2004－ＤＰＳ－117　(16)
2004－CSEC－　24　(16)
2004／3／4

Autonomous Topology Optimization and Recovery for Peer-to-Peer Networks

Haoyi Wan　Kato Takeshi　Norihiro Ishikawa
Multimedia Laboratories,　NTT DoCoMo Inc.
Johan Hjelm*　Kazuhiro Miyatsu　Hajime Kasahara
*Ericsson Research　Ericsson Research, Japan

Abstract

This paper proposes an autonomous topology optimization and recovery mechanism for peer-to-peer networks. The general topology of peer-to-peer overlay networks is constructed randomly without considering the characteristics of physical links. Thus searching and routing between peer nodes is influenced. In our research, peer nodes optimize the topology of peer-to-peer networks based on the metrics of physical links. And under the assumption that peer nodes leave the network or fail,　how to recover the split of the network topology by neighbor peer nodes is discussed.

ピアツーピアネットワークの自律的なトポロジー最適化及びリカバリーに関する研究

万　皓毅　　加藤　剛志　　石川　憲洋
（株）NTT ドコモ　マルチメディア研究所
ヨハン・ジェレム*　　宮津　和弘　　笠原　元
*エリクソン・リサーチ　日本エリクソン

概　要

本研究では、ピアツーピアネットワークの自律的なトポロジー最適化及びリカバリー方式の検討を行った。一般的にピアツーピア論理ネットワークのトポロジーはランダム的に構成され、 物理リンクの状況を考慮していないため、ピアノード間の情報検索及びルーチングに影響を及ぼす。本研究では、物理リンクのメトリックスを基に、ピアノードが自律的にネットワークトポロジーを最適化する手法について提案した。更に、ピアノードが故障もしくはネットワークから離脱した場合の状況を想定し、ネットワークトポロジーのリカバリー手法について検討した。

1. Introduction

Peer-to-Peer (P2P) systems have attracted much attention and many similar systems such as Gnutella [1], Freenet [2] and JXTA [3] have been proposed since Napster music file exchanging service was widely used. P2P systems can be classified into structured and unstructured systems[4]. Unstructured P2P systems, such as Gnutella, are popularly used in the Internet, because they require no centralized directories and no control over network topology. The effectiveness of information searching and routing among peer nodes is one of the most important indicators to evaluate a P2P system. In unstructured P2P networks, all the peer nodes form a P2P overlay network over a physical network. When a new node wants to join a P2P network, it tries to connect with the peer nodes by using the IP addresses provided by a bootstrapping node. After the new peer node connects into the P2P network, it will ping the connected node periodically and obtain the IP addresses of the adjacent nodes to build connections with them. A mismatching problem occurs between the P2P overlay network topology and the underlying physical network topology when a new node joins the network randomly according to the process described above. And it generates excessive traffic load in the Internet infrastructure and influences the performance of a P2P systems when searching information or routing in the P2P networks. In this paper we focus on unstructured

P2P systems and try to improve searching and routing effectiveness by solving the topology mismatching problem.

All of the peer nodes in an unstructured P2P network can join or leave the network at their convenience. This "freedom" might split the network into several fractions because of the sudden departure of the peer nodes or certain unpredictable failure of the peer nodes. The ability to recover the split networks is one of the key requirements of P2P networks. So far few researches have been done to satisfy this requirement. [5] proposes a random virtual neighbor node to recovery network when the split is detected. This method can not guarantee optimized network topology. And it will cause other problems, such as reconfiguration of network.

To optimize P2P network topology, it is possible by calculating metrics between every pair of peer nodes. In a large scale network, it is not realistic to calculate metrics of every pair of nodes. Hence, a heuristic mechanism is required to optimize the network topology. In this paper, we propose autonomous topology optimization and recovery mechanisms. With these mechanisms, a peer node autonomously reconfigures the topology based on the neighbor nodes' information within the range of 2 hops. Not only the virtual links of P2P level, but also the hop numbers of physical links between peers are considered in the proposed autonomous topology optimization mechanism. In addition, recovery mechanism could be easily realized by using the same information.

We organize this paper as follows. Section 2 summarizes the requirements on network optimization and recovery mechanisms. Section 3 and section 4 depicts the details of the proposed mechanisms. Section 5 is the evaluation considerations. We show the related works in section 6 and conclude this paper in Section 7.

2. Requirements

Requirements on autonomous topology optimization and recovery mechanisms are listed as the following.

Autonomous Topology Optimization based on local information

Peer nodes should be able to optimize and recover the network topology autonomously based on the local information because it is not realistic to obtain the information of the whole P2P network in terms of the traffic load.

Scalability

Proposed mechanisms should be able to be deployed large scale P2P networks which is constituted by thousands to tens of thousands peer nodes.

Applicability to heterogeneous networks

Proposed mechanisms should be able to be deployed across heterogeneous networks, including the Internet, but also ad-hoc networks, home networks and so on.

Adaptation to dynamic change of network topology

Proposed mechanisms should be rapidly able to adapt to the dynamic change of the topology of P2P networks due to frequent join and leave of peer nodes.

3. Network Topology Optimization

Based on the above requirements, we propose a network topology optimization mechanism. We describe the basic design concept and its detailed mechanisms in this section.

3.1 Design Concept

3.1.1 Exchanging of Local Topology Information

A complete understanding of the whole network topology is meaningful to optimize the network topology. However, in P2P networks, exchanging of the topology information of the whole network results in heavy traffic load, since the topology of a P2P network changes very frequently. Hence, for P2P networks, heuristic topology optimization based on local topology information is feasible. To this end, local topology information around a node should be available. In our mechanism, each node periodically broadcasts its own information to other nodes within the range of 2 hops. As a result, each node holds local topology information within the range of 2 hops. Fig. 1 shows an example of this design concept. Exchanging of topology information is controlled in a limited area so that such traffic will not increase the load of the overall network.

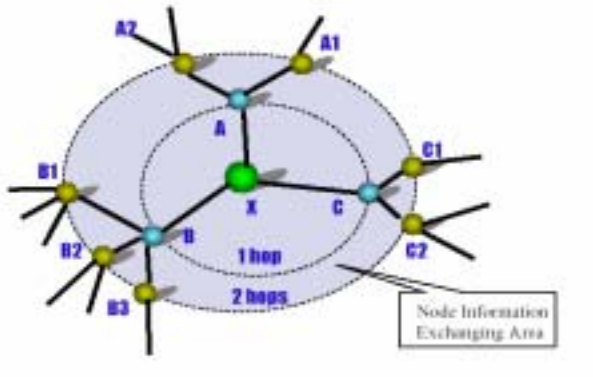

Fig. 1 The node information exchanging area

3.1.2 Criteria of Topology Optimization

Considering the physical link characteristics for topology optimization of P2P networks, metrics is introduced into our proposal. Metrics is defined as a cost between two adjacent nodes. The metrics can be hop count, bandwidth or delay of the physical link between any pair of nodes. Which metrics to be used depends on the user's choice when designing a P2P network. While the proposed mechanism is independent of any particular metrics, we use the hop count of the physical link between any pair of nodes as metrics in this paper. The hop count between any pair of adjacent nodes can be measured by Traceroute

in the case of the Internet.

Based on this consideration, we propose a mechanism which takes lower layer characteristics into account to optimize the network topology. The hop count from one node to its neighbor nodes within the range of 2 hops is used in our mechanism.

It is assumed that all of the nodes can adjust their connections with adjacent nodes. Since each node can start optimization independently, collision may occur when any pair of two adjacent nodes tries to optimize their local topologies simultaneously. To avoid it, a directional graph is used in our mechanism. This directional graph is created according to the sequence of the node participation in the network. A parent-child relationship is built between any two adjacent nodes. Each node becomes a child node when it connects to a node and joins a P2P network. The connected node becomes a parent node. Each node only optimizes the topology along its parent's direction. A directed topology graph converted from Fig. 1 is shown in Fig. 2. Node A is the parent of Node X that is the parent of Node B and Node C. Node X only considers the topology optimization along its connection with Node A and optimizations of other connections are conducted by Node B and Node C.

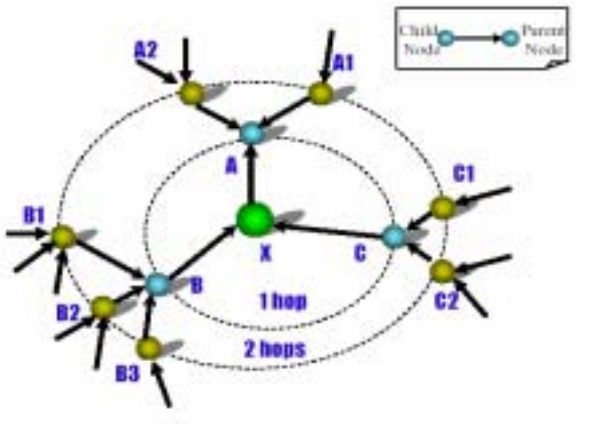

Fig. 2 Directional topology graph

3.1.3 Node Information for Exchanging

In this subsection, we will explain the details on what kind of node information should be exchanged and how the information is used in the topology optimization and recovery mechanisms.

Table 1 shows basic node information which will be used to exchange with other adjacent nodes. The table includes six elements. NodeID is the identifier of a node in a P2P network. It is a global unique ID. NodeAddress is the physical address in a network such as IP address. The next element is maximum connections a node can hold. It is a parameter that shows the ability of a node. This value of the parameter can be statistically defined by users. The following three elements are related to an adjacent node of the node. NeighborID is adjacent node's NodeID. Relation depicts the relationship with an adjacent node. In this example, Node A is the parent of Node X while Node B and Node C is the child of Node X. We use hop count as metrics in this paper.

Node ID	Node Address	Max Connection	Neighbor ID	Relation	Metrics
X	xxxx	5	A	Parent	3
			B	Child	4
			C	Child	2

Table 1 Basic node information

The basic node information shown in Table 1 is broadcasted within the range of 2 hops periodically. Hence, each node will make up a table shown in Table 2 which is a description of local topology information within the range of 2 hops. The grey part of this table is the information of the node itself (Node X).

Node ID	Node Address	Max Connection	Neighbor ID	Relation	Metrics
X	xxxx	5	A	Parent	3
			B	Child	4
			C	Child	2
A	xxxx	4	A1	Child	3
			A2	Child	4
			X	Child	3
A1	…	…	…	…	…
A2	…	…	…	…	…
B	xxxx	3	X	Parent	4
			B1	Child	4
			B2	Child	2
			B3	Child	1
B1	…	…	…	…	…
B2	…	…	…	…	…
B3	…	…	…	…	…
C	xxxx	7	X	Parent	2
			C1	Child	4
			C2	Child	4
C1	…	…	…	…	…
C2	…	…	…	…	…

Table 2 Neighbor nodes' information within the range of 2 hops

3.2 Network Optimization Mechanism

Based on the consideration in section 3.1, the following formula is used for network optimization processes.

$$M(x,k) = \frac{\sum_{i=0}^{n-1} (M_{ki} \times N_i)}{\sum_{i=0}^{n-1} N_i}$$

M(x, k) is the evaluation formula if node *x* releases the connection with its parent node and reconnect with node *k*. This connection with node *k* is called as a virtual connection. *Mki* means the metrics from node *x* to node *i* via node *k*. *Ni* is the number of nodes outside the range of 2 hops from node *x*, which are connected to node *i*. *n* is the number of nodes within the range of 2 hops from node *x*.

Fig. 3 is an example of our topology optimization mechanism. Node A is the parent node of Node X. The metrics between Node A and its neighbors is obtained by node information exchanging process which is described in section 3.1. When Node X starts its optimization process, it follows the next three steps to optimize its local network topology.

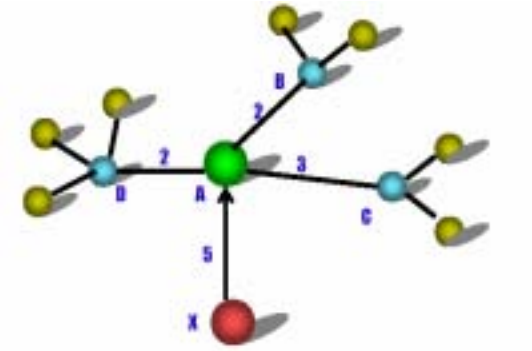

Fig. 3 Example of a node connection

Step 1: Calculate the mean metrics from node x to every adjacent node in the P2P network as Fig. 4 shows.

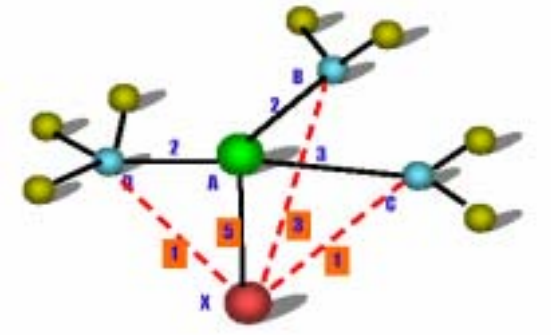

Fig. 4 Virtual connection metric calculating

In this step, virtual connections are built by Node X to calculate the metrics to its potential adjacent nodes along its parent node direction. Traceroute is used to calculate the value of this parameter as we mentioned in section 3.1.2. Then the process is moved to step 2 to build a set of tables for calculating the evaluation formula *M(x, k)* based on the assumption that Node X is connected to Node K.

Step 2: Connection table is built for further computation

In step 2, using the metrics with potential adjacent nodes a table is built for each potential adjacent node.

For Node K, Table k is built. *Mki* is set to a cell at the "Node i" row and "Metric Value" column. *Ni* is set to a cell at "Node i" row and "N. of neighbor" column.

Virtual connection table for Node X (See Table 3) is built using the P2P network shown in Fig. 3. The calculation result *M(x, k)* according to Table 3 is as follows:

M(X, A) = 78/11, M(X, B) =59/11,

M(X, C) =50/11, M(X, D) =40/11

When using the virtual connection with Node D, the evaluation formula *M(x, k)* is minimum.

Table 3 Virtual connection table for Node X

Step 3: The Node with the minimum value is selected and the connection is set up to this node.

As a result, Node X releases its connection with Node A and reconnects to Node D. Fig. 5 shows the reconnection of Node X.

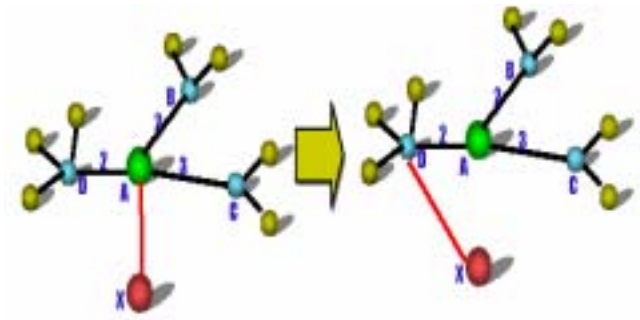

Fig. 5 Reconnection of Node X

4. Network Recovery Mechanism

4.1 Issues

In P2P networks, when a peer node leaves the networks or fails, a recovery mechanism is necessary to recover the split of the network. For example, when Node X fails in the P2P network shown in Fig. 2, the network is divided into 3 networks (see Fig. 6).

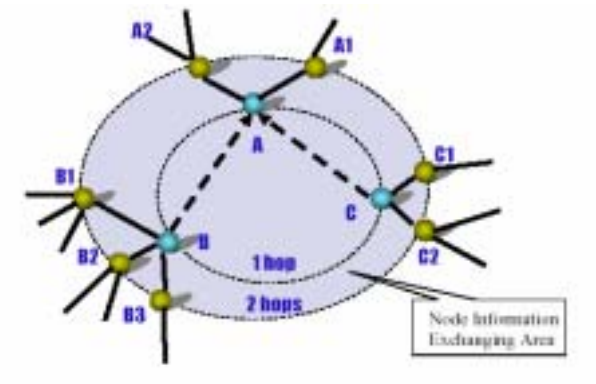

Fig. 6 Split Recovery of P2P Networks

4.2 Basic Mechanism

Topology optimization mechanism and network recovery mechanism have some similarities in terms of network reorganization. Therefore, both mechanisms should be common as much as possible. To this end, we take the similar approach as the topology optimization mechanism proposed in Section 3. For network recovery, we use the same table (i.e. Table 2) that is proposed for topology optimization

According to Table 2, every pair of the adjacent node holds a relationship of parent and child. The recovery mechanism is only executed along the direction to a parent node to prevent from the collision of recovery actions from two nodes. Recovery action is always initiated from a child node along its parent's direction. For example, Node B and Node C initiate recovery actions in Fig. 6.

The proposed mechanism is executed according to the following three steps.

Step1: When connecting with an adjacent node, a peer node establishes a parent-child relationship with its adjacent node as described in Section 3.1.

Step2: Each peer node broadcasts basic node information (i.e. Table 1) including IP address and its relationship with its adjacent peer nodes within the range of 2 hops. Hence, a table of local topology information (i.e. Table 2) can be created based on the received information as described in Section3.1.

Step 3: A peer node initiates recovery process when detecting departure of its adjacent node. Only a child node initiates recovery action. Using a table of local topology information, a child node tries to connect with a parent node of a departure node. For example, In Fig. 6 Node B and C connect with Node A.

4.3 Considerations

In addition to the basic mechanism described in 4.2, the following special cases should be considered.

(a) Case of Root Node

When a departure node does not have a parent node as shown in Fig. 7, the basic mechanism can not be directly applied. In this case, child nodes share local topology information. Such information includes maximum available connections of other child nodes. For example, since Node X has 3 connections and its maximum connections are 5 connections, its maximum available connections are 2 connections. Using such information, one child node is selected as a new parent node. For example, a node whose number of maximum available connections is largest is selected as a new parent node. In this case, for example, if the maximum available connections of Node A and Node C are 4 and Node B is 7, then Node A and Node C will connect with Node B to recover from the split of the P2P network.

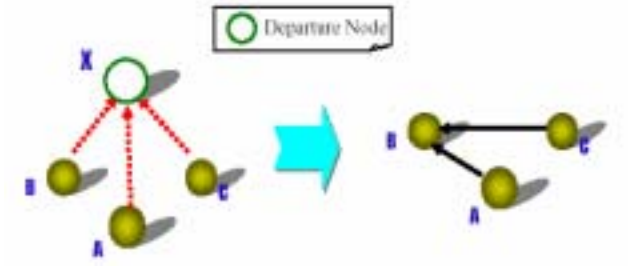

Fig. 7 Case of root node

(b) Case of Multiple Parent Nodes

The recovery mechanism should consider the cases where multiple parent nodes exist, as shown in Fig. 8. In Fig. 8, when Node X which has two parent nodes fails, child Nodes B and C of Node X try to connect to with Node X's two parent nodes (i.e. Node A1 and Node A2) to recover from the split of the network.

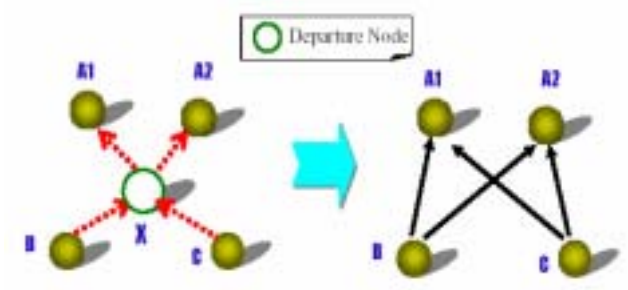

Fig. 8 Case of multiple parent nodes

5. Evaluation Considerations

To evaluate our mechanisms, a P2P simulator is under developing. We will evaluate proposed topology optimization and recovery mechanisms using a P2P simulator.

Our evaluation indicators include average hop count and average node connectivity. Average hop count between any pair of nodes relates to network traffic load and delay. It is an important evaluation indicator concerning P2P networks. Average node connectivity reflects average connection numbers of each node, and it shows the distribution of powerful nodes in a network, which is helpful to protect a network from attacks or system failures.

Other evaluation indicators such as time consumption to optimize a network, traffic consumption by exchanging node information and node information exchanging interval will be considered in future works.

The proposed mechanisms will also be implemented into Jupiter [6], a P2P platform for mobile Internet developed by NTT DoCoMo and Ericsson in the future.

6. Related works

There have been some researches regarding topology optimization of a P2P network. [7] proposes a distributed flow control and topology construction algorithm that (1) restricts the flow of queries into each node, so they don't become overloaded and (2) dynamically evolves the overlay topology, so that queries flow towards the nodes that have sufficient capacity to handle them.

In [8], each node independently defines a level of satisfaction. This quantity between 0 and 1 that represents how satisfied a node is with its current set of neighbors. As long as a node is not fully satisfied, the topology adaptation continues to search for appropriate neighbors to improve the satisfaction level. Finally, high capacity nodes are indeed the ones with high degree and that low capacity nodes are within short reach of higher capacity ones.

Both [7] and [8] do not consider the characteristics of lower physical links (e.g. hop count and delay) while adapting the overlay topology.

7. Conclusions

This paper proposes topology optimization and recovery mechanisms of unstructured P2P networks. We showed the requirements of those mechanisms. Based on the requirements, we propose autonomous topology optimization and recovery mechanisms using local topology information, considering the characteristics of lower physical links. Finally, we presented the methods of how to evaluate the performance of our proposal using a P2P simulator.

References

[1] Matei Ripeanu, "Peer-to-Peer Architecture Case Study: Gnutella Network", Technical Report, University of Chicago, 2001

[2] http://freenetproject.org/

[3] http://www.jxta.org/

[4] Q. Lv, P. Cao, E. Cohen, K. Li and S. Shenker. "Search and replication in unstructured peer-to-peer networks", Proceedings of the 16th ACM International Conference on Supercomputing, June 2002

[5] Pedram Keyani, Brian Larson, Muthukumar Senthil, "Peer Pressure: Distributed Recovery from Attacks in Peer-to-Peer Systems", The IFIP Workshop on Peer-to-Peer Computing, 2002

[6] Takeshi Kato, Norihiro Ishikawa, Hiromitsu Sumino, Johan Hjelm, Kazuhiro Miyatsu, Singo Murakami, "Jupiter: Peer-to-Peer Networking Platform toward Ubiquitous Communications", IPSJ SIGNotes UBIquitous computing system Abstract No.002-015

[7] Qin Lv, Sylvia Ratnasamy and Scott Shenker, "Can Heterogeneity Make Gnutella Scalable?", In Proceedings of the 1st International Workshop on Peer-to-Peer Systems, March 2002

[8] Yatin Chawathe, Sylvia Ratnasamy, Lee Breslau, Nick Lanham and Scott Shenker, "Making Gnutella-like P2P Systems Scalable", Proceedings of ACM SIGCOMM 2003