77652 Offenburg, Germany; dominik.welte@hs-offenburg.de

\* Correspondence: rantwi31@st.knust.edu.gh (R.A.); axel.sikora@hs-offenburg.de (A.S.)

† These authors contributed equally to this work.

**Abstract:** The increase of the Internet of Things (IoT) calls for secure solutions for industrial applications. The security of IoT can be potentially improved by blockchain. However, blockchain technology suffers scalability issues which hinders integration with IoT. Solutions to blockchain's scalability issues, such as minimizing the computational complexity of consensus algorithms or blockchain storage requirements, have received attention. However, to realize the full potential of blockchain in IoT, the inefficiencies of its inter-peer communication must also be addressed. For example, blockchain uses a flooding technique to share blocks, resulting in duplicates and inefficient bandwidth usage. Moreover, blockchain peers use a random neighbor selection (RNS) technique to decide on other peers with whom to exchange blockchain data. As a result, the peer-to-peer (P2P) topology formation limits the effective achievable throughput. This paper provides a survey on the state-of-the-art network structures and communication mechanisms used in blockchain and establishes the need for network-based optimization. Additionally, it discusses the blockchain architecture and its layers categorizes existing literature into the layers and provides a survey on the state-of-the-art optimization frameworks, analyzing their effectiveness and ability to scale. Finally, this paper presents recommendations for future work.

**Keywords:** blockchain; IIoT; network optimization; peer-to-peer; topology; gossip protocol



**Citation:** Antwi, R.; Gadze, J.D.;

Tchao, E.T.; Sikora, A.;

Nunoo-Mensah, H.; Agbemenu, A.S.;

Obour Agyekum, K.O.-B.;

Agyemang, J.O.; Welte, D.; Keelson, E.

A Survey on Network Optimization Techniques for Blockchain Systems.

*Algorithms* **2022**, *15*, 193. <https://doi.org/10.3390/a15060193>

Academic Editors: Shipping Chen, Kaimeng Ding and Yanmei Zhang

Received: 13 May 2022

Accepted: 2 June 2022

Published: 4 June 2022

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

The ubiquity of the Internet of Things (IoT) paradigm has revolutionized many fields of today's life, such as health care, smart homes, supply chain, smart metering and manufacturing [1]. Furthermore, the application of IoT in industrial applications, also known as the Industrial Internet of Things (IIoT), has paved the way for the advancements promised by the fourth industrial revolution (Industry 4.0) [2]. In IIoT, sensors collect data from industrial machines or processes, whereas actuators act on the raw or pre-processed data provided by sensors or other processes. Furthermore, aside from automation, analysis of the data generated gives insights into system efficiency, productivity and the quality of products that can also be used for long-term planning [3].

However, many of these devices have limited resources such as storage, memory and processing power [4]. According to market research, 32-bit, 16-bit and 8-bit Microcontroller Units (MCU) account for 41%, 35% and 24% of the market [5]. Most IoT devices are built

with these MCUs, with Random Access Memory (RAM) sizes of less than 1 MB and a few hundred megahertz clock rates. For example, ESP8266 is a 32-bit MCU, with an 80 MHz clock speed, 80 KB of RAM and 1 MB of flash storage [6,7]. Due to the resource constraints, security, storage and data processing are often handled in the cloud. This dependency on the cloud introduces centralization and contradicts the idea of making services required by IIoT always accessible even when some service providers become unavailable.

Furthermore, cloud-based architectures have data integrity issues and present unacceptable delays for latency-sensitive applications. However, the emergence of distributed computing architectures such as edge and fog computing eliminates centralization and presents acceptable latency for applications requiring high Quality of Service (QoS). Nevertheless, security is still an issue in these architectures since it requires mutual trust between nodes and does not provide data integrity guarantees or protection against malicious attacks [8].

The inability of existing IIoT architectures to ensure trust, data security and integrity makes blockchain an excellent add-on for IIoT architectures. Blockchain is an append-only distributed ledger technology that is collaboratively maintained by the participating nodes without the need to trust any specific peer. Data are first submitted to the blockchain peers as a transaction. Next, many of such transactions are combined into a block that the participating peers must validate. Peers then execute a consensus algorithm to agree on the block's validity before it is committed to the main ledger. Finally, every two consecutive blocks are interlinked using the block's hash [9]. Thus, the blockchain can secure data against unauthorized modification, making it immutable, traceable and trustworthy. Once a block is confirmed valid, it is then broadcast to all peers in the network to ensure the global consistency of the ledger.

Despite blockchain's security and availability guarantees, scalability issues are the main concerns inhibiting its wide usage in the IIoT ecosystem [10]. Plain vanilla blockchain has a transaction processing capacity limited to a few transactions per second. For example, bitcoin can process seven transactions per second, whereas Ethereum can process about thirty transactions per second [11]. This transactional throughput is unacceptable in IIoT since millions of devices are usually deployed, generating vast volumes of data with strict latency requirements.

One of the main reasons for the limited processing capacity is the kind of consensus algorithm mostly employed by the blockchain. The consensus algorithm used in blockchains requires solving a complex puzzle [12]. Therefore, most existing solutions to this scalability problem use lighter and faster consensus algorithms to achieve high transactional throughput [13–17]. Other existing solutions approach blockchain scalability from a storage perspective and propose storage optimization schemes to reduce the storage requirement of peers [18–28].

However, the ability of blockchain to process transactions also significantly depends on the reliability of the underlying network and its mechanisms for sharing data among the peers and how fast peers can communicate [29,30]. For instance, blockchains use a flooding technique to broadcast blocks called gossip. Although the gossip broadcast is robust for distributed systems, it inevitably leads to duplications and inefficient bandwidth use [31]. Therefore, as more peers join the blockchain network, duplication and bandwidth use increase due to a higher likelihood of overlapping peers chosen for the gossiping process. This will cause performance overheads in an IIoT setting where bandwidth is also constrained [32,33].

Moreover, in real IIoT deployments, different devices will generate data at varying rates. Some devices generate data at slow rates, whereas others may generate data rapidly. The co-existence of many such devices in a network [34] could pose a problem to the blockchain, as the current implementations of gossip could lead to a high degree of duplication. Furthermore, peers exchange data before and after a block is verified.

Furthermore, blockchain peers employ a random neighbor selection mechanism to decide which other peers to exchange data with. However, this may result in the selection

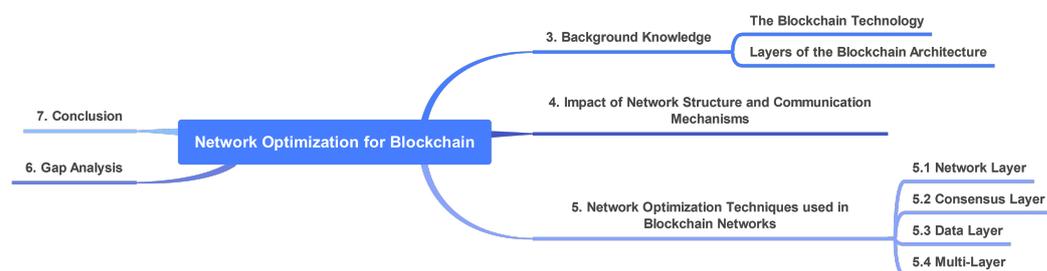
of neighbors with unfavorable communication links. As a result, as the number of peers and transactions grows, the number of forks and invalid transactions grows because newly generated blocks may not reach other peers who may submit competing proposals.

Therefore, there is a need to optimize the blockchain's network layer and improve its ability to scale [35]. Various mitigation measures have been proposed in the literature, and this paper reviews existing optimization methods and analyzes their effectiveness in solving blockchain network issues and their ability to scale.

There are existing reviews on network optimization that present frameworks for non-blockchain networks [36–44]. There are also reviews for blockchain optimization frameworks, which discuss optimized consensus algorithms, thus reducing computational complexity and storage requirements [45–53]. However, only a few works have attempted to review network optimization in conjunction with blockchain [3,54–56], and they are not detailed enough. In this work, we review state-of-the-art works on network optimization for blockchain. The main contributions of this survey are as follows: We

- establish the need for network-based optimizations for blockchain
- introduce background knowledge on the blockchain architecture and discuss the layers
- categorize reviewed works based on the blockchain architecture
- discuss network optimization in blockchain, analyze their effectiveness and ability to scale
- finally, we present some open issues and directions for future work

The rest of this paper is organized as depicted in Figure 1. Section 3 presents background knowledge on blockchain architecture and its layers. Furthermore, Section 4 discusses studies conducted on blockchain's network characteristics, then network optimization is discussed in Section 5. Research gaps are discussed in Section 6, and the paper is finally concluded in Section 7.



**Figure 1.** Structure of Survey.

## 2. Survey Methodology

This survey considers the results from Scopus, IEEE Xplore, ScienceDirect and Springer Link. Querying the databases and analyzing the papers took place from September 2021 to March 2022. The keywords used and the corresponding results returned are summarized in Table 1. Since ScienceDirect allows the exportation of only a hundred references at a time, its results were limited to publications in Engineering, Computer Science and Mathematics. Then, only the top 300 results were considered. SpringerLink also permits the exportation of the top 1000 results; therefore, the results returned were limited separately to journal articles and conference papers, and the top 1000 results were exported for each. Finally, all results from Scopus and IEEE were exported. The number of papers imported by the reference manager is shown in Table 2.

A total of 9811 references were left after removing duplicates. During the title screening process, irrelevant documents such as the cover or front matter of the conference proceedings were excluded. In addition to this, documents whose titles suggested that their full manuscript described an application of blockchain for storage, trust or security in IoT or non-IoT were excluded. Furthermore, sharding-centric publications were excluded. This reduced the number of papers to 242.

**Table 1.** Search Query used and the Total Number of Results Returned from Selected Databases.

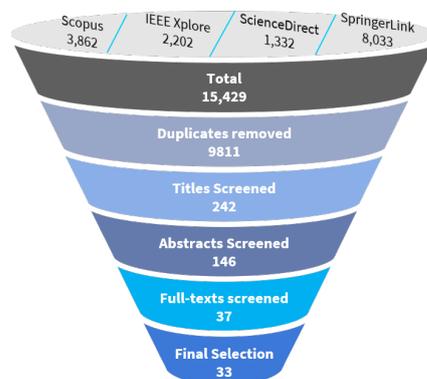
Query	Keywords	Scopus	IEEE Xplore	Science Direct	Springer Link	Total
Query 1	“blockchain” AND (iot OR iiot) AND integration	556	262	3017	5519	9354
Query 2	“blockchain” AND “scalability”	1913	1075	6280	5224	14,492
Query 3	“blockchain” AND (iot OR iiot) AND (optimization OR optimize)	363	224	2746	5134	8467
Query 4	“blockchain” AND “gossip”	50	26	134	478	688
Query 5	“blockchain” AND (peer-to-peer OR p2p OR network OR gossip ) AND (optimization OR optimize)	992	650	4583	8598	14,823
Total		3874	2237	16,760	24,953	47,825

**Table 2.** Total Number of References Imported into the Reference Manager.

Query	Keywords	Scopus	IEEE Xplore	Science Direct	Springer Link	Total
Query 1	“blockchain” AND (iot OR iiot) AND integration	556	262	300	1866	2984
Query 2	“blockchain” AND “scalability”	1913	1049	298	1986	5246
Query 3	“blockchain” AND (iot OR iiot ) AND (optimization OR optimize)	363	224	300	1992	2879
Query 4	“blockchain” AND “gossip”	49	26	134	195	404
Query 5	“blockchain” AND (peer-to-peer OR p2p OR network OR gossip ) AND (optimization OR optimize)	981	641	300	1994	3916
Total		3862	2202	1332	8033	15,429

The abstracts of the remaining papers were screened to identify works relevant to blockchain’s network structure or communication. Therefore, during this phase, network optimization works that employed blockchain as a security enabler were excluded. As a result, only 146 remained for full-text screening.

Finally, the full-text screening process selected publications focusing on the communication complexity of consensus algorithms, improving the P2P structure or reducing the total bandwidth consumed by blockchain when operational. This further reduced the number of papers selected for this survey to 33. A summary of the screening process is shown in Figure 2. Tables 3 and 4 summarize the exclusion and inclusion criteria, respectively.



**Figure 2.** Survey Methodology.

**Table 3.** Exclusion Criteria.

No.	Criteria
1	The article is not published in English.
2	The article uses blockchain to secure IoT applications.
3	The article uses blockchain to secure a non-blockchain network optimization.
4	The article presents a sharding-based blockchain optimization.

**Table 4.** Inclusion Criteria.

No.	Criteria
1	The study emphasizes the communication complexity of consensus algorithms.
2	The paper proposes optimization for protocols used to share transactions or blocks.
3	The paper presents an optimized neighbor selection or peer-to-peer topology.
4	The paper proposes compression schemes for blockchain data before transmission.

### 3. Background Knowledge

This section briefly introduces blockchain technology, its architecture and discusses the layers it comprises. The section then discusses blockchain's P2P structure and the gossip dissemination protocol.

Blockchain is a peer-to-peer distributed ledger that stores transactions in a chain of blocks. It was initially developed as the technology powering the Bitcoin cryptocurrency [53]. It aims to remove the need for a trusted third party who will validate transactions [57,58]. Therefore, transactions are timestamped and verified by all participating nodes and then added as a block in the main chain using cryptographic tools. As shown in Figure 3, the blockchain architecture consists of the control, consensus, data and network layers.

1. **Control Layer:** The control layer serves as an interface between applications and the ledger. These applications include finance, data storage, voting, securing IoT, logistics and supply chain tracking [10,51]. They interact with the ledger by invoking smart contracts to trigger transactions. Smart contracts are programmable scripts that interact with the ledger by reading or writing data to the ledger when invoked. On-chain and off-chain are the primary processing models at the control layer.
2. **Consensus Layer:** The algorithms blockchain peers use to agree on the validity of blocks and transactions exists at the consensus layer. Peers in public blockchains use Proof of X (X => Work, Stake and many more) algorithms to reach a consensus [53]. All peers in public blockchains are eligible to participate in the consensus process. However, in private blockchains, validating blocks and transactions are performed mainly by different peers with specific roles. In Hyperledger Fabric (HLF), these roles include orderers, endorsers, validators and leaders [31]. New transactions are first sent to peers who can endorse transactions by executing smart contracts. After endorsement, the transaction is sent to the leader node and maintains a connection to the ordering nodes. Next, ordering nodes add the transactions to a block and send them back to the leader node to be broadcasted in the network. Finally, every peer will have to validate the block before committing it to its local ledger.
3. **Data Layer:** The data layer defines the structure of transactions, blocks and the cryptographic mechanisms that link them together. A number of transactions are combined together into a block. The size of a block depends on an explicitly defined size or on a time interval at which blocks must be produced [59].
4. **The Network Layer:** The network layer mainly consists of the network structure and the communication mechanisms. The network structure is responsible for establishing and managing the peer-to-peer network structure, while the distribution of transactions or blocks to peers in the network is handled by the communication mechanisms using gossip. Forming the P2P network requires new peers to randomly select some existing peers as their neighbors (to exchange blockchain data). Upon joining the

network, a new peer connects to other peers whose addresses are hard-coded into its configurations. Then, the new peer requests the addresses of the existing peer’s neighbors and finally selects some peers randomly as neighbors to gossip with. Peers utilize the gossip protocol in the network to distribute new blocks or transactions [60]. The gossiping dissemination occurs in rounds and is prone to duplicate transmissions. As shown in Figure 4, Node L shares a newly generated block with its neighbours (Peer 1, 2 and 3). Peer 1 also shares with its neighbours (4 5 7), Peer 2 shares with (6 7 8) and Peer 3 then shares it with (7 9). After this round, it can be realized that Node 7 received the block three times.

Control Layer	Processing models	Contracts	
Consensus layer	PoX	Raft	PBFT
Data layer	Blocks	Transactions	
Network layer	Network Structure	Communication Mechanisms	

Figure 3. The Blockchain Architecture.

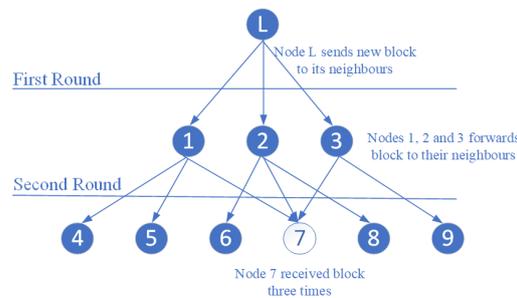


Figure 4. The Gossip Dissemination Protocol.

#### 4. The Impact of Network Structure and Communication Mechanisms on Performance

This section discusses various studies on the network characteristics of blockchain networks. These studies present findings on redundant traffic generated by blockchain, causes and impact of propagation delay and the communication complexity of consensus algorithms.

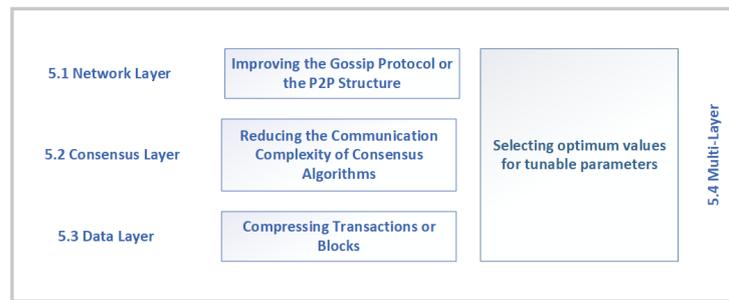
Various studies have been conducted to investigate the characteristics of blockchain’s P2P structure, gossip mechanism and their impact on performance. Kiffer et al. [61] studied the Ethereum blockchain and discovered that there was a high churn rate in its P2P. Hence, only a few peers propagated new blocks since churning nodes are usually unable to reconstruct blocks in time. Furthermore, the location of a peer determines how fast it receives a new block. The increased block propagation time due to physical distance can lead to higher occurrences of forks and invalid transactions [29]. It is, therefore, recommended that during the neighbor selection stage, a peer selects neighbors that are physically close to it [59,62,63]. In blockchain’s P2P network, peers use the gossip protocol to share blocks and transactions. Since the gossip protocol causes duplicate transmissions [31], other studies analyzed the blockchain’s traffic to observe the degree of duplication, the impact of network size and topology on the blockchain’s traffic. For example, the authors in [64,65] revealed that traffic redundancy increases with an increase in network size. We can attribute this to the fact that there is an increased probability of overlapping neighbors for different peers. A summary of relevant studies and their findings are presented in Table 5.

**Table 5.** Studies on Blockchain’s Network Characteristics 1 = Mathematical Analysis and Simulations, 2 = Measurement on Real Network, 3 = Varied Network Size, 4 = Varied Block Size and Interval, 5 = Analyzed Transactional Throughput, 6 = Analyzed Latency, 7 = Analyzed P2P Structure, 8 = Analyzed Traffic Load.

Ref.	1	2	3	4	5	6	7	8	Blockchain	Study Objectives	Key Findings
[64]	✓		✓					✓	Bitcoin	Impact of network structure on traffic redundancy	Redundancy increases with number of peers
[61]		✓					✓		Ethereum	Investigate P2P network structure, block propagation mechanisms and the evolution of network over time	Peer location determines block delivery latency High churn in P2P network Only a few nodes propagate new blocks
[47]		✓			✓			✓	EOSIO Tezos XRP Ledger	Compare transactional throughput to traffic generated	About 76% of traffic used for consensus
[59]	✓	✓		✓	✓	✓			HLF	Study impact of HLF’s parameters on latency	Block generation latency increases with transaction arrival rate Increase in number of peer organizations increases latency
[32]	✓		✓	✓				✓		Investigate feasibility of a wireless blockchain network	PBFT has the highest communication complexity
[66]	✓	✓		✓		✓			Bitcoin	Impact of miner’s geographical distribution	Propagation delay increases with an increase in block size Higher propagation delay leads to higher occurrence of forks
[29]	✓		✓			✓	✓		Bitcoin	Impact of P2P structure and gossip on latency	Forking rates reduces with a reduction in block propagation time
[67]		✓						✓		Investigate peer communication and dynamicity of peers	The proportion of always-online peers determine variability of topology
[68]		✓			✓				Bitcoin	Causes of delay in bitcoin	Physical network characteristics impact block delivery time Internet usage at peak times reduces delivery
[62]		✓	✓	✓	✓	✓			HLF	Impact of HLF parameters	Transaction latency has linear relationship with network latency Bandwidth requirements of leader node is proportional to number of nodes
[63]		✓	✓		✓				HLF	Investigate the impact of delay on HLF performance	Network delay causes inconsistencies and block synchronization problems
[30]		✓						✓	Bitcoin	Characteristics of Bitcoin’s P2P network	High churn rate Churning leads to increased block propagation time

### 5. Network Optimization Techniques Used in Blockchain Networks

In this section, we will introduce the concept of network optimization as used in blockchain networks and review various optimization frameworks proposed in the literature. The reviewed works have been categorized based on the layer on which they have been implemented. These layers include the consensus layer, the network layer and data layer, as shown in Figure 5.



**Figure 5.** Categorization of Network Optimization Frameworks in Blockchain.

In blockchain networks, network optimization refers to any technique that enhances blockchain’s performance by speeding up the communication between peers, reducing bandwidth consumption or deploying techniques that can help sustain acceptable performance for large-scale applications. Existing approaches to network optimization in blockchain improve the P2P topology, minimize duplicates caused by gossiping or by reducing the size of data exchanged between peers.

Optimization works at the consensus layer include works that improve their communication pattern and hence, making them communicate in a small amount of time, reduce the possibility of congestion or consume less bandwidth. Works at the network layer either improve the gossip algorithm or the peer-to-peer network structure. Data layer approaches are primarily about using schemes to minimize the size of the data that are exchanged on the blockchain.

5.1. Network Layer

Relevant works at the network layer enhance either the gossip algorithm or the P2P structure. Research works that optimize the gossip algorithm do so by reducing the duplicates associated with it or by increasing the speed of block dissemination. The P2P-based approaches replace the Random Neighbor Selection of blockchain with a neighbor selection scheme based on propagation delay encountered when communicating with the peer [69]. Propagation delay or proximity also serves as the basis of P2P approaches, which use clustering. Most of the P2P-based approaches are distributed, but a few of them employ Software Defined Networking (SDN) concepts in managing the P2P topology. A summary of these approaches is presented in Table 6.

**Table 6.** Comparison of Network Layer Techniques. PC = Public, PR = Private, BI = Blockchain-IoT, CT = Centralized Topology, DT = Distributed Topology, HY = Semi-Distributed Topology, DY = Dynamic Topology, LY = Latency, BW = Bandwidth, Bandwidth = Reduction in Total Network Bandwidth due to Minimization of Duplicates or Reducing Overhead Associated with Distributed P2P Topology Management.

Ref.	PC	PR	BI	CT	DT	HY	DY	LY	BW	Main Contributions	Limitations
[70]		✓							33%	Gossip Algorithm with reduced degree of duplication	Results were collected using only nine peers
[71]									33%	DHT for blockchain P2P Compression scheme to reduce size of transmitted data	

Table 6. Cont.

Ref.	PC	PR	BI	CT	DT	HY	DY	LY	BW	Main Contributions	Limitations
[31]		✓						>100%	40%	Gossip algorithm with increased speed and reduced bandwidth consumption	Could still be bandwidth inefficient at large scales due to block duplication
[72,73]										Gossip algorithm with increased speed and reduced bandwidth consumption	
[74]	✓		✓						88%	Mathematical model to assess impact of wireless connectivity on block synchronization and smart contract execution	
[75]	✓								0.88%	Probability-based gossiping	Excessive and repeated send rate of INV messages can be an overhead
[69]					✓				12.5%	Neighbor selection algorithm based on proximity	Susceptible to crowding and over reliance on one node
[76]									75%	Weight-based neighbor selection	
[77]					✓					Tree-based topology for broadcast on the blockchain	No results presented on the effectiveness of the work
[78]	✓								7%	DHT P2P Topology	
[79]					✓					Intercrossing-net P2P topology for block broadcast	Overhead due to constant reassembly of blocks
[80]						✓	✓			Cluster-based P2P topology	Susceptible to crowding and over reliance on one node
[81]										Clustered P2P topology	Susceptible to eclipse attacks
[82]				✓						Semi-distributed P2P topology management	Susceptible to crowding and over reliance on one node
[83]	✓		✓	✓		✓	✓		>100%	SDN-inspired topology management. Mathematical modelling of P2P topology	Increase in topology calculation time with increase in the number of peers.
[84]	✓						✓	>100%	>100%	Semi-distributed topology management	Increase in topology calculation time with network size. Overhead incurred by frequent leader election

### 5.1.1. Gossip Optimization

Gossip-based optimization frameworks are discussed in this section. These strategies try to limit the amount of duplication introduced by the gossip algorithm or the amount of time spent gossiping. He et al. [70] proposed an algorithm accompanied by a modification of the structure of blocks used by peers when gossiping. The modification was introduced to minimize the quantity of duplication caused by the gossip protocol. They introduced a list to keep track of all nodes previously selected for gossiping. Therefore, when a peer receives a block, it does not send the blocks to their neighbors who are already on the history list. Though their approach was effective at minimizing bandwidth wasted on duplicates by about 33%, it was tested on a network consisting of only nine nodes. Their approach may not scale well with the number of peers if the network receives huge volumes of

transactions due to the overhead incurred on checking a possibly long history list for every transaction. Therefore, the effectiveness of this approach should be verified by increasing the number of nodes.

The authors of [71] used a method similar to He et al., except that it also includes the path in the network that the block has traversed. A Distributed Hash Table (DHT) was used for the P2P network. Their approach could reduce bandwidth wastage caused by duplicates by about 33% and also scales well with peers since, for a large network size, their approach still had significant bandwidth savings. Furthermore, the results presented suggested that it had a negligible effect on block verification latency.

Berendea et al. [31] and Shaleva et al. [72,73] altered the steps and communication pattern of the gossip algorithm used by the Hyperledger Framework and Neo Blockchain, respectively, to reduce its bandwidth usage and also to reduce the amount of time it uses. In their approach, whenever a peer receives a new block, it constantly shares it with a new set of uniformly and randomly selected neighbors until the dissemination counter is exhausted. Since this approach increases the probability that all peers get the block in a few gossip rounds, the pull component of gossip (peers requesting blocks they lack) was removed to save bandwidth. The authors minimized the total network bandwidth by about 40%. Their approach realized an improvement in transactional throughput due to its speed and reduction in the number of invalid transactions by 36%. At large scale, their approach might be susceptible to a decrease in bandwidth efficiency due to duplicate transmissions. Therefore, this method can further be improved by minimizing duplicate transmissions.

Vu et al. [75] used the INV message and GETDATA response used in bitcoin to calculate a probability that will influence which neighbors will likely receive the new block. A peer sends a number of inventory (INV) messages and measures the number of responses received. The ratio of sent and received forms the probability that is used to decide which neighbor gets the new block. Since all peers will have a variable number of neighbors, peers with more neighbors will most likely respond to INV messages from a lot of their neighbors and will, therefore, respond poorly to each and every neighbor. Peers with few neighbors, however, will respond to INV messages from few peers and will, therefore, respond better, hence gaining a higher probability. Furthermore, a peer with more neighbors will most likely receive a new block from some neighbors, and hence, peers with few neighbors were given a higher priority by the authors. The results presented indicate that this approach did not yield any appreciable increase in transactional throughput. However, their approach yielded about a 15% reduction in the number of messages transmitted by a peer. When compared to the default gossip in bitcoin, their approach realized about a 0.88% reduction in duplicates. Furthermore, probability calculations are not discarded but kept for future transmissions, as well as network size.

### 5.1.2. P2P Topology Optimization

RNS is mainly replaced by more informed selection criteria in the P2P topology optimization. This modification helps to improve blockchain metrics, such as average finality time. The average finality time measures how long it takes for most peers to get a newly created block. Because a long time equals a lower transactional throughput, these strategies choose neighbors with the shortest communication delay.

The authors in [69,76] used a score-based algorithm to influence how the underlying peer-to-peer of the bitcoin network is formed. Peers with the smallest propagation delay were given a high score and also selected as neighbors. After receiving ten blocks, each peer updates its neighbors by randomly selecting new neighbors and only adding if it has high scores as determined by the algorithm. By shortening block propagation time, Aoki et al. [69] realized a reduction in latency by about 12.5%, whereas Santiago et al. [76] reduced latency by 75%.

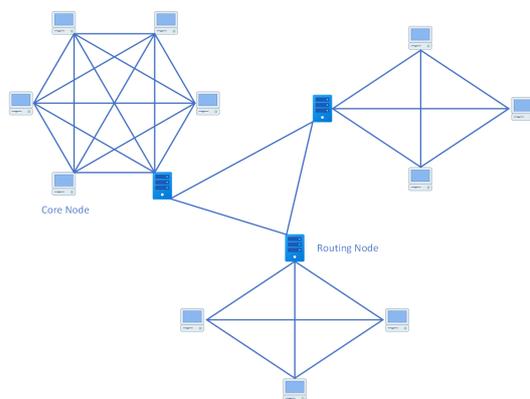
At a higher scale, however, this could lead to too much reliance on a peer when it has the shortest propagation times and can consequently degrade the node's performance. In high-throughput networks, the periodic peer update process could increase the network's

overall bandwidth. This approach can be made more efficient by putting in measures to prevent any peer from having too many neighbors (crowding). Bandwidth-efficient peer update techniques could also be introduced to minimize excessive consumption of network resources at high scale.

Kan et al. [77] developed a scheme to improve data broadcasting in a blockchain network. Their approach organized the peer-to-peer network into a tree structure. When a node receives a new block from its parent node, it broadcasts it to its children node. Since there is no interconnection between children nodes of any two parent nodes, their method ensured that duplicate transmissions did not occur. This tree structure can, therefore, simultaneously increase bandwidth efficiency, transactional throughput and also scales well with network size. Using a DHT, Kaneko et al. [78] reduced redundancy by 7%.

Yang et al. [79] proposed the Ari protocol as a solution that modeled the peer-to-peer network as an inter-crossing net instead of other approaches that employed a unidirectional tree-based P2P topology [77]. Blocks are first split into equal parts, and a serial number is assigned to each piece before they are broadcast into the network. The transmitted blocks take the shortest path in the inter-crossing net to their destination peers; this improved network-wide latency by about 50%. The splitting technique applied also helped achieve some savings in bandwidth. However, when it grows, the split-and-scatter technique can bottleneck the blockchain's performance since nodes may have to wait a longer time to receive all parts of blocks that have been scattered in the network. It will also make inefficient use of bandwidth since more peers will have to frequently request block parts from different peers.

BlockP2P by Hao et al. [80] is a technique proposed to improve transactional throughput by minimizing latency in the blockchain network. As shown in Figure 6, the main idea behind this approach is to group blockchain peers into a cluster based on the geographical location. This technique leads to a cluster of small diameter and, therefore, reduces propagation time within the cluster. Furthermore, since the network consists of several clusters, the authors designated some nodes, called routing nodes, which connect to routing nodes of other clusters to ensure full connectivity. The authors gained a net increase in transactional throughput due to the reduced latency of about 90%. When a small network size is considered, the clustering approach has better bandwidth efficiency than Random Neighbor Selection. However, when the network grows, crowding can occur in a cluster and reduce intra-cluster efficiency. Although Al-Musharaf et al. [81] also used a similar approach, their work was susceptible to network partitioning and eclipse attacks since a star topology was used within a cluster.



**Figure 6.** Clustering of Peers by Hao et al. [80].

Huang et al. [82] introduced the concept of “iTracker”. The iTracker is an entity owned and operated by the ISPs that provide internet to peers in the blockchain network. When a node wants to connect to another node in a different cluster, the iTracker of the originating node connects to the iTracker in charge of the destination cluster. The iTracker returns the node with the smallest propagation time to the originating peer. Since the iTracker is

responsible for topology, there is better bandwidth usage due to reduced overhead caused by fully distributed topology management. Similar to other techniques that employ this clustering approach, it also offers improved transactional throughput and could also suffer reduced performance at a large scale when intra-cluster crowding is not mitigated.

Deshpande et al. [83] used a centralized approach, as shown in Figure 7, to reduce the overhead incurred by the distributed network management of a vanilla blockchain. Dedicated servers generated the P2P topology and assigned neighbors to each peer. This method provided a flexible way to manage the network and make it possible to tune the topology in real time. The topology generation used both clustering and Random Neighbor Selection, but, unlike other clustering-based approaches, they introduced a constraint to limit intra-cluster crowding. The SDN-inspired topology management resulted in significant savings in bandwidth when compared to traffic generated by topology and network management in distributed approaches. Even though the authors did not evaluate their work in terms of blockchain-related metrics, this approach has the potential to improve transactional throughput. This is because nodes have fewer responsibilities and can, therefore, dedicate all their resources to process more transactions. Moreover, the topology can also be tuned to minimize latency and speed up transaction confirmation. It is, however, noteworthy that topology computation time increases with an increase in network size. Therefore, research consideration should be given to multiple controllers to help the network scale up.

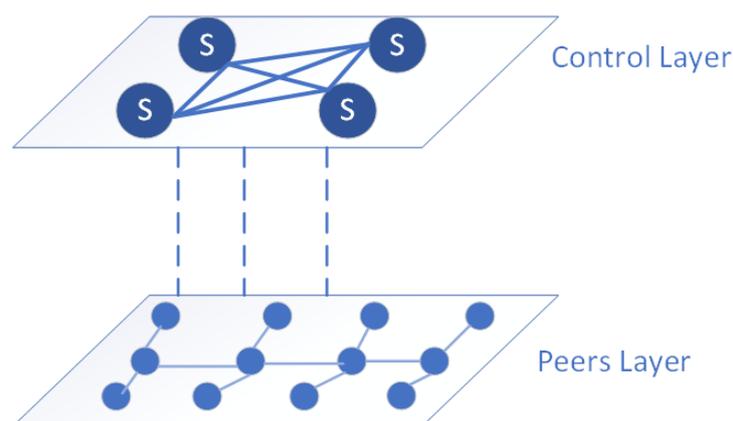
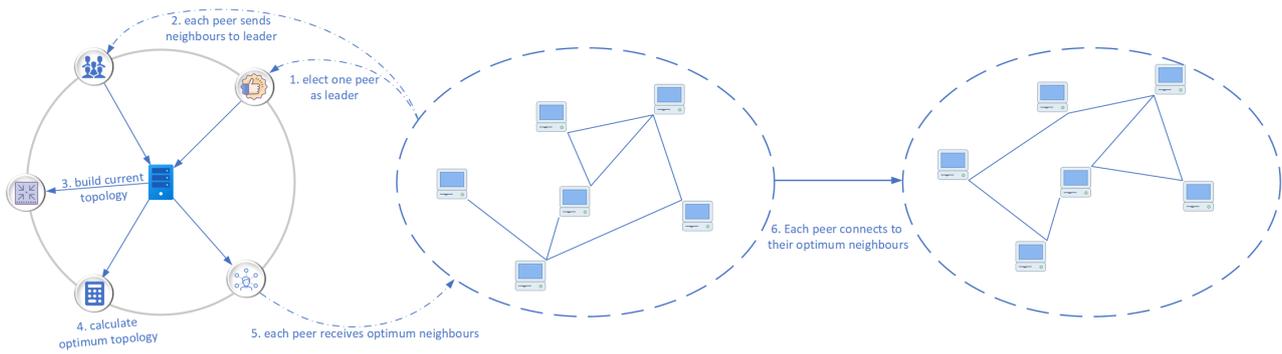


Figure 7. SDN-Inspired Topology Management by Deshpande et al. [83].

Baniata et al. [84] used a hybrid architecture to manage the P2P network of blockchain, as shown in Figure 8. One peer is voted into a leadership role to take charge of topology management. Normal peers send their neighbor list to the leader. The leader then uses the neighbor lists to calculate a Minimum Spanning Tree (MST) and sends back the optimum neighbors to each peer. Propagation delay is the cost used to calculate the MST problem. Therefore, each peer receives neighbors who offer the best propagation delay and ultimately leads to an improvement in transactional throughput. Similar to Deshpande et al., decoupling the P2P management leads to decent savings in bandwidth. However, when a new leader is elected, it will have to request the neighbor list of all peers again and compute the topology. Since this is an inefficient use of bandwidth, measures can be put in place to ensure that topology and membership will not be lost if the current leader becomes unavailable. Moreover, the MST calculation time increases with increasing peer membership. Therefore, multi-leader scenarios should be considered to reduce topology calculation time as the network scales up.



**Figure 8.** Semi-Distributed Topology Management Scheme used by Baniata et al. [84].

5.2. Consensus Layer

The reviewed works at this layer focused on reducing latency by verifying blocks selectively, reducing the number of communication steps required to reach consensus. Furthermore, other techniques replace the single leader in private blockchain consensus with two or more leaders. A summary of these approaches has been presented in Table 7.

**Table 7.** Comparison of Consensus Layer Techniques. PC = Public Blockchain, PR = Private Blockchain, BI = Blockchain-IoT, LY = Latency, TT = Transactional Throughput, BW = Bandwidth, Bandwidth = Reduction in Total Network Bandwidth due to the Minimization of Duplicates or Reducing Overhead Caused by Distributed P2P Topology Management.

Ref.	PC	PR	BI	LY	TT	BW	Main Contributions	Limitations
[14]				10%			Probabilistic verification algorithm to determine when to validate blocks	Trustworthiness of nodes not checked
[85]	✓			>100%			Trust-based verification Guarantor selection algorithm	
[86]							Reputation-based block verification	
[87]		✓			>100%		Byzantine Agreement consensus algorithm with two rounds of communication	Performance degrades with increase in peers.
[88]		✓					HSBFT consensus algorithm	No evaluation performed for the technique
[89]		✓				30%	Multi-Raft consensus algorithm	
[28]		✓	✓	>100%	68.75%		SPBFT consensus algorithm	Evaluated against a network with a few nodes
[90]		✓		>100%			Multi-leader PBFT algorithm	
[91]	✓		✓				A closed-loop algorithm to determine the optimum blocks to mine	Higher occurrences of forking at high scale

5.2.1. Selective Verification

Li et al. [14] proposed to reduce the average network latency by reducing the amount of time required to verify blocks using a probabilistic verification method. They introduced a validation degree metric, which other nodes used to determine whether or not to validate a block. According to the metric, if a block does not require validation, then it can be transmitted immediately to other peers. This probabilistic verification leads to an increase in transactional throughput and also has good scalability. However, since not all peers will validate transactions, the authors should have factored in the trustworthiness of peers.

However, unlike Li et al., the authors in [85,86] checked and used the trustworthiness or reputation of peers to decide whether or not to verify new blocks. Their approaches have good scalability, smaller latency and increased transactional rate when compared to the bitcoin network.

### 5.2.2. Communication Complexity

The authors in this category proposed consensus algorithms with a reduced number of communications required. For example, Locher et al. [87] presented a Byzantine Agreement-based consensus algorithm that requires  $1/4$  nodes and two rounds of communication to reach a consensus. The reduced number of steps should translate to a smaller latency and bandwidth consumption; however, the results presented suggest that effective throughput was reduced when the number of peers increased, and, hence, it is not scalable.

Jiang et al. [88] proposed a consensus algorithm called High-Performance and Scalable Byzantine Fault Tolerance (HSBFT). Although no evaluation was performed for the proposed technique, it may have a reduced bandwidth usage since communication complexity was reduced from  $O(n^2)$  to  $O(n)$ .

### 5.2.3. Multi-Leader

Cao et al. [89] analyzed raft vs. multi-raft for network latency and leader's traffic. It was observed that the traffic generated in raft was greater than multi-raft. This improvement is only visible in high-throughput networks with balanced nodes across multiple leaders. About a 30% reduction in total network traffic was due to the balancing. The overhead on the leader is due to the total number of simultaneous connections and log synchronization. Multi-raft also has a lower cpu at a higher scale and, therefore, is scalable.

The authors of [28] proposed, and enhanced the Practical Byzantine Fault Tolerant (PBFT) [92] algorithm called "score-voting based byzantine fault tolerance". In this scheme, the authors used a reward and punishment scheme to select nodes which can participate in the consensus process. The reduced bandwidth is due to the minimization of broadcasts during the consensus process. Consensus nodes do not broadcast but unicast to the primary node, thus, reducing communication complexity. Their technique proved superior to (PBFT), Window-based BFT (WBFT) [93] and Reputation-based BFT (RBFT) [94] in terms of the number of communications required, delay and transactional throughput. They realized an increase in transactional throughput from 160 tps to 270 tps and a reduced latency of 100 ms when compared to PBFT, which had a latency of 325ms.

Jalalzai et al. [90] voted a group of nodes, called a committee, to undergo the consensus process. Members of the committee were randomly selected to ensure fairness. The bandwidth saving and the scalability of this approach depends on the size of the committee. For a network consisting of 130 nodes, their approach has a latency of 5 s when compared to PBFT, which achieved a latency of 15 s.

### 5.2.4. Congestion

Casado-Vara et al. [91] designed a framework for blockchain-based IoT systems. They artificially adjusted the PoW algorithm's difficulty as a means to limit congestion in the network. They used a queuing system between the side chains, and the main blockchain decides the optimal number of blocks that must be admitted and mined at any particular time in the main blockchain. By reducing PoW's difficulty, an increase in transactional throughput was realized. However, this work suffers from bandwidth inefficiencies of a traditional blockchain. Since this approach uses a reduced PoW difficulty to prevent queue saturation, there will be a higher occurrence of forking rates at high scales.

## 5.3. Data Layer

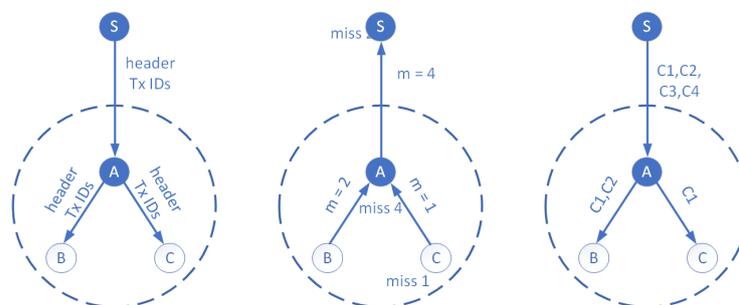
Optimizations at the data layer primarily apply a form of compression to blockchain data or traffic to reduce the bandwidth consumed when blocks or transactions are shared in the network. A summary of these approaches has been presented in Table 8.

**Table 8.** Comparison of Data Layer Techniques. PC = Public Blockchain, PR = Private Blockchain, BI = Blockchain-IoT, BW = Bandwidth. Bandwidth = Reduction in Total Network Bandwidth due to Minimization of Duplicates or Reducing Overhead Caused by Distributed P2P Topology Management.

Ref.	PC	PR	BI	BW	Main Contributions	Limitations
[95]		✓		1.87%	XOR-based compression of blockchain traffic	Requires peers to have multiple network interfaces
[96]	✓			82%	Erasure coding technique to compress transactions before transmission	Only effective when dealing with a few transactions
[97]	✓				Erasure coding technique to compress compact blocks	
[98]		✓	✓	>100%	Network coding technique to compress compact blocks	
[99]	✓				A block structure with transactions replaced by their hashes	Requires peers to have a similar transaction pool. Will consume higher bandwidth if dissimilarity is high

Ahn et al. [95] used a simple packet aggregation scheme to compress the traffic generated in the network. Their approach required a blockchain node in the network to have multiple network interfaces. They then used an XOR operation to join the blockchain traffic with the same destination. The authors reported that their technique reduced the PBFT processing time by 23%; a corresponding increase in transactional throughput was also realized. The results presented also showed a reduction in bandwidth by about 1.87%. These improvements may not be achieved in a real network (especially at a large scale) since it requires all peers to have multiple network interfaces.

Jin et al. [96] devised an erasure-coding technique to compress blocks before transmission. Their technique works by first clustering the blockchain nodes; each cluster elects a cluster head, as shown in Figure 9. If, for instance, a node has a new block to share, it transmits the block’s headers and id’s of its transactions. The cluster head then broadcasts this received data in its cluster. Next, cluster members send all the transactions they lack; their leader relays them to the originating peer. The originating peer then encodes all requested transactions and sends them back to each cluster member to recover the transactions they need. The clustering of nodes leads to an improved throughput, whereas the coding technique greatly reduced bandwidth by about 82%. However, the coding technique is only effective when dealing with a few transactions. It will, therefore, not scale well with an increase in transactions. Zhang et al. [97] proposed a similar approach, although not in a clustered network.



**Figure 9.** Clustering Technique used by Jin et al. [96].

Cebe et al. [98] also used a coding technique to minimize the size of blocks before it is transmitted to other peers in the network. The authors, first of all, split a block into chunks. Then, a group of chunks, called a generation, is combined into one packet and transmitted to other peers. In their experimental setup, consisting of five wireless IoT peers, the total traffic transmitted per node was reduced from about 51 kB to about 9 kB. However, when

the number of transactions or peers increases, the task of decoding and reassembling blocks could burden peers.

Zhao et al. [99] proposed lightblock, a modified and lighter block structure. In their approach, peers maintain a pool of transactions, and, hence, when any other peer receives a hash of a transaction, it should be able to recover the full transaction from the pool. Therefore, the authors replaced all transactions in a block with its hash, hence making it smaller and occupying a smaller bandwidth when transmitted. In the unlikely case that all peers have similar transaction pools, their approach will drastically reduce the requirements to transmit blocks. However, when there are dissimilarities in transaction pools, this approach is likely to consume more bandwidth than the normal blockchain operation and will, therefore, not scale well with peers or transactional load.

#### 5.4. Multi-Layer

Some blockchain frameworks, such as Hyperledger Fabric, allow administrators to manually configure some blockchain parameters, such as block size and block generation interval. Selecting optimum values for these parameters is an area that researchers have exploited to improve blockchain performance. Hang et al. [100] studied the impact of these tunable parameters on the blockchain network and presented a guideline for building an optimum blockchain network. Even though this guideline promises to build a blockchain with an improved throughput, it may not scale well because the guidelines were provided from a network with less than ten peers. Moreover, a configuration that is optimum for a network with tens of peers may not be optimum for a network with hundreds of peers. Table 9 summarizes the relevant works in this category.

**Table 9.** Comparison of Multi-Layer Techniques. PC = Public Blockchain, PR = Private Blockchain, BI = Blockchain-IoT, CL = Consensus Layer, DL = Data Layer, LY = Latency, TT = Transactional Throughput.

Ref.	PC	PR	BI	CL	DL	LY	TT	Main Contributions	Limitations
[100]		✓		✓	✓	36.7%	20%	Methodology to construct optimum network based on empirical data	May not scale well with peers
[101]		✓	✓	✓	✓		11.9%	A DRL algorithm to select optimum values for tunable parameters of the blockchain	Single agent may not scale well with an increase in peers

Liu et al. [101] used a Deep Reinforcement Learning (DRL) technique to proactively select a consensus algorithm, block size, interval and producers. They dynamically adjusted the consensus algorithm used by the blockchain network, block-producing peers, the rate at which their system generated blocks and the size of a block. The state-space of their DRL approach includes the size of a transaction, computational strength of a node and data transmission rate. The action taken by the agent is primarily changing either one or multiple parameters (consensus algorithm, block size, producers and interval). As a result, the agent interacts with the environment to increase transactional throughput, finality and more. The usage of consensus algorithms with a smaller traffic footprint led to a net reduction in bandwidth. Even though this work is effective throughput and bandwidth, further investigation should be conducted into multi-agent approaches since a single agent will not scale very well with network growth. Moreover, constantly interacting with the network can hamper its performance. It is, therefore, necessary to investigate how to achieve a trade-off between maximized performance and minimized interaction.

## 6. Gap Analysis

Despite all the efforts that have been put into making blockchain scalable, there is still room for further research. This section discusses some identified gaps and directions for possible future work.

1. Some blockchain frameworks allow network administrators to configure block size, interval and other parameters manually. Therefore, to construct an optimum blockchain network with improved transactional throughput, some researchers conducted extensive measurements to determine the impact of parameters on the blockchain's ability to process transactions. After analyzing the experimental results, guidelines on selecting the best value for each parameter were presented. However, since these guidelines were extracted from a setup consisting of a few nodes, the approach may not suit the blockchain when it scales up with the number of peers. Therefore, researchers could investigate optimization algorithms that will take the network's size and other relevant parameters as input and automatically determine and apply the optimum value for all the configurable blockchain parameters.
2. Deep Reinforcement Learning has been applied to automatically select tunable blockchain parameters, such as consensus algorithm and block size. In principle, a DRL agent persistently interacts with its environment and takes actions that will converge to an optimal state. However, persistently interacting with the blockchain network will impede its normal operation and subsequently bottleneck its transactional throughput. Therefore, future work could research how to achieve a trade-off between maximized performance and minimized interaction.
3. Solutions proposed as a better alternative to the random neighbor selection used in current blockchain implementations suggests selecting neighbors based on proximity or latency of communication. Hence, each peer selects neighbors offering the least communication latency during the neighbor selection stage. Furthermore, if many peers consider a specific peer as having low-latency communication, they will all select it. Consequently, it will lead to that peer having too many neighbors and, consequently, overloading it. Therefore, further research could examine neighbor selection strategies and limit the number of neighbors based on network size, a peer's computing and network resources.
4. Blockchain is a distributed technology, hence its P2P network management is also distributed, thus, all peers are responsible for setting up and managing the P2P network. However, this approach has significant network management overhead and does not easily lend itself to a dynamic reconfiguration. Therefore, to make P2P management more flexible and minimize its management overhead, researchers could consider developing intelligent semi-distributed techniques [84] to manage the P2P of blockchain networks. Thus, some peers in the blockchain are assigned special responsibilities. For example, they are tasked with calculating the optimum P2P topology and selecting suitable neighbors for every peer in the network.
5. Using semi-distributed strategies for P2P network management will be computationally intensive when the blockchain peers increases. Hence when only a single peer is tasked with the P2P topology calculation, it would require more time to achieve the optimum topology. This would affect the real-time requirements of the network. Therefore, future research could investigate the optimum number of peers required for a network of a given size and share P2P topology calculation and management across the selected peers.

## 7. Conclusions

As a summary, this paper introduces the blockchain architecture and its layers and reviews the existing optimization work in the literature. The proposed optimization frameworks are implemented at consensus, data, network or a combination of layers. As a result, the blockchain's communication complexity, bandwidth or latency was reduced. Although these proposals are reasonable steps toward realizing highly scalable and performant blockchain systems suitable for IIoT integration, they are fraught with inherent flaws. For example, optimizations based on neighbor selection may lead to over-dependence on a peer with the least propagation time. Putting in measures to prevent possible overcrowding will ensure that no peer is overloaded. Works that reduce blockchain communication

complexity (Section 5.2.2) result in a reduced bandwidth. However, better savings can be achieved when it is coupled with other bandwidth-saving approaches, such as duplicates minimization (Section 4) and coding techniques (Section 5.3), before transmission. Furthermore, better latencies may be achieved when communication complexity reduction-based techniques are combined with more informed neighbor selection. Therefore, more research is needed in this area to explore every uncharted territory.

**Author Contributions:** Conceptualization, R.A., J.D.G., J.O.A. and K.O.-B.O.A.; methodology, R.A., A.S. and E.T.T.; formal analysis, R.A., D.W. and A.S.; investigation, R.A., J.D.G. and E.T.T.; writing—original draft preparation, R.A.; writing—review and editing, A.S., H.N.-M., A.S.A., E.K., J.D.G. and E.T.T.; visualization, R.A.; supervision, J.D.G. and E.T.T.; project administration, E.T.T. and A.S.; funding acquisition, E.T.T. and A.S. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received funding from the German Federal Ministry of Research and Education (Bundesministerium für Bildung und Forschung, BMBF) and the German Academic Exchange Service (Deutscher Akademischer Austauschdienst, DAAD). This paper was written as part of the Distributed IoT-Platforms for Safe Food Production in Education, Research and Industry (Dipper) project, which is co-financed by the BMBF (Förderkennzeichen: 01DG21017) and DAAD (Projekt-ID: 57557211).

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** No new data were created or analyzed in this study. Data sharing is not applicable to this article.

**Conflicts of Interest:** The authors declare no conflict of interest.

## Abbreviations

The following abbreviations are used in this manuscript:

RNS	Random Neighbor Selection
P2P	Peer-to-Peer
IoT	Internet of Things
IIoT	Industrial IoT
QoS	Quality of Service
HLF	Hyperledger Fabric
SDN	Software Defined Networking
DHT	Distributed Hash Tables
INV	Inventory
MST	Minimum Spanning Tree
PBFT	Practical Byzantine Fault Tolerance
DRL	Deep Reinforcement Learning

## References

- Sethi, P.; Sarangi, S.R. Internet of Things: Architectures, Protocols, and Applications. *J. Electr. Comput. Eng.* **2017**, *2017*, 1–25. [CrossRef]
- Sethi, R.; Bhushan, B.; Sharma, N.; Kumar, R.; Kaushik, I. *Applicability of Industrial IoT in Diversified Sectors: Evolution, Applications and Challenges*; Springer: Singapore, 2021; pp. 45–67. [CrossRef]
- Huo, R.; Zeng, S.; Wang, Z.; Shang, J.; Chen, W.; Huang, T.; Wang, S.; Yu, F.R.; Liu, Y. A Comprehensive Survey on Blockchain in Industrial Internet of Things: Motivations, Research Progresses, and Future Challenges. *IEEE Commun. Surv. Tutor.* **2022**, *1*, 88–122. [CrossRef]
- Sirojan, T.; Phung, T.; Ambikairajah, E. Enabling Deep Learning on Embedded Systems for IoT Sensor Data Analytics: Opportunities and Challenges. In Proceedings of the 2018 IEEE International Conference on Information and Automation for Sustainability (ICIAfS), Colombo, Sri Lanka, 21–22 December 2018; pp. 1–5. [CrossRef]
- Research, P. MCU Market, Microcontroller Market Size, Share, Growth, Forecast 2030. 2022. Available online: <https://www.precedenceresearch.com/microcontroller-mcu-market> (accessed on 31 May 2022).
- Singh, D.; Sandhu, A.; Thakur, A.; Priyank, N. An Overview of IoT Hardware Development Platforms. *Int. J. Emerg. Technol.* **2020**, *11*, 155–163.

7. Tewari, N.; Deepak, N.; Joshi, M.; Bhatt, J.S. Comparative Study of IoT Development Boards in 2021: Choosing right Hardware for IoT Projects. In Proceedings of the 2021 2nd International Conference on Intelligent Engineering and Management (ICIEM), London, UK, 28–30 April 2021; pp. 357–361. [[CrossRef](#)]
8. Bouachir, O.; Aloqaily, M.; Tseng, L.; Boukerche, A. Blockchain and Fog Computing for Cyberphysical Systems: The Case of Smart Industry. *Computer* **2020**, *53*, 36–45. [[CrossRef](#)]
9. Uddin, M.; Stranieri, A.; Gondal, I.; Balasubramanian, V. A Survey on the Adoption of Blockchain in IoT: Challenges and Solutions. *Blockchain Res. Appl.* **2021**, *2*, 100006. [[CrossRef](#)]
10. Makridakis, S.; Christodoulou, K. Blockchain: Current Challenges and Future Prospects/Applications. *Future Internet* **2019**, *11*, 258. [[CrossRef](#)]
11. Fan, C.; Ghaemi, S.; Khazaei, H.; Musilek, P. Performance Evaluation of Blockchain Systems: A Systematic Survey. *IEEE Access* **2020**, *8*, 126927–126950. [[CrossRef](#)]
12. Sagirlar, G.; Carminati, B.; Ferrari, E.; Sheehan, J.D.; Ragnoli, E. Hybrid-IoT: Hybrid Blockchain Architecture for Internet of Things-PoW Sub-Blockchains. In Proceedings of the 2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), Halifax, NS, Canada, 30 July–3 August 2018; pp. 1007–1016. [[CrossRef](#)]
13. Bach, L.M.; Mihaljevic, B.; Zagar, M. Comparative analysis of blockchain consensus algorithms. In Proceedings of the 2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), Opatija, Croatia, 21–25 May 2018; pp. 1545–1550. [[CrossRef](#)]
14. Li, M.; Qin, Y.; Liu, B.; Chu, X. Enhancing the efficiency and scalability of blockchain through probabilistic verification and clustering. *Inf. Process. Manag.* **2021**, *58*, 102650. [[CrossRef](#)]
15. Wang, W.; Hoang, D.T.; Hu, P.; Xiong, Z.; Niyato, D.; Wang, P.; Wen, Y.; Kim, D.I. A Survey on Consensus Mechanisms and Mining Strategy Management in Blockchain Networks. *IEEE Access* **2019**, *7*, 22328–22370. [[CrossRef](#)]
16. Zheng, Z.; Xie, S.; Dai, H.; Chen, X.; Wang, H. An Overview of Blockchain Technology: Architecture, Consensus, and Future Trends. In Proceedings of the 2017 IEEE International Congress on Big Data (BigData Congress), Honolulu, HI, USA, 25–30 June 2017; pp. 557–564. [[CrossRef](#)]
17. Nguyen, G.T.; Kim, K. A survey about consensus algorithms used in Blockchain. *J. Inf. Process. Syst.* **2018**, *14*, 101–128. [[CrossRef](#)]
18. Nartey, C.; Tchao, E.T.; Gadze, J.D.; Yeboah-Akowuah, B.; Nunoo-Mensah, H.; Welte, D.; Sikora, A. Blockchain-IoT peer device storage optimization using an advanced time-variant multi-objective particle swarm optimization algorithm. *EURASIP J. Wirel. Commun. Netw.* **2022**, *2022*, 1–27. [[CrossRef](#)]
19. Ding, N.; Zhao, Y. Lightweight Blockchain Based on Storage Resource Optimization for Internet of Vehicles. In Proceedings of the 2021 IEEE International Intelligent Transportation Systems Conference (ITSC), Indianapolis, IN, USA, 19–22 September 2021; pp. 1063–1068. [[CrossRef](#)]
20. Zuo, Y.; Jin, S.; Zhang, S.; Zhang, Y. Blockchain Storage and Computation Offloading for Cooperative Mobile-Edge Computing. *IEEE Internet Things J.* **2021**, *8*, 9084–9098. [[CrossRef](#)]
21. Zhou, Q.; Huang, H.; Zheng, Z.; Bian, J. Solutions to Scalability of Blockchain: A Survey. *IEEE Access* **2020**, *8*, 16440–16455. [[CrossRef](#)]
22. Yu, B.; Li, X.; Zhao, H. Virtual block group: A scalable blockchain model with partial node storage and distributed hash table. *Comput. J.* **2021**, *63*, 1524–1536. [[CrossRef](#)]
23. Qi, X.; Zhang, Z.; Jin, C.; Zhou, A. A reliable storage partition for permissioned blockchain. *IEEE Trans. Knowl. Data Eng.* **2021**, *33*, 14–27. [[CrossRef](#)]
24. Xu, M.; Feng, G.; Ren, Y.; Zhang, X. On Cloud Storage Optimization of Blockchain with a Clustering-Based Genetic Algorithm. *IEEE Internet Things J.* **2020**, *7*, 8547–8558. [[CrossRef](#)]
25. Mazlan, A.A.; Daud, S.M.; Sam, S.M.; Abas, H.; Rasid, S.Z.A.; Yusof, M.F. Scalability Challenges in Healthcare Blockchain System-A Systematic Review. *IEEE Access* **2020**, *8*, 23663–23673. [[CrossRef](#)]
26. Mei, H.; Gao, Z.; Guo, Z.; Zhao, M.; Yang, J. Storage mechanism optimization in blockchain system based on residual number system. *IEEE Access* **2019**, *7*, 114539–114546. [[CrossRef](#)]
27. Zeng, Q.; Xie, K.; Li, Y.; Guan, X.; Zhou, C.; Ma, S. Storage Optimization Algorithm for Publication Blockchain. *Adv. Intell. Syst. Comput.* **2018**, *905*, 828–835. [[CrossRef](#)]
28. Li, C.; Zhang, J.; Yang, X.; Youlong, L. Lightweight blockchain consensus mechanism and storage optimization for resource-constrained IoT devices. *Inf. Process. Manag.* **2021**, *58*, 102602. [[CrossRef](#)]
29. Misic, J.; Misic, V.B.; Chang, X.; Motlagh, S.G.; Ali, M.Z. Block Delivery Time in Bitcoin Distribution Network. In Proceedings of the ICC 2019-2019 IEEE International Conference on Communications (ICC), Shanghai, China, 20–24 May 2019. [[CrossRef](#)]
30. Imtiaz, M.A.; Starobinski, D.; Trachtenberg, A.; Younis, N. Churn in the Bitcoin Network. *IEEE Trans. Netw. Serv. Manag.* **2021**, *18*, 1598–1615. [[CrossRef](#)]
31. Berendea, N.; Mercier, H.; Onica, E.; Riviere, E. Fair and Efficient Gossip in Hyperledger Fabric. In Proceedings of the 2020 IEEE 40th International Conference on Distributed Computing Systems (ICDCS), Singapore, 29 November–1 December 2020; pp. 190–200. [[CrossRef](#)]
32. Zhang, L.; Xu, H.; Onireti, O.; Imran, M.A.; Cao, B. How Much Communication Resource is Needed to Run a Wireless Blockchain Network? *IEEE Netw.* **2022**, *1*, 128–135. [[CrossRef](#)]

33. Municio, E.; Latre, S.; Marquez-Barja, J.M. Extending Network Programmability to the Things Overlay Using Distributed Industrial IoT Protocols. *IEEE Trans. Ind. Inform.* **2021**, *17*, 251–259. [[CrossRef](#)]
34. Khan, W.Z.; Rehman, M.H.; Zangoti, H.M.; Afzal, M.K.; Armi, N.; Salah, K. Industrial internet of things: Recent advances, enabling technologies and open challenges. *Comput. Electr. Eng.* **2020**, *81*, 106522. [[CrossRef](#)]
35. Smetanin, S.; Ometov, A.; Komarov, M.; Masek, P.; Koucheryavy, Y. Blockchain Evaluation Approaches: State-of-the-Art and Future Perspective. *Sensors* **2020**, *20*, 3358. [[CrossRef](#)]
36. Srinidhi, N.N.; Dilip Kumar, S.M.; Venugopal, K.R. Network optimizations in the Internet of Things: A review. *Eng. Sci. Technol. Int. J.* **2019**, *22*, 1–21. [[CrossRef](#)]
37. Agnihotri, S.; Ramkumar, K.R. A survey and comparative analysis of the various routing protocols of Internet of Things. *Int. J. Pervasive Comput. Commun.* **2017**, *13*, 264–281. [[CrossRef](#)]
38. Tang, F.; Mao, B.; Kawamoto, Y.; Kato, N. Survey on Machine Learning for Intelligent End-to-End Communication Toward 6G: From Network Access, Routing to Traffic Control and Streaming Adaption. *IEEE Commun. Surv. Tutor.* **2021**, *23*, 1578–1598. [[CrossRef](#)]
39. Mammeri, Z. Reinforcement learning based routing in networks: Review and classification of approaches. *IEEE Access* **2019**, *7*, 55916–55950. [[CrossRef](#)]
40. Mayasari, R.; Syambas, N.R. Machine learning on named data network: A survey routing and forwarding strategy. In Proceedings of the 2020 14th International Conference on Telecommunication Systems, Services, and Applications (TSSA), Bandung, Indonesia, 4–5 November 2020. [[CrossRef](#)]
41. Kumari, P.; Sahana, S.K. QoS-Based ACO Routing Protocols in MANETs: A Review. In Proceedings of the Fourth International Conference on Microelectronics, Computing and Communication Systems, Ranchi, India, 11–12 May 2019; pp. 329–340. [[CrossRef](#)]
42. Kumari, P.; Sahana, S.K. An Efficient Swarm-Based Multicast Routing Technique—Review. *Adv. Intell. Syst. Comput.* **2019**, *711*, 123–134. [[CrossRef](#)]
43. Manikannu, J.; Nagarajan, V. A survey of energy efficient routing and optimization techniques in wireless sensor networks. In Proceedings of the 2017 International Conference on Communication and Signal Processing (ICCSP), Chennai, India, 6–8 April 2017; pp. 2075–2080. [[CrossRef](#)]
44. Islam, S.; Muslim, N.; Atwood, J.W. A Survey on Multicasting in Software-Defined Networking. *IEEE Commun. Surv. Tutor.* **2018**, *20*, 355–387. [[CrossRef](#)]
45. Cao, B.; Zhang, Z.; Feng, D.; Zhang, S.; Zhang, L.; Peng, M.; Li, Y. Performance analysis and comparison of PoW, PoS and DAG based blockchains. *Digit. Commun. Netw.* **2020**, *6*, 480–485. [[CrossRef](#)]
46. Monrat, A.A.; Schelen, O.; Andersson, K. Performance Evaluation of Permissioned Blockchain Platforms. In Proceedings of the 2020 IEEE Asia-Pacific Conference on Computer Science and Data Engineering (CSDE), Gold Coast, Australia, 16–18 December 2020. [[CrossRef](#)]
47. Perez, D.; Xu, J.; Livshits, B. Revisiting Transactional Statistics of High-scalability Blockchains. In Proceedings of the ACM Internet Measurement Conference, Virtual, 27–29 October 2020; Volume 15. [[CrossRef](#)]
48. Wu, Y.; Song, L.; Liu, L.; Li, J.; Li, X.; Zhou, L. Consensus Mechanism of IoT Based on Blockchain Technology. *Shock Vib.* **2020**, *2020*, 8846429. [[CrossRef](#)]
49. Fazlali, M.; Eftekhari, S.M.; Dehshibi, M.M.; Malazi, H.T.; Nosrati, M. Raft Consensus Algorithm: An Effective Substitute for Paxos in High Throughput P2P-based Systems. *arXiv* **2019**, arXiv:1911.01231.
50. Shrimali, B.; Patel, H.B. Blockchain state-of-the-art: Architecture, use cases, consensus, challenges and opportunities. *J. King Saud Univ.-Comput. Inf. Sci.* **2021**, *in press*. [[CrossRef](#)]
51. Ali Syed, T.; Alzahrani, A.; Jan, S.; Siddiqui, M.S.; Nadeem, A.; Alghamdi, T. A Comparative Analysis of Blockchain Architecture and its Applications: Problems and Recommendations. *IEEE Access* **2019**, *7*, 176838–176869. [[CrossRef](#)]
52. Nguyen, T.G.; Phan, T.V.; Hoang, D.T.; Nguyen, T.N.; So-In, C. Federated Deep Reinforcement Learning for Traffic Monitoring in SDN-Based IoT Networks. *IEEE Trans. Cogn. Commun. Netw.* **2021**, *7731*, 1–17. [[CrossRef](#)]
53. Monrat, A.A.; Schelen, O.; Andersson, K. A survey of blockchain from the perspectives of applications, challenges, and opportunities. *IEEE Access* **2019**, *7*, 117134–117151. [[CrossRef](#)]
54. Sanka, A.I.; Cheung, R.C. A systematic review of blockchain scalability: Issues, solutions, analysis and future research. *J. Netw. Comput. Appl.* **2021**, *195*, 103232. [[CrossRef](#)]
55. Alrehaili, A.; Namoun, A. A Comparative Analysis of Scalability Issues within Blockchain-based Solutions in the Internet of Things. *Environments* **2021**, *12*, 480–490. [[CrossRef](#)]
56. Xie, J.; Yu, F.R.; Huang, T.; Xie, R.; Liu, J.; Liu, Y. A Survey on the Scalability of Blockchain Systems. *IEEE Netw.* **2019**, *33*, 166–173. [[CrossRef](#)]
57. Leible, S.; Schlager, S.; Schubotz, M.; Gipp, B. A Review on Blockchain Technology and Blockchain Projects Fostering Open Science. *Front. Blockchain* **2019**, *2*, 16. [[CrossRef](#)]
58. Litke, A.; Anagnostopoulos, D.; Varvarigou, T. Blockchains for Supply Chain Management: Architectural Elements and Challenges Towards a Global Scale Deployment. *Logistics* **2019**, *3*, 5. [[CrossRef](#)]
59. Xu, X.; Sun, G.; Luo, L.; Cao, H.; Yu, H.; Vasilakos, A.V. Latency performance modeling and analysis for hyperledger fabric blockchain network. *Inf. Process. Manag.* **2021**, *58*, 102436. [[CrossRef](#)]

60. Loizou, N.; Richtarik, P. Revisiting Randomized Gossip Algorithms: General Framework, Convergence Rates and Novel Block and Accelerated Protocols. *IEEE Trans. Inf. Theory* **2021**, *67*, 8300–8324. [[CrossRef](#)]
61. Kiffer, L.; Salman, A.; Levin, D.; Misllove, A.; Nita-Rotaru, C. Under the Hood of the Ethereum Gossip Protocol. In *International Conference on Financial Cryptography and Data Security*; Springer: Berlin/Heidelberg, Germany, 2021; pp. 437–456. ISBN 9783662643303. [[CrossRef](#)]
62. Guggenberger, T.; Sedlmeir, J.; Fridgen, G.; Luckow, A. An In-Depth Investigation of Performance Characteristics of Hyperledger Fabric. *arXiv* **2021**, arXiv:2102.07731.
63. Nguyen, T.S.L.; Jourjon, G.; Potop-Butucaru, M.; Thai, K.L. Impact of network delays on Hyperledger Fabric. In Proceedings of the IEEE INFOCOM 2019-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), Paris, France, 29 April–2 May 2019; pp. 222–227. [[CrossRef](#)]
64. Zhang, Y.H.; Liu, X.F. Traffic Redundancy in Blockchain Systems: The Impact of Logical and Physical Network Structures. In Proceedings of the 2021 IEEE International Symposium on Circuits and Systems (ISCAS), Daegu, Korea, 22–28 May 2021; pp. 1–5. [[CrossRef](#)]
65. Elagin, V.; Spirikina, A.; Levakov, A.; Belozertsev, I. Blockchain Behavioral Traffic Model as a Tool to Influence Service IT Security. *Future Internet* **2020**, *12*, 68. [[CrossRef](#)]
66. Rohrer, E.; Tschorsch, F. Blockchain layer zero: Characterizing the bitcoin network through measurements, models, and simulations. In Proceedings of the 2021 IEEE 46th Conference on Local Computer Networks (LCN), Edmonton, AB, Canada, 4–7 October 2021; pp. 9–16. [[CrossRef](#)]
67. Zhang, Y.; Tan, R.; Kong, X.; Tan, Q.; Liu, X. Bitcoin Node Discovery: Large-Scale Empirical Evaluation of Network Churn. In *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*; Sun, X., Pan, Z., Bertino, E., Eds.; Springer International Publishing: Cham, Switzerland, 2019; pp. 385–395. [[CrossRef](#)]
68. Kim, A.; Kim, J.; Essaid, M.; Park, S.; Ju, H. Analysis of Compact Block Propagation Delay in Bitcoin Network. In Proceedings of the 2021 22nd Asia-Pacific Network Operations and Management Symposium (APNOMS), Tainan, Taiwan, 8–10 September 2021; pp. 313–318. [[CrossRef](#)]
69. Aoki, Y.; Shudo, K. Proximity Neighbor Selection in Blockchain Networks. In Proceedings of the 2019 IEEE International Conference on Blockchain (Blockchain), Atlanta, GA, USA, 14–17 July 2019; pp. 52–58. [[CrossRef](#)]
70. He, X.; Cui, Y.; Jiang, Y. An Improved Gossip Algorithm Based on Semi-Distributed Blockchain Network. In Proceedings of the 2019 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC), Guilin, China, 17–19 October 2019; pp. 24–27. [[CrossRef](#)]
71. Yu, B.; Li, X.; Zhao, H.; Zhou, T. A scalable blockchain network model with transmission paths and neighbor node subareas. *Computing* **2021**, *1*, 1–25. [[CrossRef](#)]
72. Shaleva, A.; Korkhov, V. Evaluation of the Neo P2P Blockchain Network Protocol Efficiency. In Proceedings of the International Conference on Computational Science and Its Applications, Cagliari, Italy, 13–16 September 2021; Springer: Cham, Switzerland, 2021; pp. 56–71, ISBN 9783030869724. [[CrossRef](#)]
73. Shaleva, A.; Korkhov, V. Efficient gossip-based protocol in the neo blockchain network. In Proceedings of the 9th International Conference “Distributed Computing and Grid Technologies in Science and Education”, Dubna, Russia, 5–9 July 2021; Volume 3041, pp. 508–513. [[CrossRef](#)]
74. Danzi, P.; Kalor, A.E.; Stefanovic, C.; Popovski, P. Analysis of the Communication Traffic for Blockchain Synchronization of IoT Devices. In Proceedings of the 2018 IEEE International Conference on Communications (ICC), Kansas City, MO, USA, 20–24 May 2018; pp. 1–7. [[CrossRef](#)]
75. Vu, H.; Tewari, H. An Efficient Peer-to-Peer Bitcoin Protocol with Probabilistic Flooding. In Proceedings of the International Conference for Emerging Technologies in Computing, London, UK, 19–20 August 2019; Springer: Cham, Switzerland, 2019; Volume 285, pp. 29–45, ISBN 9783030239428. [[CrossRef](#)]
76. Santiago, C.; Lee, C. Accelerating Message Propagation in Blockchain Networks. In Proceedings of the 2020 International Conference on Information and Communication Technology Convergence (ICTC), Jeju, Korea, 21–23 October 2020; pp. 157–160. [[CrossRef](#)]
77. Kan, J.; Zou, L.; Liu, B.; Huang, X. Boost Blockchain Broadcast Propagation with Tree Routing. In *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*; Springer: Cham, Switzerland, 2018; pp. 77–85. [[CrossRef](#)]
78. Kaneko, Y.; Asaka, T. DHT clustering for load balancing considering blockchain data size. In Proceedings of the 2018 Sixth International Symposium on Computing and Networking Workshops (CANDARW), Takayama, Japan, 27–30 November 2018; pp. 71–74. [[CrossRef](#)]
79. Yang, X.; Shi, L. Ari: A P2P optimization for blockchain systems. In Proceedings of the 2019 17th International Conference on Privacy, Security and Trust (PST), Fredericton, NB, Canada, 26–28 August 2019; pp. 1–6. [[CrossRef](#)]
80. Hao, W.; Zeng, J.; Dai, X.; Xiao, J.; Hua, Q.; Chen, H.; Li, K.C.; Jin, H. BlockP2P: Enabling Fast Blockchain Broadcast with Scalable Peer-to-Peer Network Topology. In *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*; Springer: Cham, Switzerland, 2019; pp. 223–237. [[CrossRef](#)]

81. Al-Musharaf, A.J.; Al-Alak, S.M.; Al-Mashhadi, H.M. Improving Blockchain Consensus Mechanism via Network Clusters. In Proceedings of the 2021 1st Babylon International Conference on Information Technology and Science (BICITS), Babil, Iraq, 28–29 April 2021; pp. 293–298. [\[CrossRef\]](#)
82. Huang, J.; Tan, L.; Mao, S.; Yu, K. Blockchain Network Propagation Mechanism Based on P4P Architecture. *Secur. Commun. Netw.* **2021**, *2021*, 1–12. [\[CrossRef\]](#)
83. Deshpande, V.; Badis, H.; George, L. Efficient topology control of blockchain peer to peer network based on SDN paradigm. *Peer-to-Peer Netw. Appl.* **2021**, *15*, 267–289. [\[CrossRef\]](#)
84. Baniata, H.; Anaqreh, A.; Kertesz, A. DONS: Dynamic Optimized Neighbor Selection for smart blockchain networks. *Future Gener. Comput. Syst.* **2022**, *130*, 75–90. [\[CrossRef\]](#)
85. Chen, J.; Qin, Y. Reducing Block Propagation Delay in Blockchain Networks via Guarantee Verification. In Proceedings of the 2021 IEEE 29th International Conference on Network Protocols (ICNP), Dallas, TX, USA, 1–5 November 2021. [\[CrossRef\]](#)
86. Zhang, M.; Cheng, Y.; Deng, X.; Wang, B.; Xie, J.; Yang, Y.; Zhang, J. Accelerating Transactions Relay in Blockchain Networks via Reputation. In Proceedings of the 2021 IEEE/ACM 29th International Symposium on Quality of Service (IWQOS), Tokyo, Japan, 25–28 June 2021. [\[CrossRef\]](#)
87. Locher, T. Fast Byzantine Agreement for Permissioned Distributed Ledgers. In Proceedings of the 32nd ACM Symposium on Parallelism in Algorithms and Architectures, Virtual, 15–17 July 2020; pp. 371–382. [\[CrossRef\]](#)
88. Jiang, Y.; Lian, Z. High performance and scalable byzantine fault tolerance. In Proceedings of the 2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC), Chengdu, China, 15–17 March 2019; pp. 1195–1202. [\[CrossRef\]](#)
89. Cao, N.; Jiang, D.; Liu, Y.; Zhou, Y.; Du, H.; Qiao, X.; Xia, Y.; Zhu, D.; Yu, F.; Bi, W. Revisit Raft Consistency Protocol on Private Blockchain System in High Network Latency. In Proceedings of the International Conference on Artificial Intelligence and Security, Dublin, Ireland, 19–23 July 2021; Springer: Cham, Switzerland, 2021; Volume 1423, pp. 571–579, ISBN 9783030786175. [\[CrossRef\]](#)
90. Jalalzai, M.M.; Busch, C.; Richard, G.G. Consistent BFT Performance for Blockchains. In Proceedings of the 2019 49th Annual IEEE/IFIP International Conference on Dependable Systems and Networks—Supplemental Volume (DSN-S), Portland, OR, USA, 24–27 June 2019; pp. 17–18. [\[CrossRef\]](#)
91. Casado-Vara, R.; Chamoso, P.; De la Prieta, F.; Prieto, J.; Corchado, J.M. Non-linear adaptive closed-loop control system for improved efficiency in IoT-blockchain management. *Inf. Fusion* **2019**, *49*, 227–239. [\[CrossRef\]](#)
92. Castro, M.; Liskov, B. Practical byzantine fault tolerance and proactive recovery. *ACM Trans. Comput. Syst.* **2002**, *20*, 398–461. [\[CrossRef\]](#)
93. Jalalzai, M.M.; Busch, C. Window Based BFT Blockchain Consensus. In Proceedings of the 2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), Halifax, NS, Canada, 30 July–3 August 2018; pp. 971–979. [\[CrossRef\]](#)
94. Lei, K.; Zhang, Q.; Xu, L.; Qi, Z. Reputation-Based Byzantine Fault-Tolerance for Consortium Blockchain. In Proceedings of the 2018 IEEE 24th International Conference on Parallel and Distributed Systems (ICPADS), Singapore, 11–13 December 2018; pp. 604–611. [\[CrossRef\]](#)
95. Ahn, S.; Kim, T.; Kwon, Y.; Cho, S. Packet aggregation scheme to mitigate the network congestion in blockchain networks. In Proceedings of the 2020 International Conference on Electronics, Information, and Communication (ICEIC), Barcelona, Spain, 19–22 January 2020. [\[CrossRef\]](#)
96. Jin, M.; Chen, X.; Lin, S.J. Reducing the Bandwidth of Block Propagation in Bitcoin Network With Erasure Coding. *IEEE Access* **2019**, *7*, 175606–175613. [\[CrossRef\]](#)
97. Zhang, L.; Wang, T.; Liew, S.C. Speeding up block propagation in Bitcoin network: Uncoded and coded designs. *Comput. Netw.* **2022**, *206*, 108791. [\[CrossRef\]](#)
98. Cebe, M.; Kaplan, B.; Akkaya, K. A network coding based information spreading approach for permissioned blockchain in IoT settings. In Proceedings of the ACM International Conference Proceeding Series, New York, NY, USA, 5–7 November 2018; pp. 470–475. [\[CrossRef\]](#)
99. Zhao, C.; Wang, T.; Zhang, S. LightBlock: Reducing Bandwidth Required to Synchronize Blocks in Ethereum Network. In Proceedings of the 2021 International Conference on Communications, Information System and Computer Engineering (CISCE), Beijing, China, 14–16 May 2021; pp. 868–874. [\[CrossRef\]](#)
100. Hang, L.; Kim, D.H. Optimal blockchain network construction methodology based on analysis of configurable components for enhancing Hyperledger Fabric performance. *Blockchain Res. Appl.* **2021**, *2*, 100009. [\[CrossRef\]](#)
101. Liu, M.; Yu, F.R.; Teng, Y.; Leung, V.C.; Song, M. Performance optimization for blockchain-enabled industrial internet of things (iiot) systems: A deep reinforcement learning approach. *IEEE Trans. Ind. Inform.* **2019**, *15*, 3559–3570. [\[CrossRef\]](#)