

Streaming Cost Minimization-based Optimal Peer Selection Method

Qing-jun Yu^{1,2}, Bao-chu Yu^{1,3}

1 Dalian University of Technology, Dalian, P. R. China

2 Dalian Nuesoft Institute of Information, Dalian, P. R. China

3 Dalian Fisheries College, Dalian, P. R. China

1. e-mail yuqingjun@neusoft.edu.cn, 2. e-mail baochuyu@dlut.student.edu.cn

Abstract: In this paper, we first proposed a modified incentive mechanism to encourage the cooperative behavior among peers. Then, we defined a simple streaming cost function by considering the heterogeneity of the Internet, the limitation of end-to-end delay, the unpredictable behavior of peers and the peers' social properties, and proposed an optimal peer selection algorithm by minimizing the total streaming cost subject to the download rate constraint. Finally, we showed that the proposed scheme can significantly improve the performance of a P2P streaming system.

Keywords: peer-to-peer; peer selection; incentive mechanism; streaming cost

1 Introduction

Peer-to-Peer (P2P) has become increasingly popular during the past decades. Especially, as one of the most perspective P2P applications, P2P streaming has aroused much interest both in research fields and in industries in recent years [1]-[7].

P2P streaming system allows users to playback a media file while downloading and uploading it. In a P2P streaming system, a subset of peers own some pieces of a certain media file, and deliver the specified pieces to those requesting peers. Meanwhile, those requesting peers playback this media file while downloading the requesting pieces and providing the uploading service to other peers. Obviously, the performance of P2P streaming application depends to a great extent on the efficiency of its peer selection algorithm.

Most existing P2P streaming systems usually adopt various heuristics peer selection methods. In GnuStream [3], peer selection algorithm was built on the basis of bandwidth available. In CoolStreaming [4], a gossiping protocol was employed to devise a smart peer selection algorithm and a low-overhead scheduling algorithm. In PROMISE [5], a topology-aware peer selection technique was utilized to achieve significant gain in streaming quality. Although the above mentioned peer selection algo-

rithms can work efficiently, all of them don't take peers' social properties such as reputation and trust into account, and fall short of achieving global optimality.

In addition, there also exist some theoretical methods for formulating the peer selection problem. For example, to select an optimal peer from a set of peers in P2P streaming, Adler et al. [6] proposed two linear programming models that aim at minimizing the total streaming cost subject to continuous playback, but they did not consider an incentive mechanism and the problem of content assignment.

To improve the efficiency of peer selection in a P2P streaming system, we will first introduce a modified incentive mechanism and defined a simple streaming cost by taking into account the heterogeneity of the Internet, the limitation of end-to-end delay, the unpredictable behavior of peers and the peers' social properties in this paper. Then, we propose an optimal peer selection algorithm for P2P streaming by minimizing the total streaming cost subject to the download rate constraint.

In the next section, we present the proposed streaming cost minimization-based optimal peer selection algorithm. We also discuss its properties and performance afterwards. Finally, we conclude the paper with a summary and give our future work

2 Incentive Mechanism

Various incentive mechanisms have been proposed to encourage the cooperative behavior among peers [7], [8],

Funded by Key Research Project of Liaoning Province Bureau of Science and Technology (辽宁省科学技术计划重大项目): 2008217004, China's Post-Doctoral Science Fund (中国博士后基金) 200704111071.

[9], [10], which are based on payment, reputation, score, punishment, or service differentiation. In a payment-based system [9], peers can earn rewards that can be used for future download if they provide uploading service to other users. In a reputation-based system [10], peers earn reputation by sharing resources and the reputation determines download quality. In a service differentiation-based system, contributors are rewarded with flexibility and choice while free-riders are given limited options in peer selection. However, all of them don't consider the global reputation of each peer and the local trust between the peers simultaneously, and can't efficiently settle the heterogeneity problem of peers. To meet these requirements, we present a modified incentive mechanism hereinafter.

First of all, we assume that each peer possesses a global reputation and maintains a local trust list, also known as buddy list. The global reputation is recognized by all users in this P2P stream system and could be saved in a central server. The local buddy list is used to manage some peers locally and record their trust. Both the global reputation and the local trust of a supplying peer are used to evaluate its uploading quality and reliability.

For a given peer P_0 , its global reputation could be determined according to its upload rate, total upload amount, or behavior during the process of P2P streaming session.

For a given buddy P_i of P_0 , its local trust could be determined according to its average upload rate, upload amount, or behavior relative to P_0 , which memorizes the past transactions between these two peers.

Obviously, a peer can gain its global reputation and local trust through providing the uploading service to those requesting peers. Certainly, both of them could also be obtained by other methods, even can be bought with money.

Upon that, we propose an incentive mechanism, which described as follows,

(1) The more resource a peer owns, the more and the higher quality service it will enjoy.

(2) A peer can earn its global reputation by uploading resources and behaving fairly. The more resources a peer uploads to other peers totally and the better behavior a peer shows, the higher reputation it earns. Moreover, a peer with a high reputation has a higher probability to enjoy better service.

(3) A peer with a high global reputation has a high probability to enjoy better service, even if its network

environment is bad and its computer performance is limited.

(4) A peer can earn its local trust by providing uploading service and truly available bandwidth to a certain peer. The more resources and the larger available bandwidth a peer provide to another peer, and the better it behaves to that peer, the more trusts it earns from that peer.

(5) Likewise, a peer with high trust related to a certain peer has a high probability to enjoy better service from that peer.

3 Streaming Cost

During the process of P2P streaming session, we think a peer is consuming its streaming cost while enjoying the downloading service from those requested peers. According to the proposed incentive mechanism in the previous section, it is natural to assume that a requesting peer can easily obtain what it wants at less cost from a supplying peer with high reputation, great trust, many expected resources and high available out-bound bandwidth. Therefore, we can utilize these four factors to measure the streaming cost.

For a given requesting peer P_0 , let PS denote its expected piece set, P_i denote a supplying peer in its buddy list, and let a_i , b_{0i} , PS_i , r_{0i} denote the global reputation of P_i , the local trust of P_i relative to P_0 , the piece set owned by P_i and the available bandwidth assigned to P_0 by P_i , respectively. Then, the total streaming cost can be expressed as a weighted sum function, as follows,

$$f(P_i) = \frac{a_g * t_g * b_g}{a_i * b_{0i} * r_{0i} * n} f_1(PS, PS_i) \quad (1)$$

where n denotes the number of the expected pieces in PS , the function $f_1(.)$ is used to count the number of the expected pieces that PS_i owns, and a_g , t_g , b_g denote the threshold values for the global reputation, the local trust, and the available bandwidth respectively, which are greater than zero and could be chosen according to experience. We also can split PS into priority set and general set, as shown in Figure 1, and compute the streaming cost more accurately.

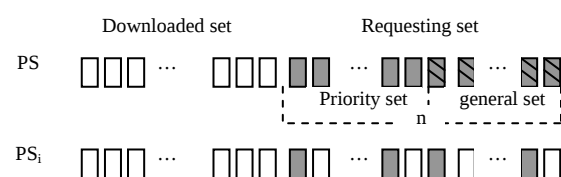


Figure 1. the illustration of PS and PS_i

4 The Proposed Optimal Peer Selection Algorithm

Consider a requesting peer P_0 with in-bound bandwidth R_d that want to playback-while-downloading a certain media file. Let R denote the playback rate of the media data, $\{P_1, P_2, \dots, P_M\}$ denote the peer list owned by P_0 . To maintain a continuous playback the requesting peer must ensure that the download rate is larger than or equal to R . Thus, our optimal peer selection problem is to determine a subset of peers $\{P_1', P_2', \dots, P_N'\}$ ($N < M$) from the peer list that minimize the total streaming cost subject to the download rate constraint, which can be described as the following optimization problem:

$$\begin{aligned} \min \quad & \sum_{i=1}^N f(P_i') \\ \text{s.t.} \quad & R \leq \sum_{i=1}^N r_{0i}' \leq R_d \end{aligned} \quad (2)$$

where r_{0i}' is the upload rate that peer P_i' assigns to P_0 , and $f(\cdot)$ is the streaming cost function defined as (1).

To solve this optimal problem, we give a pseudo code of the proposed algorithm, as follows

Input:

PlayRate: the playback rate;
ExpectedSet[]: the expected piece set;
Deadline[i]: the playback deadline of piece i;
PieceSize: the size of file piece;
PieceNum: the total number of pieces in the file;
PieceSet[k][]: the piece set owned by peer k;
PeerNum: the number of peers in the peer set;
PeerSet[]: the peer set;
Bandwidth[k]: the available bandwidth assigned by peer k;
Reputation[k]: the global reputation score of peer k;
Trust[k]: the local trust degree of peer k;
 α : a coefficient;

Algorithm:

for piece $i \in \text{ExpectedSet}$ **do**

 // determine which peers possess piece i.

$n \leftarrow 0$;

for j to PeerNumber **do**

if PieceSet[j][i] = 1 **then** // if peer j possesses piece i.

$n \leftarrow n + \text{PieceSet[j][i]}$;

 Temp[n] $\leftarrow$ j; // store peer j who possesses piece i.

end if;

end for j;

if $n = 0$ **then** // no peer possesses piece i.

 continue; // jump out the loop;

else if $n = 1$ **then** // only one peer possesses piece i.

$k \leftarrow \arg_r\{\text{PieceSet[r][i]} = 1\}$;

 // send a request message to peer k to ask for piece i;

 SupplyingPeer[i] $\leftarrow$ k;

else // there are more than one peer who possess piece i.

$p \leftarrow 0$;

 // determine which peer can get piece i in time;

for $m = 1$ to n **do**

$k \leftarrow \text{Temp}[m]$;

 // measure the download time of piece i for peer k;

 DownloadTime[m] $\leftarrow \alpha * \text{PieceSize} / \text{Bandwidth}[k]$;

if DownloadTime[m] $\leq$ Deadline[i] - CurrentTime

then

$p \leftarrow p + 1$;

 Temp1[p] $\leftarrow$ k; // store peer k temporarily.

end if;

end for m;

if $p = 0$ **then** // no peer can get piece i in time.

$q \leftarrow 0$;

 //find out which peer can obtain piece i at minimal

cost.

for $m = 1$ to n **do**

$k \leftarrow \text{Temp}[m]$;

 // compute the cost of peer k for obtaining piece i;

 cost[m] $\leftarrow$ f(i, Reputation[k], Trust[k], Band-

width[k]);

if cost[m] $\leq$ mincost **then**

 mincost $\leftarrow$ cost[m];

$q \leftarrow k$; // store peer k temporarily.

end if;

end for m;

 SupplyingPeer[i] $\leftarrow$ q;

else if $p = 1$ **then** // only one peer can get piece i in time.

$k \leftarrow \text{Temp1}[p]$;

 // send a request message to peer k to ask for piece i;

 SupplyingPeer[i] $\leftarrow$ k;

else // there are more than one peer can get piece i in

time.

$q \leftarrow 0$;

 mincost $\leftarrow$ 10000; // initialize minimal cost.

```

//find out which peer can obtain Piece i at minimal cost.
for m = 1 to p do
    k ← Temp1[m];
    // compute the cost of peer k for obtaining Piece i;
    cost[m] ← f(i, Reputation[k], Trust[k], Band-
width[k]);
    if cost[m] ≤ mincost then
        mincost ← cost[m];
        q ← k; // store peer k temporarily.
    end if;
end for m;
SupplyingPeer[i] ← q;
end if;
end if;
end for i;

```

Output:

SupplyingPeer[i]: the expected supplying peer for piece i.

5 Scheme Analysis and Evaluation

According to the above-mentioned peer selection scheme, we can conclude that it can not only significantly improve the efficiency of peer selection by solving an optimal problem, but also guarantee the playback continuity by satisfying a playback constraint. In addition, it can reward contributors with more flexibility and more choice by introducing a new incentive mechanism, resulting in high quality streaming sessions whilst free-riders are given limited options in peer selection.

In order to evaluate the proposed algorithm, we have developed a P2P streaming system demo, run and tested the proposed algorithm and several other common used mechanisms, including a random peer selection scheme [8], the end-to-end peer selection approach in GnuStream [3], and the heuristic algorithm in CoolStreaming [4].

During the test, we deploy 1 seeder and 4 peers, and use a large movie file in WMV format, whose playback rate is 423 Kbps. As far as playback continuity is concerned, the proposed method shows the better result than other three peer selection schemes according to many times test. Moreover, we also can conclude that the random peer selection scheme results in the worst streaming quality, especially for large scale media file, the end-to-end peer selection mechanism in GnuStream [3] behaves better than the random scheme, but less than the heuristic peer

selection approach in CoolStreaming [4] from the observed results. The reason is that GnuStream [3] limits the randomness of peer selection by using network bandwidth as a metric, and CoolStreaming [4] enhances the reliability of peer selection by considering both bandwidth and available time.

6 Conclusion

In this paper, we propose an optimal peer selection algorithm by minimizing the total streaming cost subject to a download rate constraint during a P2P streaming session. This scheme is not only able to meet the requirement of playback-while-downloading, but also is very suitable to encourage the cooperative behavior among peers in a P2P network. Our further work will focus on the improvement of the proposed algorithm and the study of other critical algorithms in P2P streaming application.

References

- [1] D. Xu, M. Hefeeda, S. Hambrusch, and B. Bhargava. On peer-to-peer media streaming. *IEEE International Conference on Distributed Computing Systems*, Vienna, Austria, 2002, P363-371.
- [2] D. A. Tran, K. A. Hua, and T. T. Do. A peer-to-peer architecture for media streaming. *IEEE J. Select. Areas in Comm.*, Jan. 2004, (22), P121-133.
- [3] Xinyan Zhang, Jiangchuan Liu, Bo Li, and Tak-Shing Peter Yum. CoolStreaming/DONet: A data-driven overlay network for efficient live media streaming. *Proceedings of IEEE INFOCOM'05*, Miami, FL, March, 2005, P2102-2011.
- [4] X. Jiang, Y. Dong, D. Xu, and B. Bhargava. Gnustream: a P2P media streaming prototype. *Proc. of IEEE Int. Conf. on Multimedia & Expo*, Washington, DC, USA, July 06-09, 2003, P325-328.
- [5] Hefeeda, M., Habib, A., Botev B., Xu D., Bhargava, B. PROMISE: Peer-to-Peer Media Streaming Using Collect-Cast. *Proceedings of the eleventh ACM international conference on Multimedia*, Washington, DC, USA, July 06-09, 2003, P325-328.
- [6] M. Adler, R. Kumar, K. Ross, D. Rubenstein, T. Suel, and D. D. Yao. Optimal peer selection for P2P downloading and streaming. *Proc. IEEE INFOCOM*, Miami, FL, Mar. 2005, P1538-1549.
- [7] Habib A., Chuang, J. Service differentiated peer selection: an incentive mechanism for peer-to-peer media streaming. *IEEE Transactions on Multimedia*, June 2006, 8(3), P610-621.
- [8] B. Cohen. Incentives build robustness in bittorrent. *Proceedings of the 1st Workshop on the Economics of Peer-to-Peer Systems*, Berkeley, CA, USA, June 2003, P116-121.
- [9] P. Golle, K. Leyton-Brown, and I. Mironov. Incentives for sharing in peer-to-peer networks. *Proceedings of the 3rd ACM conference on Electronic Commerce*, Tampa, Florida, USA, October 14-17, 2001, P264-267.
- [10] M. Gupta, P. Judge, and M. Ammar. A reputation system for peer-to-peer networks. In *Proceedings of ACM NOSSDAV*, Monterey, CA, USA, 2003, P144-152.