Received: Added at production | Revised: Added at production | Accepted: Added at production

DOI: xxx/xxxx

RESEARCH ARTICLE

A Physical Topology for Optimizing Partition Tolerance in Consortium Blockchains to Reach CAP Guarantee Bound

Han Wang[1] | Hui Li*[1,2] | Qiongwei Ye[3] | Ping Lu[4] | Yong Yang[5,6] | Peter Han Joo Chong[7] | Xiaoli Chu[8] | Qi Lv[1,6] | Abla Smahi[1]

[1]Shenzhen Graduate School, Peking University, Shenzhen, China
[2]Jiujiang University, Jiujiang, China
[3]Yunnan University of Finance & Economics, Kunming, China
[4]Zhongxing Telecom Equipment, Shenzhen, China
[5]Foshan University, Foshan, China
[6]Foshan Saisichan Technology Co., LTD, Foshan, China
[7]Department of Electrical and Electronic Engineering, Auckland University of Technology, Auckland, New Zealand
[8]Department of Electronic and Electrical Engineering, The University of Sheffield, Sheffield, U.K.

Correspondence
*Hui Li, Shenzhen Graduate School, Peking University, Shenzhen, China. Email: lih64@pkusz.edu.cn

Abstract

Decentralized cryptocurrency systems, known as blockchains, have shown promise as an infrastructure for mutually distrustful parties to securely agree on transactions. Nevertheless, blockchain systems are constrained by the CAP Trilemma. Due to performance degradation, it is impossible to address this issue by improving simply the consensus layer or the network layer. To alleviate the CAP constraint in consortium blockchains, we propose a topological construction method to optimize the physical layer based on multi-dimensional hypercubes with excellent partition tolerance in probability. The basic topology has the advantage of solving the mismatch problem between the overlay network and the underlying network. It is further extended to hierarchical recursive topologies with more intermediate links or short links to balance the reliability requirement with the cost of building the physical network. We prove that the proposed hypercube topology has better partition tolerance than the regular rooted tree and ring lattice topologies, and effectively fits the upper-layer protocols at the consensus and network layers. As a result, combined with suitable transmission and consensus protocols that satisfy strong consistency and availability, the proposed topology-constructed blockchain can reach the CAP guarantee bound.

KEYWORDS:
consortium blockchain, CAP Trilemma, partition tolerance, physical topology, hierarchical recursion

1 | INTRODUCTION

Bitcoin[1], as the first widely-deployed, decentralized global cryptocurrency, has sparked hundreds of variant systems. The core technological innovation underlining these systems is the decentralized infrastructure known as the blockchain. Although numerous protocols have been proposed to improve the performance of blockchains, none of them eliminate the CAP (Consistency, Availability, and Partition Tolerance) Trilemma[2] of the distributed theory. Specifically, consistency, availability, and partition tolerance concurrently cannot be strongly satisfied simultaneously in the distributed system.

A typical blockchain system generally consists of seven layers[3,4], namely, the physical layer, data layer, network layer, consensus layer, incentive layer, contract layer, and application layer. In order to balance the three properties in CAP Trilemma, some approaches optimize the consensus layer. The original Nakamoto consensus[1] uses the longest chain rule to select the main chain and satisfies availability when partitioning. However, forks turn strong consistency into final consistency within a period, which results in long confirming time and limited throughput of the overall system. In public blockchains, several alternative

ledger structures[5,6,7,8,9] being investigated to make the most of computing power. Due to their weak or final consistency, the performance of these protocols is still limited by the network synchronization rate. Once blocks are generated faster than the network synchronization rate, the number of forks will expand and the system will be insecure. In consortium blockchains and private blockchains, BFT (Byzantine Fault Tolerant) consensus protocols such as PBFT (Practical Byzantine Fault Tolerance)[10] and PPoV (Parallel Proof of Vote)[11] are used to avoid forks[12]. Although satisfying strong consistency, BFT-like protocols suffer from communication complexity and scalability. Moreover, some performance-driven consensus protocols[13,14,15,16,17,18] guarantee consistency and throughput by delegation mechanism but decrease the number of core nodes participating in the consensus. A. Lewis-Pye and T. Roughgarden[19] have further proved an analog of the CAP Trilemma at the consensus layer that no protocol is both adaptive and has finality in the unsized and partially synchronous network.

In addition to optimizing consensus protocols, there are other network-layer-related works[20]. The fully distributed public blockchain utilized the P2P (Peer-to-Peer) network, particularly the unstructured one, to enable the consensus-reaching dissemination of transactions and blocks. However, its performance suffers from the mismatch problem between the overlay links and the underlying physical network topology, resulting in a large volume of redundant traffic and loss of reliability. R. Li and H. Asaeda[21] replace P2P networks with ICN (Information-Centric Networking) for communication, but incorporate a specific centralized node for group management that is susceptible to single-point of failure. Currently, consortium blockchains also tend to adopt P2P network technologies. Due to the semi-central nature, structured overlays are more suitable than unstructured overlays for consortium blockchains, while the mismatch problem is as severe as for public blockchains. Therefore, working alone at the network layer is insufficient.

In this paper, we propose a method to construct a topology at the physical layer that alleviates CAP constraints for the overall consortium blockchain system, in particular for the network and consensus layers. The basic physical topology is based on the multi-dimensional hypercube, which can alleviate the mismatch problem. We then design hierarchical recursive physical topologies to make the approach cost-effective and implementable. In addition, we propose a quantitative model for the partition tolerance property. The analytical results show that our approach guarantees excellent partition tolerance in probability. Our experiments also show that the proposed physical topology does not affect the performance of the upper-layer protocols. Combined with protocols with good consistency and availability at the network layer and the consensus layer, such as Gossip[22] and PPoV[11], the system constructed by the proposed method can reach the CAP guarantee bound.

Roadmap. In Section 2, we introduce P2P networks in blockchains. Section 3 presents basic and recursive methods for topological construction based on hypercubes. We discuss partition tolerance properties and link consumption in Section 4. We then deploy the proposed topology and evaluate the performance in Section 5. Finally, we conclude our work in Section 6.

2 | RELATED WORK

As a means of easing the CAP constraints at the network layer, P2P overlay is the most widespread form of networking in blockchain systems. It is utilized by the network layer because it is distributed, can withstand single point of failure, and possesses equality, autonomy, and decentralization. In a P2P network, each node is both a server and a client. The message exchange relies on a group of clients rather than a central server, so nodes should participate in the relay. The implementation of the service is performed directly between the nodes. Since services are distributed among nodes, the failure of some nodes or links has minimal effect on the remainder of the network. On the other hand, queuing in communication is low, which reduces the resource and time cost caused by centralization. Depending on whether the network is centralized or not, the P2P networks are classified as fully distributed or semi-distributed.

2.1 | Fully Distributed P2P Network

Nodes in a fully distributed P2P network are free to join and leave, and there is no central node. Fully distributed P2P networks include both structured and unstructured networks. The difference is whether or not the node addresses are structured. In unstructured P2P networks, the entire network forms a random graph structure without fixed network topology and structured uniform node addresses. The typical blockchain application of a fully distributed unstructured P2P network is Bitcoin. In contrast, structured P2P networks define the topological relationships of nodes, and the structure of the network is guaranteed by certain protocols between nodes. The typical blockchain application of a fully distributed structured P2P network is Ethereum[23].

2.2 | Semi-Distributed P2P Network

Semi-distributed P2P networks combine the features of structured and unstructured networks. The advantage of semi-distributed P2P networks is that they are efficient, scalable, and easy to manage. Its typical blockchain application is Hyperledger[24].

The semi-distributed P2P network is often combined with the delegated mechanism, which also appears in the consensus algorithm. Specifically, it divides nodes into super and ordinary nodes based on evaluation criteria such as computing power, bandwidth, and retention time. Super nodes endorse and supervise ordinary nodes of the organization or institution to which the node belongs, and participate in the core consensus process. Ordinary nodes make transactions through super nodes, but do not participate in bookkeeping and voting. The super nodes are equal to each other and there is no frequent joining and leaving problem. At the same time, the number of super nodes in the consortium blockchain is significantly less than that of the public blockchain. Although the original Bitcoin was based on an unstructured and fully distributed P2P network and flooding mechanism, a semi-distributed P2P network is better suited as the networking mode of the consortium blockchain from the standpoint of network efficiency.

2.3 | P2P Network Topology in Blockchains

Several researches have focused on optimizing the topology of overlay networks, especially semi-distributed P2P networks, to speed up the propagation of transactions and blocks in the blockchain. C. Decker and R. Wattenhofer[25] constructed a subgraph of stars that served as a central communication hub in the P2P network. It reduced the number of route hops between nodes. M. Fadhil, G. Owenson and M. Adda[26] proposed a clustering protocol for the semi-distributed P2P networks based on super nodes in blockchains, called BCBSN (Bitcoin Clustering Based on Super Node). Clustering on the basis of node locality, the propagation delay of transactions and blocks within the same cluster is reduced. The LBC (Location Based Clustering) protocol[27] and the BCBPT (Bitcoin Clustering Based on Ping Time) protocol[28] were further proposed. Nodes in a blockchain network are clustered according to physical location metrics such as their geographical location and ping time to reduce the propagation delay of neighboring nodes.

While these methods have considered the impact of the physical layer on the performance, they do not fully address the mismatch between the overlay and the underlying topology. Moreover, the above methods are complementary improvements to the current public blockchain networks. Unlike nodes in public blockchains, consortium blockchains nodes are more controlled, hence it is feasible to consider the physical layer directly while networking a consortium blockchain.

3 | PHYSICAL TOPOLOGY CONSTRUCTION METHODS

For structured P2P networks in consortium blockchains, we construct a matching physical topology based on a multi-dimensional hypercube. The construction method is also applicable to super nodes of semi-distributed P2P networks. In a semi-distributed P2P network, super and ordinary nodes can be connected in a tree topology to enable management, which is not described in detail here.

In the original hypercube topology for a base $b = 2$, each node has the same responsibility. The network diameter, defined as the shortest path between most distant nodes in terms of node hops, is $log_2 N$, which is less than $O(N)$ (N is the number of nodes)[29]. As a result, the hypercube-based topology has advantages in decentralized networking.

3.1 | Basic Physical Topology

We first employ the multi-dimensional hypercube or its variants to construct the basic physical topology. A complete hypercube is kind of a closed, compact and convex graph, whose 1-dimensional skeleton is composed of a group of line segments of equal length aligned to each dimension in the space where they are located, in which the relative line segments are parallel to each other, while the line segments intersecting at a point are orthogonal to each other. Figure 1 shows examples of topologies constructed based on hypercubes in 3 to 5 dimensions, where crosses indicate that the corresponding nodes or links are unassigned or invalid in the network. The blue and purple nodes and links represent the low-dimensional hypercube before and after the move, respectively, and the green links show the path of the move.

For each node, an ID is assigned to uniquely identify a node. At this point, the hypercube topology supports the establishment of links between pairs of nodes at distance 2^i to improve query efficiency, that is, the logical and physical distances between

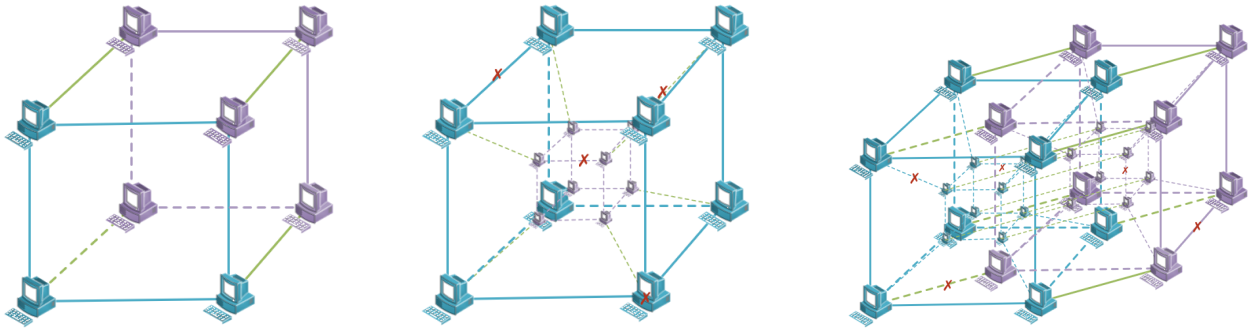

Figure 1 Examples of topology construction with complete or incomplete hypercubes evolving from 3 to 5 dimension. (a) A complete 3-dimensional cube; (b) An incomplete 4-dimensional hypercube; (c) An incomplete 5-dimensional hypercube.

pairs of nodes whose IDs differ by only 1 bit are equal to 1. Thus, the hypercube-based physical topology is a good match for overlay networks with the same IDs.

In the proposed physical topology, invalid links will be repaired actively in a finite time. If the network is partitioned unfortunately, the isolated node will request data synchronization from its neighbors before resuming to work.

3.2 | Hierarchical Recursive Physical Topology

As the network size grows, the use of the basic hypercube topology leads to excessive redundancy of links. In addition, the development of sharding techniques[30] prompts nodes to form different domains in the upper layers, which increases the complexity of networking. Therefore, recursion is a good method for scalability. The recursive topology should preserve the advantage of strong regularity of hypercubes.

In this section, we propose the hierarchical recursive topology, which is generated from the basic hypercube-based topology in two steps, including recursion and interconnection. Each recursion turns the original node into a domain. Nodes in the same domain can construct a hypercube-based physical topology described in Section 3.1 or an arbitrary topology. The number of the domain at the r-th ($r \geq 2$) level is defined as i_r, which is equal to the smallest node number n_{min} within it. After recursion, the new node number increases by one digit from the original node number, so the first $(r-1)$ digits of i_r represent the recursive attribution of the domain from the first level to the $(r-1)$-th level. Assuming that the node number of the hypercube in the first-level domain $i_1 = 0$ is $\{n_0\} = \{0, 1, 2, ..., (2^{dim_0} - 1)\}$, then the node number in the domain i_r after $(r-1)$ recursions is $\{n_{i_r}\} = \{n_{i_{r-1}}\}\&\{0, 1, 2, ..., (2^{dim_{i_r}} - 1)\}$, where & means splicing. The corresponding nodes of each domain at the r-th level are connected into the topological relationship of the domain at the $(r-1)$-th level with physical links, thus the interconnection between domains is completed.

Based on the above, we design three recursive methods to construct hierarchical physical topologies: completely symmetric, semi-symmetric, and asymmetric. In a completely symmetric way, each recursion takes hypercubes of the same dimension. In a semi-symmetric way, hypercubes of the same dimension are used within the same level and hypercubes of different dimensions are used between levels. In an asymmetric way, each domain takes a hypercube of a different dimension or whatever. Figure 2 shows the examples of hierarchical recursive physical topologies.

We use Figure 3 as a simple example to describe the two steps of the construction of a completely symmetric topology for 2-dimensional hypercubes. The numbers inside the circles indicate the number of nodes, and the numbers with the underscore indicate the number of domains. Suppose that the 4 nodes are numbered as $\{n_{i_1}\} = \{n_0\} = \{0, 1, 2, 3\}$. In the first step, each node recursively becomes a 2-dimensional hypercube. There are 4 domains at the second level and 16 nodes, respectively numbered as $\{n_{i_2}\} = \{n_{00}, n_{10}, n_{20}, n_{30}\} = \{00, 01, 02, 03\} \cup \{10, 11, 12, 13\} \cup \{20, 21, 22, 23\} \cup \{30, 31, 32, 33\}$. In the second step, we connect the 4 nodes in the same domain to the nodes in the other two domains associated with them with the same last digit. Solid lines represent physical links between nodes, and dashed lines represent logical links between domains. Specifically, the node 00 is linked to the nodes 10 and 30, the node 01 to the nodes 11 and 31, the node 02 to the nodes 12 and 32, the node 03 to the nodes 13 and 33, and so on.

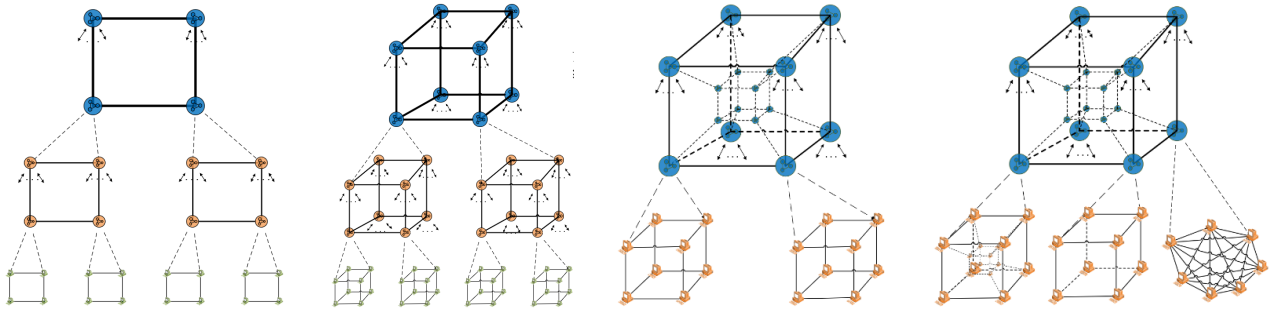

Figure 2 Examples of hierarchical recursive topology construction. (a) A lowest boundary situation; (b) A completely symmetric situation; (c) A semi-symmetric situation; (d) An asymmetric situation that there are 4, 8 and 4 domains with 4-dimensional hypercube, 3-dimensional hyper-cube and 7-potint full connection topology at the second level respectively.

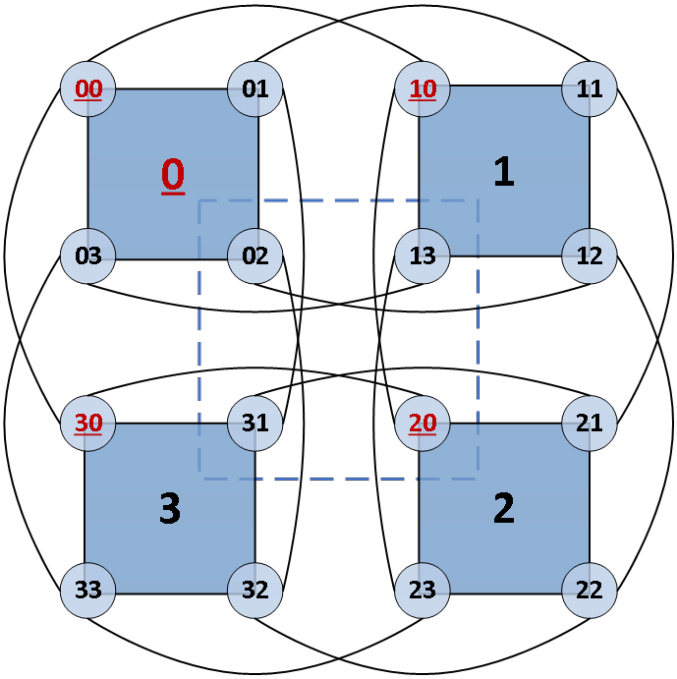

Figure 3 Two steps to construct a completely symmetric topology for 2-dimensional hypercubes.

4 | THEORETICAL ANALYSIS

In this section, we quantify the partition tolerance property by modeling the proposed topologies. We assume that physical links between nodes are secure. Other assumptions for topological construction and analysis are listed below:

1. Each involved link has only two states: work and failure;
2. Failure and repair of links are independent processes without memory;
3. MTBF (Mean Time Between Failures) and MTTR (Mean Time to Repair) of each link are independent, and their mean values are constant;
4. MTBF is much larger than MTTR;
5. If a partition has enough participating nodes, it is a good partition that works flawlessly. If not, it is a wrong partition.

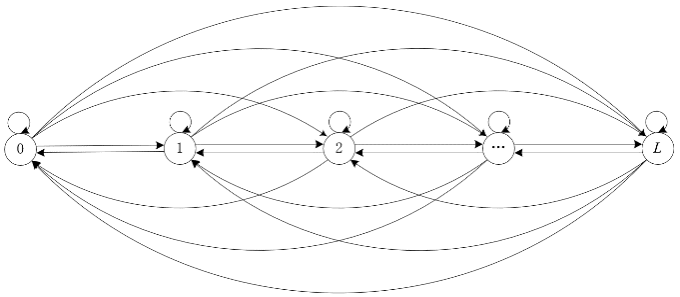

Figure 4 The Markov chain of the basic hypercube-based physical topology.

Considering that the partition tolerance property reflects the reliability of the blockchain, we propose two metrics in the quantitative model: the partition tolerance probability and the average minimum repair time. The partition tolerance probability is defined as the probability that a good partition exists in the network. The average minimum repair time is defined as the minimum time expected for the network to resume normal communication.

4.1 | Basic Physical Topology

4.1.1 | Quantitative Model for the Partition Tolerance Property

We first calculate the partition tolerance probability of the basic physical topology. According to assumptions 1-5, since the state change of each link follows "work-failure-repair", a discrete-time Markov process can be utilized for mathematical modeling the partition tolerance problem[31]. Since the failure and repair processes are independent and memoryless, Figure 4 shows the Markov chain of the basic hypercube-based topology consisting of N nodes and L links. The system state X denotes the number of invalid links in the network.

The transition matrix of the Markov model is denoted as $P_{(L+1)\times(L+1)}$, whose element p_{ji} represents the probability of the system state transiting from X_i to X_j. We calculate the state transition probability p_{ji} as:

$$\begin{aligned} p_{ji} &= P\left(X_j \mid X_i\right) \\ &= \sum_{m} P[(i-m)\ invalid\ links\ are\ repaired] \cdot P[(j-m)\ working\ links\ are\ invalid] \\ &= \sum_{m=\max\{i+j-L,0\}}^{m=\min\{i,j\}} \binom{i}{m} \mu^{i-m}(1-\mu)^m \cdot \binom{L-i}{j-m} \lambda^{j-m}(1-\lambda)^{L-i-j+m}, \end{aligned} \tag{1}$$

where the variable $m \in [max\{i+j-L,0\}, min\{i,j\}]$ is the number of unrepaired invalid links during the transition process, $\lambda = \frac{1}{MTBF}$ is the probability of a link to fail in the unit time, and $\mu = \frac{1}{MTTR}$ is the probability of repairing it in the unit time.

Since Figure 4 is a fully connected graph, the transition matrix P satisfies randomness, irreducibility, and aperiodicity. According to the limit theorem of the Markov chain [32], the above Markov process eventually converges to a steady-state independent of the initial distribution. We obtain the steady-state probability vector $\pi = [\pi_0, \pi_1, \pi_2, , \pi_L]$ through the typical partitioning algorithm [33] with $O(L)$ iterations.

We then estimate the partition tolerance probability by sampling. For each sample in steady-state π_i, the maximum number of nodes in the connected components is calculated to judge whether it is a wrong partition. The overall partition tolerance probability is:

$$p = 1 - \sum_{i=1}^{i=L} \pi_i \cdot P\{wrong\ partition \mid \pi_i\}. \tag{2}$$

Further, we compute the minimum repair time for the basic physical topology. In the proposed topology, invalid links between large partitions should be repaired preferentially. Under this strategy, we calculate the time for the system to resume its regular work, and get the overall average minimum repair time by taking the minimum repair time as the weight for each sample in Equation (2).

Algorithm 1 Calculate partition tolerance probability and average minimum repair time

Input:

The topological graph, $G = (V, L)$;
The sampling number, N_1, N_2;
The minimum number of nodes in a good partition, k;
The $(l+1) \times (l+1)$ transition matrix, P;
The probability of repairing an invalid link in the unit time, μ

Output:

The partition tolerance probability, p;
The average minimum repair time, $time$;

Initial $p\_badpart = 0; time = 0;$
$\pi = steady\_state(P);$
for $bad\_num = 0 : length(L)$ **do**
  $bad\_count = 0; min\_time = 0;$
  for $i = 1 : N_1$ **do** //Sample N_1 times.
    Remove any bad_num edges from graph G to get graph $subG$;
    $[bins, binsizes]$ = the connected component parameters of graph $subG$;
    if $max(bins) > 1 \&\& max(binsizes) < k$ **then** //This sample is a wrong partition.
      $bad\_count++; repair\_time = 0;$
      for $repair\_num = 0 : bad\_num$ **do** //Simulate the repair process of the sample.
        for $j = 1 : N_2$ **do** //Sample N_2 times.
          Repair any $repair\_num$ edges to get graph $subGG$;
          $[bins, binsizes]$ = the connected component parameters of graph $subGG$;
          if $max(bins) \leq 1 \parallel max(binsizes) \geq k$ **then** //This sample is repaired.
            $repair\_time = repair\_num;$
            break;
          end if
        end for
        if $repair\_time > 0$ **then**
          break;
        end if
      end for
      $min\_time += repair\_time;$
    end if
  end for
  $p\_badpart += \pi_{bad\_num} \cdot bad\_count / N_1;$
  $time += min\_time \cdot \pi_{bad\_num};$
end for
$time = time / \mu / p\_badpart / N_1;$
$p = 1 - p\_badpart.$

Algorithm 1 describes a pseudo-code for computing the partition tolerance probability and the average minimum repair time of a basic hypercube-based topology, where V and L are the point set and the edge set of a multi-dimensional hypercube.

4.1.2 | Simulation

We simulate the quantitative model in Section 4.1.1 based on a digital optical cable communication system that automatically switches between primary and secondary. According to the international standard, the communication systems (including physical links and repeaters) with the three different distances of $\{5000km, 3000km, 420km\}$ should meet the following indicators, respectively:

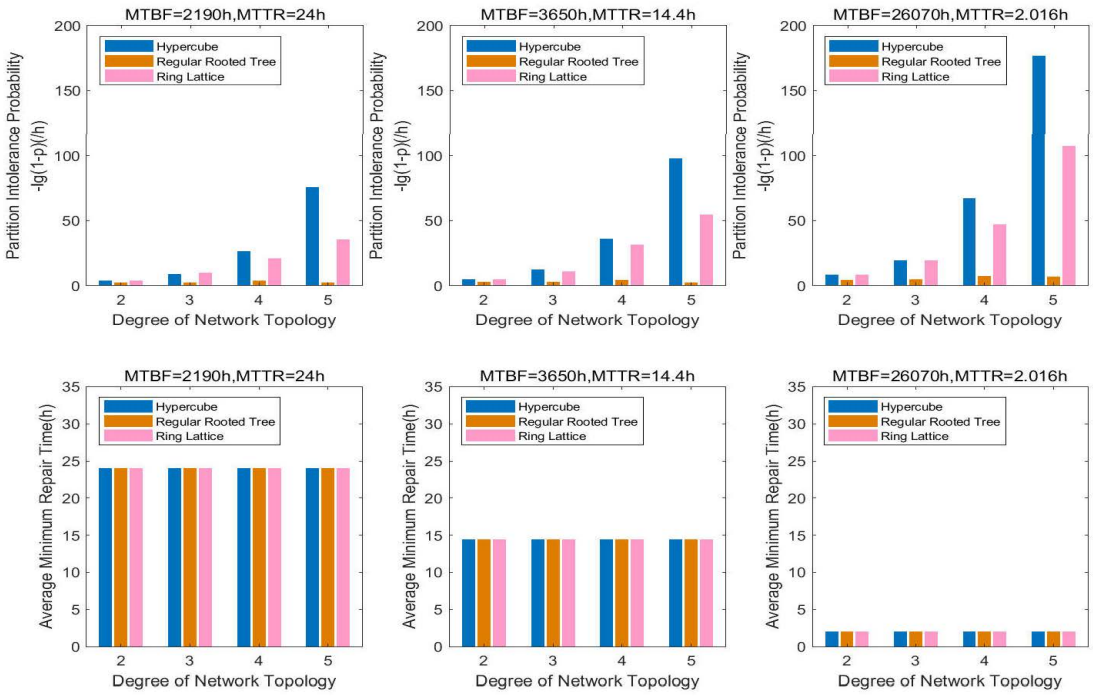

Figure 5 Partition tolerance probability and average minimum repair time for the basic topologies of multi-dimensional hypercubes, regular rooted trees, and ring lattices.

$$(MTBF, MTTR) \in \{(2190, 24), (3650, 14.4), (26070, 2.016)\}(h), \tag{3}$$

where (h) is the unit and is short for $(hour)$.

So we take the pair of parameters as $(\lambda, \mu) \in \{(4.5662 \times 10^{-4}, 4.1667 \times 10^{-2}), (2.7397 \times 10^{-4}, 6.9444 \times 10^{-2}), (3.8358 \times 10^{-5}, 4.9603 \times 10^{-1})\}(/h)$. In the consortium blockchain with the PPoV consensus [11], the minimum number of nodes in a good partition is not less than $k = \lfloor \frac{v}{2} \rfloor + 1$. Figure 5 compares the partition tolerance probability and the average minimum repair time for the basic multi-dimensional hypercube topology with the regular rooted tree topology and the ring lattice topology under the same degree.

The results show that with the increase of degree in the network, the partition tolerance probability increases rapidly without additional average minimum repair time. The proposed topology meets higher partition tolerance probability than the regular rooted tree topology and the ring lattice topology. In addition, the average minimum repair time for different topologies are similar, approximately equal to MTTR. Therefore, blockchains using the basic hypercube-based topology can obtain excellent reliability in the partitioned network.

4.2 | Hierarchical Recursive Physical Topology

Although the partition tolerance of the basic topology is excellent, since the hypercube is symmetric, it requires a large number of long-distance links. In this section, we analyze the partition tolerance property and link consumption of the hierarchical recursive topology.

4.2.1 | Partition Tolerance Property

Denote the partition tolerance probability of a basic topology at the m-th ($1 \leq m \leq r$) level as p_{i_m}. The partition tolerance of the domain i_r is not only affected by the topology it adopts at the r-th level but also related to the recursive path to which it belongs from the first level to the $(r-1)$-th level. Thus, the overall partition tolerance probability is:

$$p = 1 - \sum_{m=1}^{m=r} \sum_{(i_1,i_2,i_3,\ldots,i_m)} p_{i_1} p_{i_2} p_{i_3} \cdots p_{i_{m-1}} (1 - p_{i_m}). \tag{4}$$

Similarly, let t_{i_m} denote the average minimum repair time of a basic topology at the m-th level, and the overall average minimum repair time is:

$$t = \sum_{m=1}^{m=r} \sum_{(i_1,i_2,i_3,\ldots,i_m)} t_{i_m} \cdot \frac{p_{i_1} p_{i_2} p_{i_3} \cdots p_{i_{m-1}} (1 - p_{i_m})}{1 - p}. \tag{5}$$

According to Equation (4) and (5), the recursive path from the first level to the $(r-1)$-th level has more influence on the overall partition tolerance than the topology of the domain itself at the r-th level. Therefore, a suggestion is that the topology with high partition tolerance probability and low average minimum repair time should be adopted in the upper recursion path.

4.2.2 | Link Consumption

Considering that the number of nodes in the asymmetric recursive method is difficult to determine, in this section we only analyze the completely symmetric and the semi-symmetric recursive methods.

In a completely symmetric topology, we define the dimension of hypercubes in each domain as dim_{i_m}. Since each recursion takes the same dimension, for any $(i_1, i_2, i_3, \ldots, i_r)$, dim_{i_m} is a fixed value and is denoted as dim. Obviously, the number of nodes for $(r-1)$ recursions is $N_{symm,r} = 2^{dim \times r}$. The links in the topology consist of relatively short intradomain links and relatively long interdomain links, that is:

$$\begin{cases} L_{symm,r} = 2^{dim-1} \times dim \times 2^{(r-1)\times dim} + L_{symm,(r-1)} \times 2^{dim}, \\ L_{symm,1} = 2^{dim-1} \times dim. \end{cases} \tag{6}$$

Equation (6) is equivalent to:

$$\begin{cases} \dfrac{L_{symm,r}}{2^{r\times dim}} = \dfrac{dim}{2} + \dfrac{L_{symm,(r-1)}}{2^{(r-1)\times dim}}, \\ L_{symm,1} = 2^{dim-1} \times dim, \end{cases} \tag{7}$$

$$\frac{L_{symm,r}}{2^{r\times dim}} = \frac{L_{symm,1}}{2^{dim}} + \frac{dim}{2} \times (r-1) = \frac{r \times dim}{2}. \tag{8}$$

So, the total number of links is:

$$L_{symm,r} = 2^{r\times dim-1} \times r \times dim. \tag{9}$$

In a semi-symmetric topology, since domains in the same level use hypercubes of the same dimension, for any i_m, dim_{i_m} is a fixed value. Similarly, the number of nodes for $(r-1)$ recursions is $N_{semi,r} = 2^{\sum_{m=1}^{m=r} dim_{i_m}}$, and the number of links satisfies:

$$\begin{cases} L_{semi,r} = 2^{dim_{i_r}-1} \times dim_{i_r} \times 2^{\sum_{m=1}^{m=r-1} dim_{i_m}} + L_{semi,(r-1)} \times 2^{dim_{i_r}}, \\ L_{semi,1} = 2^{dim_{i_1}-1} \times dim_{i_1}. \end{cases} \tag{10}$$

So, the total number of links is:

$$L_{semi,r} = 2^{\sum_{m=1}^{m=r} dim_{i_m}-1} \times \sum_{m=1}^{m=r} dim_{i_m}. \tag{11}$$

Table 1 compares the properties of link consumption and partition tolerance under different physical topology construction methods. It is assumed that the 0th, 1st and 2nd recursion adopts links of 5000km, 3000km and 420km, respectively, and the indicators in Equation (3) are satisfied.

While each recursion does not reduce the total number of links, it actually uses more intermediate or short links between nodes and fewer long links than the ring lattice and the basic hypercube-based topologies. With the same number of recursions,

Table 1 Comparison of link consumption and partition tolerance under different physical topology construction methods.

Number of nodes	Topology construction method		Number of links			Partition tolerance probability -lg(1-p)(/h)	Average minimum repair time
			5000km	3000km	420km		
64	Regular rooted tree		63	0	0	2.79	24
	Ring lattice		192	0	0	43.7	24
	Hypercube	0 recursion (6)	192	0	0	87	24
		1 completely symmetric recursion (3-3)	96	96	0	8.62	24
		2 completely symmetric recursions (2-2-2)	64	64	64	3.56	21.4
		1 semi-symmetric recursion (4-2)	128	64	0	3.53	14.4
4096	Regular rooted tree		4095	0	0	3.93	24
	Ring lattice		24576	0	0	107	24
	Hypercube	0 recursion (12)	24576	0	0	227	24
		1 completely symmetric recursion (6-6)	12288	12288	0	87	24
		2 completely symmetric recursions (4-4-4)	8192	8192	8192	26.4	24
		2 semi-symmetric recursions (5-4-3)	10240	8192	6144	16.6	2.02

the higher the dimension of the hypercube in the recursion path, the less repair time is required. On the other hand, although the hypercube-based topology has a higher number of links than the regular rooted tree topology, its partition tolerance property is much better and already meets the needs of practical consortium blockchains. Blockchain projects are free to choose hierarchical recursive methods according to their own reliability and cost requirements.

5 | EXPERIMENTS

In this section, we systematically evaluate the performance of the proposed physical topology to support the Gossip protocol and the PPoV consensus protocol in the consortium blockchain.

5.1 | Overlaying the Gossip Protocol at the Network Layer

We build the network with the basic physical topology in the PeerSim-1.0.5 simulator[34]. For comparison, we also implement the regular rooted tree topology and the ring lattice topology. Upon the simulated physical layer, we execute a simple Gossip protocol[22] and show its performance. In the Gossip protocol, each node periodically selects a random subset of 4 neighbors

from its cached set to initiate a message exchange operation. Since the out-degree is a fixed value equal to the cache size, we consider only the in-degree. Each simulation runs for 5000 cycles, and every cycle has 100 units of time.

When the network size is $N = 16$, and the delay is 0 and 50%, the number of forwarded messages in the whole network is shown in Figure 6.(a) and 6.(b), respectively. In the ideal network with delay=0, the transmission performance of the three topologies compared is similar. In the network with delay=50%, the performance of the hypercube and the ring lattice topologies is similar and significantly better than that of the regular rooted tree topology.

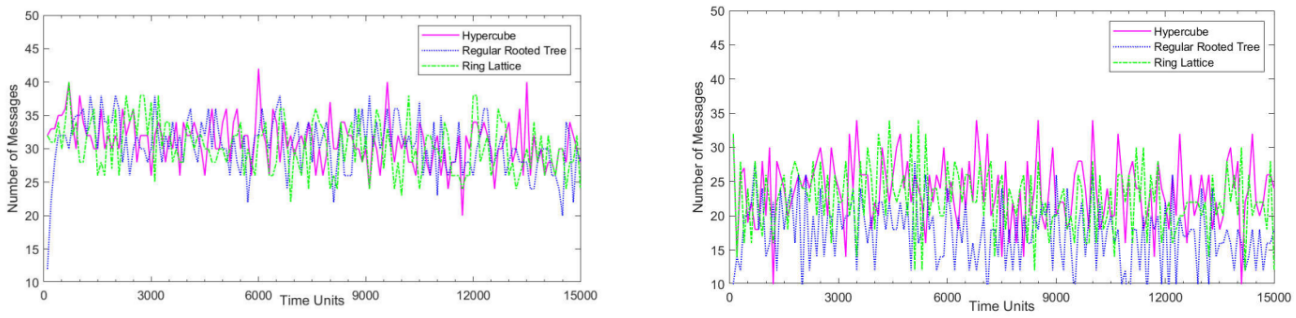

Figure 6 The number of forwarded Gossip messages when N=16. (a) Delay=0; (b) Delay=50%.

As the network size increases, the average number of forwarded messages increases approximately linearly, as shown in Figure 7. Experimental results show that the hypercube-based physical topology brings almost no additional forwarding redundancy to the network layer.

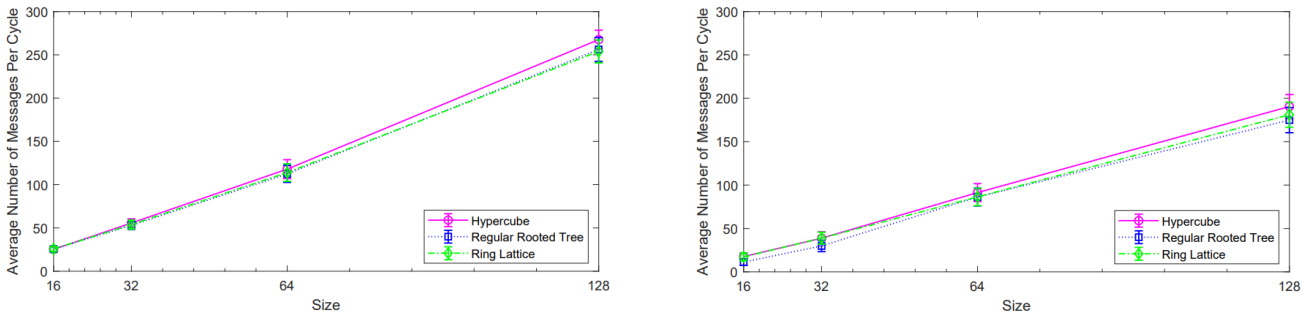

Figure 7 The average number of forwarded Gossip messages for different number of nodes. (a) Delay=0; (b) Delay=50%.

We further prototype the hierarchical recursive physical topology in the PeerSim-1.0.5 simulator. For the four examples in Figure 2, we test their performance separately, as shown in Figure 8. It can be seen that the 50% delay leads to an approximately 50% reduction in the total number of forwarded messages, so the Gossip protocol on the hierarchical recursive topology remains compliant with the original network characteristics. The results also show that the recursion increases the number of forwarded messages by 10%-20% compared to the basic hypercube-based topology. This is because the Gossip protocol tends to create loops in the domain when recursing, resulting in a large number of redundant messages in the network.

5.2 | Overlaying the PPoV Protocol at the Consensus Layer

We deploy the proposed physical topology across eight servers in China, the United Kingdom, New Zealand, and Malaysia. Each server has two 8-core CPUs and 10 Gbps network bandwidth. We next perform the PPoV protocol[11] over different topologies. PPoV is a consortium consensus protocol with strong consistency and high availability on fault-free networks.

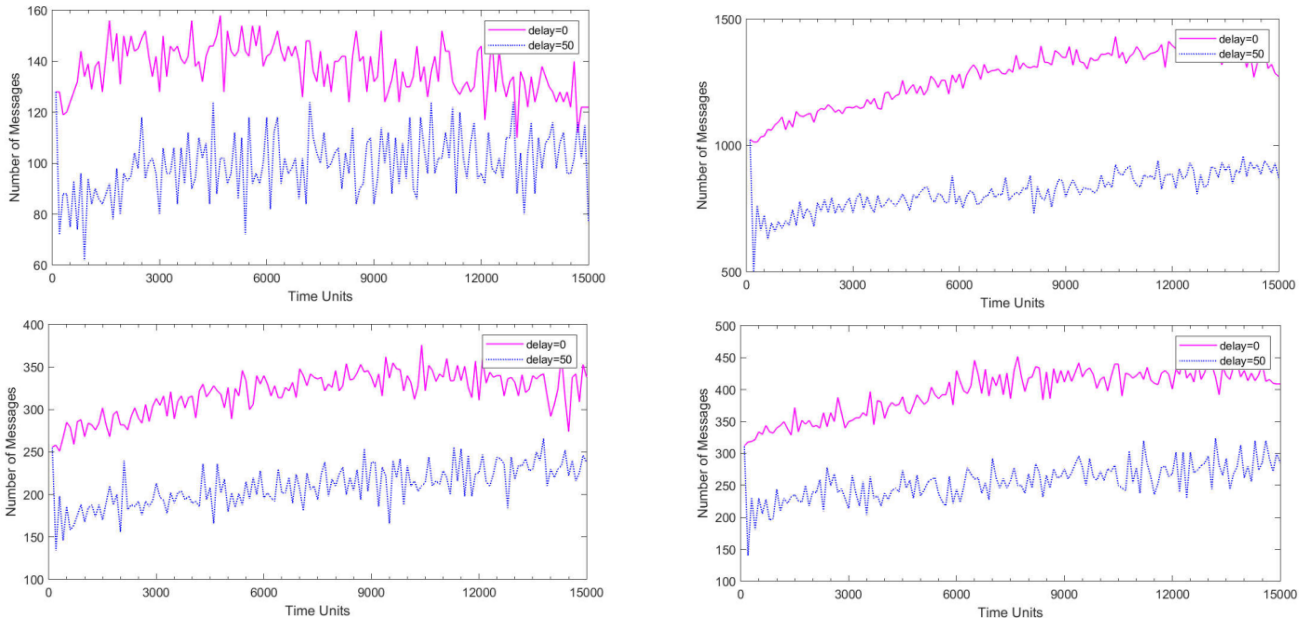

Figure 8 The number of forwarded Gossip messages with hierarchical recursive physical topologies. (a) The lowest boundary topology with N=64; (b) The completely symmetric topology with N=512; (c) The semi-symmetric topology with N=128; (d) The asymmetric topology with N=156.

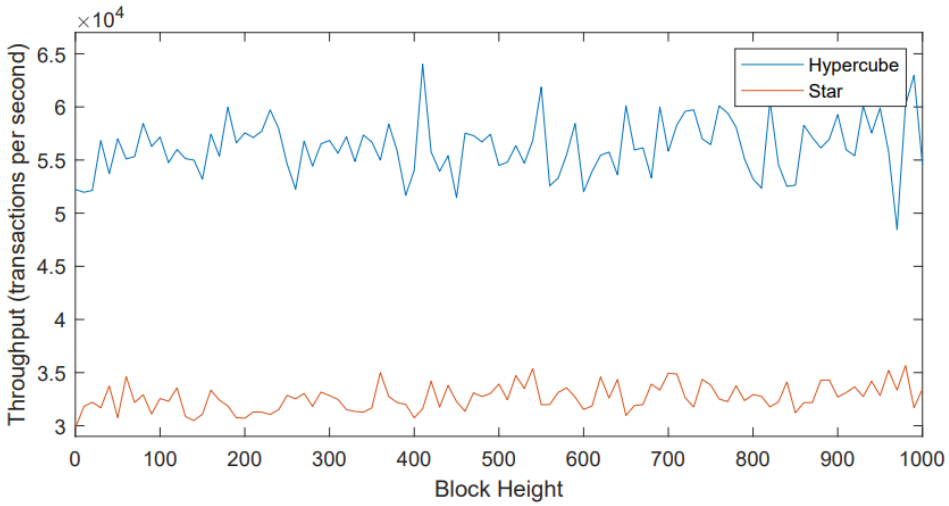

Figure 9 The throughput of the PPoV consensus protocol when N=4 with changing leader nodes.

As a non-forked BFT consensus protocol in the consortium blockchain, any attempt by an attacker to change the consistency of PPoV consensus ultimately leads to a longer consensus time or even a timeout, so we use throughput as an indicator. Our experiment generates an average of 60,000 transactions per second, each with a size of 24 Bytes. A block can store up to 10,000 transactions, and its maximum size is 235MB.

Due to resource limitations, we do not run any ordinary node. To evaluate the impact of physical topologies on the consensus process only, we turn off signature verification and transaction execution.

Figure 9 compares the efficiency of the consensus protocol for the basic hypercube-based topology and the star topology when a network size of $N = 4$. It can be seen that the throughput of the basic hypercube-based topology is better than that of the star topology and approximately equal to the transaction generation rate.

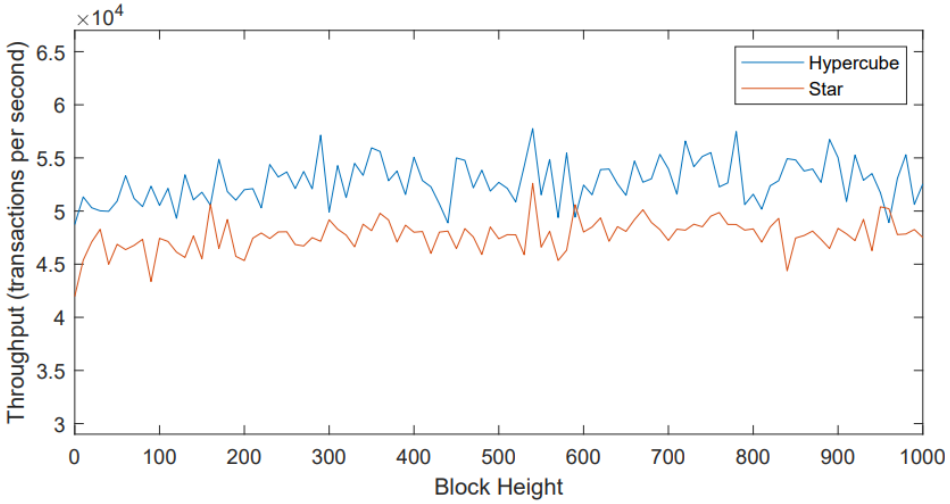

Figure 10 The throughput of the PPoV consensus protocol for N=16 with a fixed leader node.

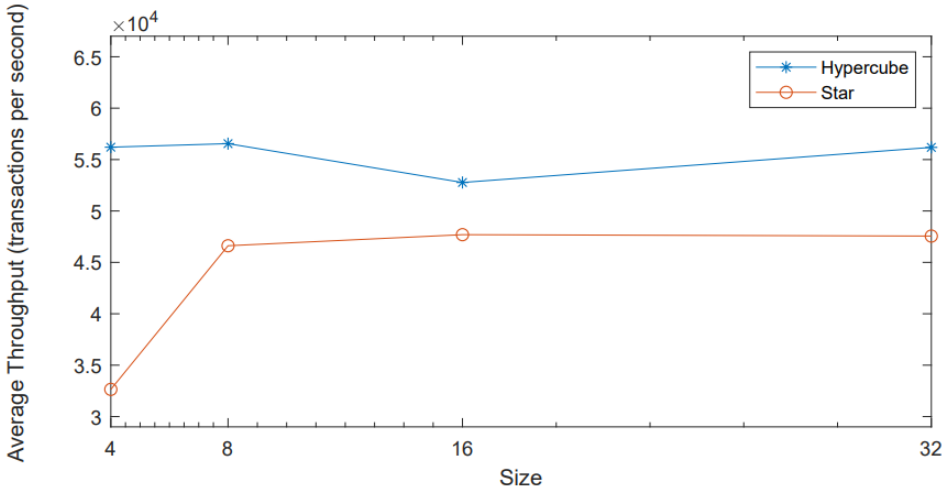

Figure 11 The average throughput of the PPoV consensus protocol in the whole network.

In the above 4-node experiment, the PPoV consensus algorithm periodically selects a super node as the leader node at random, which occupies a high bandwidth. In the following large-scale experiments, in order to obtain relatively stable and plausible throughputs, we realize the best-case scenario where the central node of the star topology is always the leader node during the measurement period and the rotation period is greater than 1000 consensus rounds. Figure 10 compares the efficiency of the consensus protocol on the hierarchical recursive topology of Figure 3 and the star topology when the network size is $N = 16$. The results in Figure 10 show that with a fixed leader node, the difference between the hypercube and star topologies is smaller than in Figure 9. Therefore, the proposed physical topology not only does not affect the performance of the upper-layer protocol, but also scales well.

According to the observation during the experiment, each super node can achieve 100% utilization of a single CPU under the above parameters, regardless of the number of nodes. In this case, the throughput is only affected by the network transmission rate. Figure 11 shows the average throughput in networks of different sizes. It can be seen that the performance of the consensus protocol on top of the hypercube-based topology is more stable compared to the star topology.

6 | CONCLUSIONS

In this paper, we propose a novel physical topology for consortium blockchains based on multi-dimensional hypercubes to optimize the partition tolerance. We also extend the basic topology to a hierarchical recursive topology with intermediate and short links. Through analyzing the partition tolerance by metrics of the partition tolerance probability and the average minimum repair time with the convergent Markov model and simulations on the digital optical cable communication system, we prove that the proposed topology meets better partition tolerance than the regular rooted tree topology and the ring lattice topology. Hierarchical recursion enables hypercube topologies to satisfy excellent partition tolerance with less network overhead. Experimental results from the prototype show that since the proposed topology is at the physical layer, there is no need to modify the upper-layer protocols. That is, blockchains constructed by the proposed topology can reach the CAP guarantee bound with appropriate transmission and consensus protocols satisfying strong consistency and availability.

ACKNOWLEDGMENTS

This work was supported by the National Keystone Research and Development Program of China [2017YFB0803204]; Foshan Innovation Team [2018IT100082]; Basic Research Enhancement Program of China [2021-JCJQ-JJ-0483]; China Environment for Network Innovation GJFGW [2020]386, SZFGW [2019]261; Guangdong Province Research and Development Key Program [2019B010137001]; Guangdong Province Basic Research [2022A1515010836]; Shenzhen Research Programs [JCYJ20220531093206015, JCYJ20210324122013036, JCYJ20190808155607340]; Shenzhen Fundamental Research Program [GXWD20201231165807007-20200807164903001]; ZTE Funding [2019ZTE03-01]; Huawei Funding [TC20201222002].

AUTHOR CONTRIBUTIONS

All authors contributed to the conception and design of the study. Material preparation and investigation were performed by Han Wang. Project administration and supervision were done by Hui Li and Ping Lu. Visualization was performed by Qiongwei Ye and Yong Yang. Peter Han Joo Chong and Xiaoli Chu contributed resources. The coding was done by Qi Lv. The first draft of this manuscript was written by Han Wang, and reviewing and editing were done by Abla Smahi. All authors read and approved the final manuscript.

FUNDING INFORMATION

None reported.

CONFLICT OF INTEREST STATEMENT

The authors declare no potential conflict of interests.

DATA AVAILABILITY STATEMENT

Data sharing is not applicable to this article as no datasets were generated or analyzed during the current study.

References

1. Nakamoto S. Bitcoin: A peer-to-peer electronic cash system. *Decentralized Business Review* 2008: 21260. doi: 10.2139/ssrn.3440802

2. Gilbert S, Lynch N. Brewer's conjecture and the feasibility of consistent, available, partition-tolerant web services. *SIGACT News* 2002; 33(2): 51–59. doi: 10.1145/564585.564601

3. Yuan Y, Wang FY. Towards blockchain-based intelligent transportation systems. 2016

4. Wang S, Yuan Y, Wang X, Li J, Qin R, Wang FY. An Overview of Smart Contract: Architecture, Applications, and Future Trends. In: : 108-113

5. Sompolinsky Y, Zohar A. Secure High-Rate Transaction Processing in Bitcoin. 2015

6. Lewenberg Y, Sompolinsky Y, Zohar A. Inclusive Block Chain Protocols. 2015

7. Sompolinsky Y, Lewenberg Y, Zohar A. Spectre: A fast and scalable cryptocurrency protocol. *IACR Cryptol. ePrint Arch.* 2016; 2016: 1159.

8. Yu H, Nikolic I, Hou R, Saxena P. OHIE: Blockchain Scaling Made Simple. 2020

9. Li C, Li P, Zhou D, et al. A decentralized blockchain with high throughput and fast confirmation. 2020

10. Castro M, Liskov B. Practical byzantine fault tolerance and proactive recovery. *ACM Trans. Comput. Syst.* 2002; 20(4): 398–461. doi: 10.1145/571637.571640

11. Bai Y, Zhi Y, Li H, Wang H, Lu P, Ma C. On Parallel Mechanism of Consortium Blockchain: Take PoV as an example. 2021

12. Vukolić M. The Quest for Scalable Blockchain Fabric: Proof-of-Work vs. BFT Replication. 2016

13. Delegated Proof of Stake (DPOS). report, 2019.

14. Armknecht F, Karame GO, Mandal A, Youssef F, Zenner E. Ripple: Overview and Outlook. 2015

15. Kiayias A, Russell A, David B, Oliynykov R. Ouroboros: A Provably Secure Proof-of-Stake Blockchain Protocol. 2017

16. Sukhwani H, Martínez JM, Chang X, Trivedi KS, Rindos A. Performance Modeling of PBFT Consensus Process for Permissioned Blockchain Network (Hyperledger Fabric). 2017

17. Gilad Y, Hemo R, Micali S, Vlachos G, Zeldovich N. Algorand: Scaling Byzantine Agreements for Cryptocurrencies. 2017

18. Yin M, Malkhi D, Reiter MK, Gueta GG, Abraham I. HotStuff: BFT Consensus with Linearity and Responsiveness. 2019

19. Lewis-Pye A, Roughgarden T. Resource pools and the cap theorem. *arXiv preprint* 2020. doi: 10.48550/arXiv.2006.10698

20. Schollmeier R. A definition of peer-to-peer networking for the classification of peer-to-peer architectures and applications. 2001

21. Li R, Asaeda H. DIBN: A Decentralized Information-Centric Blockchain Network. 2019

22. Jelasity M, Montresor A, Babaoglu O. Gossip-based aggregation in large dynamic networks. *ACM Transactions on Computer Systems (TOCS)* 2005; 23(3): 219-252. doi: 10.1145/1082469.1082470

23. Wood DD. Ethereum: A secure decentralised generalised transaction ledger. 2014.

24. Cachin C. Architecture of the Hyperledger Blockchain Fabric. 2016.

25. Decker C, Wattenhofer R. Information propagation in the Bitcoin network. 2013

26. Fadhil M, Owenson G, Adda M. A Bitcoin Model for Evaluation of Clustering to Improve Propagation Delay in Bitcoin Network. 2016

27. Fadhil M, Owenson G, Adda M. Locality based approach to improve propagation delay on the Bitcoin peer-to-peer network. 2017

28. Fadhil M, Owen G, Adda M. Proximity Awareness Approach to Enhance Propagation Delay on the Bitcoin Peer-to-Peer Network. 2017

29. Schlosser M, Sintek M, Decker S, Nejdl W. HyperCuP — Hypercubes, Ontologies, and Efficient Search on Peer-to-Peer Networks. 2003

30. Kokoris-Kogias E, Jovanovic P, Gasser L, Gailly N, Syta E, Ford B. Omniledger: A secure, scale-out, decentralized ledger via sharding. 2018.

31. Grover WD. High availability path design in ring-based optical networks. *IEEE/ACM Transactions on Networking* 1999; 7(4): 558-574. doi: 10.1109/90.793028

32. Serfozo R. *Basics of applied stochastic processes*. Springer Science Business Media . 2009

33. Sheskin TJ. A Markov chain partitioning algorithm for computing steady state probabilities. *Operations Research* 1985; 33(1): 228-235. doi: 10.1287/opre.33.1.228

34. Montresor A, Jelasity M. PeerSim: A scalable P2P simulator. 2009

How to cite this article: Han W, Hui L, Qiongwei Y, Ping L, Yong Y, Peter HJC, Xiaoli C, Qi L, and Abla S (2023), A Physical Topology for Optimizing Partition Tolerance in Consortium Blockchains to Reach CAP Guarantee Bound, *Trans Emerging Tel Tech*, *2023;00:1–6*.