Efficient Overlay Network for P2P Content Sharing Based on Network Identifier

Chanmo Park and JongWon Kim

Networked Media Lab. Department of Information and Communications,
Gwangju Institute of Science and Technology (GIST), Gwangju, 500-712, Korea
{cmpark,jongwon}@netmedia.gist.ac.kr

Abstract. In this paper, we propose an efficient overlay network for P2P (peer-to-peer) content sharing by incorporating network identifier of each peer in the construction of CAN (content addressable network)-variant DHT (distributed hash table)-based overlay. The network identifier that partially reflects the locality of each peer within the global Internet can help us to aggregate distributed peers towards more structured set of virtual peers. Through myns-based simulations coupled with GT-ITM topology, the improved efficiency of the proposed overlay construction is verified.

1 Introduction

P2P (peer-to-peer) concepts and networks for sharing files among peers have been popular since Napster [1] started its service at mid 1999. With the popularity of P2P, many researchers have been interested in media streaming systems based on P2P file sharing that serve as a directory server for locating the file and the owner of it. To discover contents for P2P media sharing systems, centralized or distributed approaches might be used. Napster, the centralized approach, has a centralized server that serves as a directory server and clients that download files by sending queries to nodes that own the file. In general, distributed content discovery approaches based on overlay network can be classified into two categories: unstructured and structured approach. Under unstructured Gnutella and KaZaA [1], all peers simultaneously take roles of a client, a server, and a message routing node on the overlay network. Structured approaches [1] such as Chord, Tapestry, Pastry, and CAN (content addressable network) [2] construct structured overlay network using DHT (distributed hash table). The structured overlay networks with DHT are based on the virtual coordinate space into which contents are mapped by hash functions. Thus, peers on the structured overlay network with DHT store content information (i.e., pointers to contents) on other nodes according to the assigned range within the virtual coordinate space. Routing messages are exchanged by forwarding along neighbor peers whose range are closer to a destination node. Therefore, the performance of structured overlay network with DHT is determined by the number of hops needed to route messages to the destination node. It is important to improve the performance of

K. Aizawa, Y. Nakamura, and S. Satoh (Eds.): PCM 2004, LNCS 3332, pp. 373–380, 2004.

content discovery to enhance P2P-based media streaming systems [3,4]. In this paper, we focus on improving the underlying P2P content sharing overlay network with DHT by reducing the number of hops for message routing. We use the network identifiers of participating peers to aggregate them into groups. In the proposed approach, the adapted virtual coordinate space is 2-dimensional and is occupied by participating peers according to the mapping rule based on their network identifiers. Participating peers having the same network identifier are mapped into the same virtual coordinate space. We expect this grouping to help as to reduce the number of hops for message routing.

The remaining part of this paper is organized as follows. In Section 2, we briefly review DHT's focusing on the CAN. The proposed overlay construction with the network identifier is described in Section 3. Section 4 shows the simulation environment and results. Finally, we conclude the paper in Section 5.

2 Background and Related Works

The CAN [2] is based on a d-dimensional Cartesian coordinate space on a d-torus. The Cartesian space of CAN is completely logical. The entire coordinate space is dynamically partitioned among all the nodes in the system such that every node owns its individual, distinct zone within the overall space. The virtual coordinate space is used to store $(key, value)$ pairs as follows: to store a pair (K_1, V_1), key K_1 is deterministically mapped onto a point P in the coordinate space using a unform hash function. The corresponding $(key,value)$ pair is then stored at the node that owns the zone within which the point P lies. To retrieve an entry corresponding to key K_1, any node can apply the same deterministic hash function to map K_1 onto point P and then retrieve the corresponding value from the point P. If the point P is not owned by the requesting node or its immediate neighbors, the request must be routed through the CAN infrastructure until it reaches the node in whose zone P lies. Efficient routing is therefore a critical aspect of a CAN. Nodes in the CAN self-organize into an overlay network that represents this virtual coordinate space. A node learns and maintains the IP addresses of those nodes that hold coordinate zones adjoining its own zone. This set of immediate neighbors in the coordinate space serves as a coordinate routing table that enables routing between arbitrary points in this space. However, it is known that the resulting performance of CAN may suffer inefficiency problem when the number of zones is increased too much (as it adds a zone whenever a new node joins) [5]. To improve the routing efficiency of the CAN, one approach is to reduce each CAN-hop latency by using network proximity (i.e., via landmark-based or DNS-based measurement) [2]. The other is to reduce routing path length (i.e., number of hops) by increasing dimension of virtual coordinate space or by increasing reality (i.e., coverage) of each zone [5]. Although the former makes routing among zones to be effective in terms of the latency stretch (i.e, the ratio of CAN routing delay to IP routing delay), the construction of overlay network requires additional efforts to reflect the physical network topology. However, in this case, improvement in routing is limited because of little knowledge about neighboring zones [6]. On the contrary, the latter can easily boost the performance of routing while keeping CAN's low maintenance cost. It also allows us to construct the overlay network without considering the physical network topology [5]. In this paper, we are mixing both approaches to improves routing performance. The proposed overlay network is reflecting network proximity by adopting network identifier. At the same time it propose an overlay network that consists of bigger size zones. Thus, the actual design needs to pursue both approaches with correct balance.

3 P2P Content Sharing Overlay Network Based on Network Identifier

3.1 Design Overview

The proposed overlay network is motivated by reducing the number of zones compared to CAN-based one. The proposed overlay network uses virtual 2-dimensional coordinate space with a distributed hash table to store $(key, value)$ pairs ranging from 0 to $2^{16}-1$ in term of x and y coordinate, respectively. Nodes which are willing to connect a overlay network are shown in Fig. 1(a). To reduce the number of zones in CAN-based overlay, we aggregate nodes of similar characteristics into a zone. As a feature to control aggregation, we introduce network identifier of each peer which is composed of 4 bytes. Note that in general nodes in a LAN (local area network) shares the same network identifier. For that reason, we use network identifier as a factor to aggregate nodes into a zone. Thus, nodes with the same network identifier are aggregated into a zone. To allocate peers and contents on the 2-dimensional space, contents and peers are assigned ContentIDs and NodeIDs respectively. Contents are hashed up into ContentIDs (values ranging from 0 to $2^{32} - 1$) and then mapped into 2-dimensional space with ContentIDs. To map NodeIDs and ContentIDs into x and y coordinate (each 2 bytes), they are separated with higher and lower 2 bytes. Each 2 bytes are corresponding to x and y coordinate on the virtual 2-dimensional coordinate

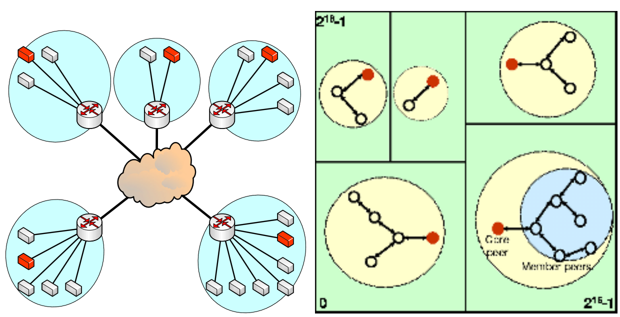

Fig. 1. (a) Network topology. (b) Proposed overlay network for P2P content sharing based on network identifier.

space. Peers are mapped into a zone, the virtual coordinate space, according to their NodeIDs (network identifiers) and own the zone. Although the proposed 2-dimensional space is originated from the concept of CAN, there are differences. Our approach uses fixed dimensional space, 2-dimension. But CAN uses n-dimensional space. We intentionally reduce the dimensional space to help peers' network identifiers to be mapped into space intuitively. To reduce the number of routing hops, mapping of peers in our approach provides aggregation of peers with the same network identifiers into a zone.
A zone owns its individual and distinct space within the overall space. A zone is composed of only one core peer and member peers because there are nodes with the same network identifier. If the number of peers is larger than one within a zone, peers store the same $(key, value)$ pairs on their own storages and should maintain a overlay network between a core and member(s). A core peer represents the his zone to neighboring zones and exchanges messages between neighboring zones like a peer in CAN with hiding members in his zone. If there are member nodes in a zone, a shared binary tree is constructed and used to exchange messages within the zone. Peers in proposed overlay network are required to keep hash table to store content index over the content sharing overlay network, peer's zone range, core peer's information (IP address and port number), and a routing table which includes neighboring zones and member peers in his zone. Fig. 1(b) shows the proposed overlay network which is constructed with Fig. 1(a). There are 5 zones and a core peer exists in each zones which including member peers.

3.2 Constructing Overlay Network

When joining the P2P content sharing overlay network, an joining peer contacts a RP (Rendezvous Point) peer for bootstrapping. After obtaining list of connected peers, the joining peer sends a *connect* request with its NodeID to a peer which is closest to its NodeID among the list. When receives a *connect* request message, a peer acts according to its role (a core or a member in a zone). In case of member peer, it just forwards the message to the core peer within the same zone because only the core peer manages a zone. In case of a core peer, it compares NodeID in *connect* request message with range of its zone. If the NodeID does not belongs to the zone, the message is forwarded to a neighboring zone until reaching to the destination node according to message routing in Section 3.4. When the message reaches a core peer with a zone covering its NodeID, the core peer splits his zone while each zone owns their own NodeIDs. Zone splitting will be presented in Section 3.3. Every peers in P2P content sharing overlay network must hold a routing table: a list of neighboring zones. This routing table is used to route every messages until reaching the destination peer and maintain the overlay network. we define neighbors as zones which overlap each other's x or y coordinates. Therefore, after splitting a zone into two, change of the zone is known to their neighbors and to the members if there is at least one member peer within the zone. As the result of splitting the zone, new core takes a part of zone and hash table from old zone.

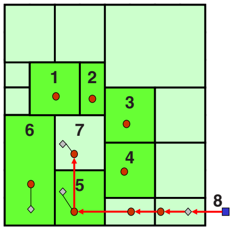

Fig. 2. (a) Connect request routing when peer 8 joins to zone 7 as zone 8. (b) After splitting zone 7 into zone 7 and 8.

3.3 Zone Management

Now, we describe the zone management which is used when new peers join in order to preserve the overlay network. When a joining peer's NodeID belongs to a zone, zone split is performed by the core peer in the zone. The zone is split in a direction of x or y axis according to the following.
To select the direction of split, a core peer calculates the absolute differences of x and y coordinate between their NodeIDs and then compares two absolute difference values. The zone is split along the axis of larger absolute difference. Border between two zones is chosen by mean value between two values of x or y coordinate according to direction of split zone. It is required that range of each zone should cover network identifiers of a core peer and member peers. After splitting a zone, neighbors of two zones are rearranged in order to maintain the P2P content sharing overlay network. For an example, neighbors of zone 7 in Fig. 2(a) are zone {1,2,3,4,5,6}. After splitting the zone 7 in Fig. 2(b), neighbors of zone 7 are zone {1,5,6,7,8} and neighbors of zone 8 are zone {1,2,3,4,5,7}. The change from zone splitting should be informed to each zone's neighbors and then makes neighbors modify their own neighboring zones. Also, any change of a zone should be notified to member peers within the zone. Fig. 2(b) illustrates the notification of changes to its own member peers and neighboring zones. A joining peer's NodeID is the same with core peers's NodeID and the joining peer is added to as a member peer in the zone. A shared binary tree is constructed and maintained among member peers within a zone. This tree is used to deliver messages between member peers within a zone without any message duplication. Fig. 3 shows the shared binary tree between member peers and result after peer 6 joining and routing table of member peers within the same zone. This tree is constructed by taking advantage of order of member peers' IP addresses and stored in each peers. Update messages between members in a zone are passed along thick arrow lines in Fig. 3 when a peer joins and leaves.

3.4 Message Routing

Message routing is used to locate destination when inserting contents into and retrieving contents from P2P content sharing overlay network based on the

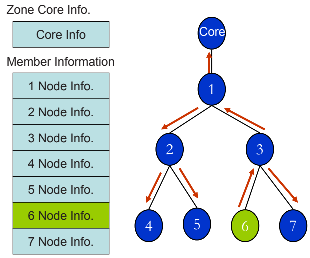

Fig. 3. Shared binary tree between member peers within a zone.

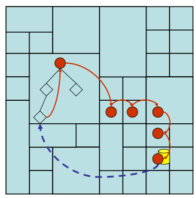

Fig. 4. Message routing in the proposed overlay network.

DHT. Our Approach defines two functions: $insert(key, value)$ and $value = retrieve(key)$ where keys can be name of content and $value$ is an address of content owner. $Insert$ function provides means of distributing information of contents and $retrieve$ function means of getting information of actual content owner. Both $insert$ and $retrieve$ function use hash function with a key to get the (x, y) pair coordinate on the space. With this (x, y) pair coordinate, a message (i.e, insert and retrieve) is routed to the destination. Message routing in our approach is similar to that of CAN. Fig. 4 illustrates message routing when retrieving contents. To get a certain content, a node uses the $retrieve$ function with key. In $retrieve$ function, hash function is performed to get x and y coordinate. Message with this (x, y) coordinate is sent to a core peer directly. If the core peer does not manage a zone containing (x, y), it just forwards to a neighbor zone which is closer to x. After reaching x, core peers select a neighbor zone whose range contains x and closer to y as next routing hop. When reaching the destination of $retrieve$ message, a core peer send $value$ to the requesting peer.

4 Experiment and Results

We use GT-ITM to create network topology with 10,000 nodes and experiment with randomly selected nodes among them. Simulator which is based on myns [7] is implemented to measure performance of message routing and verify our approach. To simplify our experiment, we use 2 bytes IP addresses so that x and y coordinate is ranged from 0 to $2^8 - 1$. Average number of neighboring zones in each zone are shown in Fig. 5. Each node maintains a routing table for their

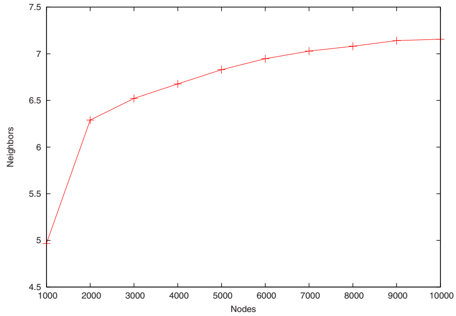

Fig. 5. Average number of neighbors.

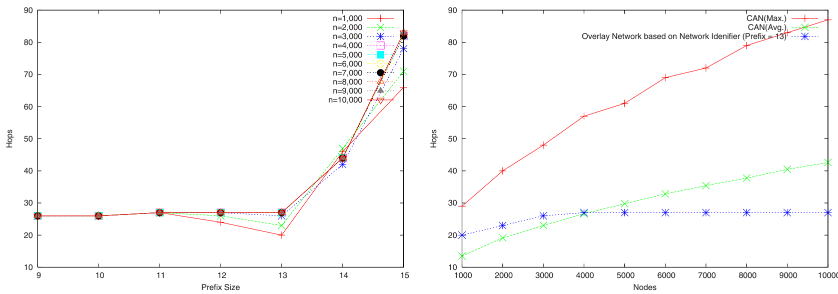

Fig. 6. (a)The number of message routing hops when changing length of network identifiers. (b) Comparison of the network identifier based overlay network with CAN.

neighbor zones' information such as ranges of neighbor zones, IP addresses and port numbers of neighbor core nodes. The result shows average neighboring zones are about 7 even though the number of nodes increases. Therefore, it can be realized that our proposal P2P network is scalable.

Fig. 6(a) shows the effect of grouping as decreasing the length of network identifiers. The number of prefix means IP masks in Classless Inter-Domain Routing. Therefore, decreasing the number of prefix means that the number of nodes having the same identifier increase and more nodes are grouped into a zone. When the number of prefix is 15 in Fig. 6(a), the number of message routing hops are more than 67 regardless of the number of participating nodes. This means that most of nodes occupy their own zone and there are many zones involving in message routing. The result shows that the number of message routing hops are declining to 28 hops as the number of prefix decreases. When the number of prefix is less than 13, there is no gain. It shows the number of prefix, 13, is enough for grouping nodes. Therefore, we prove that grouping nodes with network identifiers help improving the performance of P2P content sharing overlay network. Fig. 6(b) shows the performance of network identifier based overlay network with 13 prefix is better than CAN as the number of nodes increases.

5 Conclusion

In this paper, we showed that grouping by network identifiers for P2P content sharing overlay network based on DHT improves the performance of message routing. Also, our P2P content sharing overlay network is scalable even though the number of nodes increases by average number of neighbors. Nodes on the same local network become a group so that physical network topology was reflected on our P2P network. In the future, we have to solve limitation of grouping by network identifiers because reducing the length of network identifier is not easy.

Acknowledgement. This work was supported in part by MIC through RBRC at GIST and in part by MOE through BK21.

References

1. S. Androutsellis-Theotokis and D. Spinellis, "A survey of peer-to-peer file sharing technologies," *Athens Univ. of Economics and Business*, 2002.
2. S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker, "A scalable content-addressable network," *in Proc. ACM SIGCOMM*, 2001.
3. V. N. Padmanabhan, H. J. Wang, and P. A. Chou, "Resilient peer-to-peer streaming," *in Proc. IEEE ICNP 2003*, Nov. 2003.
4. M. Hefeeda, A. Habib, B. Botev, D. Xu, and B. Bhargava, "Promise: peer-to-peer media streaming using Collectcast," *in Proc. ACM Multimedia 2003*, Nov. 2003.
5. Z. Xu and Z. Zhang, "Building low-maintenance expressways for P2P systems", *Hewlett-Packard Labs: Palo Alto. HPL-2002-41*, 2002.
6. Z. Xu, C. Tang, and Z. Zhang, "Building topology-aware overlays using global soft-state", *in Proc. The 23rd International Conference on Distributed Computing Systems*, May 2003.
7. S. Banerjee, myns, `http://www.cs.umd.edu/~suman`.