Efficient Query Induction for Content Searching in Unstructured Peer-to-Peer Network

Kenta Asano Hiroki Watanabe Shinji Sugawara
Faculty of Eng., Chiba Institute of Technology
Narashino, 275–0016, Japan

***Abstract*—When considering content sharing in unstructured Peer-to-Peer network, flooding is usually carried out for contents searching. However, due to the increase of the number of queries, the network is congested, and the improvement of this point has been discussed. In the former research, a method which records the routes of the past content delivery and makes the new queries for the same content follow the trail was proposed. In this research, the conventional method is improved and proposed as a new method which controls query inducing directions according to the number of hops to the destination peers. By using this improved method, network cost during contents' searching and transferring is expected to be reduced compared with the conventional method. The effectiveness of the proposed method is evaluated by computer simulations.**

I. Introduction

Generally, unstructured Peer-to-Peer (P2P) based systems are more superior in fault tolerance than hybrid P2P based systems, and can be operated more freely without restriction of network structure than structured P2P based systems. In this paper, we propose a content searching method for efficient content sharing, assuming use of unstructured P2P which can be operated stably even when the number of users increases or decreases and peers' join and leave occur.

II. Conventional Method in Previous Research

In conventional content searching in unstructured P2P content sharing systems, flooding is widely used[1]. However, if this method is simply applied, there is a problem that the network load required for the content searching increases. Therefore, in our previous research, we proposed a method to induce queries to the requested content holding peer by using the past contents' movement history, and succeeded in suppressing the increase of the query without lowering the search success rate[2]. The outline of this method is as follows.

TTL (Time To Live) value [1] is set for each query to be sent. In addition, each peer holds TR (Transmission Record) that is the record of contents passed through the peer in the past, i.e., when and which content is transmitted to which adjacent peer.

When a content request occurs, the query is sent to all adjacent peers, and usual flooding is executed as long as there is no TR for the content in the peers receiving the query. If there is a TR in the query receiving peer, the query is transmitted only in the direction (downstream direction) in which the content was transmitted in the past. If the content exists in the most downstream peer to which the TR has been traced, the content is transferred in reverse order of the route of the query, and the query disappears. Otherwise, TR is traced in reverse and the query is sent to the most upstream peer. If the content exists in the most upstream peer, the content is transferred in the reverse order of the route of the query, and the direction of TR is inverted and the query disappears. If there is no content, the involved TRs are all deleted and the query disappears. Since queries whose TTL has become zero also disappear, the search fails unless the content reaches the requesting peer after a certain period of time.

III. Proposed Method

In the conventional method mentioned in section II, a query is induced always downstream by TRs at first. Then, even when the most upstream peer having the requested content exists upstream one hop from the requesting peer, the query is induced to the downstream peer direction, and the search path tends to be long. This causes increments of searching time, searching history (TR), and the length of search paths, and growth of network load for content transmission.

In the proposed method, a function to send a query selecting a direction with fewer hops, either transmission to the most upstream peer or that to the most downstream one, is newly added to the conventional method. Outline of the system's procedure executed when a content is requested by a peer in proposed method is shown below.

Procedure of Content Requesting:

1) Create a query which will be sent for the content searching at the content requesting peer.
2) Check TTL value of the query. If $TTL > 0$, go to 3), otherwise (i.e., if $TTL = 0$), The query disappears.
3) The peer is sent by flooding. If a peer which received one of the queries possesses a requested content, the peer executes the *procedure of content discovery* and the query disappears. If the peer possesses a TR of requested content, go to 4). If not, go back to 2).
4) Refer to the TR updated the most lately, and if the content possessing histories of both the most upstream and the most downstream peers are not expired, go to 5). Otherwise, go to 6).
5) Count the number of hops to the both peers. If the most upstream peer is near to the current peer, go to 7). Otherwise, go to 8).

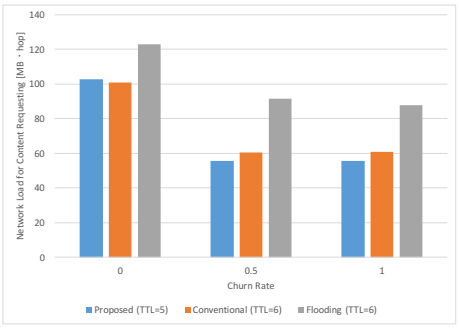

Fig. 1. Relationship between Churn Rate and Network Load for Content Requesting.

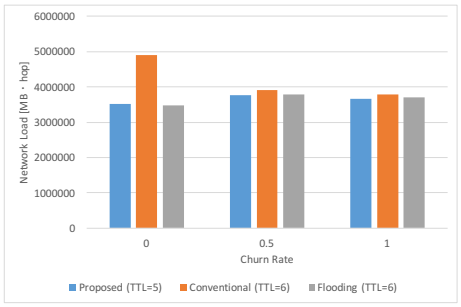

Fig. 2. Relationship between Churn Rate and Total Network Load.

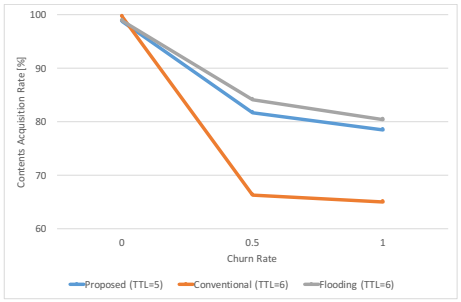

Fig. 3. Relationship between Churn Rate and Contents Acquisition Rate.

6) If only the history of the content in the most upstream peer is not expired, go to 7). If only that in the most downstream peer is not expired, go to 8). If both of them are expired, delete the TRs and the query.
7) Induce the query upstream. If a peer which received the query has the content, the peer executes the *procedure of content discovery* and the query disappears. In the case that upstream link of TRs to the most upstream peer is interrupted, or there is no requested content in the most upstream peer, invalidate the TRs and go to 8). However, in the case of already failing in both upstream and downstream inductions, delete the query.
8) Induce the query downstream. If a peer which received the query has the content, the peer executes the *procedure of content discovery* and the query disappears. In the case that downstream link of TRs to the most downstream peer is interrupted, or there is no requested content in the most downstream peer, invalidate the TRs and go back to 7). However, in the case of already failing in both upstream and downstream inductions, delete the query.

In the *procedure of content discovery*, after confirming whether the content can be sent to the requesting peer, it actually is transmitted. When requested content is found in multiple peers, permission of content transmission is given only to one peer. For the peers which did not obtain the permission, this procedure ends. At the time of transferring the content, TR is assigned to the peers on the transmission route. Finally, a replica of the requested content is placed on the requesting peer and counting of the number of times of its reference is started.

IV. Evaluation

In this paper, we compared the network loads for content requesting (Fig. 1) and total network loads including content transferring (Fig. 2), and content acquisition rates required among the proposed method, conventional method and flooding, while changing the peers' churn rate (Fig. 3). As a result, proposed method achieves almost the same content acquisition rate with flooding, which is much higher than the conventional method. Network load for content transferring is well restricted by the proposed method to almost the same extent with the case of using conventional method. Flooding costs much more. Network load for content transferring is also well suppressed by proposed method to almost the same extent with the case of using the other two methods.

Therefore, by appropriately setting the TTL value, the proposed method shows almost the same achievement compared to the dominant outcomes of the two other methods in all cases of content acquisition rate, and network costs for content requesting and transferring. And it can be said that the proposed method is sufficiently effective.

V. Conclusion

In this research, we propose a method to minimize network load for content requesting and retrieving while maintaining content acquisition rate by controlling the query sending direction in unstructured P2P network. The effectiveness of the proposal is shown by computer simulations.

Acknowledgment

This work was partially supported by JSPS KAKENHI Grant Number JP17K00134.

References

[1] I. Stoica, R. Morris, D. Karger, M. Frans Kaashoek, H. Balakrishnan, "Chord: A Scalable Peer-to-Peer Lookup Service for Internet Applications," ACM SIGCOMM'01, pp. 149-160, Aug. 2001.
[2] T. Tomimatsu, S. Sugawara, Y. Ishibashi, "Query guidance with transmission records for efficient content searching in unstructured peer-to-peer network," in Proc. IEEE 2013 International Communications Quality and Reliability Workshop (CQR2013), May 2013.