Content Sharing among UPnP Gateways on Unstructured P2P Network Using Dynamic Overlay Topology Optimization

Ema Kawamoto[1], Kohei Kadowaki[1], Takahiro Koita[1] and Kenya Sato[1]

[1]Department of Information Systems Design, Doshisha University, Kyotanabe, Kyoto 610-0321, Japan

Such home networking technologies as UPnP have become common for users to connect their devices at home. In this paper, we propose Dynamic Overlay Topology Optimizing Content Search (DOTOCS) that enables flexible content searches while optimizing P2P overlay networks. DOTOCS allows nodes to change their logical links to reach other nodes that have frequent data transfers in fewer hops, providing users with a better chance of finding their desired contents in quicker downloading time. For this reason, DOTOCS is useful to establish efficient connections among UPnP gateways on unstructured P2P networks for content sharing. We implemented DOTOCS using a P2P simulator to evaluate and measure its performance. The simulation results proved that DOTOCS increases search performance by 30 to 40% compared with a simple content search method that omits optimization.

Index Terms—**Content Sharing, Network servers, Peer-to-Peer networks, Topology, Universal Plug and Play**

I. Introduction

Recently, more and more people are using home networking technology to connect their home appliances and digital devices (e.g., TVs, computers, cameras, PDAs, and iPhones) for data sharing, communication, and entertainment purposes. UPnP [1] is one major home networking technology that allows devices to connect seamlessly to simplify network implementation in the home. On the other hand, sharing contents containing audio, video, data, or anything in a digital format using P2P technology has also become very common. With the advent of home networking and P2P technologies, expectations continue to grow that in the near future people will be able to share contents not only within home networks but also among digital devices in different home networks over such global TCP/IP networks as the Internet. To meet such demand, there are several proposals that describe solutions for content sharing among UPnP devices in different local networks [2]-[4]. It is assumed that it will not be long before people can share contents regardless of what local network the device that has the desired content belongs to. In this paper, we propose Dynamic Overlay Topology Optimizing Content Search (DOTOCS) to establish a P2P overlay network that efficiently connects UPnP gateways so that users have more chances of finding requested contents and can quickly download contents.

P2P networks are famous for their high scalability as opposed to client-server architecture [5]. However, since the location of contents in P2P networks can't be managed, the searching process for contents is complicated. Two major search methods are already used in many P2P applications. One is applied in structured P2P networks that use Distributed Hash Tables (DHT) such as Chord [6], and the other is called the bucket-relay method, which is applied in unstructured P2P networks, Gnutella, for example [7]. The DHT method, however, can't make flexible searches. On the other hand, the bucket-relay method allows flexible content searches but they take time, and users may not be able to find their desired content even though it exists in the network.

II. Dotocs

A. Overview

To overcome these problems, we propose DOTOCS, which enables flexible content searches and efficient content download by optimizing P2P overlay networks. Note that DOTOCS functions on the assumption that each node has its own "taste" for the content being sought, meaning each node tends to search for the same type of content every time. To enable flexible searches, nodes flood queries when searching for contents, as in the bucket-relay method. To enable efficient content download, each node makes a *response-traffic table* to record the number of responses that passed through the node itself. From that record, nodes predict what type of content a neighbor node has. By doing so, nodes can automatically locate themselves near the nodes with similar content, which we refer to as *dynamic overlay topology optimization*. In other words, DOTOCS gradually optimizes a network during content searches. Figure 1 shows the pictorial concept of this optimization. The network before optimization is depicted on the left and after on the right. The numbers indicate the type of content held by the peers. The figure reveals that even though nodes are randomly linked with each other before the optimization, regardless of the type of content, nodes of similar content are probably closely linked after the optimization, allowing queries to take fewer hops to reach their destinations and giving nodes more chances to discover their desired contents.

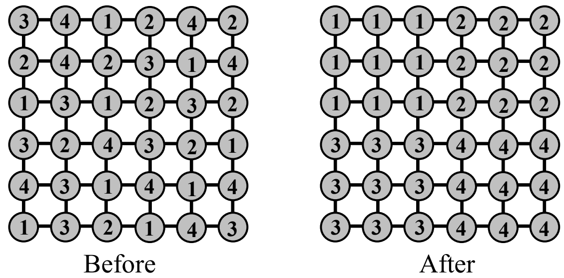


Figure 1: Optimization concept

B. Communication between UPnP Gateways

The goal of UPnP is to allow devices to connect seamlessly and to simplify the implementation of networks in the home. Under the UPnP protocol, a UPnP-enabled device multicasts an SSDP (Simple Service Discovery Protocol) packet to advertise its services or to search for devices of interest on the network. Since multicast packets can be sent only to devices on the same local network, this causes a problem when a device wants to control a device in a different local network. Even if an SSDP packet could be routed to a UPnP gateway in a different local network and was sent to control points in that local network, the control points would not be able to send a response back because the payload of the packet contains the sender's local IP address, which is not routable in other local networks.

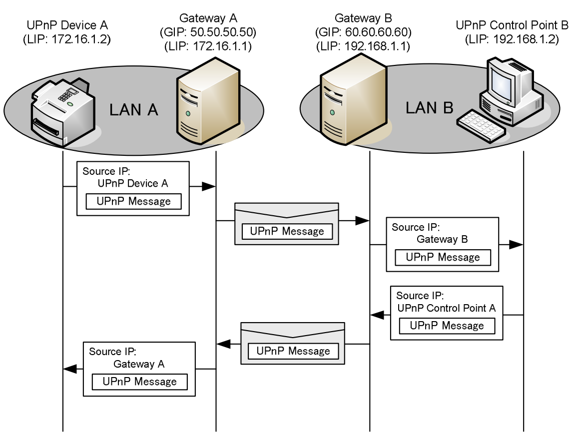


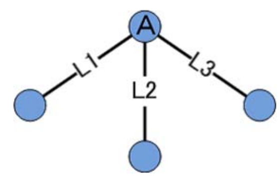
Figure 2: Message exchange between different local networks

Reference [8] proposes a device interaction method over the Internet to solve the problem by encapsulating a SSDP packet into SOAP message. Figure 2 shows two local networks of which gateways are connected through a global network. Assume device A has just joined a network and multicasted a SSDP packet to devices on the same local network. When gateway A receives the SSDP packet from device A, it encapsulates the SSDP packet into a SOAP message and sends it to gateway B over the global network. After gateway B receives the SOAP message, the web service in the gateway extracts the SSDP packet and multicasts it to all the devices on its local network. Since the SSDP packet contains the local IP address of device A's, the web service change device A's local IP address to gateway B's local IP address. By doing so, control point B regards gateway B's local address as device A's local address. Therefore, when control point B sends a message to device A, the message is first sent to gateway B, and then sent to gateway A, and finally sent to device A, which make it possible for control point B to manipulate device A even though they are in different local networks.

C. Dynamic Overlay Topology Optimization: Link Change Procedures

1) Response Recording

Dynamic overlay topology optimization is performed by dynamically changing links. Nodes change their links, if necessary, by looking up the number of responses that have crossed them in the response-traffic table to place themselves near the nodes with similar content. Figure 3 shows an example of node A's response-traffic table node. Since node A is connected to three nodes, it records the number of responses from L1, L2, and L3 regardless whether the responses were addressed to A itself. Node A also records the number of responses that were transferred between the two neighboring nodes. Therefore, in this case, node A keeps a record of the number of responses transferred between L1 and L2, L1 and L3, and L2 and L3.



L1	L2	L3	L1→L2	L1→L3	L2→L3
T(L1)	T(L2)	T(L3)	T(L1,L2)	T(L1,L3)	T(L2,L3)

Figure 3: Node A's response-traffic table

There are three conditions for the link changes: *threshold-outnumbered case*, *no-traffic-link case*, and *no-response-back case*.

2) Threshold-outnumbered Case

The 1st condition, the threshold-outnumbered case, is fulfilled when the number of responses transferred between the two neighboring nodes within a certain amount of time outnumbers the threshold. When this condition is met, the node deletes the link with the least response traffic and lets the two neighboring nodes establish a new link between them. The goal of this link change is to reduce the hop count between nodes with similar content. When a node has n links {L1, L2, ... Ln}, as shown in Figure 4, threshold Cm of link Lm can be calculated by the following formula. T(Lm,Lt) in the formula indicates the number of responses transferred from links Lm to Lt:

$$Cm = \left\{ \sum_{t=1}^{n} T(Lm, Lt) \right\} / 2 \,, \tag{1}$$

Figure 5 illustrates how node A changes its links. If the number of responses transferred between nodes B and C is larger than the threshold of links L2 or L3, node A deletes the link with the least response traffic and establishes a new link between nodes B and C. In DOTOCS, nodes exchange SOAP messages using HTTP to change their links as shown in figure 6. In this case, node A sends node C 'DELETE A' message to have node C delete the link to A and 'ADD B' message that contains node B's IP address to have node C establish a link to node B. After receiving the messages from node A, node C sends node B 'ADD B' message to ask node B to establish a link to it. If node B response it with 'ADD B OK' message, node C response to node A with 'DELETE A OK' message, which indicates that node A and node C are ready to delete the link between them. After the link is deleted, node B and node C establish a link between them.

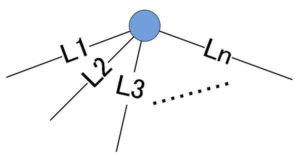


Figure 4: A node with n links

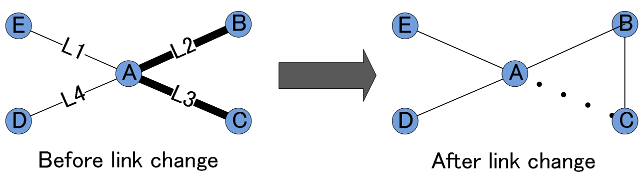


Figure 5: Link change when 1st condition is met

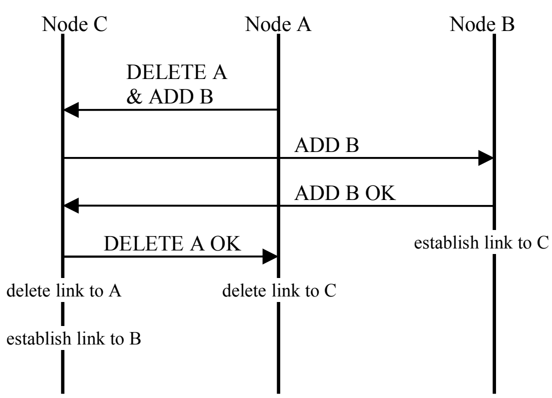


Figure 6: SOAP message exchange for link change

3) *No-traffic-link Case*

The 2nd condition, the no-traffic-link case, is fulfilled when there is a link through which no response has been transferred within a certain amount of time, as shown in Figure 7. In this case, no responses were exchanged between nodes A and B and between A and C. In that case, node A assumes that the links to nodes B and C are useless because no network uses them to download contents. Therefore, node A deletes the links and then randomly looks for new neighbors to establish links to them. The SOAP message exchange procedure for this link change is shown in figure 8. First, node A sends 'DELETE A' message to node B. When node A gets a positive response from node B, the both nodes deletes the link between them. Then, node A sends 'ADD A' message to a new neighbor node. When node A gets a positive response from the neighbor node, they establish a link between them. When deleting the link to C, node A follows the same procedure as it has done to node B.

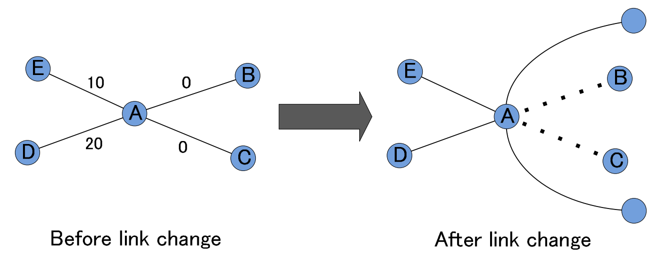


Figure 7: Link change when 2nd condition is met

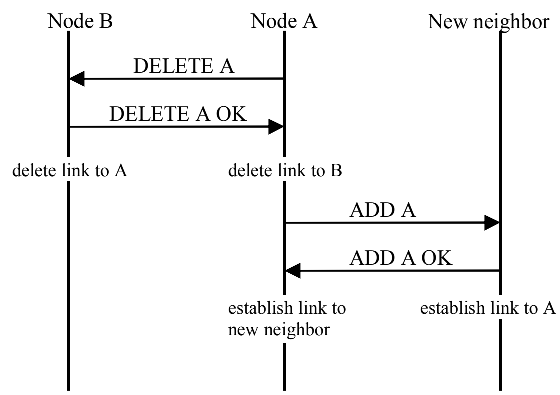


Figure 8: SOAP message exchange for link change

4) *No-response-back Case*

The 3rd condition, the no-response-back case, is fulfilled when a node has not received any responses addressed to it. Figure 9 shows an example of this case where node A receives responses from all its neighboring nodes, but none were addressed to node A. In this case, node A deletes all of its links and instead establishes new links to random neighboring nodes. The procedure for these link changes is basically the same as the one in figure 8.

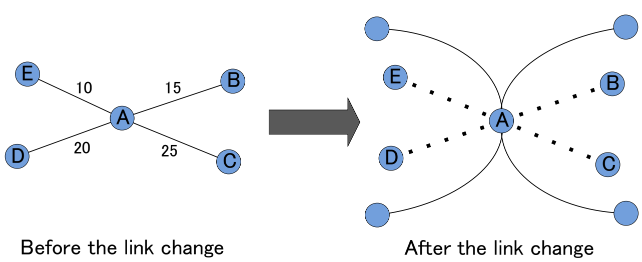


Figure 9: Link change when 3rd condition is met

III. Performance Evaluation

A. *Implementation*

We implemented DOTOCS using a simulator called *Peersim* [9], an open-to-public P2P simulator written in Java. Table 1 shows our initialization parameters to the simulation. As already mentioned, since DOTOCS functions on the assumption that each node has its own "taste" for the content being searched for, in the first stage of the simulation, nodes are classified into categories based on their content; during the simulation they search for the same type of content every time.

Table 1: Parameters used in simulation

Initialization Parameters	
Nodes	400
TTL	4
Neighboring nodes	4
Types of content	4, 25, 100
Number of queries issued every cycle	20

B. Evaluation

To evaluate the performance of DOTOCS, we compared it with a simple content search method (SCSM) that floods queries when searching for contents but does not change links during the searches. The difference between DOTOCS and SCSM concerns whether they perform dynamic overlay topology optimization. During the simulation, we measured each of the two methods' content discovery rates and compared them. Figure 10 shows the content search rates of the two methods. The nodes found the desired content about 40% of the time using SCSM. On the other hand, the nodes found their desired content slightly less than 70% of the time using DOTOCS when there were four types of content in the network, slightly more than 70% when there were 25 types of content, and a little more than 80% with 100 types of content. These results prove that DOTOCS can improve content discovery rates by 30 to 40% depending on the number of types of content in the network. The results illustrate that the more types of content there are, the better the search performance becomes, which also indicates that the stronger tendencies nodes have, the greater contribution DOTOCS can make to optimization.

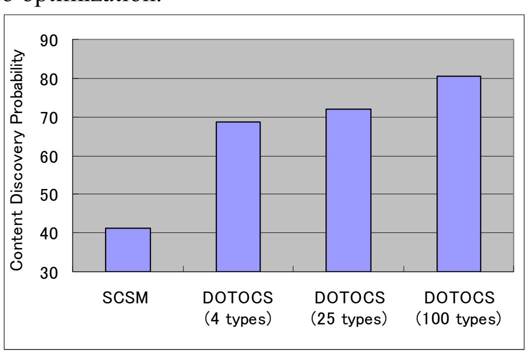


Figure 10: Comparison of DOTOCS with SCSM

Figure 11 is another type of result that shows the relation between time and the content discovery rates. To discover when the optimization is finished, we measured the content discovery rates every 50 cycles where there are 100 types of content in the network. In every cycle, nodes that received a query or a response pass over the message to its neighboring node. Therefore, a cycle can be regarded as the time for the sake of simplicity. The graph shows that content discovery rates increase as time passes by converging toward 80% approximately at the 200th cycle. This is when optimization is finished. Since 20 random nodes issue a query every cycle and there are 400 nodes in the network, the results indicate that the nodes can establish links to their best partners after issuing 10 queries individually.

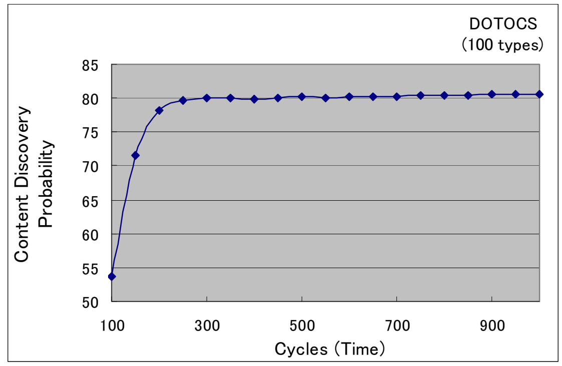


Figure 11: Time-wise optimization improvement changes

IV. Conclusion And Future Work

This paper proposed DOTOCS, a method that efficiently connects UPnP gateways to unstructured P2P networks, to reduce the time for downloading contents and improve search performance. From simulations, we verified that DOTOCS can optimize a P2P network and improve the search performance by 30 to 40% after each node issued a query more than 10 times. Based on this result, we expect that DOTOCS will help UPnP gateways establish links efficiently on an unstructured P2P network so that users will have more chance of finding desired contents and can quickly download the contents.

We haven't yet, however, applied DOTOCS to a real P2P network of UPnP gateways and measured the time required to download a file. Therefore, we must examine how to implement DOTOCS on UPnP gateways to evaluate its downloading performance.

There are several other proposals that optimize the topology of a P2P overlay network considering the characteristics of underlying physical links [10], [11]. Since the current DOTOCS fails to take account of the structure of physical networks yet, we are considering extending the related works and taking account of new metrics (e.g., hop count in physical networks, RTT, bandwidth) when making link change decisions to minimize the data transferring delay by avoiding inefficient routing in physical networks.

References

[1] UPnP Forum, "The UPnP Device Architecture Version 1.0," Apr. 2008.

[2] H. Y. Lee, and J. W. Kim. "An Approach for Content Sharing Among UPnP Devices in Different Home Networks" *IEEE Transactions on Consumer Electronics,* vol.53, 2007, pp. 1419-1426.

[3] S. Motegi, K. Tasaka, A. Idoue, H. Horiuchi, "Proposal on Wide Area DLNA Communication System," *Proc. of IEEE Consumer Communications and Networking Conference*, Jan. 2008, pp.233-237.

[4] D. Muto and T. Yoshinaga, "Wormhole Device: Software-Assisted Remote Communication Mechanism for DLNA-Based Appliances,"

Proc. of Multimedia, Distributed, Cooperative and Mobile Symposium, Jul. 2007, pp. 134-138.

[5] Q. Lv, P. Cao, E. Cohen, K. Li, and S. Shenker. “Search and Replication in Unstructured Peer-to-Peer Networks,” *Proc. of the 16th ACM Int’l Conf. on Supercomputing*, June 2002.

[6] I. Stocia, R. Morris, D. Karger, F. Kaashoek, and H. Balakrishnan, "Chord: A Scalable Peer-to-Peer Lookup Service for Internet Applications," *Proc. of SIGCOMM 2001*, August 2001, pp. 84-95.

[7] M Ripeanu, A. Iamnitchi and I. Foster, "Mapping the Gnutella Network," *IEEE Internet Computing*, January/February 2002, pp. 50-57.

[8] M. Ogawa, H. Hayakawa, T. Koita, and K. Sato. “Transparent UPnP Interactions over Global Network,” *Proc. of SPIE (ICMIT 2007)*, Vol.6794, pp.4P-1-6, 2007.

[9] PeerSim: A Peer-to-Peer Simulator. http://peersim.sourceforge.net/

[10] Y. Liu, Z. Zhuang, L. Xiao, and L. M. Ni. “AOTO: Adaptive Overlay Topology Optimization in Unstructured P2P Systems.” *Global Telecommunications Conference, 2003. GLOBECOM '03. IEEE*, vol.7, 2003, pp. 4186-4190.

[11] H. Wan, N. Ishikawa, and J. Hjelm. “Autonomous Topology Optimization for Unstructured Peer-to-Peer Networks.” *Proc. of 11th International Conference on Parallel and Distributed Systems*, Vol.1, 2005, pp. 488-494.