A Peer Selection Algorithm With Consideration of Both Network Topology Information and Node Capability in P2P Network

Tao Guo, Xu Zhou, Hui Tang
High Performance Network Laboratory,
Institute of Acoustics, Chinese Academy of Science
Beijing, China
rich0012@163.com

Zexu Wu
Department of Electrical Information,
Sichuan University
Chengdu, China
wuxexu1990@163.com

Abstract—**the features of P2P networking architecture can contribute to robustness and scalability, however, it also introduces network-oblivious traffic, which brings big pressure to the ISPs. Meanwhile, since traffic of P2P applications occupies much of bandwidth in bottleneck links, non-p2p applications may be severely harmed due to lack of bandwidth. To conquer this problem, previous works mainly focused on blocking the P2P traffic to benefit the ISPs, or optimizing peer selection algorithms which can only benefit the P2P applications but ignore the influence to the network. In this paper, we propose an algorithm, which is called PSANIC, to optimize peer selection in P2P networks with consideration of not only network topology information but also node capability. Simulation results show that PSANIC can achieve better performance than traditional DHT algorithms and the algorithm proposed in our previous work in [10], it can also reduce traffic significantly between domains in networks.**

Keywords-peer to peer;peer selection;network topology information;node capability

I. INTRODUCTION

Since P2P technology can transmit data more efficiently and make better use of network resources, things like file sharing and Instant Communication become much easier and convenient in P2P environment. Peer selection in P2P network, which means selecting other peers for a request peer to download data from, which will make great influence to the peer performance and traffic distribution of the whole network. Traditional P2P network treats all the peers equally and the peers who have the requested data copies will be selected randomly by the DHT algorithms [1][2][3][4]. These kinds of selections ignore the network information and bring huge of ISP-cross or domain-cross network traffic. The traffic of P2P applications, which have replaced the traditional HTTP applications in the aspect of Internet traffic generation, has occupied 50%~90% of Internet bandwidth nowadays. To solve this problem, many studies have been done.

In [5], a P2P content-based peer selection was introduced. Bittorrent clients try to find the total content copies stored by all intra-domain peers. If the intra-domain peers have 100% of the requested data, the clients will stop to connect the peers in extra-domains although one or more of the intra-domain peers does not have the whole data (the clients can get the data from the other bro-peers). Clients may also select peers using the algorithm proposed in [6] by taking the knowledge of underlying network into consideration; the hop counts and RTTs among these peers. Obviously, this algorithm can reduce the domain-cross network traffic. Besides these two factors including hop counts and RTTs, bandwidth is also a key aspect which we need to take into account to improve the network performance, the bottleneck bandwidth probing may work in practice such as introduced in [7]. A mechanism for optimizing P2P downloading was proposed in [8], it selects the appropriate peers, assign the reasonable tasks and make the suitable rate allocation to download requested data fast with minimum cost.. In order to make good use of the bandwidth within the AS (Autonomous System) of the network providers and alleviate P2P traffic burden over backbone, a new network architecture called P4P was introduced in [9] where the peer selection server, which is called *ptracker,* can inquire network information supplied by ISP server which is called *itracker*, and select peers according to these information.

To the best of our knowledge, few studies have been made to both reduce the AS-cross traffic and improve the P2P performance. Following factors should be considered if we try to determine the "cost" for those candidate nodes in peer sorting and selecting processes:

1) The network information reflecting geographical distances among candidate peers. This factor can make an impact on the amount of the AS-cross traffic. Like the routing cost in network layer, AS-crosss traffics are determined by the location of AS domains in which the candidate peers is located and the types of links through which the candidate peers are accessed to the Internet. More specifically, the network information about geographical information can be measured by the networking bandwidth, delay and so on.

2) The capability of nodes to join in files transferring. This factor gives an impact on the P2P application performance. Peers with strong capability we wonder to choose are expected to have more available bandwidth and less transmission delay.

978-0-7695-4077-1/10 $26.00 © 2010 IEEE
DOI 10.1109/ICICTA.2010.467

Our previous work about P2P traffic optimization in [10] is focused on selecting the ASs with minimum cost. It only depends on the link status between ASs in the core network and makes the assumption that all the nodes in the same AS have the same capability. Here we take a further step, which focuses on the node itself corresponding to the two above mentioned factors. Considering an object node *i*, as part of a delivery session, node *i* will transfer a portion of the data bytes to a client node. For such a delivery, when we sort and select the object nodes, we should take not only the cost of different ASs in the core network, but also the cost for node accessing networks and the node's capability for P2P file transferring into account.

The rest of the paper is organized as follows. Section II gives our network model, and section III details the proposed PSANIC algorithm. Section IV shows the simulation results. Finally, we conclude the paper.

II. NETWORK MODE OF THE ALGORITHM

The deployment of our network architecture is shown below:

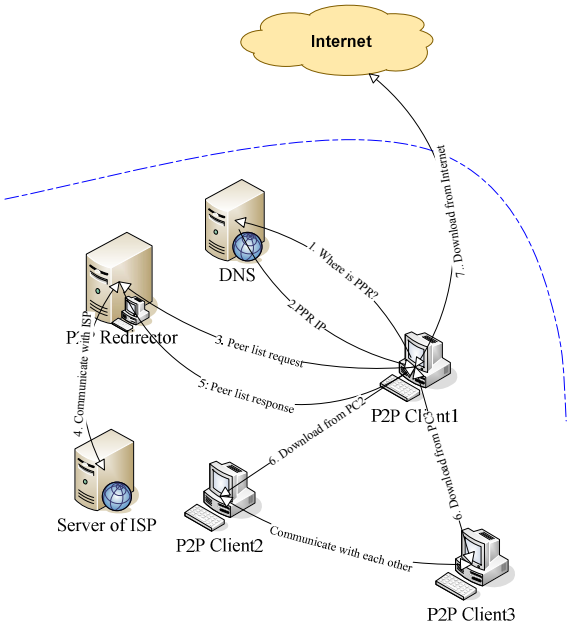

Figure 1. Deployment for one AS

Fig. 1 presents the framework for one networking domain that enables the interaction between P2P and ISP. P2P redirector, as shown above, whose function modules and interfaces are shown in Fig. 2, is deployed in each domain. It can collect the peers based on the data content through the DHT networking module, and it is also the peer index server which can provide the peer inquiry services for clients. The P2P redirector executes the peer selection algorithm (PSANIC) introduced in section III to sort and select the collected peers. Further more, it also provides the interface to P2P cache server, which is treated as the super downloading nodes to cache and proxy the data downloaded frequently and can be deployed depending on network capability.

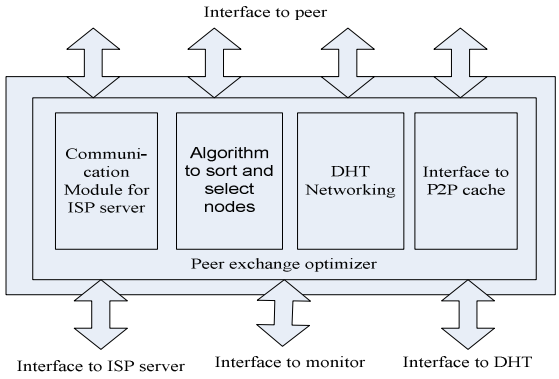

Figure 2. Modules and Interfaces of P2P redirector

Server of ISP stores the topology-related information and provides inquiry interfaces like the *itracker* in P4P construction. Both the P2P redirector and the ISP server act as logic entities and can be settled in one physical server. DNS server is generally used for domain name resolution. The process for a request client to find the appropriate data source nodes is shown in Fig. 3:

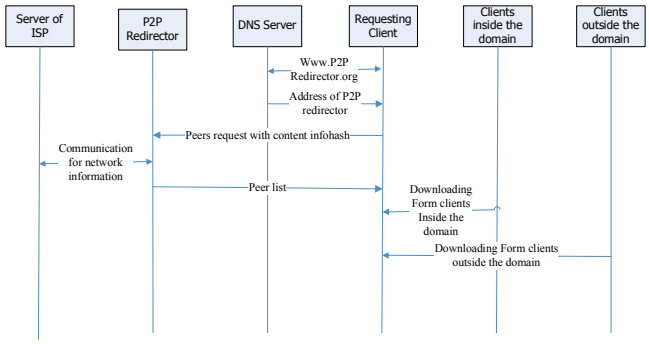

Figure 3. Processing flow for guiding the connection between P2P nodes

After all these servers complete their configurations such as assignment of an IP address, the following steps present the scheme in detail:

- Step 1: The request client sends the IP address request of the P2P redirector to the DNS, and then the DNS server sends the IP address as response to the request client.
- Step 2: After getting the IP address of the P2P redirector, the request client sends the peer list request including the content info-hash to the P2P redirector.
- Step 3: Before given the response to request client, the P2P redirector communicates with the ISP server for inquiring the topology information and sorting these candidate nodes depending on the receipt message, and then selects the set of nodes with minimum cost as the response peer list.

- Step 4: When getting the response from the P2P redirector, the request client tries to connect all of the nodes directly to download data bytes.

Since all these nodes have less cost, most of them locates in the AS where the request client locates in or nearby. Consequently, connecting to these nodes is significant to improve the network performance and alleviate the P2P workload.

III. Peer Selection Algorithm with Consideration of Both Network Topology Information and Node Capability in P2P Network

In this section, we introduce the peer selection algorithm called PSANIC. Definition of cost, which is determined by the node network information from the source node to the destination one and node capability, is the key for our sorting and selection. Thus, we should give our attention to both alleviating the network workload and improving the P2P performance.

Definition 1: The total cost from the source node to the destination node can be defined as follows:

$$Cost = \delta_1 \times C_{net\_info} + \delta_2 \times C_{node} \quad (1)$$

Where C_{net_info} is the cost calculated according to the node network information, and C_{node} is the cost depending on the node capability. δ_1 and δ_2 are the weight coefficients which can be adjusted and depends on the negotiation between P2P and ISPs. But we do not attempt to discuss how to set these weight coefficients and just present the formulation. The two terms presented in (1) are explained in part *A* and part *B* separately:

A. Cost determined by node network information

The network information includes two aspects: the AS where the object node locates in and the type for the node access to the Internet. Naturally, different from our previous work shown in [10], not only just the link status cost between ASs, but also the information for node access to the Internet can impact the network performance. Thus the C_{net_info} should be calculated in both of these two aspects. It is reasonable to assume that parameters for node access are almost the same because these intra-AS nodes have the same PoP and media access mode. Definitions for cost based on these two aspects are shown as below:

Definition 2: The cost between AS for request node and AS for candidate node, which depends on the link status, can be defined as follows:

$$CE_{ij} = \mu_1 / BandWidth_{ij}^{eq} + \mu_2 \times \sum_i^j Delay + \mu_3 \times \sum_i^j Hops \quad (2)$$

Here are the parameters' descriptions:

- CE_{ij} represents the cost just depending on the link status information between AS *i* and AS *j* .
- $BandWidth_{ij}^{eq}$ represents the equivalent bandwidth between AS *i* and AS *j*. When we deliver a packet from source to destination, the routing path may consist of one or more AS links as shown in Fig. 4. It is clear that Bandwidth 1 is the bottleneck. In this situation, we use the method as well as the traditional routing protocol OSPF to calculate the equivalent bandwidth:

$$\frac{1}{Bandwidth_{ij}^{eq}} = \sum_i^j \frac{1}{Bandwidth_k} \quad (3)$$

Where $Bandwidth_k$ represents the bandwidth of each link which is included in this routing path from AS *i to* AS *j*.

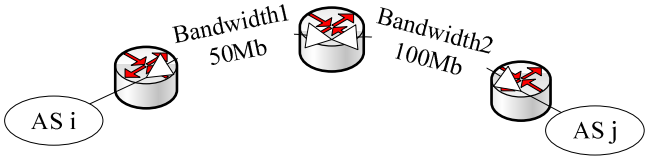

Figure 4. Sketch map for routing path bottleneck

- $\sum_i^j Delay$ represents the total delay from the source AS *i* to destination AS *j* , which includes the network links delay and the router/switcher delay. This parameter can be obtained via the ISP server, it also can be detected by the P2P redirector.
- $\sum_i^j Hops$ represents the number of the router/switcher hops from source AS *i* to destination AS *j*. More hops predicates more AS-cross P2P traffic going through the network backbone.
- μ_1 , μ_2 and μ_3 , which can be determined through the negotiation between different P2P applications and ISPs, are the weight coefficients to calculate CE, and they reflect the importance of different parameters.

Considering that the packet loss rate is tiny in the Internet core, we take it into account in definition 3. With the general sight of the whole network including all ASs, we can get the matrix of CEs, where the matrix row index denotes the source network ID and the column index denotes the destination network ID:

$$M_{CE} = \begin{bmatrix} CE_{00} & CE_{01} & CE_{02} & \cdots \\ CE_{10} & CE_{11} & CE_{12} & \cdots \\ CE_{20} & CE_{21} & CE_{22} & \cdots \\ \cdots & \cdots & \cdots & \cdots \end{bmatrix} \quad (4)$$

Definition 3: The cost for node *i* which belongs to certain AS accessing to the Internet can be defined as follows:

$$CP_i = \eta_1 \Big/ BandWidth_i^{Ac} + \eta_2 \times Delay_i^{Ac} + \eta_3 \times Lost_i^{Ac} \quad (5)$$

These are the parameters' description:

- CP_i represents the cost for nodes which belong to AS i accessing to the Internet.
- $BandWidth_i^{Ac}$ is presented for the access bandwidth of nodes which belong to AS *i*. Different kinds of access type have different access bandwidth, such as 100Mb for LAN, 1.544Mbps (T1) for ISDN in America or Japan commonly and so on.
- $Delay_i^{Ac}$ represents the delay for nodes which belong to AS *i* accessing to the Internet.
- $Lost_i^{Ac}$ represents the packet loss rate for nodes which belong to AS *i* accessing to the Internet. It is clear that the wireless access mode has the higher packet loss rate than the wire cable access modes.
- Like μ_1, μ_2 and μ_3 mentioned above, η_1, η_2 and η_3 are also the weight coefficients for different access parameters and they reflect the importance of them.

Besides the cost of links between different ASs in the core network, the access cost CP_i for the source node *i* and the access cost CP_j for destination node *j* should be added when we transmit a packet through the path from the source node to the destination one. Consequently we can get the matrix of source and destination pairs for the overall network:

$$M_{CP} = \begin{bmatrix} 2CP_0 & CP_0 + CP_1 & CP_0 + CP_2 & \dots \\ CP_1 + CP_0 & 2CP_1 & CP_1 + CP_2 & \dots \\ CD_2 + CP_0 & CD_2 + CP_1 & 2CP_2 & \dots \\ \dots & \dots & \dots & \dots \end{bmatrix} \quad (6)$$

Definition 4: The total cost C_{net_info}, which based on the node network information between AS *i* where the request node locates in and AS *j* where the object nodes locate in, could be defined as follows:

$$C_{net\_info} = CE_{ij} + CP_i + CP_j \quad (7)$$

Thus we can get the matrix M_{CT} as follows:

$$M_{CT} = M_{CE} + M_{CP} \quad (8)$$

For each row of M_{CT}, the elements of the matrix denote the total cost of the AS pairs. Less cost implies that this candidate AS is closer to the source AS, it has fewer network hops and less network delay etc. The matrix M_{CT} can be saved as a table. When the P2P redirector calculate the node cost, it will get the C_{net_info} quickly without the repeat of calculation.

B. Cost determined by node capability

Although the nodes in the same AS have almost the same access information, the number of connected sessions may be different and change along with the time. In other words, the available bandwidth and the data transferring delay may be different among these candidate nodes. Thus their capabilities are different. If the P2P redirector wants to get the intra-nodes' information of their connected session numbers to calculate the node capability, there exits two methods to achieve this goal:

1) When a certain node has been selected as one of the response peer list by the P2P redirector and been sent to the request node, the P2P redirector would record this session and numerate the number for this node.

2) The connected session number will be recorded by the P2P clients themselves and update it while a new connection comes. The client nodes will send the message which includes the number information to the P2P redirector periodically or the new connection events will trigger the notification message sending.

Definition 5: The cost determined by node capability can be defined as follows:

$$C_{node} = \frac{\gamma_1 \times NUM_{total}}{(NUM_{total} - NUM_{con}) \times Bandwidth_i^{Ac}} + \frac{\gamma_2 \times SegSize \times NUM_{total}}{(NUM_{total} - NUM_{con}) \times Bandwidth_i^{Ac}} \quad (9)$$

Where C_{node} is the intra-node cost, the formula description is as follows:

- The first term in formula (8) shows the cost due to the available bandwidth. NUM_{total} represents the maximum session number of nodes and NUM_{con} represents the connected session number.
- $SegSize$ represents the size of data segment which need to upload to other nodes. The second term in this formula shows the cost due to the data segment's transferring delay. As the same above, γ_1 and γ_2 are the weight coefficients.

After getting the C_{net_info} from the M_{CT} matrix and calculating the cost of node capability, we will get the total cost for a candidate node. We can select the less key nodes as the response nodes. Because the total cost takes both the network information and node capability into account, we have found the balance between reducing the AS-cross traffic and optimizing the node performance.

In perspective of software, the P2P redirector is capable of collecting the nodes for a certain content through the DHT

networking module, the quantity of these contents is large and we can organize them as a binary tree with the time complexity for searching by $O(\log m)$, where m is the quantity of info-hash. For the set of candidate nodes, we build a minimum heap using the total cost key to store and select nodes. It has a running time of $O(\log n)$ for the node joining or leaving the network, where n is quantity of candidate nodes. If the request clients asks for l nodes ($l<<n$) for downloading data bytes, the time complexity for selecting is $O(l \times \log n)$.

IV. SIMULATION AND ANALYSIS

In this section, we evaluate the improvement of PSANIC algorithm in P2P network. We compare the PSANIC algorithm with the traditional DHT algorithm and the algorithm proposed in [10], which just select the less cost ASs but ignore the difference of intra-AS nodes, in terms of some network parameters of the response peers. As shown in Fig. 5, with consideration of network diversity, we deploy 5 ASs corresponding to 5 kinds of different access types respectively:

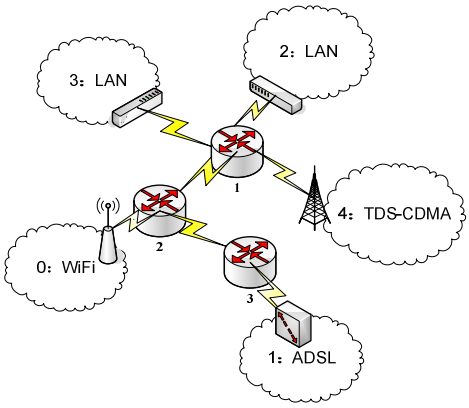

Figure 5. Deployment of the experiment system

Fig. 5 is just the topology map of the deployment and we explicate the node access parameters in detail for different ASs:

TABLE I. TABLE TYPE STYLES

Net ID	Accessing Parameters			
	Access mode	*Bandwidth(kbps)*	*Delay(us)*	*Lost (%)*
0	WiFi	150	10000	5
1	ISDN	100	4000	2
2	LAN	2000	2000	0.05
3	LAN	2000	1000	0.05
4	TDSCDMA	2	20000	10

Although the bandwidth in core network links is sufficient enough and can be capable of holding thousands of connections to transmit data bytes like the Gigabit Fiber transmission technology, the available bandwidth in these links are uncertain one time and obviously less than their capability. We suppose that the available bandwidth between router 1 and router 2 is 50Mbps and between router 2 and router 3 is 10Mbps. It is simple to calculate the equivalent bandwidth between any two of ASs. Moreover, we also assume that the data transmission delay for both of the two core links is 1000 μs and for all the three routers is 500 μs. Comparing with the DHT algorithm and our previous work, the improvement of PSANIC algorithm is shown in the following figures:

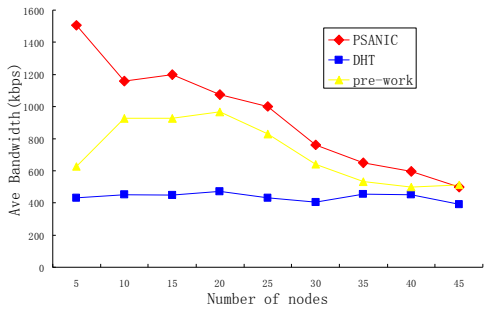

Figure 6. Comparison of average bandwidth for response nodes

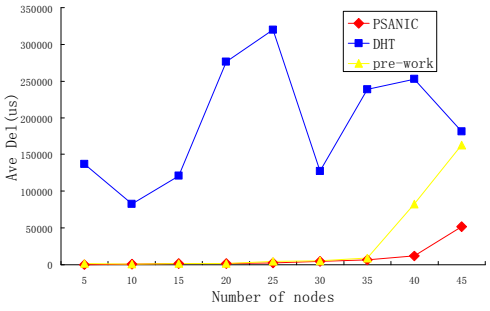

Figure 7. Comparison of average data trasmission delay for response nodes

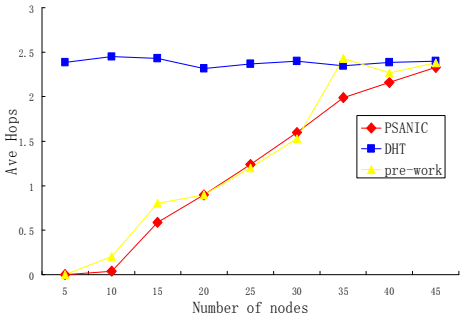

Figure 8. Comparison of average routing hops for response nodes

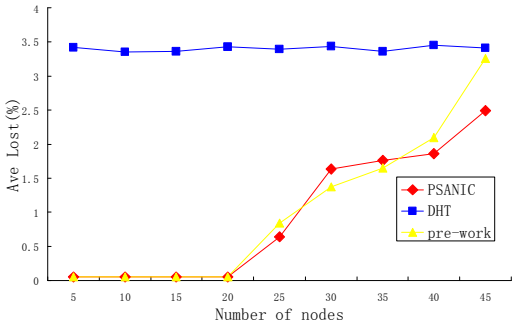

Figure 9. Comparison of average packet loss rate for response nodes

From Fig. 6-9 we can see that peer list nodes returned by the PSANIC algorithm and the algorithm proposed in [10] have obviously not only more average bandwidth, less average data transmission delay and less packet loss rate, which of these parameters implies better node performance, but also have less average routing hops, which means connecting to these nodes determines to have much less AS-cross traffic traveling through the core network than the DHT algorithm, for both of these two algorithms are aware of the network topology.

Moreover, from Fig. 6-7, we can see that the PSANIC algorithm performs better than our pre-work algorithm, although both of them are aware of the network topology. The response nodes returned by PSANIC have more average bandwidth and less data transmission delay. That is because the pre-work algorithm just selects the ASs and ignores the node capability, in other works, a node in the prior AS with low capability will have high priority than the node in the lower priority AS with high capability. Suppose each AS has 10 nodes and the request node in AS 3 asks for 20 candidate nodes, the response peer list sequence returned by all the three algorithms is shown in Fig. 10. From this figure we can find that our pre-work algorithm mostly selects all the nodes in AS 3 and in AS 2 even the nodes in these two AS have low capability, while the PSANIC algorithm selects the nodes in prior ASs with high capability. However, the DHT algorithm selects them almost randomly.

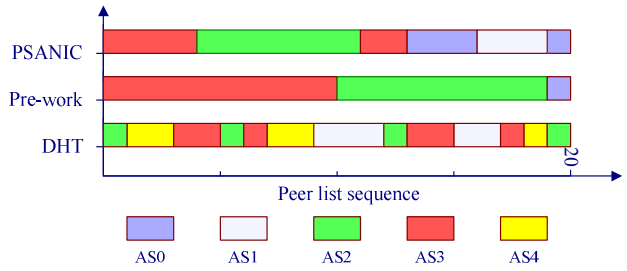

Figure 10. Peer list sequence for all the three algorithms

In terms of the average hops and the average packet loss rate, From Fig. 8-9 we can find that the PSANIC algorithm and our pre-work algorithm perform mostly the same, that is because the routing hop is mainly determined by geographical location of ASs and the packet loss rate is mainly determined by the node access mode, these two kinds of networking information are stable comparatively along with the time in Internet and have been taken into account by both PSANIC and our pre-work algorithm.

SUMMARY

In this paper, we made an in-depth investigation on the issue of peer selection, which is a fundamental issue in P2P systems. Then we proposed a peer selection algorithm called PSANIC with consideration of both network topology information and node capability. Simulation results indicate that our algorithm performs better than the DHT algorithms and pre-work algorithm. Both the P2P clients and the ISPs can benefit from our algorithm.

REFERENCES

[1] B. Zhao, J. Kubiatowicz, and A. Joseph, "Tapestry: An infrastructure for wide-area fault-tolerant location and routing," U.C. Berkeley Technical Report UCB//CSD-01-1141, April 2001.

[2] I. Stoica, R. Morris, D. Liben-Nowell, D.R. Karger, M.F. Kaashoek, F. Dabek, H. Balakrishnan, "Chord: a scalable peer-to-peer lookup protocol for Internet applications," Proceedings of the IEEE/ACM Transactions on Networking, vol. 11, pp. 17-32, Feb 2003.

[3] A. Rowstron, P. Druschel, "Pastry: Scalable, Decentralized Object Location, and Routing for Large-Scale Peer-to-Peer Systems," Proceedings of the ACM SIGCOMM, 2001.

[4] D. Loguinov, A. Kumar, V. Rai, S. Ganesh, "Graph-Theoretic Analysis of Structured Peer-to-Peer Systems: Routing Distances and Fault Resilience," Proceedings of ACM SIGCOMM, vol. 33, pp. 395-406, October 2003

[5] OY. Rong, C. Hui, "A Novel Peer Selection Algorithm to Reduce BitTorrent-like P2P Traffic between Networks," Proceedings of the Information Technology and Computer Science, 2009. vol. 2, pp. 397-401, July 2009.

[6] LI Wei, C. Shanzhi, YU Tao, "UTAPS: An Underlying Topology-Aware Peer Selection Algorithm in BitTorrent," Proceedings of the Advanced Informative Networking and Applications, 2008. pp. 539-545, March 2008.

[7] TSE. Ng, Y. Chu, S. Rao, K. Sripanidkulchai, H. Zhang, "Measurement-based optimization techniques for bandwidth-demanding peer-to-peer systems," Proceedings of the IEEE INFOCOM, vol. 3, pp. 2100-2209, 30 March-3 April 2003.

[8] Kai Han, Qingyu Guo, Jing Luo, "Optimal Peer Selection, Task Assignment and Rate Allocation for P2P Downloading," Proceedings of the Intelligent Control and Automation, vol. 1, pp. 397-401, July 2009.

[9] Haiyong Xie, Y. Ysang, A. Krishnamurthy, Yanbin Grace Liu, A. Silberschatz, "P4p: Provider portal for applications," Proceedings of the ACM SIGCOMM 2008 Conference on Data Communication, SIGCOMM'08, vol. 38, pp. 351-362, 2008.

[10] T. Guo, X. Zhou, Z. Wang, H. Tang, "Mechanism for Optimizing P2P Traffic Between Networks Base on Network Measurement," Journal of Computer Applications, Apr. 2010 in press.