Supernode selection mechanism based on location information

Yongqiong Zhu
School of Art, Wuhan Business University, Wuhan, China
yongqiongzhu@gmail.com

Abstract. In a hierarchical p2p network such as Gnutella0.6, supernodes are selected high-capacity nodes. As the central server of leaf nodes, they process queries more efficiently than a single leaf node. So how to effectively select supernodes and construct a supernode network topology is a key issue in hierarchical P2P networks. Existing research shows that this mechanism of peers randomly selecting logical neighbors without any underlying physical connection information causes a mismatch between the P2P logical overlay network and the underlying physical network. The same message traverses the same physical connection multiple times. A lot of unnecessary network traffic is generated. Therefore, this paper proposes a super node selection mechanism based on distance information. When selecting a super node, the actual physical distance between the nodes must be detected, so that the selected super nodes are logically adjacent, and their physical locations are as close as possible, thereby improving the processing capacity of the network.

1. Introduction
P2P is an overlay network that is built on the Internet. It breaks the traditional C/S model and weakens the concept of the server. Each node in the system no longer distinguishes the role relationship between the server and the client, and the nodes can directly communicate with each other. Exchange resources and services without having to go through the server. The resources and services in the network are scattered on all nodes, and the status of each node is equal, serving as a server, providing services to other nodes, and enjoying the services provided by other nodes. According to the topology of the network, P2P can be divided into three categories: centralized P2P systems, fully distributed unstructured P2P systems, and fully distributed structured P2P systems. In a fully distributed unstructured P2P system, Gnutella 0.4 is its typical representative.

In Gnutella 0.4 [1], point-to-point connections are randomly established between nodes to form an overlay. Each node has one or more neighboring nodes and maintains the connection status with its neighboring nodes. When a node joins or leaves the system, ping and pong messages are used to maintain the topology of the system. Gnutella 0.4 uses flooding to send query messages to all its neighbors. When its neighbors receive the query message, they first check whether they have the required data. If so, they return a response message along the original path. Otherwise, they continue to use flooding to forward the query message to all its neighbors until the query message is received. The implementation of this topology system is simple, but the flood search method will cause a large system load and low query efficiency. In order to improve the search efficiency, Gnutella 0.6 [2] optimized the topology of Gnutella 0.4 and proposed the concept of hierarchical topology. First, according to a certain criterion, such as node interest, the nodes are divided
into multiple clusters. Then, in each cluster, a node is selected as a super node according to the node's ability or reputation. The other nodes are ordinary nodes. All ordinary nodes are connected to the super nodes. The supernodes between the clusters are connected to each other to form the super node topology layer, which constitutes a two-layer topology structure. When searching for resources, all query messages are only forwarded on the super node layer. The super nodes process according to the shared resource index information stored in the cluster to determine whether there are required resources in the cluster. If not, the query messages are forwarding is performed on the super node layer. If a node in the cluster responds, a response message is returned, which contains information such as the location of the node where the resource is located, and the corresponding resource can be downloaded from the node in the cluster.

2. Related work

In recent years, there has been a lot of research on the improvement of p2p efficiency. In 2016, Nick Rahimi [3] propose the use of Linear Diophantine Equation (LDE) as the mathematical basis to realize a hierarchical P2P architecture. LDE provides a significantly lighter weight mechanism to create and maintain the overlaid P2P structure as compared to DHT based schemes. They have also presented algorithms for handling of new resources and peers joining and leaving the P2P network and fault tolerance in the event of peers crashing or leaving. In 2019, in order to solve the vulnerability of the network structure in the real world, Zhao H [4] constructs a P2P network model based on neighbor-neighbor lists. They propose two algorithms, network repairing and pruning, to improve the effectiveness and robustness of the network. Simulation experiments show that the proposed model and algorithms can effectively enhance the self-healing of P2P network. In the same year, Indranil Roy [5] have applied modular arithmetic, specifically residue class (RC), to design a non-DHT-based structured P2P network. It is known as pyramid tree. Search latency for its inter-group data lookup algorithm is bounded by the tree diameter and is independent of the number of the distinct resource types as well as the total number of peers present in the system. In addition, any intra-group data look up communication needs only one overlay hop. In 2020, Yingying Ren [6] utilize the idea of machine learning to select trusted data reporter to collect data. The data collection optimization is translated into how to maximize data coverage and minimize the cost under given budget in malicious network, which has rarely been considered in previous studies.

In Gnutella0.6, although SN (Super Node) stores ON (Ordinary Node) near itself, it can reduce the delay between SN and ON to a certain extent, but it is not always possible to guarantee the physical location between SN and ON in the cluster. It is similar, and it is not very effective in reducing the information retrieval delay between nodes in the region.

![Mismatch of Network topology](image)

Figure 1. Mismatch of Network topology

Yunhao Liu's [3] evaluation of popular P2P systems such as Gnutella, KaZaA, shows that P2P traffic accounts for most of the Internet traffic. [7] shows that even if 95% of the hops between any two nodes
are less than 7 and the TTL of the message is set to 7, the flooding routing algorithm in Gnutella of 5000 nodes generates 330TB / month. Most P2P traffic is caused by invalid coverage topology and unnecessary blind flooding, which affects the scalability of unstructured P2P networks. The main reason for this problem is that the mechanism that the peer randomly selects logical neighbours without any underlying physical connection information results in a mismatch between the P2P logical overlay network and the underlying physical network, which causes the same message multiple times. Traversing the same physical connection generates a lot of unnecessary network traffic. In many cases, the same query message will traverse the same physical logical connection at least twice. For example, in Figure 1: (a) is two overlay topologies, and the underlying physical topology is (b). Suppose C1 and C3 are in one autonomous system, and C2 and C4 are in another autonomous system. It is assumed that the physical delay between C1, C4 is greater than the delay of all other links in (b). Obviously, on the invalid coverage topology (a), the query message will traverse the longest link C1, C4 four times from the source.

Aiming at this problem, this paper proposes a supernode selection mechanism based on physical distance information. When selecting a super node, the actual physical distance between the nodes should be detected, so that the selected supernodes are logically adjacent and as close as possible in physical location.

3. Design

3.1. Evaluation of node distance

By evaluating the distance of the nodes, the nodes can find other nodes within the delay range. Many distances between evaluation nodes are measured using virtual coordinate services. Virtual coordinate services such as Vivaldi [8], GNP [9], etc. The basic principle of this type of service is to map the Internet into a geometric space. A group of host nodes in the network is selected as anchor nodes to provide reference coordinates for other nodes. Based on the positions of these reference coordinates, other nodes can calculate their own coordinates in cyberspace. The key to this type of network measurement method is the selection and location of anchor nodes, which plays a decisive role in the performance of the virtual coordinate service. However, some current methods can easily cause anchor nodes to become the bottleneck of the system. In dynamic P2P networks, the joining and leaving of anchor nodes also affect the accuracy of the measurement.

In this paper, the distance delay of neighbor nodes is obtained by detecting neighbor nodes. There are two modes of detection: periodic detection and event-driven detection. In the periodic approach, the neighbor distance is probed at regular intervals. After detecting the neighbor, it sends its own cost to the neighbor. Therefore cycle time is the key to affecting the performance of periodic methods. In event-driven mode, when the connection between a node and a neighbor changes, an updated cost table is sent to the neighbor, such as the node joining or leaving. This article uses event-driven detection to reduce the detection overhead in the network.

![Figure 2. Probe message format](image-url)
In order to effectively detect, this chapter designs a new message detector like Gnutella’s 0X82 message. Figure 2 shows the two formats of the probe message.

One of these two formats is a short message, which is used as a probe message from the source node of the probe. What is recorded in the short message is the IP address of the probe initiating node and the time stamp of the moment. In addition to the header of the 0X82 message, we append a 2-byte timestamp to record the current moment of the node. The other is a long message, which is used to forward the received message of the node. The relay node receives the detection short message initiated by the detection source node, and forwards it using the long message, and records the current timestamp on this message. If the current timestamp minus the source timestamp does not exceed the system threshold \( t \), the message continues forward. The delay distance between nodes is calculated by subtracting the timestamp of the received message and the timestamp of the source node.

Nodes broadcast probe messages to all their neighbors. The probe message will stop forwarding when the delay exceeds the system threshold, and return the information along the way to all nodes on the path. Then detecting the source node can know the information of all nodes within the threshold delay range. Because broadcast will generate huge communication traffic, in order to save the network’s detection traffic, this paper uses a labeling method to reduce overhead. During the detection process, the nodes will continuously select supernode connections. If the node is connected to the supernode, both the node and the supernode are marked to indicate that the two have established a connection and no longer needs to be detected. In this way, when broadcasting the query, these marked nodes will not be forwarded, so this flooding is a limited flooding.

3.2. Supernode selection mechanism

In addition to satisfying some basic conditions such as strong computing power, large storage capacity, high credibility, and long online time, the supernode must also consider the physical location of the nodes in the area under the jurisdiction of the SN. Similar to reduce the information retrieval delay between ON and SN and improve retrieval efficiency.

Network model: assuming that the nodes in the network are heterogeneous, their computing power and capacity are different, and the bandwidth of the network connection is different, so the number of nodes that each node can connect to is limited. Assume that the capacity of the node is \( C_i \), which represents the processing capacity of node \( v_i \), and each node knows its own capacity. For any two nodes \( v_i \) and \( v_j \) in the network, the distance between them is:

\[
 h_{ij} = |v_j - v_i|_{RTT} 
\]

Average capacity of nodes in the network

\[
 \bar{C} = k \times \frac{\sum_{i=1}^{n} C_i}{n} 
\]

For the node set \( v_i \) of any node \( v \) within the delay \( t \), find \( w \) with the largest weight, and

\[
 W_{ij} = \alpha h_{ij} + \beta C_{ij} 
\]

Among them \( \alpha + \beta = 1, \alpha > 0, \beta > 0 \)

3.3. Algorithm

Below we give the supernode selection algorithm.

Input: node \( v_i \), capacity \( C_i \), delay \( t, \alpha \)

1. For the nodes \( v_i \), probe its neighbors within the delay \( t \) and get \( V_i = \{v_1, v_2, \ldots, v_n\} \).
2. According formula (1), compute the $h_{ij}$ and $C_j$ of any neighbor.
3. According formula (3), find out the maximum of $W_{ij}$.
Output: $v_j$ is the selected supernode.

4. Simulation
This experiment is based on the PEERSIM [10] simulation environment. The network size is fixed at 2000 and 4000 nodes. The capacity of the nodes conforms to the uniform distribution of $[1,500]$. For each node in the network, the delay distance between nodes conforms to the normal distribution of an average of 250ms. Test performance at $t = 200$ms and $\alpha = 0.5$. Table 1 shows the settings of the experimental parameters.

| Parameters               | Default value   |
|--------------------------|-----------------|
| Number of nodes          | 2000/4000       |
| Delay between nodes $t$  | 250ms normal distribution |
| Node capacity            | $[1,500]$ evenly distributed |
| Number of files queried  | 1000            |
| $t$                      | 200ms           |
| $\alpha$                 | 0.5             |

Simulation experiments show that the algorithm can effectively reduce the number of redundant messages in the network by about 10% and can obtain the same query success rate, show as figure 3.

5. Conclusion
The supernode selection algorithm proposed in this paper does not use virtual coordinates to obtain the distance between nodes, but uses message detection. This method is more accurate than the method of virtual coordinates. Each node turns on a detector in a small area to record the relative delay information, with low overhead and simple implementation. At the same time, this kind of detection is a kind of limited flood, which can take the capacity of the nodes. Compared with the traditional method, the service capacity of the system is significantly improved.

Acknowledgements
This paper is supported by grant from the academic team of Wuhan Business University (2020TD001).
References

[1] The Gnutella Protocol Specification v0.4 Document Revision 1.2.
[2] Gnutella Protocol Specification v0.6.
[3] Rahimi N, Sinha K, Gupta B, et al. LDEPTH: A low diameter hierarchical p2p network architecture [C] //2016 IEEE 14th International Conference on Industrial Informatics (INDIN). IEEE, 2016: 832-837.
[4] Zhao H, Wang C, Zhu Y, et al. P2P Network Based on Neighbor-neighbor Lists [C] //Journal of Physics: Conference Series. 2019, 1168 (3).
[5] Roy I, Gupta B, Rekabdar B, et al. A Novel Approach Toward Designing A Non-DHT Based Structured P2P Network Architecture [C] //Proceedings of 32nd International Conference on. 2019, 63: 182-188.
[6] Ren Y, Zeng Z, Wang T, et al. A trust-based minimum cost and quality aware data collection scheme in P2P network [J]. Peer-to-Peer Networking and Applications, 2020: 1-24.
[7] Yunhao Liu, Li xiao, Location Awareness in Unstructured Peer-to-Peer Systems [J].
[8] Dabek, Frank, et al. "Vivaldi: A decentralized network coordinate system." ACM SIGCOMM Computer Communication Review. Vol. 34. No. 4. ACM, 2004.
[9] Ng, T. S., and Hui Zhang. "Towards global network positioning." Proceedings of the 1st ACM SIGCOMM Workshop on Internet Measurement. ACM, 2001.
[10] Montresor A, Jelasity M. PeerSim: A scalable P2P simulator [C] //2009 IEEE Ninth International Conference on Peer-to-Peer Computing. IEEE, 2009: 99-100.