Channel-Aware Peer Selection in Multi-View Peer-to-Peer Multimedia Streaming

Miao Wang
University of Nebraska-Lincoln, mwangcse@gmail.com

Lisong Xu
University of Nebraska-Lincoln, xu@cse.unl.edu

Byrav Ramamurthy
University of Nebraska-Lincoln, bramamurthy2@unl.edu

Proceedings of 17th International Conference on Computer Communications and Networks, 2008. ICCCN '08.
Digital Object Identifier: 10.1109/ICCCN.2008.ECP.137
Publication Year: 2008 , Page(s): 1 - 6

Channel-Aware Peer Selection in Multi-View Peer-to-Peer Multimedia Streaming

Miao Wang, Lisong Xu, and Byrav Ramamurthy
Department of Computer Science and Engineering
University of Nebraska-Lincoln
Lincoln, NE 68588-0115 USA
email: {mwang, xu, byrav}@cse.unl.edu

***Abstract*—Motivated by the success of the Picture in Picture feature of the traditional TV, several commercial Peer-to-Peer MultiMedia Streaming (P2PMMS) applications now support the multi-view feature, with which a user can simultaneously watch multiple channels on its screen. This paper considers the peer selection problem in multi-view P2PMMS. This problem has been well studied in the traditional single-view P2PMMS; however, it becomes more complicated in multi-view P2PMMS, mainly due to the fact that a peer watching multiple channels joins multiple corresponding overlays. In this paper, we propose a novel peer selection algorithm, called Channel-Aware Peer Selection (CAPS), where a peer selects its neighboring peers based on the channel subscription of the system, in order to efficiently utilize the bandwidth of all peers in the system, especially those peers watching multiple channels. The results of a large-scale simulation with 10,000 peers and 4 channels shows that CAPS can significantly improve the system performance over the straightforward Random Peer Selection (RPS), which is widely used in single-view P2PMMS networks.**

I. Introduction

Recently, the Internet has witnessed a rapid development in Peer-to-Peer MultiMedia Streaming (P2PMMS) systems. There are many successful commercial P2PMMS applications, such as PPLive [8] and PPStream [9], which broadcast multimedia programs to a large number of users all over the world. For example, PPStream claims that it can support tens of thousands peers watching a single channel simultaneously [9].

The Picture-In-Picture (PIP), an important feature of the traditional TV, has been recently introduced to the P2PMMS applications. For example, the latest version of PPStream (version 2.0.9.980) supports PIP, which enables a user to watch a channel in a big window on the screen while at the same time watching another channel in a small window. Moreover, several popular P2PMMS applications including PPLive and PPStream allow a user to open multiple instances of their applications, so that a user can simultaneously watch multiple different channels on their screens. In this paper, we refer to a P2PMMS system that allows a user to simultaneously watch multiple channels as a *multi-view P2PMMS* system.

Even though some multi-view P2PMMS applications have been recently developed, some fundamental research problems in multi-view P2PMMS have not been studied yet. For example, the peer selection problem is important in P2PMMS, since it greatly determines the overlay construction and overall system performance. This problem has been well studied in single-view P2PMMS; however, it becomes more complicated in multi-view P2PMMS, mainly due to the fact that a peer watching multiple channels joins multiple corresponding overlays.

When a peer joins a single-view P2PMMS system that maintains only a single overlay, it first contacts the tracker server to get a list of usually randomly selected candidate neighbors. Next, the peer sends a request to each peer in the list and possibly some other peers using a certain mechanism. For example, CoolStreaming [15], PPLive [8], and PRIME [6] use some kind of gossip protocol. Finally, a required peer may accept or reject the request based on its constraints, especially its bandwidth limitations.

When a peer joins a multi-view P2PMMS system that maintains multiple overlays, one for each channel, it follows three steps as in the case of single-view P2PMMS. The only difference is that if a peer watches multiple channels, then it also joins multiple corresponding overlays, and it may have different neighbors in different overlays. We refer to a peer simultaneously watching multiple channels as a *multi-view peer*, and a peer watching only a single channel as a *single-view peer*. Note that a multi-view P2PMMS system consists of both single-view peers and also multi-view peers.

This paper studies the peer selection problem in multi-view P2PMMS. In other words, we answer the following question: for each peer joining single or multiple overlays, how many multi-view neighbors and how many single-view neighbors should it select? Multi-view peers may have greater impact on the overall system performance than single-view peers, since they have the ability to reallocate upload bandwidth to neighbors in different overlays to adjust the neighbor's streaming quality [14]. Moreover, multi-view peers might have more neighbors than single-view ones, due to their joining multiple overlays.

The main contributions of our work are as follows. First, we propose a novel peer selection algorithm for multi-view P2PMMS systems, called Channel-Aware Peer Selection (CAPS), where a peer selects its neighboring peers based on the channel subscription of the system, in order to efficiently utilize the bandwidth of all peers in the system, especially those peers watching multiple channels. Second,

The work reported in this paper is supported in part by UNL Layman Fund Award.

we conduct a large scale simulation experiment with 10,000 peers and 4 channels. Our simulation results show that CAPS can significantly improve the system performance over the straightforward Random Peer Selection (RPS), which is widely used in the single-view P2PMMS system.

The rest of the paper is organized as follows. Section II introduces the related work. Section III describes the peer selection problem in both single-view and multi-view P2PMMS. Section IV proposes the Channel-Aware Peer Selection algorithm for multi-view P2PMMS. Section V presents the simulation results. Finally, conclusions are given in Section VI.

II. Related work

Even though some multi-view P2PMMS applications have been recently developed, little research has been done to study even the fundamental problems such as peer selection. Therefore, we discuss the related work from two perspectives: (1) Related research on multi-view P2PMMS system; and (2) Related research on peer selection in single-view P2PMMS.

Liang *et al.* [4] propose a framework for future IPTV supporting the multi-view feature, and they discussed automatic screen layout, content-based channel recovery, and locality-aware overlay construction. Wu *et al.* [14] consider the bandwidth allocation among peers joining different overlays. Since the upload bandwidth is essentially important for downstream peers to achieve smooth playback, each peer requests as much bandwidth as possible from its upstream neighbors, which conflicts with other peers' requests. Wu *et al.* [14] model the strategy conflict problem using game theory and propose a distributed auction based bandwidth allocation algorithm. As claimed in their paper, the obtained topology is optimal for upload bandwidth sharing; however, the topology evolution is based on a random peer selection method, which either increases the convergence time or leads to inefficient bandwidth utilization due to improper peer selection. Our work mainly focuses on peer selection, which is a phase before the bandwidth allocation.

The peer selection problem in single-view P2PMMS has been well studied. These studies fall into two categories. First, the optimal peer selection category, which aims to minimize the content delivery cost according to some constraints. Adler *et al.* [1] formulated the problem using linear programming to minimize the content distribution cost. *rStream* [13] optimizes peer selection and solves the content reconciliation problem with Rateless Codes.

The second category corresponds to the location-aware peer selection, which focuses on the mismatch between overlay topology and physical topology. Xu *et al.* [11] resort to the topologically-aware overlay construction and peer selection by establishing some landmark servers among all the peers to facilitate peer distance estimation. Liu [5] improves Xu's work using location-aware overlay construction, which enhances a peer's search scope. One drawback of this method is that it may separate the physical network into several groups or clusters, which might not be resilient to peer dynamics.

Even though the peer selection problem has been well studied in single-view P2PMMS, it is still worth paying attention to, due to the fact that a multi-view peer may join multiple overlays. To the best of our knowledge, this paper is the first one studying the peer selection problem in multi-view P2PMMS.

III. Peer Selection Problem in P2PMMS

This section describes the peer selection problem in single-view P2PMMS and multi-view P2PMMS systems.

A. Peer Selection in a Single-View P2PMMS System

Generally, a peer in a single-view P2PMMS system might encounter two kinds of bottlenecks influencing the playback satisfaction. First, the bandwidth bottleneck [6], which means that a peer is not able to obtain enough bandwidth from its upstream neighbors (a peer's *upstream* neighbors are those from which the peer obtains contents and bandwidth and this peer is their corresponding *downstream* neighbor). Bandwidth bottleneck can be eliminated by constructing an effective overlay. Second, the content bottleneck, where a peer can't get useful contents from its upstream neighbors, even though it has sufficient bandwidth. The content bottleneck can be mitigated by using an efficient block scheduling algorithm.

Peer selection is the prerequisite stage of overlay construction and therefore it is vital for eliminating the bandwidth bottleneck and for improving the overall system performance. Its goal is to facilitate a new peer in selecting proper neighboring peers from which it will likely obtain the required bandwidth. As mentioned before, peer selection has been well-studied in single-view P2PMMS systems. When a peer joins a single-view P2PMMS system, it first contacts the tracker server to get a list of usually randomly selected candidate neighbors. Next, the peer sends a request to a peer in the list and possibly some other peers using a certain mechanism. Finally, the requested peer may accept or reject the request based on its constraints, especially its bandwidth limitations.

B. Peer Selection in a Multi-View P2PMMS System

A fundamental question of peer selection for multi-view P2PMMS is whether a new peer should select multi-view peers or single-view peers as its neighbors. Before approaching the answer to this question, we first discuss some unique characteristics of a multi-view P2PMMS system.

One unique characteristic distinguishing a single-view P2PMMS system and a multi-view P2PMMS system is the existence of multi-view peers in a multi-view P2PMMS system. Since a multi-view peer is watching multiple channels simultaneously, it is capable of *adjusting* the upload bandwidth allocation among different overlays.

An example is illustrated in Figure 1. Peer A is a multi-view peer watching all three channels. Peer B is a single-view peer watching only channel 3, and peer C is another single-view peer watching only channel 2. Both peer B and peer C are neighbors of peer A. Consider the case in which if A evenly allocates its upload bandwidth among all of its neighbors, then

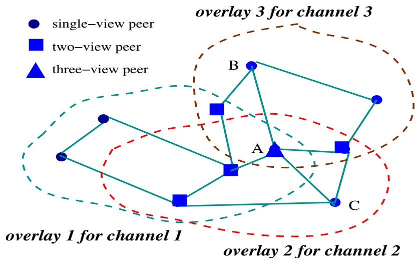

Fig. 1. A multi-view P2PMMS system with both multi-view peers (e.g. peer A) and single-view peers (e.g. peers B and C).

B does not have sufficient download bandwidth, in the sense that the total obtained bandwidth from upstream neighbors is less than the streaming rate of B's desired channel. However, C has some extra download bandwidth, which can be considered as a waste of system resources. In this case, A can help B to improve its performance by allocating more bandwidth to B and less bandwidth to C. This situation is very likely to occur in real P2PMMS systems, in that peers in P2PMMS have heterogeneous bandwidths (both upload and download bandwidths). Moreover, different channels may have different numbers of peers (e.g. some channels are less popular than others). In this situation, overlays with smaller populations can benefit from the reallocated shared bandwidth.

Intuitively, a new peer may prefer to select a multi-view peer as its neighbor due to the advantages discussed above. However, this peer selection may lead to the following two problems. First, the *unfairness problem* where peers joining the system earlier may consume all the bandwidth of multi-view peers, and thus a peer joining later may not have the opportunity to select a multi-view peer as its neighbor. In this case, only a few peers can benefit from multi-view peers. Second, the *clustering problem* where peers may cluster around multi-view peers. In this case, if a multi-view peer leaves the system or changes its channels, then a large number of its neighbors would be severely affected.

Let us now return to the question presented at the beginning of this subsection. Whether a peer chooses a multi-view peer as its neighbor or not depends on whether such a peer is beneficial for its playback quality. To solve this problem, we propose Channel-Aware Peer Selection for multi-view P2PMMS in the next section.

IV. Channel-Aware Peer Selection (CAPS) for Multi-View P2PMMS

Since multi-view peers are beneficial for downstream peers' playback quality, it is better for downstream peers to select a proper fraction of multi-view peers as their upstream neighbors, in order to avoid the clustering and unfairness problems. To achieve this goal, an important objective is to distribute the multi-view peers evenly among all of the peers in the system.

Our proposed method uses the peers' upload bandwidth information in the system to control the number of different types of neighbors, which are classified according to their number of viewed channels. The number of different groups of neighboring peers is determined by the total upload bandwidth of different groups. It means that a group with a relatively higher bandwidth than others with lower bandwidth, should be selected as neighbors by other peers with a relatively high probability. In the following paragraphs, we first introduce how to control the number of neighbors in different groups, given the bandwidth information. Then, we will discuss how to estimate the bandwidth information in the system.

Let us consider an example. Suppose that peer A watches a single channel. A can receive some global control information about the distribution of peers watching different channels. We call the control information as the *neighbor proportion control matrix*, which is a two dimension n *by* n matrix, where n is the maximal number of channels in the system. The entry m_{ij} denotes the probability of peers in group i selecting peers in group j as neighbors and $M[i][*]$ represents the *ith* row of M. For example, the matrix for A could be $M[A][*] = [0.2, 0.3, 0.5]$, where each element $M[A][i]$ represents the probability of selecting upstream neighbors watching i channels. Therefore, A can select single-view peers as neighbors with the probability 0.2 and two-view peers with the probability 0.3 and three-view peers with the probability 0.5.

The control matrix M is determined by the estimation of upload bandwidth among different peer groups, where each group corresponds to the peers watching exactly the same number of channels. The intuition behind the determination of M is to find out how to utilize the surplus bandwidth of one group to fill up the deficit bandwidth of another group, where "how to" means determining the proper fill up ratio.

As an example of determining M, suppose that there are two groups of peers G_1 and G_2 watching maximally 2 channels, with size of 1,000 peers and 2,000 peers respectively; the total upload bandwidth of G_1 is 400Mbps and total required bandwidth is 500Mbps; while the total upload bandwidth of G_2 is 2Gbps and total required bandwidth is 2.1Gbps. Also assume that peers in G_1 only watch one channel A with streaming rate 500kbps and peers in G_2 watch channels A and B. Now we determine $M[1][*]$. Based on the assumption, G_1 has 500Mbps$-$400Mbps $=$ 100Mbps deficit bandwidth and G_2 has 2.1Gbps$-$2Gbps$=$ 100Mbps surplus bandwidth, which can support 100Mbps/0.5Mbps$=$ 200 peers in G_1. Therefore $M[1][*] = [(1000 - 200)/3000, (2000 + 200)/3000]$,which indicates that peers in G_1 selects neighbors in G_1 and G_2 with the probability 0.27 and 0.73 respectively.

The essential part of control matrix determination is to estimate the bandwidth of different peer groups mentioned above. It can be estimated either by centralized algorithms or by distributed algorithms. Since the P2P streaming networks are usually Internet-scale, distributed algorithms are preferred. RandPeer [3] uses a distributed scheme for membership management. It clusters peers based on their QoS properties and supports efficient queries. Therefore, we can use RandPeer style distributed information management scheme to cluster peers based on their viewed channels. Then, when a peer joins a channel/channels, it reports its channel and bandwidth information to RandPeer and retrieves the current system in-

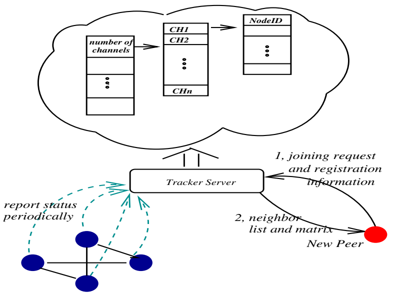
number of channels
CH1
CH2
CHn
NodeID
Tracker Server
1, joining request and registration information
2, neighbor list and matrix
New Peer
report status periodically

Fig. 2. A centralized implementation for Channel-Aware Peer Selection in multi-view P2PMMS

formation to estimate the control matrix M. In addition, there are other choices of implementing the distributed bandwidth estimation, such as Meridian [12] and Vivaldi [2].

In this paper, we focus on the impact of Channel-Aware Peer Selection in multi-view systems. Therefore, we implement a simple centralized scheme in our simulation instead of the complex distributed schemes. Figure 2 illustrates this mechanism. There is a tracker server maintaining the channel distribution information of all the peers. When a peer joins the system, it first contacts the tracker server to get a list of candidate neighbors, and more importantly, to get the peer selection control matrix. Based on this information, a peer then communicates with each peer in the list to find its neighbors. Once a peer joins the system, it periodically send its channel subscription information to the tracker server, so that the tracker server always maintains the correct information. We must emphasize that our proposed Channel-Aware Peer Selection is sensitive to bandwidth estimation schemes.

From a peer's perspective, the neighbor proportion control matrix solves the problem of whether a peer should select multi-view peers as neighbors by providing channel-awareness capability for each peer. In the global scope, the matrix controls the distribution of multi-view peers. As discussed in the related work section, there are several performance concerns of the systems, such as location awareness etc. In this paper, we focus on upload bandwidth allocation among all peers in the system.

V. Simulation Experiments and Results

In this section, we study the performance of CAPS for multi-view P2PMMS systems by simulating a large-scale P2PMMS system.

A. Simulator Design and Setup

Our simulator is based on the P2PStrmSim simulator developed by Meng Zhang [7], which has the capability of simulating a maximum of 10,000 peers joining a single overlay at the packet level (many more concurrent online peers than in ns-2). The latency matrix of the underlying topology is obtained from the Meridian Data set [10], which reflects the median of the round-trip times between 2500×2500 nodes.

We extended the simulator under the original framework, at both packet level and flow level for multi-view P2PMMS simulation. The packet level simulator fully models the P2P streaming systems including overlay construction, block scheduling and buffer modeling etc and therefore it simulates the real systems. By contrast, the flow level simulator simulates overlay construction, resource allocation, and message exchange among peers, but it does not simulate streaming at packet-level such as block scheduling and buffer modeling. Although flow level simulator can not simulate packet-level streaming, it is an efficient and accurate tool to study the overlay construction, resource allocation and neighbor selection in large scale multi-view systems. The reason is that it eliminates the streaming traffic but accurately simulates other parts of the system. In this paper, we use the flow level simulator to study the impact of channel-aware neighbor selection in multi-view systems.

We implemented a single tracker server in the simulator, which is assumed to be always up. This is done for efficiency and to exclude other irrelevant factors. The tracker server manages peer information in a multi-level map-set structure, an implementation of the structure shown in Figure 2. The key parameters are Neighbor Proportion Control (NPC) matrices, which can be modified from random to any controlled value ranging from 0 to 100%.

In order to evaluate the tracker server based method, we measure whether a peer attains sufficient upload bandwidth for sustaining all its interested channels. The *Bandwidth Satisfaction Ratio* (BSR) is defined as follows:

$$BSR_i^S = \frac{\sum_{j \in Nbr(i)} a_{ij}^S}{StrmRate^S}$$

where BSR_i^S denotes peer i's bandwidth satisfaction ratio of Channel/View S; a_{ij}^S denotes the bandwidth that peer i gets from peer j for channel/view S; and $StrmRate^S$ denotes the streaming rate of Channel/View S. $Nbr(i)$ denotes the neighboring set of peer i.

In order to evaluate our neighbor selection scheme, we fully implemented the dynamic bandwidth allocation algorithm proposed by [14] to allocate the bandwidth for existing overlays. The allocation algorithm aims to solve the bandwidth requirement conflicts among peers after constructing overlays with different neighbor selection methods. Briefly, the authors modeled the bandwidth allocation problem into dynamic auction games, in which upstream peers sell their upload bandwidths to maximize their revenue and downstream peers try to obtain required bandwidths with lowest cost. Therefore, overlays generated by good neighbor selection schemes will show good bandwidth allocation results and vice versa. The bandwidth allocation results can be used to evaluate whether a peer selection scheme is better than others.

Additionally, our simulator provides two modes of peer joining/leaving corresponding to the static mode and trace-driven mode. In the static mode, peers randomly join the overlay at the beginning of the simulation, and most of them leave the overlay at the end of simulation. In the trace-driven mode, the joining time is collected from the P2PMMS system, GridMedia [7] developed by Meng Zhang, and most of the peers will stay in the overlay until the simulation ends. The reason why we do not simulate the departure during simulation is that the simulation is at the flow level and we implemented an overlay recovery mechanism. Therefore the peer departure only affects the convergence time of the bandwidth allocation algorithm.

B. Simulation Results

Our motivation in these sets of simulation experiments is to study the impact of Channel-Aware Peer Selection in multi-view P2PMMS. In the following sets of simulation experiments, every peer can watch at most four channels simultaneously, corresponding to four groups of peers. Peers in group I watch only one channel; peers in group II watch two channels, which are the combinations of any two of the four channels, and so on. In addition, each channel has the same, fixed streaming rate.

Based on the basic simulation setting, the dimensions of the neighbor proportion control matrix, M introduced in the previous section, is 4×4, where each row i of M represents the neighbor proportion control parameter of peer group i. The central server sends the candidate neighbor list to a specific peer by looking up the specific row of matrix M indexed by the peer's group number.

Moreover, we use the upload bandwidth distribution vector D and the bandwidth type vector T to describe the upload bandwidth of a given group. For example, if $D = [0.2, 0.3, 0.7]$ and $T = [1M, 256k, 128k]$ in group I, it means that there are 20% of peers in group I have an upload bandwidth 1Mbps and 30% peers have a 256kbps bandwidth and 70% peers have a 128kbps bandwidth.

We ran our simulations in both static and trace-driven modes and only the results of static mode are shown here due to space considerations. We observed that the trend and impact of the trace-driven mode are similar to those of the static mode.

Since there are so many control factors in the simulator, we use the following setting to study the impact of channel-aware peer selection: first, all of the 10,000 peers are divided into the four groups with the population size of 4,000 in group I and 2,000 in each of the other three groups; we set the total upload bandwidth of group I, which represents the peers watching a single channel, to be insufficient to support all peers in the group; then we set the total upload bandwidth of group II, III, and IV to be sufficient. Parameters of simulations are summarized in Table I Then we change the control matrix M to show the impact of the performance for group I peers (single-view peers). Among all the simulations, we chose three representative sets with the corresponding M as follows:

$$M_{RPS} = \begin{bmatrix} 0.4 & 0.2 & 0.2 & 0.2 \\ 0.4 & 0.2 & 0.2 & 0.2 \\ 0.4 & 0.2 & 0.2 & 0.2 \\ 0.4 & 0.2 & 0.2 & 0.2 \end{bmatrix}$$

$$M_{CAPS1} = \begin{bmatrix} 0.1 & 0.27 & 0.3 & 0.33 \\ 0.1 & 0.5 & 0.2 & 0.2 \\ 0.1 & 0.2 & 0.5 & 0.2 \\ 0.1 & 0.2 & 0.2 & 0.5 \end{bmatrix}$$

$$M_{CAPS2} = \begin{bmatrix} 0.14 & 0.16 & 0.24 & 0.46 \\ 0.14 & 0.16 & 0.24 & 0.46 \\ 0.14 & 0.16 & 0.24 & 0.46 \\ 0.14 & 0.16 & 0.24 & 0.46 \end{bmatrix}$$

We use RPS, CAPS1 and CAPS2 to represent the three sets of experiments and we use the average value of BSR and the histogram of each channel in *Group I* to evaluate the impact. RPS corresponds to the Random Peer Selection, since each element of each row in the *Neighbor Proportion Control (NPC)* matrix is equal to the peer population of each group (e.g. *Group I* has a population of 4000, and therefore a peer would select a *Group I* neighbor with a probability of 0.4). CAPS1 and CAPS2 represent the Channel-Aware Peer Selection.

The matrices M_{CAPS1} is determined as follows. Since only *Group I* peers suffer bandwidth deficit, they migrates to some other groups, which have surplus bandwidth. As an example, *Group II* has 0.08 Gbps surplus bandwidth and can support 0.08Gbps/300kbps= 267 peers from *Group I*. Therefore, the probability of selecting *Group II* peers as neighbors is $0.2 + 0.07$, where $267/4000 = 0.07$. Similarly, we determine other entries of the first row. Other rows of M_{CAPS1} control peers in corresponding group to select *Group I* peers as neighbors with the same probability as row 1, but they prefer to select peers in the same group. For example, row 2 of M_{CAPS1} determines that *Group II* peers select *Group I to IV* peers as neighbors with probability 0.1, 0.5, 0.2 and 0.2 respectively. M_{CAPS2} is determined by the bandwidth distribution of each group (i.e. the neighbor selection is based on the total upload bandwidths of different groups). The goal of these simulations is to show that the peer heterogeneity is considered in channel aware neighbor selection and it is effective in improving the system resources utilization and the user satisfaction.

Simulation results are shown in two types of figures: for each channel, the average BSR of *Group I to IV* peers; for each channel, the BSR distribution of *Group I* peers. For space reason, we show the two types of figures for channel 1 and channel 2 respectively. Note, channel 1 refers to peers at least watching channel 1, so does channel 2.

From Figure 3, we can see that for channel 1, peers in *Group I to IV* corresponding to watching one to four channels, have a low BSR with RPS scheme. This is because peers select neighbors in *Group I* with a high probability using random scheme. However, *Group I* peers do not have sufficient bandwidth to support these peers and surplus bandwidths in other groups can not be used to relieve the bandwidth deficit. Figure 5 shows that the CAPS1 and CAPS2 increase at least 10% the proportion of peers, whose BSR is greater than 0.9. In addition, CAPS2 has more peers with BSR greater than 0.9, since the control matrix is determined by the distribution of the bandwidth. In Figure 3, CAPS1 has a higher average BSR than CAPS2 in *Group IV*. It is can be explained below. *Group IV* peers have much higher total upload bandwidth than other groups but they consume more bandwidth. CAPS1 controls

TABLE I
SIMULATION SETUP

Group	Population	Streaming Rate	Required Bandwidth	Total Upload Bandwidth
I	4000	300 kbps	1.2 Gbps	1.08 Gbps
II	2000	300 kbps	1.2 Gbps	1.28 Gbps
III	2000	300 kbps	1.8 Gbps	1.92 Gbps
IV	2000	300 kbps	3.2 Gbps	3.7 Gbps

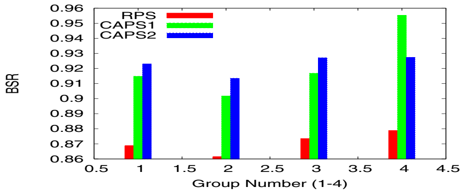

Fig. 3. Average BSR for Peers Watching At Least Channel 1

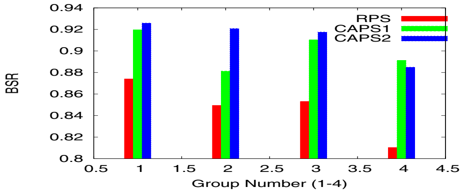

Fig. 4. Average BSR for Peers Watching At Least Channel 2

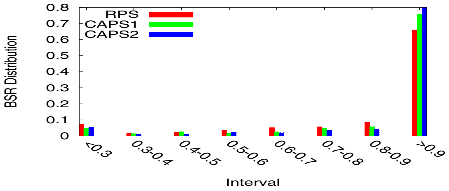

Fig. 5. BSR Distribution for Channel 1

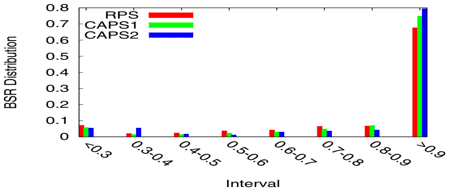

Fig. 6. BSR Distribution for Channel 2

peers to select neighbors in the same group with a higher probability. *Group IV* peers can benefit more from their own group, but the proportion of peers with BSR greater than 0.9 is lower than CAPS2. Therefore, neighbor selection algorithms should consider both the bandwidth distribution and channel information. Since channel 2 has similar trend, we omit the discussion here.

VI. CONCLUSION

In this paper, we propose a novel peer selection algorithm, called Channel-Aware Peer Selection (CAPS), where a peer selects its neighboring peers based on the channel subscription of the system, in order to efficiently utilize the bandwidth of all peers in the system. The results of a large-scale simulation study with 10,000 peers and 4 channels show that CAPS can greatly improve the system performance than the straightforward Random Peer Selection (RPS), which is widely used in single-view P2PMMS systems. The improvement is due to the reason that CAPS considers the peer heterogeneity of different channels.

Encouraged by the simulation results, in the future, we will design a dynamic information management system to collect the peer and channel information for determining the control matrix. Moreover, we will propose an optimized neighbor selection algorithm, fully considering the heterogeneity and dynamics in multi-view P2PMMS systems.

ACKNOWLEDGEMENT

The authors would like to thank Meng Zhang (Tsinghua University, China) whose efforts helped make this study possible.

REFERENCES

[1] M. Adler, R. Kumar, K. W. Ross, D. Rubenstein, T. Suel, and D. D. Yao. Optimal peer selection for p2p downloading and streaming. In *Proceedings of IEEE INFOCOM*, Miami, FL, March 2005.

[2] F. Dabek, R. Cox, F. Kaashoek, and R. Morris. Vivaldi: A decentralized network coordinate system. In *Proceedings of the ACM SIGCOMM '04 Conference*, Portland, Oregon, USA, August 2004.

[3] J. Liang and K. Nahrstedt. Randpeer: Membership management for qos sensitive peer-to-peer applications. In *Proceedings of IEEE INFOCOM 2006*, April 2006.

[4] J. Liang, B. Yu, Z. Yang, and K. Nahrstedt. A framework for future Internet-based TV broadcasting. In *Proceedings of IPTV Workshop*, Edinburgh, Scotland, May 2006.

[5] Y. Liu, L. Xiao, X. Liu, L. Ni, and X.Zhang. Location awareness in unstructured peer-to-peer systems. *IEEE Transactions on Parallel and Distributed Systems*, 16(2):163–174, February 2005.

[6] N. Magharei and R. Rejaie. PRIME: Peer-to-peer receiver-driven mesh-based streaming. In *Proceedings of IEEE INFOCOM*, Anchorage, Alaska, May 2007.

[7] P2PStrSim. http://sourceforge.net/projects/p2framework.

[8] PPLive. http://www.pplive.com.

[9] PPStream. http://www.ppstream.com.

[10] Meridian Data set. http://www.cs.cornell.edu/People/egs/meridian/data.php.

[11] Z. Xu C. Tang and Z. Zhang. Building topology-aware overlay using global soft-state. In *Proceedings of International Conference on Distributed Computing Systems (ICDCS'03)*, Providence, Rhode Island USA, May 2003.

[12] B. Wong, A. Slivkins, and E. Gün Sirer. Meridian: A lightweight network location service without virtual coordinates. In *Proceedings of ACM SIGCOMM 2005*, 2005.

[13] C. Wu and B. Li. rStream: resilient peer-to-peer streaming with rateless codes. In *Proceedings of the 13th annual ACM International Conference on Multimedia*, pages 307–310, Singapore, November 2005.

[14] C. Wu and B. Li. Strategies of conflict in coexisting streaming overlays. In *Proceedings of IEEE INFOCOM*, Anchorage, Alaska, May 2007.

[15] X. Zhang, J. Liu, B. Li, and T. Yum. DONet/CoolStreaming: A data-driven overlay network for live media streaming. In *Proceedings of IEEE INFOCOM*, Miami, FL, March 2005.