Research and Implementation of Peer-to-Peer Network Topology based on Balanced Binary Tree

Zong HU
College of Education
Ningbo University
No.818 Fenghua Road, Ningbo, Zhejiang 315211
CHINA
nbshiny@gmail.com

Abstract: - With the quick development of network, how to locate or lookup the resources in peer-to-peer network becomes a hot spot. Having analyzed disadvantages of current peer-to-peer networks, AVLNet, a new peer-to-peer network, is set forth. AVLNet topologizes peer-to-peer overlay network as a balanced binary tree. Each node in AVLNet only holds the information of its parent and children, which solves the problems caused by status parameters in unstructured network. What's more, AVLNet also weakens the relationship of nodes as in structured network, which solves the problems of frequent hashing. The paper designs peer checking in, checking out and searching strategy of AVLNet in algorithm, and implements it based on JXTA platform, which proofs the correctness and feasibility of AVLNet network both in theory and practice. What's more, it simulates AVLNet, Gnutella and Chord by MatLab, compares the performance of inexact searching in three networks and shows the advantages of AVLNet.

Key-Words: - Peer-to-Peer Network; JXTA; Balanced Binary Tree (AVL); Searching Algorithm

In order to meet the needs of large-scale network applications, collect and collaborate the increasingly distribution resources, the mode of the Internet system changes from the C / S mode to P2P, each node act as the role of client and server, and directly share and exchange of resources without the aid of the intermediate server between nodes[1]. The emergence of P2P networks make computing models transfer from centralized to distributed, and network applications core diffusion from a central server to the edge of the network terminal equipment. P2P mode allows interconnection online idle resources can be fully utilized, fault-tolerant performance of the network is also greatly improved in the P2P network, the more rapid dissemination of information, but also to optimize the utilization of network bandwidth, P2P technology has been used in file sharing, distributed computing, collaborative work, instant messaging and other many areas.

The purpose of P2P is the sharing of resources, and the premise of the resource sharing is to find and locate resources, and this is the key to P2P systems, namely the resource index. For file sharing system, the index problem is that given a keyword to find the appropriate file, and gives the location of these files, a common strategy to solve this problem is to build an overlay network (Overlay Network) at the application layer in the overlay network, through the allocation and routing mechanism to implement indexes.

Overlay network topology can be divided into unstructured and structured according to the degree of coupling topology, on the basis of analysis of the advantages and disadvantages of these two types of topology, this paper propose P2P system based on balanced binary tree, built AVLNet network. The network of innovation is: although the overlay network is structured topology, but weaken the relevance of the stored content in the network and the network topology, can improve the shortcomings of structured and unstructured P2P networks.

The paper is organized as follows: Section 1 discusses the research background, analysis the structured and unstructured P2P network topology and its corresponding discovery mechanism need to be addressed; Section 2 design P2P network topology structure based on a balanced binary tree, a progressive approach to introduction of the topological network design ideas and implementation, to determine the basic behavior of the search and implementation of programs, and the corresponding algorithm; Section 3 to achieve a balanced binary tree structure P2P network, divide into various functional modules of the network required, give their interactions, design a node to join, exit and search for messages, give the interactions of various types of packets; verify by experiment, and give a comparison of search performance; Section 4 Summary and Outlook on the innovation of this paper and the value, given the outlook for future work.

1 Background

Topology refers to the relationship between the physical or logical connection between the compute unit in the distributed system[2]. According to the commonly method researchers used, the topology of the P2P systems can classify from the dispersion and the degree of coupling [3-5]as follows:

In the unstructured P2P system topology, logical topology between nodes are usually more loosely, with the randomness. Resources (or resource information) place usually has nothing to do with the topology of the P2P system, it is generally placed in the local. Unstructured topology relatively simple achieve and maintain, and can support flexible resource search criteria, but efficient resource search is usually more difficult (usually flood search, random forwarding and selective forwarding method), applies to a large number of autonomous nodes, and quality of service is not strictly required application.

Logical topological relationship between nodes in the structured topology P2P systems, usually strictly control by a deterministic algorithm, the resources place is also a precise release by a deterministic algorithm to a specific node. Structured topology P2P system is typically using the Distributed Hash Table (DHT) [6-7] to build. Structured topology has the advantage of the resource accurate locator and a certain efficiency, has a good scalability and performance, making it suitable at high system availability requirements. But structured topology maintenance is relatively complex, usually only support exact match resources search, poor support for complex search criteria. When nodes and content dynamic change, need to modify the status of some nodes, there exist a certain route maintenance overhead.

1.1 Unstructured P2P network

Unstructured topology, for example Gnutella, the efficiency of this P2P system to search for resources is very low, many researchers are working to improve the search algorithm in this unstructured networks, such as breadth-first search(BFS), depth-first search(DFS). The starting point of these improvements are adopted to reduce the redundancy of the messages sent to improve the efficiency of search. However, due to the network system is unstructured, this limit determines the improved algorithm will still exist many question:

1. A node by setting the state variables to achieve the non-repetitive send. state variable to maintain the time setting is a problem: With the expansion of the network size, heavy and complex search tasks, in order to ensure the completeness of search results, the state variables to maintain the time will be getting longer, this will greatly reduce the network performance, and the search time will exceed the length of time the user can tolerate;

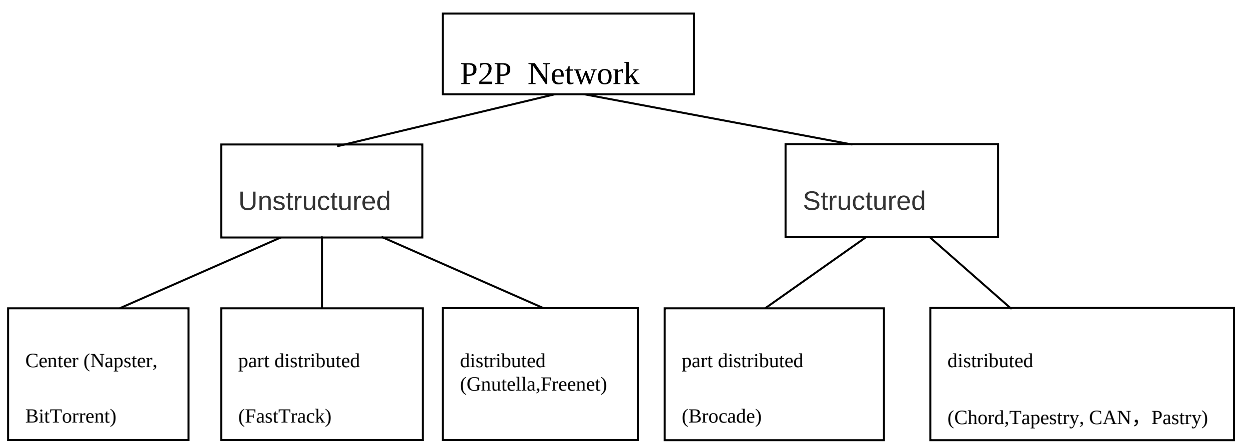


Fig.1 Category of Peer-to-Peer

2. There are a large number of search tasks, each node needs to set up and maintain multiple state variables, this will give the node a great deal of burden;

3. Unstructured P2P system in the application layer multicast, there exist a grouping of nodes and load balancing questions.

1.2 Structured P2P network

Structured topology, namely the DHT network, to fundamentally change the network topology based on the distributed hash thinking to build P2P networks, using certainty topology. DHT network has greatly improved the efficiency of the precise searches, but improve search efficiency at the same time it also brings new problems:

1. When nodes join and leave, the resources need to re-hash, in practical applications, the nodes frequently join the leave will lead to a large number of propagation of the packets, increasing the network load;

2. Initial DHT network support only exact keywords that match the query, can not support the content/semantics of complex queries. Later DHT network search algorithm has been improved to enable it to fuzzy search, but the complexity is high.

Considering the problems of unstructured and structured P2P networks, this paper propose P2P systems based on balanced binary tree to built AVLNet network. Balanced binary tree structure topology weakened the relevance between the stored content in the network and the network topology, can improve the shortcomings of structured and unstructured P2P networks.

2 Design of P2P Network based on Balanced Binary Tree

2.1 Initial design of the AVLNet network

2.1.1 Basic definitions

Balanced binary tree, known as of AVL (GMAdel'son, Vel'skii the, EMLandis) tree, or an empty binary tree or binary tree with the following properties:

1. The absolute value of height difference between its left and right sub-tree is not more than 1;

2. Its left and right sub-tree are also balanced binary tree.

Balance factors of the binary tree node define to the height of the node's left sub-tree minus the height of the right sub-tree, therefore, the balance factor of the balanced binary tree on all nodes can only be -1, 0 and 1.

According to the definition of balanced binary tree, given the initial design of the AVLNet network is defined as follows:

Define 1 (node definition). Any node of the AVLNet network includes two basic elements: relations (R, Relation), the balance factor (B, Balance). Set P_n as one node of the network, in which the relationship R express connection relations between P_n and the adjacent nodes, pointing out the position of the node's parent node and child node, as shown in Figure 2-1, the P_n node's parent node set P_{np}, left child set P_{nl} and right child set P_{nr},for the node relationship R can be expressed as an ordered collection $R_n = \{< P_{np}, P_n >, < P_n, P_{nl} >, < P_n, P_{nr} >\}$. balance factor B define the height difference between the left and right sub-tree of the node, use number to express, $B_n = 0$ in the figure.

Define 2(network define). The AVLNet network consists of a set of nodes, $AN_n = \{P_1, P_2 ......P_n\}$ express there exist n nodes in the AVLNet network; network topology (TP) consists of node locations R, $TP_n = R_1 \cup R_2 \cup ...... \cup R_n$; for any node, the absolute value of the balance factor is less than 1. As shown in figure 2-1.

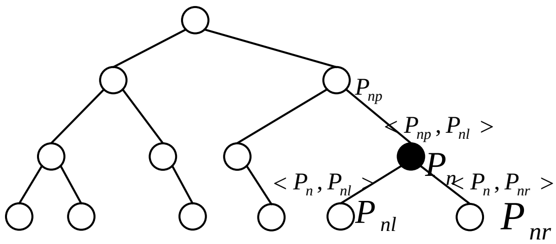


Fig.2-1 Definition of Node P_n and AVLNet

2.1.2 Node join

Nodes join the network divide into two cases, a single point to join and multi-point to join, a single point to join is the basis, so first considering a single point to join, according to the number of network nodes, also the single point to join can be divided into two:

(1)No node in the network

This situation indicates that the node is the first node to join the network, so directly into the root node. Before the node joins, the node first broadcast within the network, at a specified time no response packet, the nodes is the only node on that network, so become the root node.

(2) Network already exists in the node

In most cases, the networks exist node, when a new node join the network, the new node in the network first broadcast, network nodes receive the broadcast packets if the network node has the left and right child, the packet is ignored; if the node has no left and right child or only one child, the network node send a response packet to the new node, the new node will be received more than one response packet, then according to the information provided by the packet, select a node as a child of its to join the network.

From the above analysis can be found, when a single node to join the network, the network will provide one or more location to this node, and handed over to the node to make choice. Discussing specific node join algorithm, the first need to define the network location algorithm and node location determination algorithm, and then based on these two algorithms derived node join algorithm.

Algorithm 1 (network location algorithm). P2P network AN_n for any balanced binary tree with n nodes, at any time can be provided to the location of the new node set is:

$$Poa_n = \{x \mid y = null \ \& \ <x, y> \in TP_n\}$$

Poa_n (Position Available) express new placement in the network with n nodes. $y = null \ \& \ <x, y> \in TP_n$ express that x node has no child or only one child. In Figure 2-2, the dotted line position is the new placement that node join, collection of black nodes is Poa_n.

Algorithm 2 (node location determination algorithm) When a node to join the P2P network AN_n of a balanced binary tree with n nodes, its determined join location as follows:

$$Poj_{n+1} = \mathrm{Re}cf(\min(\{\mathrm{Re}cq(y) \mid |B_y| = \max(\{|B_x| \mid x \in Poa_n\}) \ \& \ y \in Poa_n\}))$$

.

Firstly, according to the $\max(\{|B_x| \mid x \in Poa_n\})$, calculate the maximum of the absolute value of the node's balance factor of the new placement node, as shown in figure 2-2, the value is 1, from all nodes with the largest balance factor (Figure 2-2 as a dotted line a , b, c) return response packets to choose, according to the packets arrive in chronological order selection, selection of the first to reach the packet node as the node location algorithm, the function $\mathrm{Re}cq$ used to obtain the need node to join to receive the invitation packets order to join, the function $\mathrm{Re}cf$ determine packets from certain node based on the given specific figures location, these two functions are mutually inverse function.

In summary, we can draw the node join algorithm:

Algorithm 3(node join algorithm). P2P networks AN_n of a balanced binary tree with n nodes, when node P_{n+1} apply to join the network, all nodes of Poa_n in the network AN_n send a join invitation to P_{n+1}, and P_{n+1} receive these messages, according to the algorithm 2 calculate to determine the join location Poj_{n+1}, and response to the node, build relationships R_{n+1}; each node of the network adjusted balance parameters, complete the node join process.

Based on the single node join, taking into account the multi-node join. According to the network policy of single node, for these node can accept the child node, may be receive multiple requests, how to choose the child node becomes a problem.

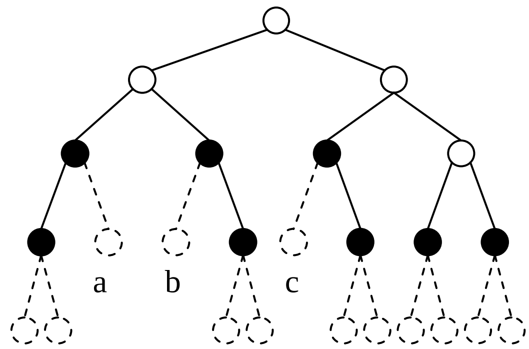


Fig.2-2 Nodes Check in AVLNet

This paper considered in accordance with the order in the message queue, select the first request of the message queue as the processing object, and reject other requests, the other node receive the reject response will be re-initiated send the join request until all requests to join the network. Using the algorithm is described as follows:

Algorithm 4 (multi-node joined avoidance algorithm). When n nodes at the same time request to join the network, they may send request to a node P, hope to be that child node. In this case, if P can only provide the location of m (m <n), then it will select the priority requests of m nodes, allowing as its child nodes, while the remaining n-m node sends avoid packet, the remaining nodes can be removed P-node from their location collection, and re-using the network to provide location algorithm to calculate the next join node. If the location collection of a node reduce to the empty set, that it will re-request to join.

2.1.3 Node leave

There have two cases in node leave, normal leave and abnormal leave, this section only consider the normal leave. When a node requests to leave the network, this node need to notify all of its associated node, in order to ensure balance binary tree network structure, the network may need to adjust. Node leave can be used the following strategy: any node apply to leave if its balance factor is greater than or equal to 0, then send out leave messages to its left sub-tree and the right child, select a leaf node from the left sub-tree, the leaf node instead of the location of the leave node. If its balance factor is less than 0, then send out leave message to its left child and right sub-tree, select a leaf node from the right sub-tree alternative to the location of the leave node.

Algorithm 5 (node deletion algorithm). For any P2P networks AN_n of balanced binary tree with n nodes, when a node P_m to apply to leave the network, you need to select a leaf node of the sub-tree to take its place. Set need to choose a node P_x, use recursion to find alternative nodes:

Step 1: set $P = P_m$;

Step 2: if the P balance factor $B > 0$ or $B = 0$ and exist left sub-tree; then select the left sub-tree; set $T = T_l$; if the balance factor $B < 0$, select the right sub-tree, set $T = T_r$; if the balance factor $B = 0$ and there are no left sub-tree, then $P_x = P$, the algorithm terminates;

Step 3: set $P = P_{Troot}$, namely the selected sub-tree root node, continue to Step 2.

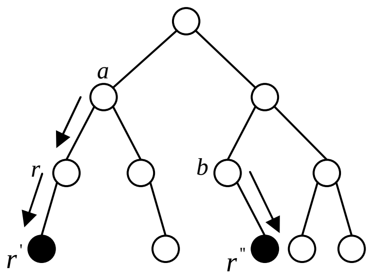


Fig.2-3 Nodes leave in AVLNet

Figure 2-3, the node a leave, first to enter step 1, set P as node a, the node a balance factor to 0 and have the left sub-tree, select left sub-tree as T. enter the step 3, set P as the root node of the left sub-tree, continue step 2, the balance factor of P is 1, greater than 0, select the left sub-tree as T. re-enter step 3, set P as the left sub-tree root node r', continue step 2, the balance factor of P to 0 and not exist the left sub-tree, the leaf node r' of alternative a finally get. Node b leave, first enter step 1, set P as node b, the balance factor of b is -1, less than 0, then select the right sub-tree as T. Enter step 3, set P as the root node r'' of the right sub-tree, continue to step 2, this time the balance factor of P to 0 and not exist the left sub-tree, the leaf node r'' of alternative b finally get.

At this point, P2P network architecture based balanced binary tree success to build.

2.2 Perfect AVLNet network

The initial design of AVLNet network is based on the balancing factor B of each node building, each node join or leave, which affect other nodes balance factor, the adjustment of the balance factor must process by the form of network sending packets, this will result in a large number of adjustment packet flooding, affecting the performance of the network. In order to reduce the impact of the balance factor for the network, this section improve definition of network, nodes join and leave algorithm, eliminates the whole impact of network action of nodes join and leave.

Construct AVLNet network based on balancing factor, nodes join and leave will cause the entire network action, in the case of the network size increases, frequent node joins, it will cause serious adjustment packet flooding, affecting the performance of the entire network. Therefore, the need to improve the initial design of the network to reduce the impact of the balance factor on network performance, eliminate the action of the whole network of node join and leave.

2.2.1 Improvements to node join cause the action of the entire network

In order to eliminate the entire network action because of node joins, consider to change from the balance factor to the network node level to describe AVLNet network, modify the basic definition is as follows:

Definition 3 (Node improved definition 1). Any one node of the P2P network based on balanced binary tree includes two basic elements: the relationship (R, Relation), the node level (H, Height). Set P_n as a node in the network, which the relation R express its connection between P_n and adjacent nodes, point out that the location of the node's parent node and child node, R can be expressed as an ordered collection $R_n = \{<P_{np}, P_n>, <P_n, P_{nl}>, <P_n, P_{nr}>\}$. Node-level H indicates that the node is in which levels of the network, using digital number, descending order of 0,1,2,3.

Definition 4 (Network improved definition). P2P networks of balanced binary tree (AVLNet, P2P Net with AVL Architecture) consist by a set of nodes, $AN_n = \{P_1, P_2 ......P_n\}$ express that there are n nodes in AVLNet network; The network topology (TP) is consist by node locations R $TP_n = R_1 \cup R_2 \cup ...... \cup R_n$; the absolute value of highest level difference of its left and right sub-tree leaf node for any node, is not greater than 1, the absolute value of the highest level difference of each node of the left and right sub-tree leaf nodes are less than or equal to 1.

According to the Improved network definition, Algorithm 2, node location determination algorithm, re-described as follows:

Algorithm 6 (node location determine improved algorithm 1) when a node need to join the P2P network AN_n of balanced binary tree with n nodes, its determined join location is: $Poj_{n+1} = \mathrm{Re}cf(\min(\{\mathrm{Re}cq(y) \mid H_y = \min(\{H_x \mid x \in Poa_n\}) \ \& \ y \in Poa_n\}))$

Original algorithm 2 based on $\max(\{|B_x| \mid x \in Poa_n\})$, calculate the maximum absolute value of the balance factor of node, which can provide location to the network, to ensure that the node of the upper layer not full first to be considered, algorithm 6 will modify it $H_y = \min(\{H_x \mid x \in Poa_n\})$, select the minimum number of levels, that is the highest level node to add a new node, its expression is to modify the balancing factor to the node level.

Under the improvement network definition, node leave become complicated because the node leave algorithm have no the balance factor, but can use sending leave messages to the left and right sub-tree to solve, until all the leaf nodes that receive the information return the leaf node information to the leave node, the nodes need to leave select a node from these nodes to replace, the select strategy similar to the basic algorithm 6, in order to leave the network, but when the node leave the network, need to traverse the node sub-tree, so as to find a suitable replacement node, more impact the network. Therefore, need further improvement.

2.2.2 Improved node leave cause the whole network action

In order to further improve the network structure of AVLNet, provide the concept of virtual nodes to solve the problem of the whole network action during node leave. Firstly, definite the virtual node, and then amend the definition of AVLNet network node.

Definition 5 (virtual nodes definition). P2P network AN_n of any balanced binary tree with n nodes, when a node P_m apply to leave the network, this node inform all nodes in its relations R_m, in its associated node generate an object that store all nodes information of P_m. This node is only occupy part of the storage space of the associated node, not the actual existence nodes, called virtual node, as shown in figure 2-4, when a node P_m leave the network, notify the parent node P_{mp}, left child node P_{ml} and right child node P_{mr}, generate an object containing all the nodes information of node P_m in P_{mp}. After the real node P_m leave, the node P_{mp} store the information of the original node P_{mp} and the node P_m information, the virtual node P_m only takes up storage space of P_{mp}, not an actual node.

Definition 6 (Node improved definition 2). $P_n = \{R_n, H_n, V_n\}$ express that a node in the P2P network of the balanced binary tree, including relational (R, Relation), node-level (H, Height), the virtual node (V, Virtual node) three basic elements. And the relation R express the connection between P_n and adjacent nodes, pointed out the location of the node's parent node and child node, R can represent a collection of ordered pairs $R_n = \{<P_{np}, P_n>, <P_n, P_{nl}>, <P_n, P_{nr}>\}$; node level H indicates that the node is in which levels of the network, use number to express; virtual node V represents a virtual node of the node contains, information of virtual nodes is fully consistent with node definition $V_m = \{R_m, H_m, V_m\}$, virtual node can contain a virtual node.

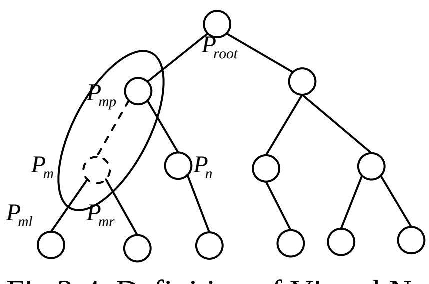


Fig.2-4 Definition of Virtual Node

In figure 2-4, the relationship among the node P_{mp}, its parent node, left and right child nodes can be expressed as $P_{mp} = \{R_{mp}, H_{mp}, V_{mp}\}$, in which $R_{mp} = \{<P_{root}, P_{mp}>, <P_{mp}, P_m>, <P_{mp}, P_n>\}$. $H_{mp} = 1$ express the level of the node is 1, $V_{mp} = \{R_m, H_m, V_m\}$ express that the virtual node P_{mp} containing is node P_m, the relation set among the node, parent node, left and right child node $R_m = \{<P_{mp}, P_m>, <P_m, P_{ml}>, <P_m, P_{mr}>\}$. $H_m = 2$ express the network level of virtual node, $V_m = NULL$ express that the virtual node are no longer continue to include the virtual node

Through definition 6 and definition 4, can easily implement node leave in the AVLNet network, without affecting the entire network, here, given the node leave algorithm based on virtual node:

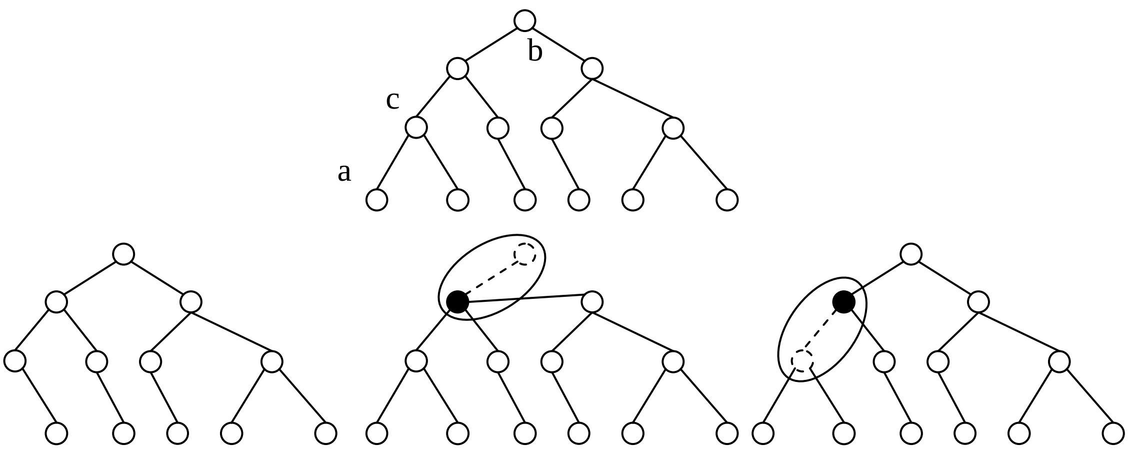
Algorithm 7 (node leave algorithm based on the virtual nodes). P2P network AN_n for any balanced binary tree with n nodes, when node P_m apply to leave the network, if it is the leaf node, then directly leave; if it is the root node, it will be sent the node information to its child nodes, create a virtual node in the left child, right child modify its relationship to make it point to the left children of its root node; if it is the other node, it will be sent node information to parent node and child nodes, parent nodes generate a virtual node, child nodes modify their relationship to point to the node's parent node.

Shown in figure 2-5, delete the leaf node of the network, get figure (a), shows the network topology. Delete the root node b, then get figure (b), shows the network topology to generate a virtual node in the left child, store the basic node information of the root node, the right child refers to the left child, black spots marked. When the deleted node is the general node c, figure (c) as shown in the network topology, the parent node of node c, to generate a virtual node, store node basic information of the node c. child node point to the node's parent node, black point marked.

According to algorithms, it is not difficult to find any one node in the network leave does not cause packet storm, but also brings a new problem, when a large number of nodes leave will lead to too much virtual node in the network, serious damage to the structure of the network, this section further amended the node join algorithm, thereby reducing the destruction of the virtual node on the network structure, here, the algorithm 1 need to modify, namely network location algorithm, improved as follows:

Algorithm 8 (network location algorithm containing the virtual node). P2P network AN_n of any balanced binary tree with n nodes, at any time can provide to the location set of the new node: $Poa_n = \{x \mid y = null \& <x, y> \in TP_n\} \cup \{P_x \mid V_x \neq null \& <x, y> \in TP_n\}$, in which, V_x is the virtual node of node P_x containing.

Unlike Algorithm 1, the algorithm incorporated into the collection $\{V_x \mid P_x \neq null \& <x, y> \in TP_n\}$, that is, in addition to the node that have no child or a child node can provide the location to the newly join node, the node containing the virtual node can provide virtual node as a new joining node location, as shown in figure 2-6, according to the algorithm 2, the bottom two layers of dotted line nodes can be placed to the newly join node. And according to the algorithm 8, the root of the virtual node, can provide a virtual node a to new node placement, merging these two collections, as shown in Figure 2-6, dotted line node of the network can provide location, black point is the collection Poa_n get from the algorithm.



(a) Del leaf node (b) Del root node (c) Del common node

Fig.2-5 Node's checking out in AVLNet with virtual nodes

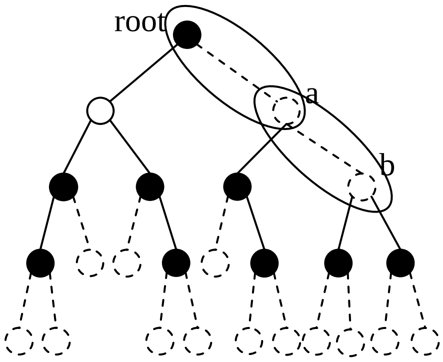


Fig.2-6 Node's checking in AVLNet with virtual nodes

For special circumstances: the virtual node contains the virtual node, the algorithm still applies. Shown in Figure 2-6, root node contains the virtual node a, virtual node a contains a sub-child virtual node b, then the location that the root node provide contains a and b.

Through algorithm 8, the network put all the virtual nodes as to the location of the new node. Binding with the location determine algorithm, namely algorithm 6, when a new node apply to join the network, select a lower level of location to join, in figure 2-6, virtual node a has higher optional priority than the next level location, and when determine the location, it will first select a root node containing virtual node a as the determined location. Therefore, we first get theorem 1:

Theorem 1. In accordance with the algorithm 8, when the node joins the network, give priority in filling the location of the virtual node.

Meanwhile, according to the algorithm 8, when the node leave, if it is not a leaf node, it must bring out virtual node, get theorem 2:

Theorem 2. The network layer of virtual node must not greater than the network level of non-virtual node location.

Through theorem 1 and 2, the virtual node bring out by node leave will priority fill by join node, nodes frequently join and leave will not affect the topology of the network, and not result in weakening of the balanced binary tree structure, which ensure the stability of the network structure.

For the special situation of the virtual node contains the virtual node, as shown in figure 2-6. Root node contains a virtual node a, a contains a sub-child virtual node b, a will be filled with new join node, then the child virtual node b of a containing will be the virtual node of the newly join node.

During the join process, if there are two nodes q and r, at the same time these two nodes request to join the network. Actual node q packets priority to reach the root node, a is replaced by the actual node q, while the actual node r avoid q, remove the location of virtual node a of root node in the location collection of r, select the virtual node b of root node, but at this point a is replaced by q, so unable to find b. In order to solve the problem, the root node will forward the join requests to the newly replaced virtual node q, then implement the action of r replace b by q.

When a large number of nodes in the network leave and have no new node to join, it will produce too much virtual node, this is bound to lead to serious degradation of the network structure, therefore the need for appropriate methods to solve the problem.

When two child of a node are both the virtual node, and have no new node to join in a long-time, the node will initiate a special search limited by the MTT, which contains the node information, other nodes receive the special search information, if there exist a virtual node, will add its own node information to the special search, until the leaf node. All leaf nodes, receive the information, can priority contact a higher level node, request to replace the virtual node, while that node have virtual node priority select the highest level of the leaf node. selected leaf node implement leave and join the network step, but do not need to request to join the operation and calculate the join location, when the leaf node leave, the parent node needs to check whether it become leaf node, if become a leaf node, the parent node should be launch the request of replace the virtual node to the node existing in the special search.

After adjustments through this method, if the virtual node of a node is not connected to any actual nodes, only exists information of the virtual nodes, then these virtual nodes can be combined or removed, thus ensuring the stability of the network structure.

2.3 AVLNet network search

2.3.1 Basic definitions

To further describe the search process, we first give the following three definitions:

Definition 7 (associated node collection). P_n is a node in P2P networks of a balanced binary tree, its associated node collection (RN, Related Nodes), namely its parent node and child node can be expressed as:

$$RN_n = \{x \mid y = P_n \& < x, y > \in R_n\} \cup \{y \mid x = P_n \& < x, y > \in R_n\}$$

Definition 8 (upstream node and downstream node). P_n and P_m are two nodes in P2P networks of a balanced binary tree, if node P_n send a message packet to node P_m, so call node P_n is the upstream node of node P_m, and node P_m is the downstream node of P_n.

Definition 9 (associated transfer node collection). P_n is a node in P2P networks of a balanced binary tree, if it receive search request of node P_m from its RN_n, then the associated transfer node collection (RTN, Related Transmit the Nodes) that is associated node collection remove the upstream node P_m, can be expressed as:

$$RTN_n = \{x \mid y = P_n \& x \neq P_m \& < x, y > \in R_n\} \cup \{y \mid x = P_n \& y \neq P_m \& < x, y > \in R_n\}$$

2.3.2 Basic search algorithm

Algorithm 9 (search algorithm). P_n is a node in P2P networks of a balanced binary tree, it send a search request $\mathrm{Re}q_n$ to the network, firstly distribute to the all nodes in associated node collection RN_n, then forward by node's RTN, subsequent all nodes receiving the search request are all forward by node's RTN, until all nodes have no node to forward. resource search is completed.

In figure 2-7, node P_1 initiates a search request, firstly sends a request message to its associated node collection $BN_1 = \{P_0, P_3, P_4\}$, arrow ① as shown in the Figure. Subsequently, node P_0 transfer search request to its associated transfer node collection $BTN_0 = \{P_2\}$, node P_3 transfer search request to its associated transfer node collection $BTN_3 = \{P_7, P_8\}$, node P_4 transfer search request to the associated transfer node collection $BTN_4 = \{P_9\}$, shown by the arrow ② below, then in turn shown in the arrow ③ and ④ transfer the search request to the remaining nodes, until traverse the entire network to complete the request.

Theorem 3. For any P2P networks AN_n of balanced binary tree with n nodes, according to the algorithm 9, the search request of any one node can be completely traverse the network, and does not generate the loop.

Theorem 4. For any P2P networks AN_n of balanced binary tree with n nodes, according to the algorithm 9, its search time complexity is $O(\log_2 n)$, and the maximum number of the search messages is $n/2$ at the same time.

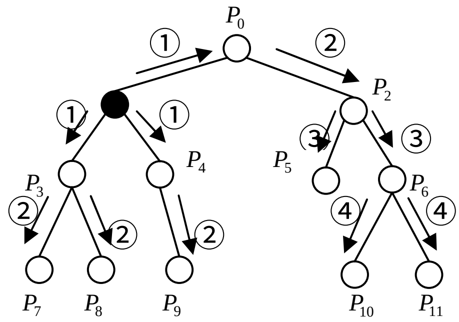


Fig.2-7 Searching Algorithm in AVLNet

2.3.3 Improved search algorithm with TTL

According to theorem 4, when the number of nodes in the network is large, the search time and the number of packet message at some point is very large, so here provide an improved search algorithm, the algorithm draws on the TCP/IP TTL concept.

Algorithm 10 (search algorithm include TTL). P_n is a node in P2P networks of a balanced binary tree, which send the search request $\mathrm{Re}q_n$ to the network, and set a maximum transmit time(MTT) in the search request message, firstly distribute to all nodes in the associated node collection RN_n, the nodes of RN_n forward in accordance with the node's RTN. All nodes receiving the search request forward in accordance with the node RTN, through one forward process, MTT of request message reduce 1, until all nodes have no node to forward or MTT value of all packets message is 0. Search process complete.

2.3.4 Second search algorithm

According to Algorithm 10, if the network structure does not change for a long time, there may exist that the resources need to search have not been able to search. Therefore, a second search algorithm to solve this problem. First, propose algorithm 11, Select Second-search Start Note algorithm.

Algorithm 11 (Second-search Start Note algorithm--SSN). P_n is a node in P2P networks of a balanced binary tree, it uses the algorithm 10 to search a resource, there have not search any resources, so need to second search. Before the second search, these nodes that MTT value is 0 in the last search return information of these nodes. P_n select the node P_m with lowest level H as the start node, if there are multiple same-level node, select the node that response packets firstly reach. If the node collection of response is $RES_{MTT=0}$ then $SSN_{n+1} = \mathrm{Re}cf(\min(\{\mathrm{Re}cq(y) \mid H_y = \min(\{H_x \mid x \in RES_{MTT=0}\}) \,\&\, y \in RES_{MTT=0}\}))$.

Algorithm 12 (second search algorithm). P_n is a node in P2P networks of a balanced binary tree, it uses the algorithm 10 to search a resource, there have not search any resources, so need to second search. According to the algorithm 11, determine search node SSN_{n+1} , P_n first forward the search message to SSN_{n+1}, go to forward through SSN_{n+1}, its forward rule is to forward through SSN_{n+1} in accordance with RTN of the first search calculations, that is not forward the packet message to the upstream node in the first search. The remaining nodes still forward according RTN , until all nodes have no node to forward or the MTT value of all packets is 0. The second search is completed.

Theorem 5. A node do not search twice in the second search.

Due to the existence of a special virtual node of the network, here need to discuss the network search based on the virtual node, the essence of the virtual node is a virtual node object of its parent, the connection method is fully using object pointer, so their search is passed the original message as a parameter to the virtual node, rather than the actual network packets message, and forward method does not change, but the virtual node does not calculate the MTT values.

3 Implementation of P2P Network Topology based on Balanced Binary Tree Structure

3.1 AVLNet network architecture

AVLNet network architecture uses a hierarchical structure, shown in figure 3-1.

Application layer: specific P2P applications, including resource downloads, online video, and more.

Structure layer: maintain the network structure, namely balanced binary tree structure, responsible for node join and leave, information resources search.

Channel layer: implement VPN in the layer of TCP/IP, each node in the P2P network is belong to a VPN.

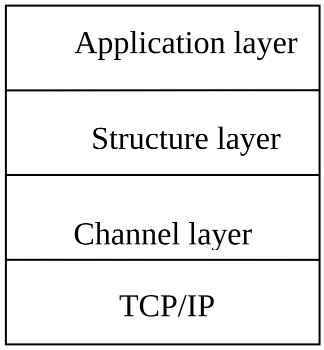


Fig.3-1 Architecture of AVLNet

TCP/IP layer: specific data transmission, can use TCP, UDP transport protocol for IP-based network data transmission.

In the four levels, the TCP/IP layer and channel layer has been implemented in JXTA, and can be used directly; while application layer is need to development different applications depending on the different application. Therefore, the structure layer is a core part, specific elaborate its design.

3.2 Structure layer implement

Node has join, leave and search, three most basic functions for the structural layer. When the node requests to join the network, the request will reach the structure layer from the application layer, structural layer requires a special processing node join module, the same when the node requests to leave or request search, need corresponding processing module. Therefore, divide structure layer into node information processing module, node join processing module, node leave processing module and the search processing module. According the algorithm as described in section II, the function of each module need to be refined, and ultimately, the structure of the basic functional block diagram as shown in figure 3-2.

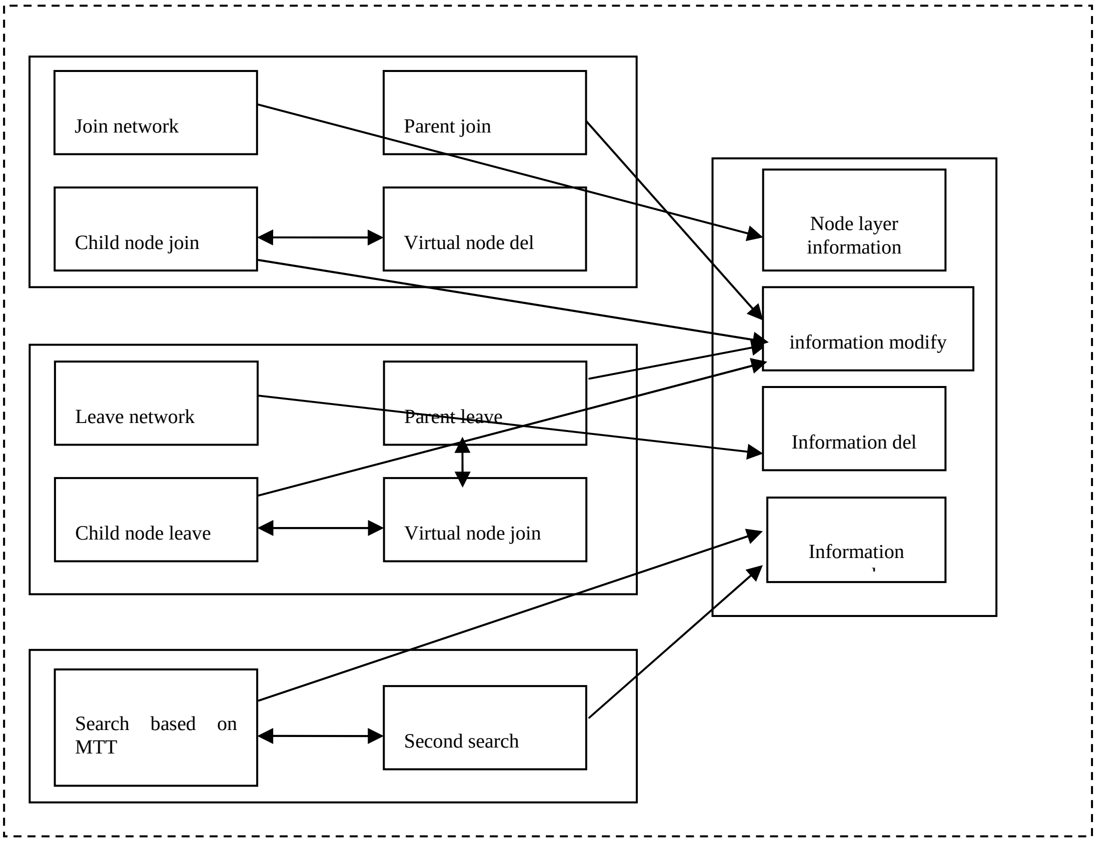


Fig.3-2 Structure in detail

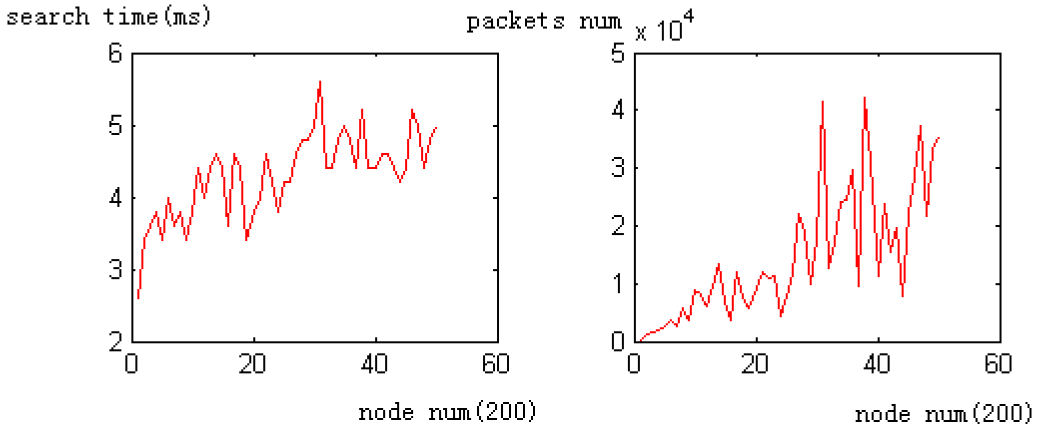
3.3 Experiment result analysis

This section simulate unstructured network Gnutella, structured network Chord, and AVLNet network fuzzy search, use Matlab6.5 software implementation of the simulation based on the following assumptions:

1. Packet message delay is 1 ms among nodes
2. Network traffic is infinite
3. Node processing time is negligible
4. the total number of node stores routing up to 16
5 the total number of nodes in the network is n, the node number as 1, 2, 3 ... n

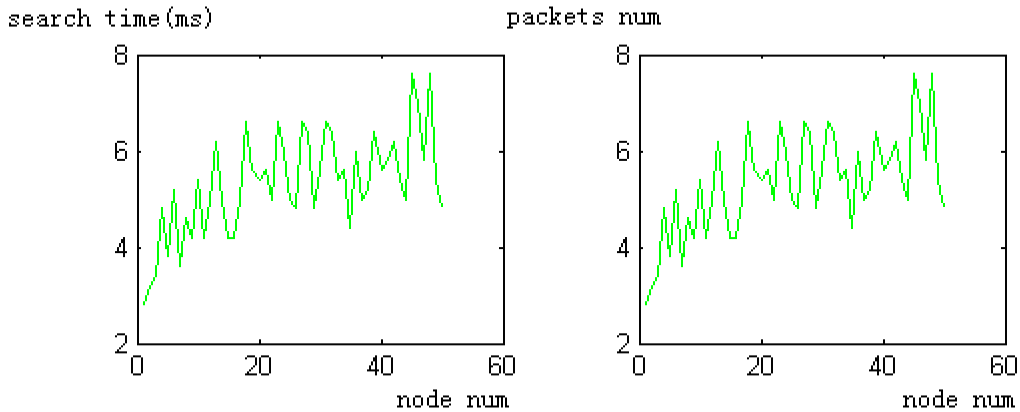
During the simulation network test, the number of nodes in the network increase from 200 to 10,000, an increase of 200 each time, a total of 50 repeated, each round test five times, according to the results of five times, record the average value. Testing process includes generating nodes routing, randomly generate search initiated node, and randomly generated target node, in the last record the time of first visit the target node as hit time, each test record data includes hit time and the number of network packets, according to the five times repeated test to calculate the average hit time and number of packet message.

The simulation results as shown in figure 3-3,3-4,3-5, the node number indicates the size of the network, 200 node per unit, the number of network nodes increase from 200 to 10,000 in increments of 200, a total of 50 round, search time express the time of searching the target node, the total number of packets count until search the target node, the total number of packets of all nodes interact, can reflect the network traffic, to show that accounting for the consumption of network resources in the search process.



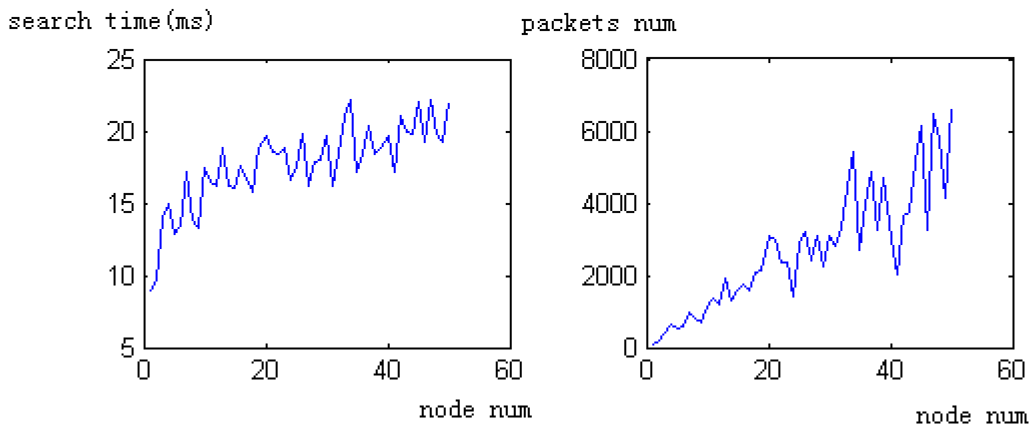
(a)Gnutella network search time (b)Gnutella network search packets num

Fig.3-3 Simulation Result of Gnutella



(a)Chord network search time (b)Chord network search packets num

Fig.3-4 Simulation Result of Chord



(a)AVLnet network search time (b)AVLnet network search packets num

Fig.3-5 Simulation Result of AVLNet

It can be seen from figures 3-3 and 3-5: compared with the Chord network, AVLNet network search are not dominant in terms of search time and the number of sending packets. The search of the Chord network is based on keyword matching, it is difficult for fuzzy search, while structure AVLNet network is weaker than DHT network, can support the fuzzy search, in order to solve the DHT network complexity higher question in content/semantic query. At the same time, through num the node, AVLNet network also retained a precise search of the DHT network, and in the AVLNet network, nodes join and leave only need to notify the adjacent nodes, the number of packets message sent is greatly reduced, a good solution to the network fluctuation caused by node frequently join and leave in the structure P2P network.

It can be seen from figure 3-4 and 3-5: compared with the Gnutella network, the AVLNet network search time-consuming is higher, the same node case is time-consuming 4-5 times; but the AVLNet search messages cost is much smaller than the Gnutella network, its the same node case, 1/3, saving a lot of network resources. Furthermore, where the simulation is assumed infinite network bandwidth, while the actual environment the network bandwidth is limited, time-consuming search of the Gnutella network will greatly increase due to the large number of packets caused by network congestion, while the AVLNet network more advantages in the search.

In conclusion, the AVLNet network has good scalability; guarantee the robustness by virtual nodes and network degradation automatically adjust algorithm, maintain complexity is small, node join and leave is a single node movement, not network-wide action; positioning efficiency and accuracy high, the average search time is $O(\log(N))$, can ensure that search all nodes with high accuracy; but the time may be slower than the DHT network; easily support complex searches, and support a variety of query patterns.

4. Summary and Prospects

In this paper, provide AVLNet network based on P2P networks of a balanced binary tree structure, design the action strategies of node joins, leave, search and the corresponding algorithm, theoretically verified the feasibility of AVLNet network, at the same time , use simulation experiment to compares the network fuzzy search performance of the Gnutella, Chord and AVLNet, shows the superiority of the the AVLNet network from the practice.

Compared to the structure network, AVLNet simple network topology determines the arrangement of convenience and availability of their applications, has a high practical value, it applies not only to the laboratory environment, more suitable for medium and large networks. But compared to unstructured network in a large number of applications, AVLNet network excellent flow characteristics, the more a good fit in the streaming media network, such as video conferencing.

The future work is to further improve the design of the AVLNet network. In addition, on the basis of the right build and improve the structure layer, development the application layer, namely P2P applications based on AVLNet network, such as resource sharing, network video can be further research and development.

References:

[1] Dejan S. Milojicic, Vana Kalogeraki, Rajan Lukose, Kiran Nagarajal, Jim Pruyne, Bruno Richard, Sami Rollins, Zhichen Xu. Peer-to-Peer Computing. HP technical report HPL-2002-57, HP Laboratories Palo Alto, March 2002

[2] Qingbo Wang. P2P Network based on specified location. Master paper of Polytechnical university of Harbin.

[3] K.Kant, R.Iyer, V.Tewari. A Framework for Classifying Peer-to-Peer Technologies. Proc. of 2nd IEEE/ACM International Symposium on Cluster Computing and the Grid (CCGRID’02), IEEE Society Press, pp.368-376, May 2002

[4] S.Androutsellis-Theotokis and D.Spinellis. A Survey of Peer-to-Peer Content Distribution Technologies. ACM Computing Surveys, Vol.36, No.4, pp.335-371, December 2004

[5] Dongsheng Li, Xicheng Lu and Chuanfu Xu. Topology Construction and Resource Discovery in Peer-to-Peer Networks. Handbook of Theoretical and Algorithmic Aspects of Ad Hoc, Sensor, and Peer-to-Peer Networks, J.Wu (ed.), Auerbach Publications, pp.733-765, 2005

[6] Y.CHawathe, S.Ratnasamy, L.Breslau, N.Lanham, and S.Shenker. Making Gnutella-like p2p systems scalable. In Proc of ACM SIGCOMM, 2003

[7] I.Clark, O.Sandberg, B.Wiley, and T.Hong. Freenet: A Distributed Anonymous Information Storage and Retrieval System. Proc. of ICSI Workshop on Design Issues in Anonymity and Unobservability, Berkeley, California, June 2000