A New P2P Network Routing Algorithm Based on ISODATA Clustering Topology

Yi Ma\textsuperscript{a}, Zhenhua Tan\textsuperscript{a}, Guiran Chang\textsuperscript{a}, Xueyi Wang\textsuperscript{a}, a*

\textsuperscript{a}Software College, Northeastern University, Shenyang City, Liaoning Province, China

Abstract

Most P2P applications use the routing algorithm that selecting the neighbor nodes at random. These routing algorithms increase the routing hops. To get better routing efficiency, a new routing algorithm named RIDC was presented in this paper. It dynamically merges nodes into different clusters in a taxonomy hierarchy, and organizes the clusters into routing overlays. By this algorithm the network performance is greatly enhanced. Preliminary evaluation shows that RIDC achieves a good convergence on a large scale of nodes.

© 2011 Published by Elsevier Ltd. Open access under CC BY-NC-ND license.
Selection and/or peer-review under responsibility of [CEIS 2011]

Keywords P2P; ISODATA; hierarchy agglomerative clustering; routing algorithm

1. Introduction

Most classical P2P network modals, for example, Freenet\textsuperscript{[1]}, Pastry\textsuperscript{[2]}, Kademlia\textsuperscript{[3]}, Chord\textsuperscript{[4]}, Tapestry\textsuperscript{[5]} and CAN\textsuperscript{[6]}, are based on Distributed Hash Table(DHT). These algorithms are effective compared to other unstructured P2P routing algorithms. Some algorithm that every node records routing message of all other nodes is even more effective, but not appropriate for large scale of nodes with more start-up time and bandwidth.

This paper presents a new routing algorithm based on ISODATA Clustering Topology (RIDC) for structured P2P overlay network. According to the communication history, clusters are merged by ISODATA (Iterative Self-Organizing Data Analysis Technique) \textsuperscript{[7]} method. Each cluster has a SubSuper-Node, by which the nodes in the same cluster can route directly. Meanwhile, the clusters are organized as
a tree by the similarity of each other in agglomerative clustering algorithm [8]. So the whole network has a good routing efficiency.

2. ISODATA Clustering

The ISODATA algorithm has some further refinements by splitting and merging of clusters than other algorithms. Clusters are merged if either the number of members in a cluster is less than a certain threshold or if the centers of two clusters are closer than a certain threshold. Clusters are split into two different clusters if the cluster standard deviation exceeds a predefined value and the number of members is twice the threshold for the minimum number of members.

The new P2P network topology in this paper is structured by several clusters which were composed by a tree hierarchy. All the nodes are subordinated to the top cluster, and every cluster can be split into sub clusters. The root node of the tree is called Super-Node, and the root node of sub cluster is SubSuper-Node. In the same cluster, every node is equal whether normal node or root node. In the tree structured hierarchy topology, every root node of sub cluster has routing information of other nodes in the same cluster.

• Definition of Communication History

To compute the similarity of nodes in P2P network, every node keeps a log file that shows the communication history.

$S_i, \lambda_i, \eta_i$ is data quantity, times and proportion of successful communications of node $i$.

We can define a 3-dimension vector $\beta(i)$, which shows amount of communication, rate of successful communication, and quantity of communication data.

$$\|i - j\| = x \sqrt{\lambda_i^2 - \lambda_j^2} + y \sqrt{\lambda_i^2 - \lambda_j^2} + z \sqrt{\eta_i^2 - \eta_j^2}$$  \hspace{1cm} (1)

• ISODATA Clustering Algorithm In P2P

In ISODATA clustering algorithm, clusters will be merged if either the number of members in a cluster are less than a certain threshold or if the centers of two clusters are closer than a certain threshold. Meanwhile, clusters can also split into two different clusters if the cluster standard deviation exceeds a predefined value and the number of members is twice the threshold for the minimum number of members. The principle is as follow.

[Algorithm 1] ISODATA Clustering Algorithm

$K$ is the anticipant amount of clusters. $N_{\theta}$ is the minimum number of a cluster. $\theta_S$ is the parameter of standard deviation. $\theta_C$ is the parameter of merging. $L$ is the maximum number can be merged by one iteration. $I$ is the iterative number.

Step1: Initialize $c$ seeds to be the cluster centers, $m_i, i=1,2,\ldots,c$.

Step2: if $\|y - m_i\| < \|y - m_j\|, i \neq j$ then $y \in \Gamma_i$

In this way, all the nodes are assigned to the $c$ clusters.

Step3: For each cluster $\Gamma_i$

if $N_i < \theta_i$ then delete $\Gamma_i, c=c-1$.

Step4:

$$m_i = \frac{1}{N_i} \sum_{y \in \Gamma_i} y, y = 1,2,\ldots,c$$  \hspace{1cm} (2)

Step5: Calculate the average distance $\bar{\delta}_i$ between node $y$ and $m_i$ in cluster $\Gamma_i$.

$$\delta_i = \frac{1}{N_i} \sum_{y \in \Gamma_i} \|y - m_i\|, i = 1,2,\ldots,c$$  \hspace{1cm} (3)

Step6: Calculate the average distance $\bar{\delta}$ of all nodes.

$$\bar{\delta} = \frac{1}{N} \sum_{i=1}^{c} N_i \delta_i, i = 1,2,\ldots,c$$  \hspace{1cm} (4)
Step 7: if \( c \leq K/2 \), goto step 8
   if \( c > = 2K \), goto step 11
   else goto step 8

Step 8: For each cluster \( \Gamma_i \), calculate the standard deviation
\[
\sigma_i = \left[ \sigma_{i1}, \sigma_{i2}, \ldots, \sigma_{id} \right]^T
\]
\[
\sigma_{ij} = \frac{1}{N_i} \sum_{y_{ij} \in \Gamma_i} (y_{ij} - m_{ij})^2
\]
\( y_{kj} \) is the \( j \)th feature of the \( k \)th node in cluster \( i \), \( m_{ij} \) is the \( j \)th feature of \( mi \), \( \sigma_{ij} \) is the standard deviation of \( j \)th feature in cluster \( i \), \( d \) is the dimension of \( y \).

Step 9: For each cluster \( \Gamma_i \), calculate the maximum feature \( \sigma_{i\text{max}}, i = 1, 2, \ldots, c \).

Step 10: For each \( \sigma_{i\text{max}} > \theta_d \)
   if \( (\sigma_i > \sigma \text{ and } N_j > 2(\theta_{Nj} + 1)) \text{ or } (c <= K/2) \)
   then split \( \Gamma_i \) into two different clusters, which cluster centers are \( m_i^+ \) and \( m_i^- \), \( c = c+1 \).
\[
m_{ij}^+ = m_{ij} + k \sigma_{ij}, 0 < k <= 1
\]
\[
m_{ij}^- = m_{ij} - k \sigma_{ij}, 0 < k <= 1
\]

Step 11: For each \( mi \) and \( mj \)
\[
\sigma_{ij} = \left\| m_i - m_j \right\|^2_{i = 1, 2, \ldots, c - 1}
\]

Step 12: For each \( \sigma_{ij} < \theta_c \), sort 1 of them \( \sigma_{i1,j1} < \sigma_{i2,j2} < \ldots < \sigma_{ij,l} \)

Step 13: Merge the smallest \( mi \) and \( mj \), \( c = c-1 \)
\[
m_i = \frac{1}{N_i + N_j} [N_i \cdot m_i + N_j \cdot m_j]
\]

Step 14: I=I-1
   If I==0 then return
   else goto step 2

Using ISODATA clustering algorithm, the nodes will be classified into several clusters, which are not too large or too small. The time efficiency of ISODATA algorithm is \( O(N^2) \).

3. Construction of Tree Structured Hierarchy Topology

The clusters obtained by ISODATA will be constructed into hierarchy topology by group-average agglomerative clustering algorithm (GAAC).

- Group-Average Agglomerative Clustering In P2P

Group-average agglomerative clustering algorithm evaluates cluster quality based on all similarities between documents, thus avoiding the pitfalls of the single-link and complete-link criteria, which equate cluster similarity with the similarity of a single pair of documents. GAAC calculates the average similarity of all pairs of documents, including pairs from the same cluster. As every node is a sub cluster of its upper cluster except the root node, the nodes can be lay out as a tree. We use a bottom-up clustering algorithm as follow.

step1: Given: a set \( X = \{x_1, \ldots, x_n\} \) of objects and a function sim
step2: for i=1 to n do \( c_i = \{x_i\} \)
step3: \( C = \{c_1, \ldots, c_n\} \)
step4: while |\( C \)|>1
step5: \( (c_{a1}, c_{a2}) = \arg \max_{(c_i, c_j) \in C} \text{sim}(c_i, c_j) \)
step6: \( c_j = c_{n1} \cup c_{n2}; \quad c = c \setminus \{c_{n1}, c_{n2}\} \cup \{c_j\} \)

In this paper, the set \( X \) is obtained by ISODATA. Group-average similarity \( S(c_j) \) can be calculated as follow.

\[
S(c_j) = \frac{1}{|c_j|(|c_j| - 1)} \sum_{x \in c_j} \sum_{y \in c_j} \text{sim}(x, y)
\]  

(11)

The group-average similarity of the united cluster can be calculated as follow.

\[
S(c_j \cup c_i) = \frac{(S(c_j) + S(c_i)) \bullet (S(c_j) + S(c_i)) - (|c_j| + |c_i|)}{(|c_j| + |c_i|)(|c_j| + |c_i| - 1)}
\]  

(12)

When GAAC was finished, we got a tree structured topology.

In each cluster there is a root node named SubSuper-Node that provides a most reliable performance. When the clustering algorithm is done, every node in the same cluster will be registered by the SubSuper-Node so that it can route directly to other nodes.

• Nodes Join and Exit

When a new node joins the P2P network, it will not be put into any cluster until the traffic exceeds a certain threshold, then it will be put into the cluster that it communicated with most frequently. When a node is leaving the network, there are two cases to discuss. If a leaf node is leaving, we can delete it from the network directly, and inform its higher SubSuper-Node. If a SubSuper-Node is leaving, we should use its backup root node to take place of it and rebuild the cluster.

4. Routing Algorithm

[Algorithm 2] Routing Algorithm

Input: P2P system, key_id of the destination recourse, node \( p \) that starts the route and a routing message \( \text{msg} \).

Output: node \( p_{\text{dst}} \) who is the last node in the route that has the key_id resource.

step1: Search for the SubSuper-Node in the cluster. Node \( p \) sends \( \text{msg} \) to its own cluster root node \( C \).

step2: If \( C \) exists, \( C \) searches in its cluster. If the there is a node owns the key_id, then go to step 5, and if not, then go to step 3. If \( C \) doesn’t exist, that means the P2P system is busy clustering and updating the new route table. So \( p \) has to wait until its new SubSuper-Node \( C \) is selected.

step3: \( C \) sends message to its higher SubSuper-Node to find the owner of key_id.

If there is a responder named \( C' \) from other cluster, then go to step 4; and if not, then go to step 6.

step4: \( C' \) searches in its own cluster. If the there is a node owns the key_id, then go to step 5; and if not, then go to step 6.

step5: Return \( p_{\text{dst}} \).

step6: Return null.

According to this algorithm, if the routing is successful, there is a good performance of routing hops with the complexity of \( O(\log N) \). The clustering is not running frequently, it will start clustering and selecting new cluster center nodes only necessary. As described, the time efficiency of RIDC algorithm is \( O(\log N) \). In most of time, we will get several clusters by ISODATA, and then these clusters will be structured as a tree, so the searching time will be the height of the tree, which is \( O(1) \).

5. Performances

We compare the routing hops with chord [4] and RIDC in a simulate system. While the nodes are increasing from 1000 to 10000, we record the routing hops, as shown in Figure 1. Obviously, with new network topology of RIDC, the routing hops remain a low level. This is the comparison in a steady network, which means there is no node join or reject from the P2P network. In RIDC, before nodes are
structured in a tree topology, they are merged into several clusters by ISODATA, so the height of the tree is much lower than $O(\log N)$, it is between $O(1)$ and $O(\log N)$.

![Comparison for routing hops](image)

**Fig. 1. Comparison for routing hops**

According to the nodes leaving algorithm, upper SubSuper-Nodes will be informed from bottom to top if a normal node is leaving the P2P system. The message number is equal to the height of the tree, so the time complexity of nodes leaving in RIDC is $O(\log N)$. If a SubSuper-Node is leaving, the backup root node in the same cluster will take place of it and update the route table from bottom to top. So the complexity of deletion in RIDC is still $O(\log N)$. Message number of nodes leaving is more than nodes join because some cluster may be reorganized as the SubSuper-Node is changed.

Compared with Chord, RIDC is efficient, and this means clustering by a tree structured hierarchy topology is better than DHT.

6. Conclusions

In conclusion, the routing algorithm RIDC has advantages as follows. (1) Design an improve hierarchy agglomerative clustering algorithm based on ISODATA, by which the P2P network are organized into tree structured hierarchy topology. (2) Optimize the rout table according to the clusters, so we can increase the performance in routing efficiency. (3) With the time complexity of $O(\log N)$, RIDC shows a good stability for medium-large size P2P system.

References


[7] L Xu-liang, Lin Wei, Xu Wei-dong, Pan Yu-long, Sun Wen yan; Applying the Fuzzy Clustering Analysis of ISODATA to the Classification of Camouflage Effectiveness[J];Acta Armamentarii;2005-05