Hyperclique: A Novel P2P Network Structure for Blockchain Systems

Xiaonan You, *Graduate Student Member, IEEE*, Shengli Zhang, *Senior Member, IEEE*, Taotao Wang, *Member, IEEE*, Soung Chang Liew, *Fellow, IEEE*

***Abstract*—This paper presents a novel P2P broadcast network called Hyperclique, which assigns a coordinate value to each peer in the network with adjustable dimensionality. Based on the concept of a clique, nodes whose coordinate values differ in only one dimension form one clique in Hyperclique. Hyperclique offers several advantages over the current P2P network topology, including a shorter network diameter and tighter network density. The node degree and average path length can be adjusted based on the network's actual conditions. Combining the coordinates with smart contract will result in extra-savings in both broadcast messages and convergence time. This paper provides a comparative analysis of the Hyperclique structure and other popular structured P2P topologies, demonstrating that Hyperclique outperforms them in improving the performance of P2P overlay networks. Additionally, the Hyperclique system supports various blockchain upper layers and is a promising and secure solution to improve the performance of blockchain networks.**

***Index Terms*—Peer-to-Peer, Overlay Network, Topology, Blockchain Network**

I. Introduction

The significance of blockchain as a secure, transparent, and decentralized infrastructure is becoming increasingly apparent as emerging application scenarios like the Metaverse [1] and Web3 [2] gain more traction. Serving as a pivotal technology for securing virtual world transactions and safeguarding virtual privacy and rights [3], [4], blockchain now faces the challenge of handling massive transactions in the expanding virtual world on an unprecedented scale. To meet this demand, substantial enhancements to the the existing level of throughput performance are imperative. Various factors, including network performance, consensus mechanism, block size, scalability solutions, and transmission latency, contribute to the throughput of blockchain. While many studies have focused on improving throughput through consensus mechanisms, the P2P network remains the bottleneck of the entire system [5], [6].

The performance of the P2P network directly impacts the speed of transaction and block propagation within the network [7]. By influencing the time it takes to broadcast messages to a majority of nodes, the P2P network imposes limitations on the speed at which consensus mechanisms can reach agreement. This limitation has serious implications, such as network fragmentation and blockchain forks. Speeding up block propagation can reduce the fork rate. [8], [9]. According to [6], in the Bitcoin [10] system, it takes an average of 12.6 seconds to broadcast a block to the entire network, 6.5 seconds to reach 50% of the nodes, and even after 40 seconds, 5% of the nodes may still not receive the block. Consequently, the performance of the P2P network sets an upper limit on the blockchain's throughput (TPS, transactions per second). Furthermore, P2P networks' broadcasting time also impacts the security of the blockchain. Large gaps in block arrival times at nodes can lead to vulnerabilities such as double-spend attacks, as observed in [6] and [11], while eclipse attacks target specific nodes at the network level, as mentioned in [12]. Therefore, it is crucial to enhance the performance of the P2P network layer in terms of both throughput and security.

From an engineering perspective, successful blockchain projects like Bitcoin [10] and Ethereum [13] have utilized existing P2P protocols with modifications as their network layer protocols. These protocols were originally designed for content distribution networks and instant messaging streams. And other projects, using POS, POA, etc., as their consensus mechanism, are also following the existing P2P protocol rather than proposing a new structure.

From an academic perspective, several works have aimed to enhance the P2P network of blockchain systems. For instance, [11] proposed a topological connectivity structure with hierarchical clustering, while [14] introduced a fractal-ring structure topology combined with geography. [15]presents algorithms for reaching the optimal topology dynamically using machine learning.

In our work, we propose a novel and scalable network layer protocol tailored specifically to blockchain. Our solution, Hyperclique, introduces a self-organizing protocol and a new blockchain broadcast topology. A Hyperclique consists of a few cliques with the same size, and all the nodes in each clique are fully connected. On the other hand, each node belongs to d cliques at the same time, hence connecting different cliques into a Hyperclique. In Hyperclique, every node forwards messages to its neighbor nodes in cliques achieving transaction broadcast. Therefore, Hyperclique ensures rapid message dissemination throughout the network within fixed time limits. With simple analysis, Hyperclique exhibits good property such as node symmetry, dimensional variability, a small diameter, security, and high interconnectedness.

In the following sections, we delve into the various aspects of Hyperclique. Section II explores related work about the current blockchain networks. Moving forward, Section

The research is partially supported by research grants from the Chinese NSF project (62171291). This work was funded by Shenzhen Key Research Project(JSGG20220831095603007, JCYJ20220818100810023, and JCYJ20220818101609021)

III provides a comprehensive explanation of Hyperclique, covering the system formulation, the coordinate system, and the procession of broadcasting through illustrative examples. Additionally, in Section IV, we delve into the theoretical performance, scalibility and security of Hyperclique and conduct a comparative analysis with other regular topologies. Finally, Section V presents our key findings and overall conclusions.

II. Related Works

Hyperclique is an overlay network, designed to support the broadcasting traffics in blockchain. An overlay network [16], [17] enables nodes to communicate, share resources or collaborate effectively regardless of the underlying physical network connections. It is created by establishing virtual connections between peers, thus forming an additional layer of connectivity.

Studies on network topologies [18]–[21] point out that, deterministic topologies are more intuitive and transparent than random topologies. Theoretically, a DHT-based system can guarantee that any data object can complete a lookup with an average hop count less than O(logN), where N is the number of peers in the system.

However, for blockchain systems, what expected from its network layer is quickly and reliably broadcasting transactions and blocks to all nodes in the network, rather than locating resources as in DHT.

Recent research [11], [14] have proposed new network layer protocols, such as ring topologies, hypercube topologies, hierarchical clusters, and geolocation. [15]presents algorithms for reaching the optimal topology dynamically using machine learning. Research studies [12], [22] suggest that nodes in the blockchain prefer to connect to highly connected nodes, but the network may experience temporary fragmentation if these 'popular' nodes are attacked or go offline.

III. Hyperclique

We proposes Hyperclique, a broadcast P2P network with structured topological connections, ensuring a predetermined upper bound on the time required for the broadcast process and the number of propagation rounds.

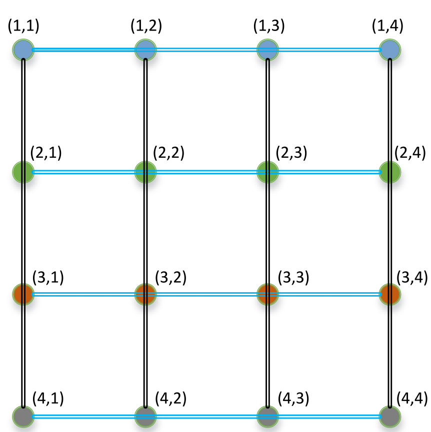

Fig. 1. A Hyperclique schematic diagram of a two-dimensional cliquesize=4

With reference to Fig. 1, the four nodes with labels $(1,1),(1,2),(1,3)$ and $(1,4)$ are fully connected and consist of a clique. A clique has the highest degree of connectivity. As in Fig. 1, there are 8 cliques and each clique has four nodes. Each node belongs to two cliques simultaneously, hence connecting the two cliques. In Fig. 1, each horizontal clique connects to the four vertical cliques through its four member nodes.

A. Illustrating Example

1) System Formulation: We consider a 3D example with N nodes. With a distributed or centralized protocol, each node is first randomly allocated a coordinate in each dimension, namely (i,j,k). The coordinates satisfy that $i,j,k \in \{1,...,M\}, M = N^{(1/3)}$.

With P2P node discovery and connecting protocols, each node, $s(i,j,k)$, connects to the peer nodes with the coordinate $(i,*,k)$, $(*,j,k)$,$(i,j,*)$, $* = 1,2,...,M$, to form the three cliques. As a result, all the peer nodes with two same coordinates, such as all the nodes like $s(i,j,*)$, connected with each other to form a clique. Finally, all the N nodes together form a Hyperclique.

2) Broadcasting with General Hyperclique: Suppose $s_0(i_0,j_0,k_0)$ needs to broadcast a message to all the nodes in the Hyperclique. We assume $s_0 \in \mathbf{I_0},\mathbf{J_0},\mathbf{K_0}$, where $\mathbf{I_0},\mathbf{J_0},\mathbf{K_0}$ are cliques as $\mathbf{I_0} = \{s_1(*_i,j_0,k_0) \mid *_i = 1,2,...,M\}$, $\mathbf{J_0} = \{s_1(i_0,*_j,k_0) \mid *_j = 1,2,...,M\}$, $\mathbf{K_0} = \{s_1(i_0,j_0,*_k) \mid *_k = 1,2,...,M\}$. The detailed and simple broadcasting protocol has three steps as follows:

1. Node $s_0(i_0,j_0,k_0)$ directly broadcast the message to its neighbor nodes, denoted as s_1, in the three cliques $\mathbf{I_0},\mathbf{J_0},\mathbf{K_0}$. Therefore, s_1 has coordinates like $(*_i,j_0,k_0),(i_0,*_j,k_0),or(i_0,j_0,*_k)$ as in Fig.2(b).
2. After receiving the message from s_0, each node s_1 forwards the message to its the other two cliques, where s_0 is not a member. For example, the nodes $s_1(i_1,j_0,k_0)$ will forward the message to nodes, s_2, with coordinates as $(i_1,*_j,k_0)$ and $(i_1,j_0,*_k)$. In this way, nodes located in the same "plane" will receive the message as in Fig.2(c).
3. Finally, the nodes s_2 will forward the received message to its last dimension cliques, where s_1 is not located. For example, the node $s_1(i_1,j_1,k_0)$ will forward the message to the nodes $s3$, with coordinate as $(i_1,j_1,*_k)$. As result, all the node in $s3$ will receive the message, for three times, as in Fig.2(d).

To implement the above protocol, each node in the Hyperclique works as follows. If the node is a source node (the message is generate by itself), it will simply broadcast the message to its neighbors in the three cliques. If it receive the same message at least three times, it will simply keep the message. If it receive the message for one or two times, it will forward the message to cliques different from the senders. For example, suppose node s is located at coordinates (i_0,j_0,k_0), and it receives a message twice from nodes at coordinates (i_1,j_0,k_0) and (i_0,j_1,k_0). Node s should forward the message to the nodes in clique different from the two senders, i.e., nodes with coordinates (i_1, j_1, k_*). The broadcasting logic of a node is given in Algorithm 1. This broadcasting scheme can be extended to Hypercliques of any dimension, where the last round of broadcast will have $(c-1)^n$ nodes receiving n copies of the message, but the total round of broadcasts will not exceed n. It is a future work to optimize the protocol so as to reduce the redundant copies of the message.

Algorithm 1 Broadcasting Logic for Each Node

```
for Every broadcasting round do
    if Node is the initial broadcast node then
        Forward message to every clique which it is located
    else Count the sum number R of received message(s)
        if R <Hyperclique dimension and≠ 0 then
            Own coordinates⊕source node(s)' coordinates
            Forward to the cliques that XOR result is 0 ▷ Forward message to the clique(s) that source node(s) is(are) not in
        else No need to forward the message
```

The six-degree separation theory and small-world effects suggest that the maximum distance between nodes in a complex network is typically no more than six. Thus, we set the maximum dimension limit of the Hyperclique system to 6, in line with this theoretical prediction. The nodes in the Hyperclique system are self-organizing without a server or super node, making it fitting for the blockchain's property of distribution.

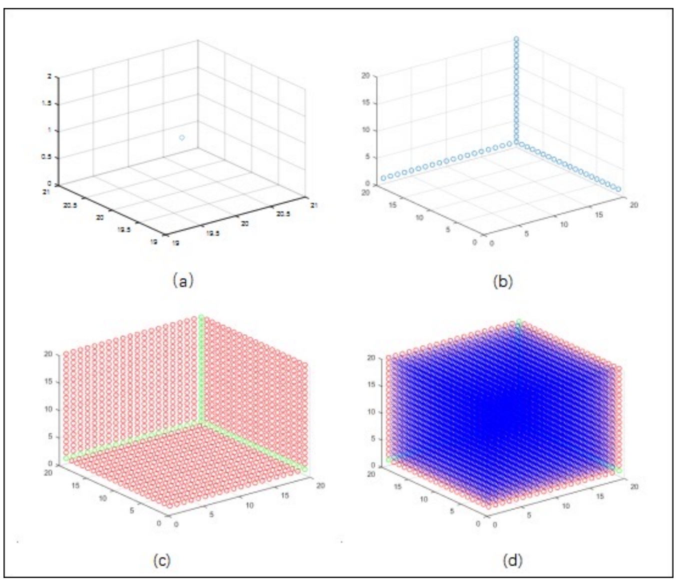

Fig. 2. Broadcast process

B. Topology Management

Our Hyperclique is designed to support a blockchain, where we can design and deploy a smart contract to manage the topology of Hyperclique. The Hyperclique network has two parameters, c : clique size, d : dimension. Depending on the number of nodes in the system and the load capacity of the network, the dimension and clique size of Hyperclique can be adjusted. The basic function of the Hyperclique management smart contract includes Hyperclique initialization, nodes joining and nodes leaving. The management smart contract, with **supplement**, **sign-out** and **queuing** capabilities, will maintain a coordinate table of node IDs against coordinates, as long as a queue for the waiting nodes.

1) Initialization: When a blockchain is deployed, the Hyperclique is initialized at the same tinme. For the sake of simplicity, assume a fixed number of N nodes are nominated to apply Hyperclique as in some PoS blockchains. For a 2-D Hyperclique the node coordinates are like (i, j), for a 3-D Hyperclique the node coordinates are like (i, j, k), or they can have higher dimensionality. Each nodes are randomly assigned one index from 1 to c^d.

Upon initial construction of the network, the table is blank and there are a total of c^d unoccupied positions. We use a compact coordinates generation algorithm 2 to calculate the coordinates of each node, where the coordinates are calcualted from the index according to a Fibonacci alignment. After initialization, the full table are known to all nodes as a genesis configuration, so that all the nodes in one clique will connected each other.

Algorithm 2 2D Compact Coordinate

```
Input: x : the index of node
Output: (i, j) : the coordinates
m = ⌊sqrt(x)⌋
n = x − m^2
if n == 0 then
    (i, j) = (m, m)
else if n − m > 0 then
    (i, j) = (m + 1, n − m)
else n − m <= 0
    (i, j) = (n, m + 1)
```

queuing When a node want to join the Hyperclique, it will call the **queuing** function of the smart contract, and waiting in the first-in, first-out (FIFO) queue. When there are nodes leaving or off line, the first node in the queue will be called to join the Hyperclique. There is also a maximum limit on the size of the queue.

2) Nodes Supplement: As long as there is a blank position in the coordinate table, the first node in the queue will be filled into the table. This event will be known by all the nodes since the table is in the smart contract. Then, the new node can connect to all the other nodes in its cliques, so as to finish the nodes supplement procedure.

3) Nodes Leave: There may be two cases for nodes leave. In the first case, a node actively leaves the Hyperclique by calling the **sign-out** function. This function will automatically call the **Nodes Supplement** procedure to fill in the blank position.

In the second case, a node may unexpectedly leave the Hyperclique, due to some reasons. To detect this event, each node need to send a periodic heartbeat to its neighboring peers within the same cliques. The primary objective of the heartbeat packet is to signal that the node is operational and available for communication. Failure to transmit the heartbeat packet is interpreted as an indication that the node has either left the network voluntarily or has gone offline. As long as one node is reported to leave by more than t nodes, the smart contract will call the **sign-out** function of this node.

IV. Hyperclique Performance

In this section, we try to analyze the properties and performance of Hyperclique from blockchain point of view. In current blockchain broadcasting environment, network islands or bottlenecks of bridge nodes can prevent some broadcast content from being disseminated to all nodes. This can result in some nodes quickly receiving messages and reaching a consensus, while others are affected by network delays and deviate from the majority. Moreover, message retention in the network can lead to a waste of network resources, as nodes continue to broadcast the message to their neighbors. Nodes with a high degree (super nodes) are also present in both Bitcoin and Ethereum, making the networks vulnerable to eclipse and Sybil attacks. The offline status of a bridge node may lead to short-term network fragmentation.

Differently, the special structure of Hyperclique can help to overcome the existing shortcomings. In Hyperclique, the message can cover all nodes within an fixed limit of broadcast rounds and time. Each node in Hyperclique are symmetric and has the same degree. With the proposed broadcasint algorithm, the nodes far from the source will receive the message from more peers, resulting in the receive reliability and low latency for the network edge.

A. Hyperclique Properties

In graph theory, a clique is a subset of vertices of an undirected graph such that every two distinct vertices in the clique are adjacent. In other words, a clique is a complete subgraph of the original graph. Clique structures are important in graph theory because they provide insights into the topology of the network, including its density and connectivity. Given a clique with c nodes and a Hyperclique of d dimensions, the numbers of summary nodes can be expressed as $N = c^d$. The total number of edges is the product of the number of edges in the clique and the number of cliques

$$E = \frac{c(c-1)}{2} \cdot c^{d-1} d$$

1) Degree and density: The degree of a node in a Hyperclique can be expressed as $k = \sum_{i \neq j}^{N} e_{ij}$, where e_{ij} represents the connection between node i and node j in the Hyperclique. Generally, the average degree is expressed as

$$k' = 2E/N = (c-1) \cdot d$$

where E represents the number of all edges , and N represents the number of network nodes. As the Hyperclique structure is symmetric, each node in the network possesses an equivalent degree, which is k'.

The network density, $\rho = 2E/N(N-1)$, is a metric that describes the number of actual connections E between nodes relative to the total number of possible connections. This measure can provide valuable insights into the network's robustness, efficiency, and security. A highly dense network is more resistant to failures, as there are more alternate paths available to reroute traffic. It also tends to be more efficient, as the direct connections between nodes reduce the number of hops required for communication. Additionally, network density can affect security, as a sparse network may be more vulnerable to attacks due to a smaller number of nodes to distribute the impact. The density of the clique structure is 1, Hyperclique is orthogonal to multiple cliques, and its density is quite high. By applying the formula to calculate network density, we can derive the density of a hypercube network.

$$\rho_{hc} = \frac{(c-1)d \cdot c^d}{c^d(c^d-1)} = \frac{(c-1)d}{c^d-1}$$

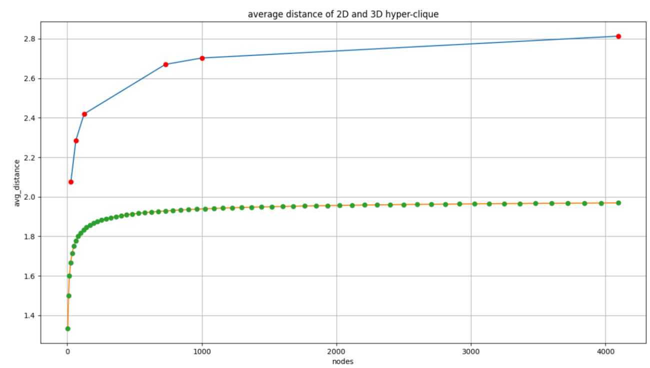

Fig. 3. Average distance in 2D and 3D Hyperclique

2) Network diameter and path length: The network diameter is defined as the maximum distance between any pair of nodes in the network, where distance is measured in terms of the shortest path between the nodes, i.e., $D = \max d_{ij}$, where d_{ij} represents the number of edges passed on the shortest path between node i and j. The network diameter affects the latency of communication and response times. Generally, a smaller diameter implies faster and more reliable communication between nodes and greater network efficiency. The network diameter of Hyperclique is exactly the dimension of Hyperclique, $D = d$.

The average path length, denoted by L, of a network is commonly defined as the average distance between any two nodes in the network. It is mathematically expressed as

$$L = \frac{1}{\frac{1}{2}N(N-1)} \sum_{i \neq j} d_{ij}.$$

In the case of hypercube structures, the average path length is jointly determined by the dimension and the size of the clique.

$$L = \frac{1}{\frac{1}{2}N(N-1)} \sum_{i=1}^{6} i \times SUM_{li},$$

where SUM_{li} is the sum number of node pairs with distance i.

$$SUM_{li} = \frac{C_d^i \cdot (c-1)^i \cdot c^d}{2}.$$

3) Clustering coefficient: The clustering coefficient is an index to describe the local connectivity of the network, which can characterize the community structure and tightness of the network. In a network, the higher the clustering coefficient of a node, the closer the neighbor nodes of the node are, which usually means that there is a more obvious community structure in the network. The clustering coefficient of a node in a network can be expressed mathematically as follows:

$$C_i = 2T_i/(k_i \cdot (k_i - 1))$$

where C_i is the clustering coefficient of node i, k_i is the number of neighbors (i.e., degree) of node i, T_i is the number of triangles (i.e., three nodes that are all connected to each other) that contain node i. In the context of blockchain networks, the clustering coefficient can be used to assess the level of decentralization of the network. A higher clustering coefficient suggests that there are more interconnected nodes and thus a higher level of decentralization, while a lower clustering coefficient suggests that the network is more centralized.The clustering coefficient of the nodes in Hyperclique is related to the dimension and clique size($c >= 2$), expressed as

$$C_i = \frac{2d \cdot C_{c-1}^2}{k_i(k_i - 1)} = \frac{(c-2)}{2(dc - d - 1)}.$$

B. Scalability

From the node degree and message overloads viewpoint, Hyperclique is scalable.

1) node degree: The degree of nodes in Hyperclique can be expressed as $k = d \cdot (c-1) = d \cdot (N^{1/d} - 1)$, where $c = N^{1/d}$. Given N, k and d appear to be a decreasing curve for $d \in [1, 6]$. As a result, as the dimension of the network increases, the degree of each node decreases. Increasing the dimension of the network can improve its scalability, but this may come at a cost of extending the coverage time due to more broadcast rounds, especially for low-dimensional networks.

Algorithm 3 Broadcasting Procedure

Broadcast message from the message source node.
Let c be the clique size and d be the dimension of the network ($d_{max} = 6$).
for $i = 1$ to $(d-1)$ **do** ▷ The i^{th} round of broadcasting
  for each of the $C_d^{i-1}(c-1)^{i-1}$ receiving nodes **do**
    Send 1 message to $(d - i + 1)(c - 1)$ nodes.

2) Total messages: The broacast precedure of Hyperclique is given in Algorithm 3 when broadcasting one message over the network. In particular, there are $C_d^i(c-1)^i$ nodes receiving the message in the i^{th} round, and each node receiving $i - 1$ copies the message.

The proposed broadcast structure is deterministic and follows a set of rules that determine the quantity of messages transmitted during each round of broadcast. This quantity is directly correlated with the dimension d of the Hyperclique and the clique size c of the Hyperclique. Table 1 illustrates the number of messages in each round of broadcasting, which is determined by the aforementioned factors.

TABLE I
NUMBER OF MESSAGES

Broadcast Rounds	Number of Messages
1^{st}	$d(c-1)$
2^{nd}	$d(d-1)(c-1)^2$
3^{rd}	$3C_d^3(c-1)^3$
4^{th}	$4C_d^4(c-1)^4$
5^{th}	$5C_d^5(c-1)^5$
6^{th}	$6C_d^6(c-1)^5$

C. Security

The P2P layer of blockchain system is particularly vulnerable to Sybil and Eclipse attacks. Several solutions have been proposed to address Sybil and Eclipse attacks in blockchain networks. For Sybil attacks, [23] detecting Sybil nodes by evaluating the witness peers of the nodes and using reputation systems and identity verification in combination with consensus mechanisms can help to restrict the behavior of malicious nodes. For Eclipse attacks, [14] proposed using Intel SGX to guarantee nodes' behavior integrity and incorporating pattern obfuscation to prevent traffic pattern analysis, which can make it more difficult for attackers to recognize vulnerable nodes. Hyperclique can support these solutions and adapt some of them to enhance the security of its blockchain architecture.

D. Compared with other topologies

Ring and tree structures are traditional topologies widely used by regular P2P networks. Fractal rings are still being tried in blockchain by [14]. Toshniwal's research [24] compares various topologies, including trees and hypercubes. Studies [24], [25] have shown that hypercube structures offer high connectivity and fault tolerance in P2P networks. While hypercube topology is commonly used in high-performance computing systems, it has also been proposed for use in overlay networks. As a structured network topology, some researchers [26]–[28] have incorporated hypercube-like structures into blockchain

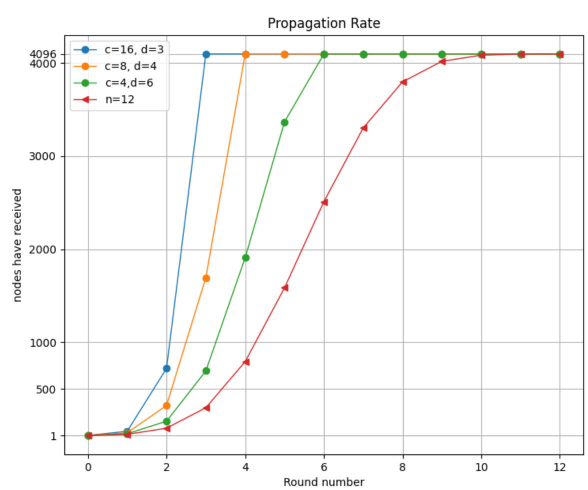

Fig. 4. Propagation rate

Therefore we mainly compare our Hyperclique performance with hypercube(n=12). As illustrated in Figure 4, when the number of nodes is fixed(4096), Hypercliques require fewer broadcast rounds to achieve full network coverage than hypercubes, with more than half fewer rounds for 3D, 4D, and 6D Hypercliques. Additionally, Hypercliques have a much higher average node degree, with $< k >$ values of 28, 18, and 12 for 3D, 4D, and 6D, respectively. This higher degree can be achieved with lower dimensions, reducing the network diameter. Conversely, if a high degree is not required, Hypercliques can be designed with a higher dimension. Nonetheless, the broadcast rounds of Hypercliques is still less than that of hypercubes.

Moreover, Table II presents a comparison of the network properties for Hyperclique, hypercube, ring and tree topologies. Our results show that when the number of nodes exceeds $2^6 = 64$ nodes, especially up to $2^{12} = 4096$ nodes, Hyperclique structures require the least number of broadcast rounds to reach the entire network, while still maintaining a controllable network diameter.

TABLE II
COMPARISON

Topology	Hyperclique	hypercube	ring	tree
Size	N	N	N	N
Degree	$(c-1)log_c^N$	log_2^N	log_2^N	Fixed value
Edges	$(c-1)Nlog_c^N/2$	$Nlog_2^N/2$	$Nlog_2^N$	$Nlog_2^N$
Diameter	log_c^N	log_2^N	log_2^N	log_2^N

c is the clique size of Hyperclique

V. Conclusion

This paper presents Hyperclique, a novel P2P broadcast structure that exhibits excellent deterministic convergence. This structure is particularly suitable for compact and message-intensive blockchain systems. The Hyperclique structure is highly scalable due to its variable dimensions, enabling nodes to reach fast consensus. Compared to other structured topologies, Hyperclique offers better scalability and convergence. This is particularly advantageous for blockchain systems to support consensus protocols.

References

[1] J. Lee, I. Yeo, and H. Lee, "Metaverse current status and prospects: Focusing on metaverse field cases," in *2022 IEEE/ACIS 7th International Conference on Big Data, Cloud Computing, and Data Science (BCD)*, 2022, pp. 332–336.

[2] W. Gavin, "Introduction to web3.0," 2023, https://ethereum.org/en/web3.

[3] T. Huynh-The, T. R. Gadekallu, W. Wang, G. Yenduri, P. Ranaweera, Q.-V. Pham, D. B. da Costa, and M. Liyanage, "Blockchain for the metaverse: A review," *Future Generation Computer Systems*, vol. 143, pp. 401–419, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167739X23000493

[4] B. Guidi and A. Michienzi, "Social games and blockchain: exploring the metaverse of decentraland," in *2022 IEEE 42nd International Conference on Distributed Computing Systems Workshops (ICDCSW)*, 2022, pp. 199–204.

[5] E. Kolyvas and S. Voulgaris, "Cougar: Fast and eclipse-resilient dissemination for blockchain networks," in *Proceedings of the 16th ACM International Conference on Distributed and Event-Based Systems*, ser. DEBS '22. New York, NY, USA: Association for Computing Machinery, 2022, p. 7–18.

[6] C. Decker and R. Wattenhofer, "Information propagation in the bitcoin network," in *IEEE P2P 2013 Proceedings*, 2013, pp. 1–10.

[7] S. M. Varma and S. T. Maguluri, "Throughput optimal routing in blockchain-based payment systems," *IEEE Transactions on Control of Network Systems*, vol. 8, no. 4, pp. 1859–1868, 2021.

[8] Y. Shahsavari, K. Zhang, and C. Talhi, "A theoretical model for fork analysis in the bitcoin network," in *2019 IEEE International Conference on Blockchain (Blockchain)*, 2019, pp. 237–244.

[9] J. Misic, V. B. Misic, X. Chang, S. G. Motlagh, and M. Z. Ali, "Block delivery time in bitcoin distribution network," in *ICC 2019 - 2019 IEEE International Conference on Communications (ICC)*, 2019, pp. 1–7.

[10] N. Satoshi, "Bitcoin: A peer-to-peer electronic cash system," 03 2009, cryptography Mailing list at https://metzdowd.com.

[11] W. Hao, J. Zeng, X. Dai, J. Xiao, Q.-S. Hua, H. Chen, K.-C. Li, and H. Jin, "Towards a trust-enhanced blockchain p2p topology for enabling fast and reliable broadcast," *IEEE Transactions on Network and Service Management*, vol. 17, no. 2, pp. 904–917, 2020.

[12] T. Cao, J. Yu, J. Decouchant, X. Luo, and P. Verissimo, "Exploring the monero peer-to-peer network," Cryptology ePrint Archive, Paper 2019/411, 2019, https://eprint.iacr.org/2019/411. [Online]. Available: https://eprint.iacr.org/2019/411

[13] V. Buterin, "Ethereum white paper:a next-generation smart contract and decentralized application platform," 2014.

[14] H. Qiu, T. Ji, S. Zhao, X. Chen, J. Qi, H. Cui, and S. Wang, "A geography-based p2p overlay network for fast and robust blockchain systems," *IEEE Transactions on Services Computing*, pp. 1–14, 2022.

[15] Y. Mao, S. Deb, S. B. Venkatakrishnan, S. Kannan, and K. Srinivasan, "Perigee: Efficient peer-to-peer network design for blockchains," in *Proceedings of the 39th Symposium on Principles of Distributed Computing*, ser. PODC '20. New York, NY, USA: Association for Computing Machinery, 2020, p. 428–437.

[16] E. K. Lua, J. Crowcroft, M. Pias, R. Sharma, and S. Lim, "A survey and comparison of peer-to-peer overlay network schemes," *IEEE Communications Surveys & Tutorials*, vol. 7, no. 2, pp. 72–93, 2005.

[17] D. Doval and D. O'Mahony, "Overlay networks: A scalable alternative for p2p," *IEEE Internet Computing*, vol. 7, no. 4, pp. 79–82, 2003.

[18] B. D. Anderson and M. Ye, "Recent advances in the modelling and analysis of opinion dynamics on influence networks," *Int. J. Autom. Comput.*, vol. 16, no. 2, p. 129–149, apr 2019. [Online]. Available: https://doi.org/10.1007/s11633-019-1169-8

[19] K. Lu, "Research on theory and application of solving central node in complex networks," in *2022 International Conference on Big Data, Information and Computer Network (BDICN)*, 2022, pp. 802–806.

[20] B. Tao, I. W.-H. Ho, and H.-N. Dai, "Complex network analysis of the bitcoin blockchain network," in *2021 IEEE International Symposium on Circuits and Systems (ISCAS)*, 2021, pp. 1–5.

[21] M. Grundmann, M. Baumstark, and H. Hartenstein, "On the peer degree distribution of the bitcoin p2p network," in *2022 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, 2022, pp. 1–5.

[22] M. Conti, E. Sandeep Kumar, C. Lal, and S. Ruj, "A survey on security and privacy issues of bitcoin," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 4, pp. 3416–3452, 2018.

[23] P. Swathi, C. Modi, and D. Patel, "Preventing sybil attack in blockchain using distributed behavior monitoring of miners," in *2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, 2019, pp. 1–6.

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

[25] X. Huang, A. F. Ramos, and Y. Deng, "Optimal circulant graphs as low-latency network topologies," 2022. [Online]. Available: https://arxiv.org/abs/2201.01342

[26] H. Wang, H. Li, Z. Wang, B. Han, M. Zhang, P. H. J. Chong, X. Chu, Y. Liu, S.-Y. Liew, and L. Wuttisittikulkij, "A p2p network topology for optimizing partition tolerance to reach the cap guarantee bound in consortium blockchains," 2021. [Online]. Available: https://arxiv.org/abs/2109.02907

[27] V. Aradhya, S. Gilbert, and A. Hobor, "Overchain: Building a robust overlay with a blockchain," 2022. [Online]. Available: https://arxiv.org/abs/2201.12809

[28] M. Zichichi, L. Serena, S. Ferretti, and G. D'Angelo, "Towards decentralized complex queries over distributed ledgers: a data marketplace use-case," in *2021 International Conference on Computer Communications and Networks (ICCCN)*, 2021, pp. 1–6.