

A GROUPING ALGORITHM of HELPERS in PEER-TO-PEER VIDEO-ON-DEMAND SYSTEMS

Xia Li*, Hua Zou*, Xinchao Zhao**, Fangchun Yang*

**State Key laboratory of Networking and Switching Technology, Beijing University of Posts & Telecommunications, Beijing, China*

***School of Sciences, Beijing University of Posts & Telecommunications, Beijing, China*

lixiabupt@yeah.net, zouhua@bupt.edu.cn, xcmmrc@gmail.com, fcyang@bupt.edu.cn

Abstract— Peer-to-peer (P2P) content distribution effectively solves the problem of large-scale streaming media delivery. But the performance of a P2P system is primarily bottlenecked, because the upload bandwidth of the participating peers is limited. Using helpers in the P2P system utilizes upload bandwidth of those idle peers uninterested in the delivered content, which further improves the performance of P2P systems, and the feasibility and validity have been testified [1]. Aiming at the p2p video-on-demand system, this thesis designs a novel grouping algorithm of helpers. In order to make more rational and effective use of helpers, the algorithm divides the helpers into different groups according to the number of each video's users, thus reducing the average view delay of whole system and improving the utilization rate of the helpers, reinforcing watch experience for users of the system. Analysis and simulation results corroborate the effectiveness of the proposed algorithm in the paper.

Keywords— P2P, streaming media, video-on-demand, helpers, grouping algorithm

I. INTRODUCTION

Nowadays, the users on the Internet have a rapid increasing demand over streaming media, and the number of worldwide active video-on-demand users is expected to add up to about 150 million in 2010 [2]. In the field of large-scale streaming media delivery, due to the limitation of computing power and network bandwidth, the centralized C/S model has the problems of single point of failure and performance bottleneck. Though IP multicast try to resolve this problem through multicast routers in the network layer, the lack of support from the majority of routers make it difficult to promote. Content delivery network (CDN) pushes the server to the network edge, enhances service quality through providing service from the nearby server, and balances the load through multiple servers, but costs high for purchasing and maintaining the server. It has been reported that YouTube is paying millions of dollars per month to internet service providers and content distribution networks [3]. P2P-

based data distribution model organizes peers through overlay network and quickly deliver data between peers by way of relay. Such mutual assistance between peers makes the overall system performance greatly improved. P2P network owns the advantages such as low-cost, self-organization, high extensibility and robustness, and so on, which is an effective approach solving large-scale data distribution problem.

The effectiveness of using the P2P approach for streaming media distribution has been proven by many implemented live streaming systems such as a university project Coolstreaming [4] and a commercial system PPLive [5]. P2P content distribution technology has also developed some applications in video-on-demand, such as PPStream VoD [6]. However, P2P network still gets bottleneck problem since a large number of nodes with limited upload bandwidth. With more requirements of the quality and quantity of media delivered in Internet, researchers need to find out more effective content distribution mechanism.

Wong introduced the idea of using helpers in P2P for the first time [7]. Being uninterested in the delivered content, a lot of idle nodes in real network are not able to join in the traditional P2P network and cooperate in the content distribution and delivery, which results in a waste of their bandwidth and resources. In Wong's research work, those idle nodes are named helpers, and used in P2P system. Helpers are added in to the neighbour nodes list of P2P system users through server's distribution or the initiative search of the users. In this way, they could assist users with the content distribution and delivery in P2P network, take full use of the idle resources and bandwidth, and effectively enhance the performance and capability of P2P system. Due to using the same user network mechanism in traditional P2P system and P2P system using helpers, user clients can easily join in the new system almost without any change. The smaller system migration costs also make the helpers of the P2P-based system a favorable prospect of utilization. Furthermore, we could enhance the capability of the system and upgrade user's watch experience by using helpers in P2P streaming media system, and designing a series

of specific helper's content delivery mechanism according to the characters of streaming media.

Aiming at the p2p video-on-demand system, we propose a grouping algorithm of helpers in this paper. According to different amount of users of each video, we divide all the helpers into different groups with different helpers' number and distribute them to the corresponding users of videos. The aim of this algorithm lies in distributing the resources of helpers to the users who need the resource most, thus enhancing the system's performance to the fullest extent. In the third part of this thesis, we analyze the helpers' distribution principle, reach a conclusion that the helpers should be allocated in priority to the users of the video with smallest number of users, and implement specific grouping algorithm as well. In the fourth part, we validate the effectiveness of the algorithm through several simulation experiments, and intuitively show the main advantages of the algorithm bringing to the system.

II. RELATED WORK

The notion of helper is first introduced to file download by Wong [7]. However, a potential problem of resource waste exists in the system, because a helper can download a very large portion of a file. In [8], Tribler, a social-based P2P system, is put forward. This system provides incentive for these helpers to help users of the system. [1] designs a BitTorrent-compliant file download system using helpers. In this paper, author analyses the performance of system under certain peer arrival/departure models. [9] regards some high-bandwidth, high-connection and controllable super nodes as helpers, which can present resistance against selfish peers and dynamically change the content for the economic aim. Wang introduces the mechanism of helpers into P2P streaming media system in [10]. The thesis shows that, in a P2P live multicast system, if introduced, helpers can allow peers to stream video at a rate surpassing the average peer upload bandwidth without increasing the server's burden by downloading a very small segment of the video in the form of erasure-coded parity packets. In [11], Wang and Zhang further expands the mechanism of helpers, and investigates a novel P2P video-on-demand architecture using helper to provide a scalable solution scheme. Efficiency of the system has also been verified through packet level simulations. In [12], Zhang proposes a decentralized P2P Video-on-Demand system. Users and helper "servelets" cooperate in a P2P manner to deliver the video stream in this system. Helpers are preloaded with only a small fraction of video data packets, and form into swarms each serving partial video content. However, none of the papers mentioned above considers the differences of efficiency when using helpers between different videos within whole VoD system.

III. DESIGN OF ALGORITHM

In this section we analyse and implement the grouping algorithm of helpers. The details are as follows.

A. Environment Description

The user network of a P2P video-on-demand system using helpers is a typical P2P streaming media network. In this network a server maintains a tracker to keep track of all the participating peers in the streaming session and to assist building the overlay network [10]. Besides, the server is a basic video content provider. When a new user joins the P2P network, it acquires a neighbour list of watching the same video from the tracker. Users who started in closer time stamps compared to the newly arrived user's own start time are favored in choosing the neighbour list to improve upload efficiency [13].

Corresponding to each of the different video, the helpers also form an unstructured overlay network. All of the helpers are required to register on the server. The tracker will real-timely keep track of the number and some available information of the helpers. Periodically, the server divides the helpers available into different groups and assigns them with users of different videos. When a new user joins the system, it can receive the assistance of a number of helpers, and these helpers can only be chosen from the assigned helpers group corresponding to the video the user wanting to watch.

B. Grouping Principle

For different on-demand video has different popularity, the number of users watching the video online at the same time varies greatly. As in the P2P system, the more users watching the same video, the better quality of service will be provided. So, it will be obviously not economical if more helpers are assigned to these hot videos. Therefore, a rational helpers grouping algorithm is necessary. To make the discussion in the thesis convenient, we suppose users watch their video from the very beginning, and will not leave the system before the end of the video. Also, all the peers in the system do not have much heterogeneity.

We assume the length of one video is t and its video-streaming rate is r . The Video download rate of certain user who is watching a video without the help of helpers is d ($d < r$). When get assistance from the helpers, the download rate increased will be h . Then, D , the time delay of before-and-after using helpers decreased, is :

$$D = \left(\frac{rt}{d} - t\right) - \left(\frac{rt}{d+h} - t\right) = rt\left(\frac{1}{d} - \frac{1}{d+h}\right) \quad (1)$$

Examine monotonicity of (1):

$$\frac{\partial D}{\partial d} = rt\left(\frac{1}{d+h} + \frac{1}{d}\right)\left(\frac{1}{d+h} - \frac{1}{d}\right) \quad (2)$$

$d, h > 0$, so $(2) < 0$. D monotonously decreases with d . A conclusion can be drawn from the deviation above, that is, when h is equivalent, video users with less download rate will get more enhancement of performance. Therefore, helpers should be assigned to the user group with the least download rate.

C. Practical Implementation

In a large-scale video-on-demand system, it is impractical to get real-time download rate of each user. While in a P2P video-on-demand system with less heterogeneity, the larger the number of users watching certain video is, the higher average download rate of the users should be when the total amount of peers controlled in a certain scale. In a P2P video-on-demand system, for the corresponding user groups of two different videos, if the numbers of each user group are p and q respectively, and when a helper is added, the total download bandwidth it can provide is k , then, theoretically speaking, the average download bandwidth that each user can get from the helper is k/p and k/q respectively. If $p > q$, then $k/p < k/q$. Therefore, the users of the video with comparatively smaller number of users can get more average download bandwidth from the helper. With the combination of (2), we can conclude that helpers should be assigned to the users group the video which has the least number of users, thus improving the system performance to the fullest extent.

Because the tracker can keep track of the number and some available information of the helpers in real time, video-on-demand system can easily assign the helpers to different user groups based on the number of helpers. After the video user group with the least number of users get some helpers, the total number of all peers of the group including the number of users and helpers corresponding to their video may be not the least, so it's impossible to allocate all of the helpers available to the user group with the least users in the beginning. Therefore, the purpose of the grouping algorithm of helpers is to find out a grouping array H , distribute the helpers in H to different videos to reduce the watch delay of the whole video-on-demand system to the minimum extent, thus increasing the smoothness for users watching videos.

We suppose that the total number of users in system is u , the number of videos that has more than 0 users is n , and the numbers of users corresponding to the videos make up a array: $U_i (i = 1 \dots n)$, $\sum_{i=1}^n U_i = u$, and $U_a \leq U_b (a < b)$. In the system, the number of helpers is m . So, the helpers' numbers need to be divided into array: $H_i (i = 1 \dots n)$, according to the users' number of different videos, which is used to distribute to the user groups relevant accordingly, thus maximizing the reduction of the total watching delay of the whole video-on-demand system, though making $\sum_{i=1}^n (H_i + U_i - \frac{u+m}{n})^2$ the least. In order to meet the needs above, In order to meet the needs above, as shown in Figure 1, we divide server into two logical objects, server and tracker, so that the figure is more legible to illustrate. All of the helpers are distributed to the user groups in accordance with different videos, and the video user group with the least number of users get helpers in priority. When the peers including users and helpers of the group go beyond other groups, the helpers will be distributed to other groups with small number of peers. Correspondingly, all of the helpers will be allocated in the end.

According to the analysis above, we adopt the grouping algorithm below to get the helpers' grouping array:

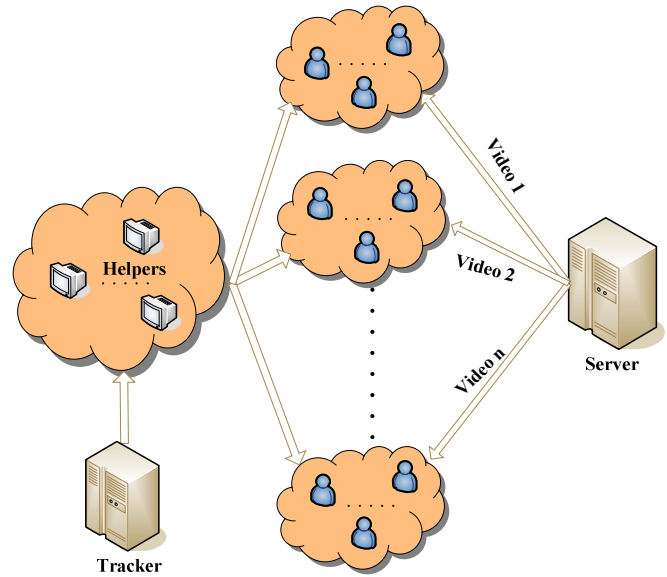


Figure 1. Illustration of grouping of helpers

```
int i = 1;
```

```
// find a video possessing the largest number of users among
the videos which can be distributed to helpers.
```

```
for (i; n; i++) {
```

```
    m +=  $\sum_{k=1}^i U_k$ 
    if (  $\frac{m}{i} < U_{i+1}$  )
```

```
        break;
```

```
}
```

```
// calculate the average amount of peers corresponding to the
videos having received the helpers allocated.
```

```
    tem =  $\frac{m + \sum_{k=1}^i U_k}{i}$ ;
```

```
    ave = floor(tem);
```

```
// distribute the helpers to users of each video.
```

```
for (j = 1; n; j++) {
```

```
    if (j ≤ i) {
```

```
         $H_j = \text{ave} - U_j$ ;
```

```
    }
```

```
    else {
```

```
         $H_j = 0$ ;
```

```
    }
```

```
}
```

```
// distribute the helpers left over to users of each video.
```

```
If (ave != tem) {
```

```
    for (f = 1; m - n * ave; f++) {
```

```
         $H_f++$ ;
```

}
}

Through the above algorithm, we guarantee the helpers should be assigned in priority to the video with the least number of users. In video-on-demand system, the server will loop again to execute the algorithm at regular intervals to adapt to changes in video popularity.

IV. SIMULATION EXPERIMENTS

In this section, we will corroborate the effectiveness of the proposed grouping algorithm through some simulation experiments.

A. Simulation Environment

The simulation experiments base on PeerSim [14] in the environment of less heterogeneous upload bandwidth and dynamic propagation delays. The total number of videos is 5, each video length is 60 seconds, and media streaming playing rate is 512kbps. The user arrival rate is 0.5 to 4 peers/second and buffer length equal to 10 seconds. The users will not drop out of the video-on-demand system until playing out all the video. The number of video users shows the pattern of an arithmetic series. All the peers including users and helpers have unlimited download bandwidth and close upload bandwidth which is 256kbps or so. The time for each simulation experiment lasts for 1200 seconds. The size of chunk is 10000bit, i.e. the approximate size (1250Byte) of a maximum IP packet payload. Propagation delays are randomly assigned from the delay matrix (2500*2500) in the Internet measurement [15] and their values will change once every two seconds.

B. Simulation Result

In the simulation experiments, we first compare the differences in average view delay rate of the system adopting grouping algorithm of helpers and that without grouping algorithm of helpers. Here, we define average view delay rate of a video as D_i , i.e. mean of average of view delay per unit time by all of the users watching the video. Therefore, the average view delay rate of whole system is $\sum_{i=1}^n D_i / n$.

Obviously, the smaller this index is, the better the system view continuity is.

From Figure 2 we can see, when the total number of users differs within the system, the average view delay rate of the system adopting grouping algorithm of helpers distinctly decreases. In the first half part of Figure 2, because the user number is relative small, helpers and server are vital in affecting the view continuity of system. Thus, the average view delay rate of system enlarges when the user number increases. While in the second half of Figure 2, the user number has become a major factor affecting the view continuity of system. When the number of users continually increases, the average view delay rate of system declines accordingly. Along with the user group scale enhancing step by step, the ratio between the number of helpers and that of

users becomes less and less. Therefore, the performance effects brought by grouping helpers become smaller. This trend is also shown in Figure 2.

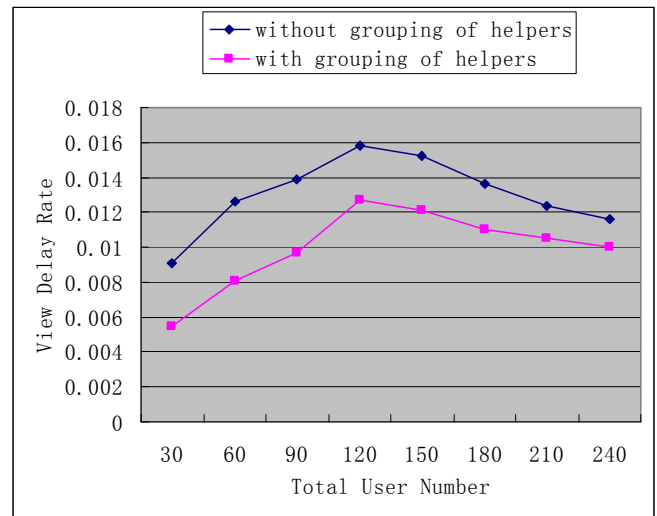


Figure 2. Comparison of average view delay continuity

Figure 3 shows the result of another simulation experiment. We compared the differences of average view interruption number of system. We define average view interruption number of a video as N_i that is, the average of view interruption numbers by all of the users watching the video until they finish the video. Then, the average view interruption

number of the system is $\sum_{i=1}^n N_i / n$. The index reflects the degree of view continuity of system from another aspect. The experiment's result shown in Figure 3 indicates that the enhancement of system's performance due to grouping of helpers is similar to the result shown in Figure 2. We can infer from Figure 3 that the average view interruption number of system decreases by 27% by adopting grouping of helpers.

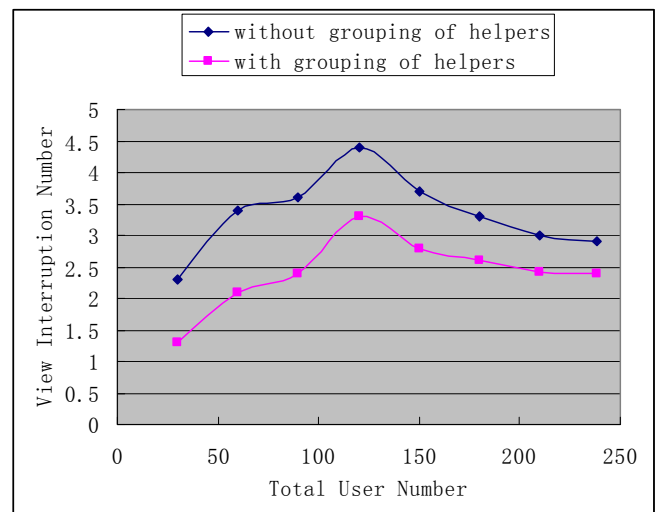


Figure 3. Comparison of average view interruption number

V. CONCLUSIONS

In this paper we propose a grouping algorithm of helpers to the P2P video-on-demand system using helpers. The algorithm effectively reduces the average view delay and the average view interruption number of whole system, and reinforces watch smoothness for users of the system. The distribution bias of helpers to non-hot videos in this algorithm, to some extent, solves the problem that non-hot videos view experience is rather poor in current video-on-demand systems. Thus the algorithm is of good practicability as well. In the future work, we will consider the heterogeneity of the network, and further optimize the grouping algorithm of helpers so as to get closer to the actual network environment, and further enhance system performance.

ACKNOWLEDGMENT

This work was supported by National Key Basic Research Program of China (973 Program) (2009CB320406), the National High Technology Research and Development Program of China ("863" Program) (2008AA01A317), the Foundation for Innovative Research Groups of the National Natural Science Foundation of China (Grant No. 60821001), Beijing Municipal Commission of Education to build the project special, National Key Basic Research Program of China (973 Program) 2009CB320504, Chinese Universities Scientific Fund (BUPT2009RC0701). And I'm extremely grateful to my teachers and wife.

REFERENCES

- [1] J. Wang, C. Yeo, V. Prabhakaran, and K. Ramchandran, "On the Role of Helpers in Peer-to-Peer File Download Systems: Design, Analysis and Simulation," in Proc. of IPTPS, 2007.
- [2] H. Zhang, J. Wang, M. Chen, and K. Ramchandran, "Scaling Peer-to-Peer Video-on-Demand Systems Using Helpers," in IEEE International Conference on Image Processing, 2009.
- [3] C. Huang, J. Li, and K.W. Ross, "Can Internet Video-on-Demand be Profitable?" in Proceedings of the 2007 conference on Applications, technologies, architectures, and protocols for computer communications, 2007, pp. 133–144.
- [4] X. Zhang, J. Liu, B. Li, and T. S. P. Yum. "Cool streaming/donet: A data-driven overlay network for efficient live media streaming", In Proceedings of IEEE INFOCOM'05, 2005.
- [5] PPLive. [Online]. Available: <http://www.pplive.com/>
- [6] PPStream. [Online]. Available: <http://www.ppstream.com/>
- [7] Wong J. "Enhancing collaborative content delivery with helpers", Master's Thesis, The University of British Columbia, 2004.
- [8] J. Pouwelse, P. Garbacki, J. Wang, and et al., "TRIBLER: A Social-Based Peer-to-Peer System," *Concurrency and Computation*, vol. 20, no. 2, pp. 127, 2008.
- [9] K. Xu, Y. H. Yang, Tao Chen. "Improving BitTorrent Network's Performance via Deploying Helpers", in EUC 2008 IEEE/IFIP International Conference, 2008, Volume 2, pp. 507 – 512.
- [10] J. Wang and K. Ramchandran, "Enhancing Peer-to-Peer Live Multicast Quality Using Helpers," in IEEE International Conference on Image Processing, 2008, pp. 2300–2303.
- [11] H. Zhang, J. Wang, M. Chen, and K. Ramchandran, "Scaling peer-to-peer video-on-demand systems using helpers," in IEEE International Conference on Image Processing, 2009.
- [12] H. Zhang, and K. Ramchandran, "A reliable decentralized peer-to-peer video-on-demand system using helpers," in Picture Coding Symposium, 2009.
- [13] J. Wang, C. Huang, and J. Li, "On ISP-friendly Rate Allocation for Peer-Assisted VoD," in MM '08: Proceeding of the 16th ACM international conference on Multimedia, 2008, pp. 279–288.
- [14] Peersim. [Online]. Available: <http://peersim.sourceforge.net>
- [15] S. Saroiu, P. Gummadi, S. Gribble. "A measurement study of peer-to-peer file sharing systems", in Multimedia Computing and Networking, 2002, pp. 156-170.