Proceedings of the 2013 Federated Conference on
Computer Science and Information Systems pp. 627–633

A SCORE-BASED PACKET RETRANSMISSION APPROACH FOR PUSH-PULL P2P STREAMING SYSTEMS*

Muge Sayit, Erdem Karayer, Kemal Deniz Teket, Yagiz Kaymak, Cihat Cetinkaya, Sercan Demirci, and Geylani Kardas
Ege University, International Computer Institute, 35100, Izmir, Turkey
{muge.fesci, yagiz.kaymak, cihat.cetinkaya, sercan.demirci, geylani.kardas}@ege.edu.tr, {erdemkarayer, denizkemal}@gmail.com

ABSTRACT

In this paper we propose an inference based packet recovery technique which considers past scores indicating retransmission success of the peers. Past scores are calculated by considering several parameters such as requested packets availability and round trip time. The importance of packets to be retransmitted is also considered in the proposed model. In order to obtain comparable results, we also implement a different retransmission approach similar to the models proposed in the literature. The ns3 simulations show that retransmission model increases the Peak Signal to Noise Ratio (PSNR) value even under high peer churn and limited resource index. Furthermore, score-based approach provides a decrease in reset counts and the number of duplicate packets, when it is compared to different retransmission approaches.

Keywords - peer-to-peer networks, live video streaming, packet recovery

1. INTRODUCTION

Peer-to-peer (P2P) video streaming systems represent one of the applications having a huge effect on the network traffic and there are remarkable live streaming systems proposed in the literature for both wired and wireless networks. Although most of these applications fall into two main categories as push-based and pull-based, several hybrid systems bringing the advantages of these two approaches have reported their success [1, 2]. In pull-based systems the nodes in the system send and receive video data chunks from one or more nodes [1] whereas each node has one parent and one or more children node in push-based systems [3]. In hybrid push-pull-based streaming; video data are divided into substreams. Nodes in the system connect one or more parents to receive these substreams and play video by combining them. During streaming, buffer maps, indicating the received blocks of each substream periodically, are exchanged between partner nodes in order to detect congestion and determine the candidate parents in case of parent re-selection.

We implement a hybrid push-pull system having the main properties of CoolStreaming [1] as to be used as an underlying framework. Although such systems have remarkable success, the system performance may degrade if there are not enough special-aimed nodes such as Content Delivery Network (CDN) nodes or super peers in the system. In this case, stream can be supported via retransmission of important packets. One may discuss if parent selection algorithms proposed for pull-based systems can be implemented in the selection of node which will send retransmission packets. Nevertheless, retransmission or packet recovery has more limited time to receive the requested packets when compared to the required time to receive packets from parents during streaming. The reason is that the packets received from the parents are generally obtained before their playout time since they are kept in buffer for a while whereas buffering time for retransmitted packet is relatively small. Since requested packets must be received in short time period, the selection of node sending retransmission packets should be realized considering more constrained time period.

In this work, we propose a new packet recovery technique based on retransmission. The contributions of this paper can be listed as follows: (*i*) We propose a pull-based retransmission model designing as to run combined with hybrid push-pull systems. It is shown that the performance of the hybrid system is significantly improved under high peer churn with the proposed model. (*ii*) The selection of the retransmission packets is based on the frame type and the selection is done in the sender side. (*iii*) Proposed model considers past behaviors of the peers, and the algorithm has low computational complexity and easy to implement.

The rest of the paper is organized as follows: In section 2, we give the summary of the related work. In section 3, we introduce the main contribution of this paper, namely score-based retransmission approach. The simulation results and conclusion are given in section 4 and 5, respectively.

2. RELATED WORK

Packet recovery techniques for missing packets can be classified into two categories, namely Forward Error Correction (FEC) and retransmission. FEC algorithms can

*This work is funded by the Scientific and Technological Research Council of Turkey (TUBITAK) Electric, Electronic and Informatics Research Group (EEEAG) under grant 111E022.

recover lost packets if the necessary number of packets is received. However, these techniques may not be useful in case of burst packet lost [4]. Furthermore, finding optimal FEC redundancy rate is a difficult problem due to the unestimated packet loss nature of P2P video streaming applications [5]. With the usage of retransmission techniques, there is no need to use redundant recovery packets introduced by FEC algorithms. Although there are well-known retransmission algorithms such as Automatic Repeat Request (ARQ) for classical server-client model [6], there are also some approaches proposed in the literature for retransmission in P2P video streaming systems [7, 8]. In [9], lost packets are obtained by retransmission at each hop from source to destination in the application overlay. This approach may cause retransmitted packets to reach lately to the destination nodes especially in leaf positions. A model in which parent nodes deciding retransmit some packets according to received NACK from children is proposed in [10]. Nevertheless, retransmission of missing packets from the same node, i.e. from the parent as proposed in [9, 10] has some disadvantages in case of parents do not have missing packets. In this case, requesting retransmission from a node other than the parent node provides higher continuity index [5, 11]. In [11], while selection of the node to request lost packets, the availability of the requested packets is not considered since this information is not available to the nodes in hybrid push/pull system. However, in pull based P2P video streaming systems, nodes have the information about the chunk availability and the selection of nodes to request lost packets can be done by taking into account this information [5]. On the other hand, when considering the high success of pull-based P2P video streaming systems, requesting missing packets from a set of nodes may improve the performance of the system. In [12], authors proposed a model for selecting this set of nodes. This selection is done by considering two criteria, the distance in terms of Round Trip Time (RTT) between requester node and the node retransmitting packets, and the position of the node retransmitting packets in the multicast tree.

This paper presents a work based on recovery of missing packets via pull-based retransmission. We prefer to use the term “retransmission parent” for the node which sends retransmission packets. Our study differs from the literature in several dimensions. First, we select more than one retransmission parents by considering the packets that they have. Second, after retransmission parent selection, we implement packet selection and give more priority to the retransmission of packets carrying I frames. Furthermore, we also propose a new algorithm to select candidate retransmission parents by considering their past behaviors, which provides increase in Quality of Experience (QoE) in terms of reset counts and decrease in message complexity.

In order to evaluate the performance of the proposed algorithm and to give the comparable results, we also implement a different retransmission approach similar to related studies proposed in the literature. The obtained results show that the performance of the proposed method exceeds the performance of the state-of-the-art solutions in terms of continuity index and total number of reset counts, as will be discussed in the following sections of this paper.

3. Score-Based Retransmission

We used substream based, i.e. hybrid push-pull-based P2P live video streaming system as an underlying framework since it is reported that users can obtain higher continuity index than that of users in pure pull systems [1]. As mentioned before, each node in the system connects one or more parents to receive substreams in hybrid push-pull-based streaming. After joining the P2P system, the nodes obtain the list of online nodes in the system by communicating with the tracker server and construct partnership table by selecting a subset of these nodes. Parents of the nodes are selected from partnership table and parent selection is done according to the buffer maps of the partners, in other words, candidate parents. Each buffer map indicates the latest received packet of the related substream, thus if the video data are partitioned into the k substreams, buffer maps are the k-dimensional vectors. Buffer maps are periodically exchanged between partner nodes in order to watch the buffer condition of the partners. We embed additional data such as playout index, i.e. the current position of the video and the buffer indicating lost and existing packets between playout index and latest received packet into these messages.

In order to recover lost packets, a node must obtain them before their playout time. After streaming started, each node tries to recover lost packets by requesting them from their partners. An example scenario showing the buffer of a peer and the packets received from the substream parents and retransmission parent is given in Fig. 1. According to the figure, the peer is subscribed four parents to obtain four substreams. The packets numbered with 55 and 60 are not received from the parents hence these can be received from the retransmission parent. Note that 52^{nd} packet is not received and will not be requested from the retransmission parent since the playout time is passed for this packet. In packet recovery process two selections are made: selection of which partner(s) to request retransmission and selection of which packet(s) to request.

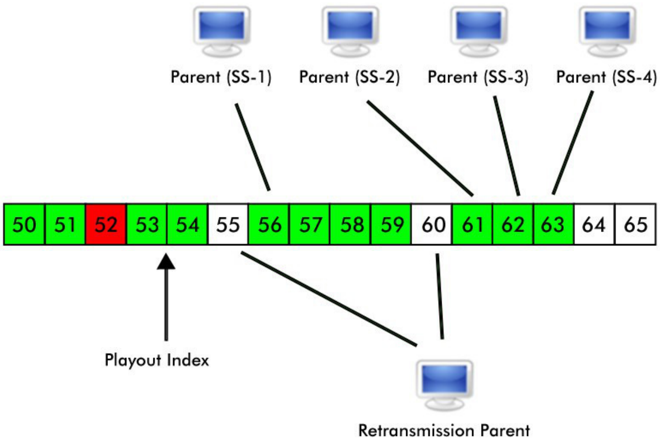

Fig. 1. An illustration of a retransmission scenario.

For retransmission parent selection, we propose a new approach based on previous scores of partners. A partner score is determined as the success ratio of received requested packets. Parent selection algorithm for determining candidate partners to request lost packets is given in Fig. 2.

```
S_requested_packets : the set of packets which will be retransmitted

for each substream k
  for each node p_i in partnership table
    fullness_p_i = number of requested packets /
                   number of requested packets existing in p_i's buffer;
  end for
  normalize fullness and RTT for all pi;
  for each node p_i in partnership table
    expected_score_p_i = α.fullness_p_i + (1-α).(1/RTT_p_i);
    expected_score_p_i = β.expected_score_p_i
                   + (1-β).(previous_score_p_i);
  end for
  //rank all p_i according to their expected scores
  construct γ(p); // ranked list
  while |S_requested_packets| > 0
    p_first = first partner in γ(p);
    S_p = the set of existing packets within requested packets
                                   in p_first's buffer;
    add packets in S_p to the list representing requests from p_first;
    S_requested_packets = S_requested_packets – S_p ;
    calculate expected scores for new set of request packets and
                                   construct γ(p);
  end while
end for
```

Fig. 2. Selection algorithm for retransmission parent.

Selection algorithm for retransmission parent starts with assigning fullness ratio of each partner in the partnership table. This value is calculated by examining the buffer map messages received from partners. In the next *for* loop, expected score for all candidate retransmission parents are calculated by considering three parameters, the fullness, the RTT value and previous expected score of the partners. Note that RTT is an important parameter to consider since requested packets need to be received before playout deadline. In the second step of the calculation of expected score, past scores of the partners is evaluated since even if a partner has high fullness and low RTT value, it may have limited available upload bandwidth or it may not be a stable node. For partners whose previous score is not calculated yet, this value is given as 0.5. After expected scores of each partner are calculated, the ranked list according to these values, $\gamma(p)$, is constructed. Requested packets which also exist at the first partner in the ranked list are put in a list representing the requested packets from the first partner in $\gamma(p)$. After the set of requested packets is reconstructed again, the expected scores are recalculated by considering new set of requested packets and $\gamma(p)$ is re-ranked. This process continues until retransmission parents are selected for all requested packets or remaining requested packets does not exist in the buffer of any partner.

Since the size of the partnership table is limited and quite small when compared with the size of the system, the space and the computational complexity of the algorithm given in Fig. 2 are negligible.

After request messages are sent, requester node waits for a period to receive the packets and to evaluate the performance of the retransmission parent for this session. This period is completed if the video playout index reaches to the playout time of the requested frame. In each buffer map exchange, retransmission parent selection is done again for the previously requested and not received packets. But, requester nodes may select the same retransmission parent with high probability since previously selected parent is not evaluated yet and still have the highest score. There are three reasons for this choice of evaluation period, in order to evaluate the performance of the retransmission parent fairly, to decrease the number of request messages and to prevent the number of duplicate packets. With the completion of this period, the success ratio S_R is calculated by dividing the number of received packets to the number of requested packets and previous score of the partners is updated by the smoothing function given in (1).

$$previous\_score = \lambda.S_R + (1\text{-}\lambda).previous\_score \quad (1)$$

Since the loss of I or P frames causes more distortion on display, sender node gives more priority to the packets containing I and P frames. In order to make nodes to detect which packet carries a part of an I or P frame, video server marks the packets according to its type and each node matches the received packet to its frame type.

4. SIMULATION

The simulations are implemented on ns3 [13] with the networks consisting of 50, 100, 150, 200 and 300 nodes. All the topologies used in simulations are generated randomly with BRITE topology generator [14] and Barabasi model [15]. Simulations are repeated several times in order to obtain averaged performance results.

There are one video server and one tracker server in the system. The video server has an upload capacity that provides the server to handle 20 percent of all peers in the system. Besides, we do not employ any CDN or super peer to support the video data dissemination. However 10 percent of the peers are selected as robust nodes which have higher bandwidth and tend to stay longer in the system. The tracker server is assigned to serve as the entry point of the streaming system. All newly joined peers connect to the tracker server first to request a random list of online peers to establish their initial connections.

In our simulations we use Foreman video sequence having QCIF resolution. The video is looped several times as to be used in 30 minutes length of simulations and encoded at 300 Kbps. We use frame copy as an error concealment method for missing frames. The original stream is divided into 4 substreams in each simulation in order to increase the potential suppliers of video. An exponential distribution which has an expected value of 1000 seconds for online and 400 seconds for offline periods is used to generate on/off intervals. Furthermore, we employ 10 percent of the nodes as free riders. The cumulative upload bandwidth distribution of all peers is given in Table I. We choose to send all messages including control and video over TCP in order to be able to connect all users even located behind a firewall. Retransmission requests are embedded to buffer map exchange messages, hence requests messages do not cause an extra load.

TABLE I. THE CUMULATIVE UPLOAD BANDWIDTH DISTRIBUTION

Percentage	Upload Bandwidth
10%	<50 Kbps
50%	<300 Kbps
90%	<1000 Kbps
100%	<3000 Kbps

In order to compare the performance of the proposed approach with the performance of other studies proposed in the literature, a different approach is also implemented for retransmission of missing packets. As proposed in [5], the selection of the retransmission parent is done randomly among the partners having requested packets and among the partners located close to the server, i.e. the position in the tree [12]. In this approach, if the node cannot receive the requested packets, it concludes that the current retransmission parent has not sufficient upload capacity and requests these packets from another node at the next buffer map exchange period. For this reason, we prefer to use the term "greedy-based" for this approach. In the simulations, buffer map messages are exchanged in every 2 seconds, hence if the requester node cannot receive retransmission packets within 2 seconds; it repeats the requests whereas in score-based retransmission, the nodes do not re-send request messages with high probability. Both greedy and score-based approaches give priority to the packets carrying I and P frames for fair comparison.

We measure PSNR, continuity index and total reset count as the video quality metrics. If a node consumes all data in its buffer, stops to receive video packets and cannot find suitable parent to connect, it resets itself, in other words, leaves the system and then joins immediately. The reset procedure causes the duration of the video playout for a while. As we observe, the time of this duration can change from 15 seconds to 20 seconds. The change on the average PSNR and the total reset count metrics are presented in Fig. 3 and Fig. 4 with respect to the change in network size. The PSNR of packet recovery techniques is higher than sole hybrid push-pull as expected. Although the greedy pull technique is slightly better than the score-based pull technique in terms of PSNR values, this difference is not easily observable at the end-user. In Fig. 4, an example frame received in hybrid push-pull, greedy-based and score-based approaches is given. As it can be seen from the figure, the distortion in the frame is similar in greedy and score-based approaches even the difference of the PSNR values of both approaches bigger than the difference of the averaged PSNR values given in Fig. 3.

The greedy pull technique has a higher control overhead and higher reset counts which can be seen in Fig. 5. The total reset counts are calculated by summing the number of resets of each node in the system. As it can be seen from the figure, the total reset count of proposed score-based pull packet recovery technique are always lower than sole hybrid push-pull and greedy pull packet recovery techniques except for 50 peers. Furthermore, resets of the system implementing score-based retransmission stay almost identical in network size of more than 100 nodes. The greedy pull packet recovery technique causes receiving the most part of the packets via retransmission. Since stream parents have not sufficient residual upload bandwidth due to the over-requested packets, hence this technique has the highest total reset count.

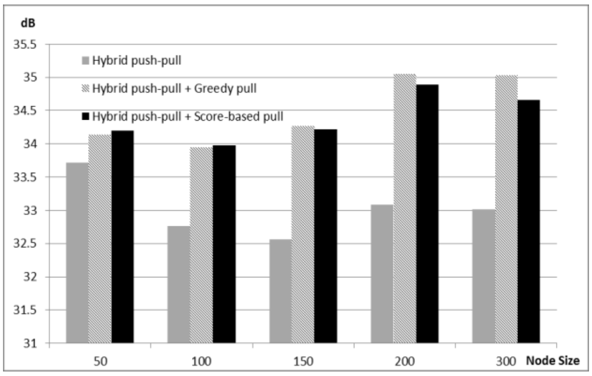

Fig. 3. Average PSNR

a. Original b. Hybrid push-pull (PSNR: 31.21 dB)

c. Greedy (PSNR: 35.30 dB) d. Score-based (PSNR: 34.66 dB)

Fig. 4. Example frames from the videos obtained in different P2P systems.

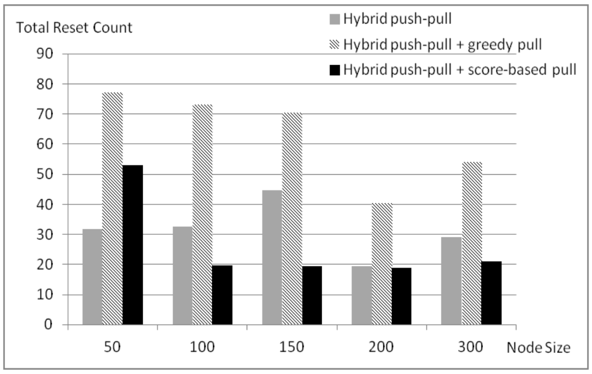

Fig. 5. Total Reset Count

In order to observe the inference performance of the score-based retransmission with respect to time, we measure two parameters, the ratio of the requested retransmission packets to the received retransmission packets and the change of continuity index, in Fig. 6 and Fig. 7, respectively. Since the greedy pull technique sends more request messages than the score-based pull, the score-based pull has higher reception ratio than the greedy pull technique on requested packets.

The graph given in Fig. 6 shows that the selection of retransmission parents is more successful in the score-based approach when compared to the greedy-based approach. This achievement provides the increase in continuity index by time as it can be seen in Fig. 7. In this figure, the graph shows that the inference process of large size of networks may be longer than that of small size of networks. Thus, it is expected that an increase in PSNR values for especially large size of networks such as network containing 300 nodes by time. Note that the most difference in PSNR values is observed between the greedy pull and the score-based pull for the network size of 300 nodes. This difference can be closed if the simulation lasts longer or if the nodes stay longer in the system. In Fig. 7, the observed change in continuity index for greedy-based retransmission shows a random pattern since it does not use any inference mechanism.

Finally, we give the comparative duplicate packet ratio values for greedy and score-based retransmission technique in order to indicate message complexity overhead in Fig. 8. The ratio is calculated by dividing the number of received packets to the number of requested packets. The graph shows that 50% of retransmitted packets are duplicate packets in the greedy-based approach.

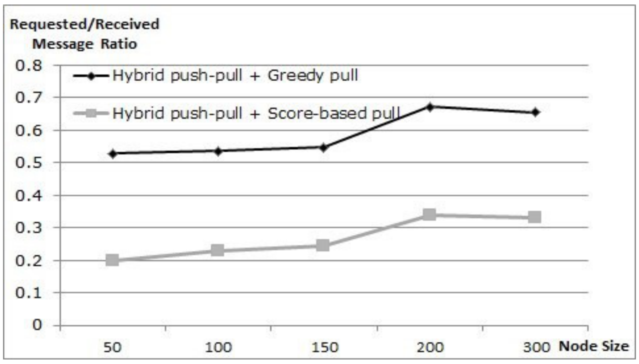

Fig. 6. Requested/Received Message Ratio

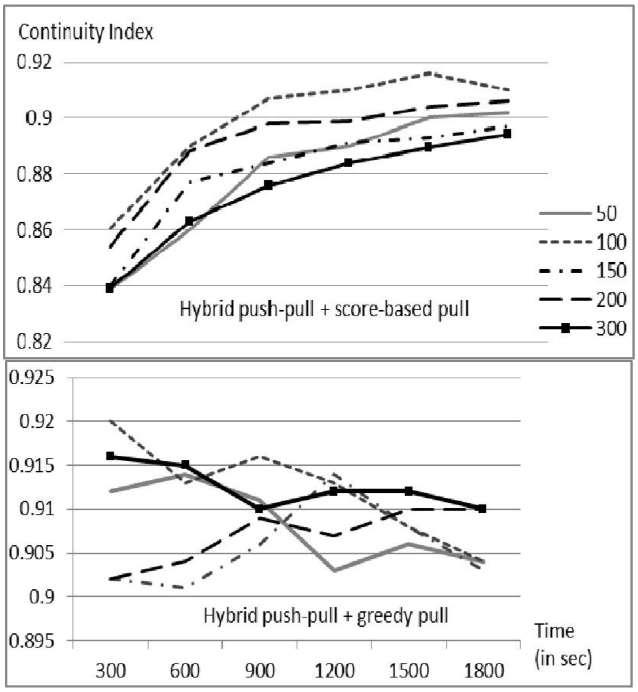

Fig. 7. Continuity Index

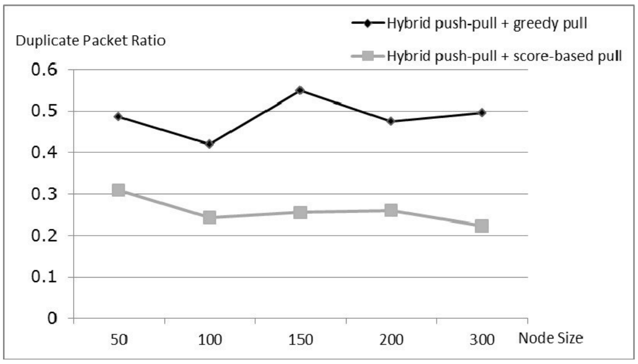

Fig. 8. Duplicate Packet Ratio

5. CONCLUSION

In this paper, we have introduced a new packet recovery technique based on retransmission for hybrid push-pull P2P live video streaming system. For selection of retransmission parent, an inference based approach has been discussed. The proposed retransmission strategies provides an increase in QoE parameters in terms of PSNR, total reset count and continuity index. Furthermore, the score-based retransmission approach achieves PSNR values similar to the greedy-based retransmission approach while the number of duplicate packets is relatively small. It is shown that the performance of the score-based approach increases by time. With the help of node clustering, it is also possible to improve the effectiveness of the proposed inference-based approach in P2P systems consisting high number of nodes.

We plan to implement the proposed score-based retransmission approach for the live P2P video streaming systems using different type of video codecs as our future direction. In this case, for example, some additional parameters such as packet priority or layer information may be considered during the selection process of the retransmission parent for scalable video codec.

6. REFERENCES

[1] S. Xie, B. Li, G.Y. Keung, and X. Zhang, “CoolStreaming: Design, Theory, and Practice,” IEEE Transactions on Multimedia 9, pp. 1661-1671, 2007.

[2] Wang F., Xiong Y., and Liu J, “mTreebone: A Collaborative Tree-Mesh Overlay Network for Multicast Video Streaming,” IEEE Transactions on Parallel and Distributed Systems, vol. 21, pp. 379-392, 2010.

[3] M. F. Sayıt, T. Tunalı, and A.M. Tekalp, “Resilient peer-to-peer streaming of scalable video over hierarchical multicast trees with backup parent pools,” Elsevier Signal Processing: Image Comm., vol. 27, pp. 113-125, 2012.

[4] M.-F. Tsai, C.-K. Shieh, T.-C. Huang, and D.-J. Deng, “Forward-Looking Forward Error Correction Mechanism for Video Streaming over Wireless Networks,” IEEE Systems Journal, vol. 5, no. 4, December 2011.

[5] H. Wehbe, G. Babonneau, and B. Cousin, “Fast Packet Recovery for PULL-Based P2P Live Streaming Systems,” The Second International Conference on Advances in P2P Systems (AP2PS'10), October 25 - 30, 2010, pp. 7-13, Florence, Italy, 2010.

[6] L. Rizzo, “Effective Erasure Codes for Reliable Computer Communication Protocols,” ACM SIGCOMM Computer Communication Review, vol. 27, pp. 24–36, 1997.

[7] E. Setton, P. Baccichet, and B. Girod, “Peer-to-Peer Live Multicast: A Video Perspective,” Proceedings of the IEEE, vol. 96, no. 1, pp. 25–38, 2008.

[8] O. Abboud, K. Pussep, A. Kovacevic, K. Mohr, S. Kaune, and R. Steinmetz, “Enabling resilient P2P video streaming: survey and analysis,” Multimedia Systems, pp. 1-21, 2011.

[9] B. Akabri.; H. Rabiee, and M. Ghanbari, "Packet Loss Recovery Schemes for Peer-to-Peer Video Streaming," Third

International Conference on Networking and Services (ICNS), IEEE Computer Society, pp. 94, 2007.
[10] P. K. Hoong and H. Matsuo, “Push-pull incentive-based P2P live media streaming system,” WTOC, vol. 7, no. 2, pp. 33-42, February 2008.
[11] C. Liu, K. Wang, and Y. Hsieh, "Efficient push-pull-based P2P multi-streaming using application level multicast," Personal Indoor and Mobile Radio Communications (PIMRC), 2010 IEEE 21st International Symposium, pp. 2586-2590, 26-30 Sept. 2010.
[12] X. Jin, H.-S. Tang, S.-H. G. Chan, and K.-L Cheng, "Deployment Issues in Scalable Island Multicast for Peer-to-Peer Streaming," IEEE Multimedia, vol. 16, no. 1, pp. 72-80, January-March 2009.
[13] Network Simulator 3, http://www.nsnam.org.
[14] “BRITE Topology Generator”, http://www.cs.bu.edu/brite.
[15] A.L. Barabasi and R. Albert, “Emergence of scaling in random networks,” Science 286, pp. 509–512, October 1999.