Scenario Analysis of Decentralized Social Networks

Chenle Xiong

***Abstract*— P2P network has been widely used in media on demand, file sharing and other fields, but the application of P2P technology to the social field is very small. Traditional social networks have centralized servers, and P2P social networks completely remove centralized servers. Therefore, this paper briefly introduces P2P network technology, BT protocol, B coding, seed file, BT client, Kadmelia algorithm, and conducts further research on the application of P2P technology to social network through seven social scenarios.**

***Index Terms*—P2P network, social network, seed file, caching mechanism.**

I. INTRODUCTION

With the rapid development of the mobile Internet, online social networking has become an extremely important part of people's lives. Centralized social networks have absolute control over users' data, and the issue of privacy leaks has made people more and more concerned. The application of P2P technology to social networks can achieve decentralization. The nodes in the network are equal to each other, and the nodes act as both client and server roles in the network. The speed of data transmission in social networks and the security of data transmission are of the utmost importance. According to 2021 statistics, 28% of Internet traffic comes from social media [1]. In order to ensure the smoothness and security of data transmission, it is necessary to design a reasonable blockchain social network caching mechanism.

Rajesh Sharma and Anwitaman Datta designed a decentralized online social network architecture based on super peers [2], the main solution is to ensure the availability of data even when the data owner is not online, so that others can The required data can be accessed even when the node is offline or down. In this architecture, there will be a super node to help nodes find other nodes to store their content. Compared with ordinary nodes, super nodes are nodes with more bandwidth, more storage space, and higher availability. Shirin Nilizadeh [3] proposed a decentralized social network architecture Cachet that uses cache to protect privacy, which retains the main functions of online social networks while providing strong security and privacy guarantees, using a mix of structured and unstructured A paradigm is used to ensure the efficient dissemination and retrieval of data. And use traditional distributed hash table to increase social connection between users.

Manuscript received July 20, 2022
Chenle Xiong School of Computer Science and Technology,Tiangong University,China.

II. MATH

Bencoding encoding is the encoding method used by BitTorrent to transmit data structures, also called B encoding. It consists of four types, namely strings, integers, lists, and dictionaries[4], As shown in Figure 1.

String type Bencoding: <length of string represented by decimal ASCII>:<string>. It starts with the length of a string, followed by a colon, followed by the actual string, e.g. 8:announce for the string "announce". Integer type Bencoding encoding: i<decimal ASCII-encoded integer>e. It starts with an 'i' for the delimiter, followed by the decimal ASCII-encoded integer value, and ends with an 'e' for the end delimiter. For example, i6789e represents the integer 6789; i-6789e represents the integer -6789; where i0e represents the integer 0, and i-0e is invalid and cannot represent the integer 0. The integers starting with 0 are invalid, such as i06789e, which cannot represent the integer 06789. List type Bencoding encoding: l<type of encoding>e. It starts with 'l' as a delimiter, followed by a list value encoded in Bencoding, which can be an integer, string, list, dictionary, and ends with an 'e' as an end delimiter. For example: l8:announce4:infoe means the two strings "announce" and "info". Dictionary type Bencoding encoding: d<encoding string><encoding element>e. It starts with 'd', the middle is the Bencoding encoding string, the value is any Bencoding encoding type, and finally ends with 'e'. For example, d6:lengthi93e3:sex5:womane means {“length”=93,”sex”=”woman”}.

TABLE I. B ENCODING TYPE

Type	Describe	Format	Example
string	character sequence	<string length in decimal ASCII>:<string>	8:announce represents the string "announce"
integer	decimal integer	i<decimal ASCII encoded integer>e	i6789e means integer 6789
list	List of Type B	l<type of encoding>e	l8:announce4:info e means the two strings "announce" and "info"
dictionary	an array of key-value pairs	d<encoded string><encoded element>e	d6:lengthi93e3:sex 5:womane means {"length"=93,"sex" ="woman"}

III. RELATED WORK

In order to solve the problems in the centralized social network, this paper designs some real social scenarios to analyze how to solve the decentralized social network[5].

Scenario 1: Suppose an unregistered user wants to join a blockchain social network. That is, the problem of adding new nodes to the network.

Scenario 2: After a user joins the network, how to let nearby users quickly learn and how to quickly learn about

nearby users.

Scenario 3: After a user joins the network, how to add friends to a stranger, and what to do when the stranger is online or offline.

Scenario 4: How users view online users and offline users.

Scenario 5: How users view square content.

Scenario 6: After an old user goes online, how to quickly obtain the messages sent to him by other users when he is offline.

Scenario 7: When a user sends information to a group, how to ensure that other members in the group receive the information.

Since there is no central server, these scenarios are all problems that need to be solved by decentralized social networks. Ensuring the smooth operation of the network is the first and foremost, and the information in the network needs to be cached by the nodes in the network[6].

IV. FEASIBILITY ANALYSIS

Scenario 1: A new node joins the network for the first time. Assuming that node A joins the decentralized social network DSN for the first time, and node A gets in touch with node B already in the decentralized social network conn, then node B is the boot node of node A and is responsible for the initialization of newly added node A. Contacting conn refers to obtaining information about a node that is already in the decentralized social network DSN, no matter what method is used. From a logical point of view, Scenario 1 is achievable. Note: Bootstrap node refers to the bootstrap node of a node. For a node A, it is called a bootstrap node. Every node in a decentralized social network may be a bootstrap node.

Scenario 2: When node A joins the network, it first establishes a connection with the bootstrap node. Node A initializes its routing table with the routing information of the bootstrap node, and then node A takes itself as the target and performs a complete query operation. After the query operation is completed, node A also obtains the closest node B to itself in the nearest node network, and finally node A refreshes its routing table by using the routing table content of node B. From a logical point of view, the second scenario is achievable.

The function of the last refresh is: because node A is already the closest node to node B, for node A, it is how to obtain node information farther than node B, that is, how to let more people know the existence of node A , and the routing table of node B can just be provided. During the process of refreshing the routing table, it also inserts its own information into the routing table of other nodes, so that more people know the existence of node A.

Scenario 3: Node A wants to add unfamiliar node C as a friend. Node A searches for the IdC in the network, keeps searching through the routing table, and finally finds the online node C. If node C is not online, it keeps searching through the routing table, but no offline node C is found. At this time, node A sends request Q to nearby nodes. Request Q refers to asking whether the node has cached IdC information. Finally, the online node D that caches the information of the offline node C can be found. Node A sends an inquiry request to node D to view the information of node C. Through the above operations, the offline stranger C is finally brushed out. From a logical point of view, Scenario 3 is achievable.

Scenario 4: How does user A view online and offline users around in the decentralized social network DSN. Each node actually caches the information of some offline nodes. As long as a new node joins the network, its own routing table will be updated. As long as the query request is continuously sent from the routing table, it can find the online and offline people around. From a logical point of view, Scenario 4 is achievable.

Scenario 5: How does user A view the content in the square. Suppose user A views the content of node B in the square. The content in the square is visible to all users, and the amount of data in this part is huge. If all of them are stored in the blockchain, the data read and write efficiency is not high, and the storage performance is not high, which will affect the system. performance requirements. Combining blockchain and IPFS technology to solve this part of content storage can greatly improve system performance. The data content and content ID are actually stored in IPFS, IPFS maps the data content to a unique hash value, and the blockchain is responsible for storage. This hash value is sufficient. According to the above analysis, it is possible for user A to view the content in the square.

Scenario 6: Assuming that after the old user A goes online, how to quickly obtain the message M sent to him by node B when he is offline. When B sends information M to user A, and user A is offline, user C needs to cache information M. User C knows that information M needs to be sent to A according to the ID of information M, and user C will be in the network. Real-time monitoring of whether user A is online, once user A is online, user C will immediately send information M to user A.

Scenario 7: Suppose user A is in a group with user B, user C, and user D. User A sends information M in the group. How is the information M sent to user B, user C, and user D? First, the user ABCD is in a group, establishes a relationship with the IDs of the four users, and indexes them together to obtain a group ID number. User A sends a message in the group, and the ID numbers of the other three users can be found according to the group ID, so that the information M is sent to the hands of these three users. According to logical analysis, scenario 7 is achievable.

V. CONCLUSION

According to the analysis of seven social scenarios, it can be concluded that a decentralized social network is achievable. Since the data in the network is cached by the nodes in the network, we will propose a set of effective caching mechanism in the follow-up work to ensure the normal operation of the network.

REFERENCES

[1] Beibei Niu,Jinzheng Ren,Xiaotao Li. Credit Scoring Using Machine Learning by Combing Social Network Information: Evidence from Peer-to-Peer Lending[J]. Information,2019,10(12).

[2] Nilizadeh S , Jahid S , Mittal P , et al. Cachet: a decentralized architecture for privacy preserving social networking with caching. 2012.

[3] J. Wang, "Fundamentals of erbium-doped fiber amplifiers arrays (Periodical style—Submitted for publication)," *IEEE J. Quantum Electron.*, submitted for publication.
[4] Yu Lin Du. Risks and Management of P2P Online Lending[J]. Advanced Materials Research,2014,3181(926-930).
[5] He Fei,Chen Mei Lian,Tang Rong,Li Ming Yong. Construction and Algorithm Analysis of a Social Network System Structure Based on P2P[J]. Applied Mechanics and Materials,2013,443(443-443).
[6] [1]Li Yan,Chen Zhuo,Zhang Hua. Improving Sharing Efficiency in Online Short Video System through Using P2P Based Mechanism[J]. Procedia Engineering,2012,29(C).

Chenle Xiong School of Computer Science and Technology,Tiangong University,China.