A Fast Channel Switching System for IPTV based on Multi-channel Preview

Tein-Yaw Chung, Cheng-Li Hsieh, Hsiao-Yi Wang, Chung-Yu Hsu, Fong-Ching Yuan
Departments of Computer Science and Engineering
Yuan Ze University
Chungli, 320 Taiwan
E-mail: csdchung@netlab.cse.yzu.edu.tw, s961432@mail.yzu.edu.tw,
s961419@mail.yzu.edu.tw, frycol@netlab.cse.yzu.edu.tw,
imyuan@saturn.yzu.edu.tw

Abstract—**In this paper, we propose a High Performance Multi-channel Preview (HPMP) System in Peer-to-Peer live media streaming to help users find their favorite channel fast. In HPMP, users can watch TV in two different modes: Preview and Watch mode. To support the Watch mode, HPMP uses a P2P mesh topology to promote bandwidth utilization similar to many existing solutions. In the Preview mode, a P2P timed forest scheme (TFS) is proposed to reduce tracker server overhead and minimize topology maintenance overhead. Finally, we use OMNet++ to simulate the performance of HPMP and compare with a system without a preview service. The simulation results show that HPMP outperforms existing non-preview systems.**

Keywords- Channel switching time; IPTV; Live streaming; Multi-channel Preview; P2P; Timed Forest Scheme;

I. Introduction

In recent years, the development of P2P Multimedia Streaming System (P2PMMS) [1][2][3][4] is getting more and more popular and has attracted attention both in academic and industrial circles. Now a day, there are a large number of users using PPLive [5] or PPStream [6] to watch video channel. With the increasing number of channels, users have more and more choices to watch their favorite channel. However, how do we help users to find their favorite channel fast? Most existing solutions [7][8] change the topology or buffer processing methods to shorten the channel switching delay but increase the overhead of server.

On the other hand, in the existing P2PMMS, real-time playback and video-on-demands (VoD) [9][10] program are indicated as words. Users need to select one channel before they can watch the channel. Consequently, it increases substantially the frequency of channel switching. In this paper, we propose a High Performance Multi-channel Preview (HPMP) System in Peer-to-Peer live media streaming to help users find their favorite channel fast. In HPMP, users can watch TV in two different modes: Preview and Watch mode. Figure 1 shows the interface of the Preview mode, in which the users can choose the channel type first and then HPMP can provide many channel previews with low quality videos. After a user chooses a channel, the user switches to the Watch mode with a full quality channel.

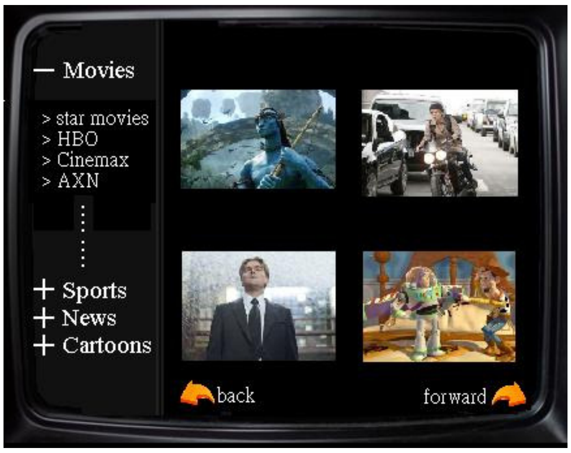

Fig. 1 Interface schematic under HPMP

There are four goals to implement HPMP: 1) help users to find their favorite channel fast, 2) decrease the bandwidth consumption during searching for users' favorite channel, 3) Reduce the server's overhead and the peer's connection delay when peers switch channel, 4) provide a stable quality for users to watch multi-channel preview.

To support the Watch mode, HPMP uses a P2P mesh topology to promote bandwidth utilization similar to many existing solutions. In the Preview mode, a P2P timed forest scheme (TFS) is proposed to reduce tracker server overhead and minimize topology maintenance overhead. TFS is a novel scheme that constructs a tree for a group of users with the same channel preview and the root switches channel preview group on behalf of all users in the tree periodically. Thus, TFS has nice features to fulfill our goal.

In this paper, we address the issues of how to control and construct a preview tree and how to allocate upload bandwidth in the Preview mode and Watch mode. Architecture and algorithms are presented to implement HPMP. The rest of the paper is organized as follows. Section II briefly review related work. Section III presents our designs for architecture and algorithms of HPMP. Section IV shows our simulation results. Section V concludes the paper.

Proc. of the 4th Annual International Conference on Computer Games, Multimedia and Allied Technology (CGAT 2011)
Editor: Dr. Edmond Prakash
Co-Editor (CCV 2011): Prof. San Murugesan
Copyright © 2011 GSTF & CGAT
ISBN: 978-981-08-8644-8

II. RELATED WORKS

In the past, researchers have proposed channel change acceleration mechanisms in IPTV such as Instant Channel Change (ICC) [11], or a buffering technique on channel changing servers to reduce the channel change delay. The ICC solution provides a unicast stream with only I-frame so that Set-top Box (STB) can receive data fast. When a STB receives the first I-frame, it can start to display video. However, this solution requires extra bandwidth when channel change happened. Moreover, it increases the management overhead in acceleration mechanisms.

The other solution is to provide Picture-in-Picture (PIP) stream from neighboring nodes to a STB. Users can watch a full quality video and also a low quality video simultaneously. And this PIP solution has acceptable bandwidth overhead since the PIP streams require very small bandwidth. However, the drawback of PIP stream is that it has a limited number of channels to watch at the same time.

Compared with these above methods, our design provides a novel mechanism HPMP to allow user to preview many channels with low quality video. It can reduce the frequency of channel switching and channel hop distances.

III. HPMP FOR P2P MULTIMEDIA STREAMING SYSTEM

In HPMP, channels are classified into M types, i.e., $A=\{A_i \mid 0<i\le M\}$. For example, Figure 1 illustrates the M types channel on the left of the interface and each channel type A_i includes N related channels $B=\{B_{ij} \mid 0<i\le M, 0\le j\le N\}$. When a client peer joins the system on the preview mode and chooses channel type A_i, HPMP will select C (we assume C=4 in the following description) channels from B_{ij} and shows them in the main screen. Note that there are "back" and "forward" buttons below the main screen to change channel fast.

Figure 2 shows the operation process of the HPMP system. There are three operational modes in HPMP: 1) Wait mode: in this mode, users wait to choose the channel that client peer wants to see after it joins the system, 2) Preview mode: In this mode, HPMP provides C low quality streams to a client peer to choose what it wants to watch. Once a client peer on the preview mode presses the "back" or "forward" button, it will come back to the wait mode and rejoin the next preview channel tree. 3) Watch mode: in this mode, users start to watch a channel in a full screen. It will be back to the preview mode when a client peer wants to select some other channels.

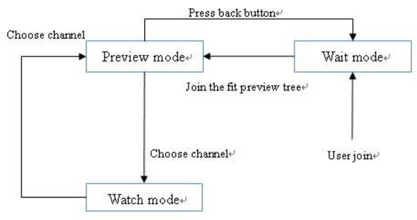

Fig. 2 The operation process of HPMP system

To reduce the overhead for channel preview, we presented a novel method named Timed Forest Scheme (TFS) for group preview. By this way, HPMP can help clients finding fast the channel they want to watch and use their bandwidth efficiently.

Figure 3 shows an example for the architecture of the HPMP system, in which the status of a client peer is either in the Watch mode or the Preview mode when it joined the HPMP system. Here, we focus on the design of the Preview mode and construct generally a mesh-based topology on the Watch mode using gossip [12] for P2P message exchange. All client peers on the Watch mode will calculate their own quality value Q by Eq. (1) periodically and send the value to the tracker server. The tracker server keeps a source peer list $S=\{s_{ij} \mid 0<i\le M, 0\le j\le N\}$, which is periodically updated and sorted based on the value Q.

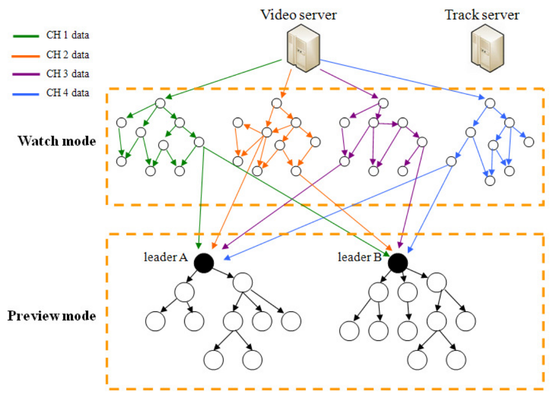

Fig. 3 The architecture of HPMP System

Now, we propose a design of TFS when the system is in the Preview mode. The concept of TFS is shown in Figure 4.

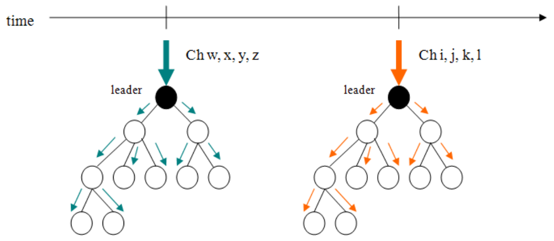

Fig. 4 Timed Forest Scheme on Preview mode to change channel

In the preview mode, three mechanisms are used: 1) Preview Tree Feeding (PTF), 2) Preview Peer Joining (PPJ), 3) Preview Tree Maintenance (PTM). The TFS scheme creates a tree based topology when client peers join the preview mode of the HPMP system. When a client peer joins HPMP on the Preview mode and becomes a leader peer (or root of a TFS tree), the Preview Tree Feeding process is executed. In PTF, the tracker server will transmit sub-lists $s\in S$ with C different channels to the new root. After that, the leader peer will calculate the R value by Eq. (2) and choose source peers with the minimal R value on C different channels in sub-lists $s\in S$, and then push streams down to their leaves. The R represents the stability of connection between the leader peer and the source peer; a connection with a smaller R has higher stability.

After T time units, the leader peer will automatically hop to next source peers of C channels recommended by the tracker server and change channels. The architecture can be regarded as a forest since it contains many trees and each tree changes channels simultaneously every time T units. In this way, TFS can help users to reduce the connection time and the overhead of tracker server when all peers change preview channels. Algorithm 1 shows the working principle of the root (leader) in TFS.

Algorithm 1 Leader Preview Tree Feeding in TFS

```
1: T := the period for hopping to next source peers of C
        Channels
2: N := the total number of preview channels
3: $C_s$ := temporary set to save the candidate source peers
         on channels C
4: Leader peer takes source peers in set $C_s$ from tracker
   server for C channels in the N channels
5: delay T
6: while TRUE do
7:    disconnect with previous C source peers
8:    take a new candidate source peers set $C_s$ from
      tracker server
9:    connect with the new C source peers $C_s$
10:delay T
11:end while
```

To increase the stability of the Preview mode, we limit the depth of the TFS tree in the HPMP system. A new join peer will create a new preview tree and register as a leader when the depth of all preview trees reaches the limitation D; otherwise, it will join the preview tree with minimal depth. Besides, the new preview tree will take the data of next C channels. An example is shown in Figure 5, in which the new leader will take the data of channel 9, 10, 11 and 12 if all depth of existing preview trees is D (D=3 in the example).

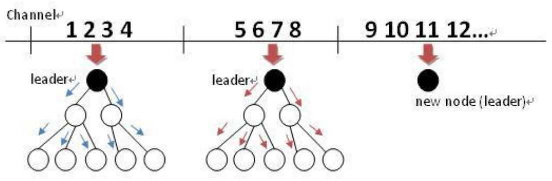

Fig. 5 The construction of leader

On the other hand, a new leader will take the data of channel 1,2,3,4, the same as the fist preview tree, when there are only 12 channels and all depth of existing trees is D.

In addition, there exists an exceptional problem when the number of peers on the watch mode and the preview mode are not balance. That is we do not have enough source peers to support the preview streams for all TFS trees. To solve this problem, we increase the depth D of the preview tree to make each root provide more preview streams and hence can support many more peers joining the preview mode. The detailed algorithm of preview peer joining is shown in Algorithm 2.

Algorithm 2 Preview Peer Joining in TFS

```
1: $C_{peer}^{min\_depth}$ : save a peer with minimal depth in preview tree
2: $C_{child}^{num}$ : the current number of children supported by
      $C_{peer}^{min\_depth}$
3: $C_{child}^{max}$ : the maximal number of children supported by
      $C_{peer}^{min\_depth}$
4: $C_{c\_list}$ : a list of children of $C_{peer}^{min\_depth}$
5: $C_{root}^{depth}$ : the depth of root
6: $leader_{list}$ : the list of current leader
7: $D$ : the maximal depth set by the HPMP system
8: $C_{peer}^{min\_depth} \prec=$ peer with minimal depth in $leader_{list}$
9:   if $C_{root}^{depth} = D$ then
10:      be a leader peer
11:      if there are no available source peers then
12:        increase $D$ by one
13:        rejoin the preview tree and go to step 8
14:      end if
15:      else
16:        join the list of $leader_{list}$ and create a new preview
           tree
17:      end else
18: end if
19: else
20:   send a connection request to peer $C_{peer}^{min\_depth}$
21:   while $C_{child}^{num} = C_{child}^{max}$ do
22:      $C_{peer}^{min\_depth} \prec=$ peer with minimal depth in $C_{c\_list}$
23:   end while
24:   send a connection request to peer $C_{peer}^{min\_depth}$
25:   connect to peer $C_{peer}^{min\_depth}$
26:end else
```

Peers leave a preview tree dynamically. There are three ways for preview peers to leave the Preview mode of the HPMP system. The first one is to press the small screen and join the Watch mode. The second one is to choose another channel type and join the other type of preview tree. The third one is to press the “back” or “forward” button. A preview peer will rejoin the Preview mode in the HPMP system when it detects the later two conditions with the preview peer join process.

On the other hand, when a preview peer detects that its parent node has left, it and its sub-tree, if any, will rejoin a preview tree. First, it will select another preview tree with the same C channel and join if its parent is leader peer or the limitation D of the same preview tree can accommodate the depth of its sub-tree. Otherwise, the preview peer will become a leader and create a new tree when there are not appropriate preview tree to join. The detailed algorithm is shown in Algorithm 3.

Algorithm 3 Tree maintenance when parent node leaves in TFS

```
1: $C_{peer}^{\min\_depth}$ : save a peer with minimal depth in preview tree
2: $C_{child}^{num}$ : the current number of children supported by
        $C_{peer}^{\min\_depth}$
3: $C_{child}^{\max}$ : the maximal number of children supported by
        $C_{peer}^{\min\_depth}$
4: $C_{c\_list}$ : a list of children of $C_{peer}^{\min\_depth}$
5: $C_{root}^{depth}$ : the depth of root
6: $leader_{list}$ : the list of current leader
7: $D$ : the maximal deep set by the HPMP system
8: $p_{depth}^{current}$ : the tree depth of current preview peer
9: $C_{peer}^{\min\_depth} \prec=$ peer with minimal depth on the same
   channels C in $leader_{list}$
10:  if $C_{peer}^{\min\_depth}$ does not exist then
11:      join the list of $leader_{list}$
12:      create a new preview tree
13:  end if
14:  else
15:    if $p_{depth}^{current} + C_{root}^{depth} \geq D$ then
16:       join the list of $leader_{list}$
17:       create a new preview tree
12:    end if
13:    else
14:      while $C_{child}^{num} = C_{child}^{\max}$ do
15:          $C_{peer}^{\min\_depth} \prec=$ peer with minimal depth in $C_{c\_list}$
16:      end while
17:      send a connection request to peer $C_{peer}^{\min\_depth}$
18:      connect to peer $C_{peer}^{\min\_depth}$
19:    end else
20:  end else
```

How a leader peer selects the most stable source peer on the watch mode is important. So we design a model to calculate the quality Q of a watch peer and the value R of a preview peer by Eq. (1) and Eq. (2) respectively to find the stable peers in the HPMP system. Here, the quality Q and value R are denoted as follows.

$$Q = \alpha \frac{B^U}{B_{UMax}} + \beta \frac{L_{T\max}}{L_T} \tag{1}$$

$$R = Q + (1 - \alpha - \beta)RTT \tag{2}$$

where $0 < \alpha, \beta < 1, \alpha + \beta \leq 1$. The variables and some properties involved in the potential function are defined as shown in Table 1. Note that the quality Q considers the distance between peers, i.e., the upload bandwidth B, the living time of peer's L, in one specific channel. For Eq. (1), the first term considers the source peer with lower upload bandwidth utilization and the second term considers the source peer with the longest online time. We merge the above-mentioned model to find stable source peers.

TABLE I. NOTATION DEFINITIONS

Symbol	Meaning
RTT	The round-trip time delay between a source peer and leader peers on the preview mode.
B^U	the upload bandwidth has been used in the watch mode.
B_{UMax}	The maximal upload bandwidth of each peer on the watch mode.
L_T	The living time of each peer on the watch mode in the system.
L_{Tmax}	The longest living time of peers on the watch mode.

When a peer joins the watch mode, it calculates its own quality Q periodically and reports to the tracker server. In this way, the tracker server will save better peers in the source list.

On the preview mode, a leader will take the source list from the tracker server and calculate the value R to choose a source peer with minimal value R. If the source peer leaves the system, the leader peer will find another source peer and keep all its tree structure without change. In other word, it reduces the connection time if all peers are made reconnect to another tree.

IV. SIMULATION RESULTS

In this section, we perform extensive simulations to study the performance of the HPMP system. We simulate distributions with different upload bandwidth using network simulator OMNeT++ [13]. We use a full quality video stream whose rate is 2 Mbps and the rate of low quality of preview stream is 128 Kbps. The number of channels is 20. The incoming access link bandwidth for all peers is set to 2 Mbps so that each peer can easily receive the full quality rate of a video. The number of total peers joining the HPMP system is 200. And Figure 6 shows the time distribution of peers joining system and favorite channel of each peer is randomly selected between 1 to 20. In the HPMP system, the period T of preview channel auto-switching is 8 seconds. Each time users can watch C preview channels and C is set as 4. In the traditional scheme, a user needs to change channel by himself/herself, and each time a full quality video is presented. We set that every user starts to change channel in order from Channel 1 up to his/her favorite channel because they do not know beforehand what the content of each channel is.

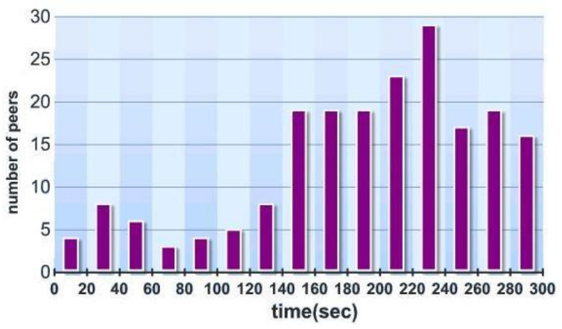

Fig. 6 The distribution of Peers joining the HPMP system

The outgoing bandwidth of peers can be set as one of four values: 0.5Mbps, 1Mbps, 2Mbps and 3Mbps as shown in Table 2. To analyze the effect between the depth of preview tree and average hop distance time, we set a half of peers in SN1 with higher upload bandwidth and a half of peers in SN2, 3 and 4 with lower upload bandwidth so that the number of peers on the watch mode and the preview mode are not balance when large amount peers suddenly join the system. We use four scenarios, each has different distributions of 200 peers across these four groups. The parameter of average hop distance time is defined as the average hopping time that a client starting from the first channel until it reaches its favorite channel. In the simulation, we assume that the leaders of the preview trees always find enough source peers to connect.

TABLE II. SCENARIOS FOR COMPARING DIFFERENT UPLOAD BANDWIDTH AND DEPTH OF PREVIEW THREE UNDER HPMP AND TRADITIONAL MODE

Upload BW	**SN 1**	**SN 2**	**SN 3**	**SN 4**
0.5Mbps(peers)	20	100	100	100
1Mbps(peers)	30	50	50	50
2Mbps(peers)	50	30	30	30
3Mbps(peers)	100	20	20	20
Tree's depth	8	6	8	10
	Average hop distance time(sec)			
HPMP	79.579	48.623	66.527	82.065
Traditional mode	138.09	149.41		

Table 2 shows the result with different upload bandwidth and depth of preview tree under HPMP and traditional mode. Obviously, compared with the traditional mode in the same scenario, the peers using HPMP to find their favorite channel is much faster than that of traditional mode users. Compared with SN2, 3 and 4 in HPMP, we observed that the smaller the depth of a preview tree is; the smaller average hop distance time we get. Moreover, we found that the average hop distance of SN1 is larger than SN3 at the same depth of preview tree.

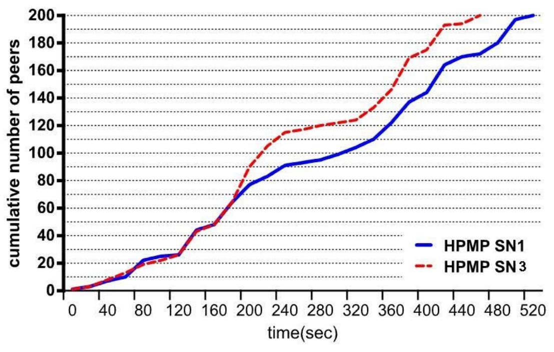

Fig. 7 Cumulative number of peers finding favorite channel on different upload bandwidth

Figure 7 shows the cumulative number of peers that find their own favorite channel between SN1 and SN3. The peers in SN3 will get their own favorite channels early than SN1. Because a preview tree that created by peers with higher upload bandwidth in SN1 will support more peers, it causes more waiting time during the process of peers joining (Algorithm 2) and makes a leader spend more time to find an appropriate position for insertion.

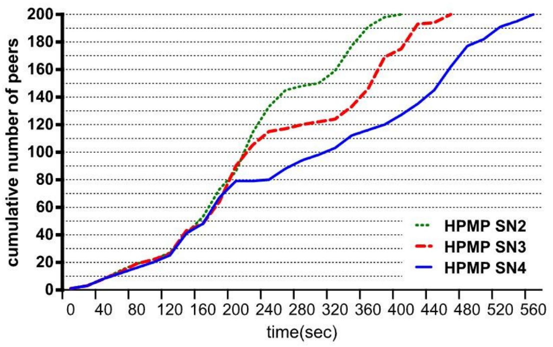

Fig. 8 Cumulative number of peers finding favorite channel on different depth of preview tree

Figure 8 shows the cumulative number of peers that find their own favorite channel between SN2, SN3 and SN4. We can find that peers in preview tree with less tree depth (like SN2) will reach their own favorite channels fast. Finally, the depth of preview tree tradeoffs between the average hop distance time and maintenance overhead.

IV. CONCLUSIONS

In this paper, we provide a novel method named HPMP to improve channel change time in P2P live media streaming. HPMP reduces the channel hopping time efficiently and decreases the tracker server's overhead. The simulation results demonstrate that the average hop distance of HPMP is much less than that of the traditional method. In addition, by using our system, the users can find the channel they want to watch quickly.

ACKNOWLEDGMENT

The authors would like to thank the National Science Council, Taiwan, R.O.C. for financially supporting this research under Contract No. NSC 99-2221-E-155-040.

REFERENCES

[1] C. Wu and B. Li, "Optimal Peer Selection for Minimum-Delay Peer-to-Peer Streaming with Rateless Codes" in Proc. ACM P2PMMS , Nov. 2005.

[2] M. Wang, L. Xu, and B. Ramamurthy, "Channel-aware peer selection in multi-view peer-to-peer multimedia streaming", In Proc. IEEE ICCCN, pp. 1-6, 2008.

[3] Won J. Jeon and Klara Nahrstedt, "Peer-to-peer multimedia streaming and caching service," in Proc. IEEE ICME, pp. 57-60, vol. 2, August 2002.

[4] P. Shah and J.-F. Paris, "Peer-to-Peer Multimedia Streaming Using BitTorrent," In Proc. IEEE IPCCC, pp. 340-347, 2007.

[5] PPLive, http://www.pplive.com/.

[6] PPStream, http://www.ppstream.com/.

[7] Y. Zhu, W. Liu, L. Dong, W. Zeng, and H. Yu, "High Performance Adaptive Video Services based on Bitstream Switching for IPTV Systems," in Proc. IEEE CCNC, pp. 1-5, 2009.

[8] C. H. Chang, P. J. Wu and C. N. Lee, "Smart-Fit: Peer-to-Peer Topology Construction Strategy for Live Video Streaming Toward Minimal Delay," in Proc. ICS, Nov. 2008.

[9] Windows Media Server, http://www.microsoft.com/windows/windowsmedia/

[10] Real Player Service, http://www.realnetworks.com/

[11] D. Banodkar, K. K. Ramakrishnan, S. Kalyanaraman, A. Gerber, and O. spatscheck, "Multicast instant channel change in IPTV systems," in Proc. COMSWA, pp. 370-379, 2008.

[12] M. Zaharia and S. Keshav," Gossip-based Search Selection in Hybrid Peer-to-Peer Networks," in Proc. IPTPS, 2006.

[13] OMNET++, http://www.omnetpp.org/