

A Feedback Control Model based Hierarchical VoD System

Jiali You*, Xiaolin Li**, Jun Song**, Jinlin Wang*

*National Network New Media Engineering Research Center, Institute of Acoustics,
Chinese Academy of Sciences, China

**University of Chinese Academy of Sciences, China

{youjl, lixl, songj, wangjl}@dsp.ac.cn

Abstract—P2P video-on-demand(VoD) system shows its advantage in network scaling. In VoD systems, how to organize the peers and how to schedule data among peers are popular research topics. However, the relationship between these two parts and whether the service quality affects the overlay structure is not described exactly. In this paper, we consider the overlay organization and data scheduling process as two different layers, and design a hierarchical VoD system. In addition, a novel feedback control model is proposed to link these two layers, in which the history service information of nodes can be used efficiently. Moreover, based on this model, the peers are prone to connect the peers who can provide them satisfying service. Simulation proofs that our feedback model can help the overlay organize the peers as a more efficient structure than before, in which the data is easy to be shared with each other. Compared with the system without the feedback information, our system shows that the server stress can be significantly reduced, and the starting delay is also improved obviously.

Keywords—P2P, video-on-demand, feedback, control model, peer selection

I. INTRODUCTION

Recently, Internet shows a powerful action for the real-time media distribution, which has been an important part of people's daily life. A lot of Websites, which provide media services, perform more and more popular than before. Currently, the traditional client/server(CS) structure and Peer-to-Peer(P2P) technologies are usually used. Compared with the C/S approach, P2P technologies present great scalability and service for millions of users worldwide. Thereby, it is deeply investigated for a lot of applications, and the P2P based approaches has already been proved to be useful in file sharing and real-time media distribution.

The P2P based real-time media distribution can be classified into two categories: live streaming and video-on-demand (VOD). In live streaming systems[1-5], the source servers broadcast live shows or TV programs, and all the clients play the content at a same pace. In each live program, the clients cannot interact with the system to decide which part of the program will be watched and all of them have the similar downloading targets with the same pace. Therefore, they can efficiently share their buffers to accelerate the downloading processes, which can alleviate the server stress significantly. A lot of successful examples prove that the P2P

technology is efficiently in live streaming systems, including PPLive[2], PPStream[3] and CoolStreaming[4]. Different from the live streaming, VOD[6-15] is an interactive multimedia service, whose clients can choose any preferred part to watch. In a VOD system, the users can operate the VCR controls, such as play, pause, fast forward, and rewind. The users who are watching the same part of a video can cooperate with each other by P2P technologies. Because when a VCR control is triggered, the expected content of a client is changed. The old cooperating peers may not contain the target content, and the peers should obtain the data from server or other peers. This process may incur extra stress on the server side and also bring the problem of how to get data efficiently from other peers. For easy of expression, in the rest of this paper, we use the terms "peer" and "node" interchangeably.

In P2P based live streaming systems, the peers watching the same program are organized as an overlay structure, in which the peers may help each other in real time. In VOD systems, the peers with the close playbacks are structured as a swarm. Because the users may watch different parts of a video, several swarms will be generated. When a VCR operation is incurred, the server of the VOD system may give the client some peers who can help this peer entry the target swarm, and an overlay maintaining algorithm will be implemented to update the neighbors of peers. From the above expression, we know that the neighbors are very crucial in P2P based media systems. In some research work [2, 15], the authors proved that the more peers in a same program, the better average performance will be obtained. However, in real applications, the amount of users in different programs is not the same, and the capability of the neighbors is not balance. If the neighbors cannot provide enough capability, like upload bandwidth or content, the node should request the server for the data block, which may lead to high server stress and non-fluent playing experience. To solve the above problem, in this paper, we design a framework for a VOD system which is a feedback model based hierarchical structure. This model can drive the overlay organization as a more efficient one than before, and the structure of the overlay may be helpful for the nodes in future. The simulation shows that our VoD system has the positive performance, and the resource utilization of the whole network is significantly improved. To summarize, the paper's main contributions include:

- A feedback control model based hierarchical structure is designed, which includes an overlay layer and a data scheduling layer. Besides, a feedback control model is used in the middle of the two layers to process the interaction between them. The data scheduling layer returns the service information of neighbors to the overlay layer. And the overlay layer selects the new nodes which may have the capability to supply the data based on the feedback. This structure can organize the nodes reasonable and make a benefit circle for the VoD system.
- In the overlay maintenance process, we maintain an extra routing table for each peer, which stores the information of the history service peers. In our system, routing table and history service node table are merged together into a node pool. Based on the feedback model, the neighbors are selected from this pool and return to the scheduling layer. The history routing table is a complement of the routing table in the overlay organization process, which may obviously improve the user experience of the VoD system.

The remainder of this paper is structured as follows. Section II describes the framework of our system and the corresponding maintenance process. The feedback control model is discussed in Section III. Section IV evaluates our framework and algorithm. Finally, conclusions and future work are drawn in Section V.

II. THE FRAMEWORK OF THE HIERARCHICAL VoD STRUCTURE

In our VoD system, a two-layer VoD system is designed, which includes overlay layer and scheduling layer. Moreover, a feedback control model between these two layers is proposed to modify the performance of the system gradually

A. The Framework of the VoD System

Figure 1 gives an overview of our VoD system, in which the whole process from user operation to media playing is presented.

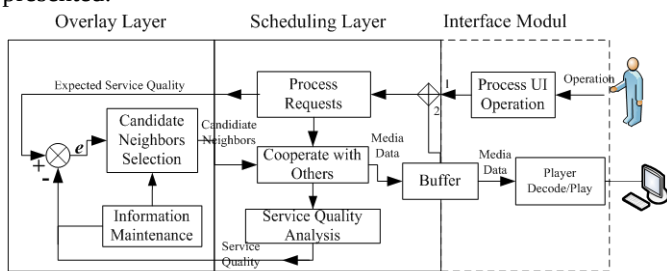


Figure 1. The interaction among different parts in the hierarchical VoD system

This figure is composed by two layers and one interface module. The effect of overlay layer is to organize the peers as a structure, in which peers are easy to route to different positions and cooperate with each other. In this layer, the information maintenance module is working to record the updated state of the peers. The purpose of the data scheduling layer is to share the data among peers who have close playbacks. In this layer, how to select the cooperating peer is

an important problem. Besides, some parameters related to the service quality of peers are observed, such as bandwidth and latency. In this VoD system, the use requests and media playing process are both processed in the interface module.

When a user clicks some video, the operation information will be send to the scheduling layer, in which the identifiers of the requested data blocks are calculated. Then, scheduling layer starts the scheduling process, and asks for the cooperation nodes from the neighbors that prepared by the overlay layer. If some neighbors have the needed data, the data scheduling algorithm will select the appropriate peer as a service providing node. If neighbors do not have some data block or cannot provide the data on time, a request will be send to the server.

In this structure, the overlay layer is to organize the nodes and provide the neighbors for the scheduling layer. And the scheduling layer tries its best to get the data from these nodes. Each peer has these two layers, and the information about service quality is transferred between them. To maintain the overlay structure, the overlay layer in each node should check whether its neighbors are still useful periodically. If some nodes cannot meet the need of the future scheduling requirement or cannot be available for a long time, a neighbor decision process will be evoked, and some new neighbors will be selected. In our paper, we use a feedback control system to model above process, which may help peers get more benefit from the appropriate neighbors. Details of the feedback control model will be introduced in next section.

B. The Maintenance of the Overlay Layer

In VoD systems, the video is divided into several sequential chunks in time domain, and the peers watching the same chunk are organized as a swarm. The purpose of the overlay layer is to organize the nodes with the close playing points as a swarm to cooperate and share the data with each other conveniently. Therefore, when a new node enters the system, it can get some node information that have the close playback to it from the bootstrap peers and share data among them.

In our system, the overlay layer can be analyzed in two aspects. One is how to route quickly in the network; the other is how to get good service quality among nodes. Thus, two routing tables are designed, including a routing list and a history service node list. In our system, the peers with the close playback structure a swarm by gossip protocol. Thus, the routing list is generated based on the gossip protocol and just used for routing in network. The history service node list records the nodes that have provided data to the current node in the history, and the corresponding service information is also in it. In the overlay layer, these two lists compose a node pool, and the neighbors for data scheduling will be selected from it. In this process, scheduling layer reports the service information periodically, which is used for deciding candidate neighbors by the feedback control model.

C. Scheduling Algorithm

In our system, pull-based scheduling algorithm[1] is used. The basic idea of this algorithm is that the data block should be transmitted from the neighbor node np to the current node p

only if p requests the data from np . Besides, rarest-first strategy is applied to queue the requests. This scheduling algorithm is a simple and basic one. In order to verify the efficiency of the feedback control model, we just use it here. Although there are some algorithms which can improve the performance of scheduling process, it is out of the scope of our paper

In addition, the played content is stored in the disk and used for the cooperation of neighbors. It cannot be deleted until the peer changes the channel or the reserved disk is full.

III. A FEEDBACK CONTROL MODEL BETWEEN TWO LAYERS

In the two-layer structure, the service quality of the history neighbors is returned to the overlay layer periodically. If the current neighbors cannot meet the expected need of the future scheduling, the old neighbors should be replaced by the new ones which are selected from the node pool that is build up by the routing table and the history service node table. Thus, a selection criterion is applied in overlay to decide which nodes should be selected as the new neighbors. Then, the new nodes are used for the data scheduling layer in the future. These steps are continually repeated. This process is similar to a closed-loop control system, in which the expected input and output signal of the system is the performance of data scheduling, and the error signal of the system means whether the candidate neighbors should be replaced. Therefore, we use a feedback control model to describe it.

The feedback control model is composed of two loops like figure 2. The outer loop is transfer the playing quality from the scheduling layer, which is used to decide whether the neighbors should be updated. The other information of neighbors is updated in the inner loop control, which participates calculating the new neighbors in the selection process. Therefore, the new candidate neighbors are decided based on this feedback model.

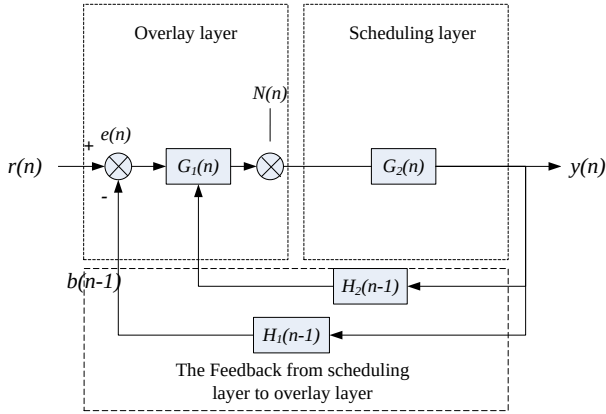


Figure 2. The feedback control model

In figure 2, $r(n)$ is the expected playing bit rate and the $y(n)$ is the real one in the observation time n . $G_1(n)$ and $G_2(n)$ are the process of neighbors selection and data scheduling, separately. $H_1(n)$ and $H_2(n)$ are the feedback process. And $N(n)$ is the noise induced by overlay layer. In the real system, the feedback control process is as follows.

1. In our system, we define $r(n) = \begin{cases} 1 & r(n-1) = br \\ 1+\Delta & r(n-1) > br \end{cases}$,

where br is the bit rate, $r(n-1)$ is the average requested bit rate of the last round scheduling and Δ is a parameter to reflect the buffer filling speed.

2. Then, $r(n)$ is compared with $b(n-1)$. Here, $b(n-1)$ is used to represent the average received bit rate of the last round scheduling, which is returned by the feedback strategy of the scheduling layer. Based on the result of the comparison, $e(n)$ can be obtained as formula (1). Here, $e(n)$ is defined as an error signal to represent whether some neighbors should be replaced or not.

$$e(n) = f(r(n) - b(n-1)) \parallel C(c_{last} - c_{current})$$

$$s.t. \ f(x, y) = \begin{cases} 1 & x - y \geq \rho \\ 0 & otherwise \end{cases} \quad (1)$$

$$C(x) = \begin{cases} 1 & x \neq c_{degree} \\ 0 & otherwise \end{cases}$$

where ρ is a system parameter; c_{last} and $c_{current}$ mean the number of neighbors in the last scheduling round and current time separately; c_{degree} is the maximum degree of each node. From the above formulas, if $e(n)=0$, the neighbors should not be changed; otherwise, the neighbor list should be replaced or filled by the new ones.

3. After the $e(n)$ is obtained, the $G_1(n)$ process will be evoked, whose target is to prepare the neighbors for the next scheduling round. If $e(n) = 0$, the neighbors are not changed and just their information, like service ability, should be updated. If $e(n)=1$, the new candidate neighbors should be selected from the node pool of the overlay layer to replace the old neighbors who cannot satisfy the needed of the scheduling layer. This process is the key point in our system. Several factors are considered. And formula(2) is used to calculate the priority of the nodes. All nodes in the node pool and the old candidate neighbor list should be sorted by (2), and formula(3) is applied to select the appropriate candidate nodes.

$$f_{oi}(b_i, d_i, s_i) = \beta_1 b_i + \beta_2 d_i + \beta_3 s_i + \beta_4 q_i$$

$$s.t. \ \sum_{k=1}^4 \beta_k = 1 \quad (2)$$

$$\max \sum_{i=1}^{c_{degree}} f_{oi}(b_i(n-1), d_i(n-1), s_i(n-1))$$

$$\sum_{i=1}^{c_{degree}} b_i(n-1) \geq B_D \quad (a) \quad (3)$$

$$s.t. \ \forall i \leq c_{degree}, d_i(n-1) < D_R \quad (b)$$

$$\forall i \leq c_{degree}, s_i(n-1) > \varepsilon \quad (c)$$

Here, b_i is the bit rate that provided by node i in the last scheduling round; d_i is the reciprocal of the delay between current node and the node i ; $s_i = c_i / c_{degree}$, in which c_i is the degree of the node i and c_{degree} is a pre-set value; q_i is the proportion of the content reserved by node i . Here,

c_{degree} is the number of the target selected candidate nodes. B_D , D_R and ε are the pre-set threshold of the different parameters.

In this step, candidate nodes are decided, which are the potential data providers in the scheduling of next round. Each node estimates the data providing capability based on its history experience. Moreover, default of the δ value indicates the system disturbance error, i.e. $N(n)$, which may be caused by the experience measuring process. After this process, some or all old neighbors may be replaced.

4. Then, the selected nodes are the input of $G_2(n)$ which is the process of data scheduling. In this paper, pull-based algorithm is used, which is introduced in section II.

5. $H_1(n)$ and $H_2(n)$ is the feedback process that the statistic information of the last scheduling round is returned to the overlay layer.

From the above process, several factors, such as bandwidth and delay, are considered in the overlay organization process, which can avoid the nodes that just provide unstable service selected as the neighbors.

IV. SIMULATION AND ANALYSIS

A. Simulation Setup

To verify the effect of our feedback model, we design an experiment to simulate the different situations. In VoD systems, some research proofs that the amount of the users follows the poisson distribution. Therefore, in our paper, we design a VoD system to simulate the algorithm. In this system, 1000 peers enter a channel with $\lambda = 1$. The bit rate of the program is 500kbps. All nodes are divided into four parts. One part contains 15% nodes with 1Mbps upload bandwidth; one part is 15% nodes with 512kbps upload bandwidth; and the others are 30% 384kbps and 40% 256kbps upload bandwidth nodes. In our simulation, we assume that the download bandwidth is not the bottleneck of the system. And the delay among peers is from a network log which is tested in the Internet. The length of the program is 2400s, and we observe the performance until 3000s.

B. Performance of the System

In a VoD system, the server stress and the playing quality are two important parameters to evaluate the performance of the system. If a system has low server stress and satisfying playing quality, it can save a lot of bandwidth cost and may be a good solution. Therefore, we compare the server stress of two different systems in figure 3. And the playing quality is shown in figure 4. In our comparison, the original system means that no feedback model is used.

From figure 3, we found that after the feedback control model is used, the server stress can be reduced a lot. Moreover, the two lines of the playing quality are almost the same as figure 4 shown. The probable reason is that the feedback model can help the system to organize the overlay based on their capabilities. Therefore, it is easy to set the appropriate

nodes as the neighbors for each node, which can provide satisfying service quality for nodes.

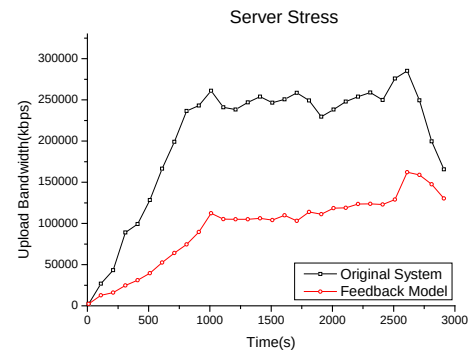


Figure 3. Comparison of the server stress of different systems

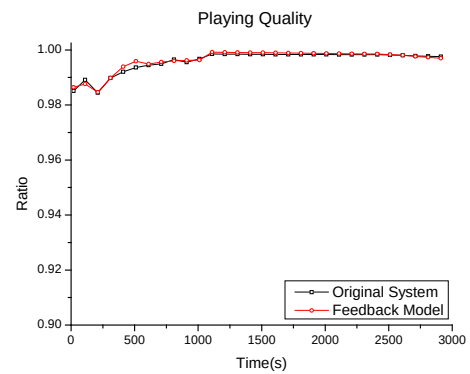


Figure 4. Comparison of the playing quality of different systems

C. Other Evaluation Metric of the System

Starting delay is usually used as another important metric to evaluate the performance of the VoD or living streaming systems. In our paper, when a node enters a channel, if the beginning part of the buffer is filled full, it will play the content. Here, 16s buffer should be filled before playing. This metric can evaluate the filling speed for scheduling buffer. Figure 5 shows the delay comparison of the systems with and without the feedback model.

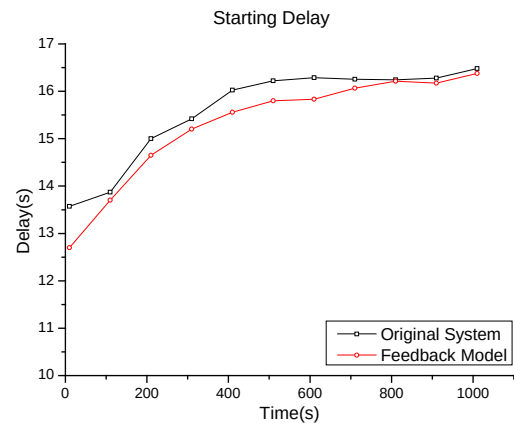


Figure 5. The comparison of starting delay from different systems

From this figure, we analyze that there are two reasons for the slightly lower starting delay than the original system. One is that the nodes which have some requested content and enough capability is prone to be selected. They can provide better service than the randomly selected neighbors by the overlay maintaining process, and accelerate the buffer filling process. The other is that the history service list can give the node some neighbors quickly, and it also saves the time in the node entering process.

D. The Impact of the Different Network Parameters

In the feedback control model, the key step is to select the neighbors from the node pool. In this process, nodes in the node pool should be sorted based on their history capability, which is introduced in section IV. To calculate the priority of each node, 4 factors are considered. However, how to weight the different factor may affect the results. To set appropriate parameters in our system, we also analyze how the parameters affect the performance of systems. Table I gives the different value of parameters. Figure 6 presents the average server stress with different parameters, which is the mean value from the time of the first node entering to the observing time. And figure 7 gives the starting delay of them.

TABLE 1. THE SETTING OF DIFFERENT PARAMETERS

	Proportion of Content	Last Round Bit rate	Delay	Degree
Main_Content	0.7	0.1	0.1	0.1
Main_Bitrate	0.1	0.7	0.1	0.1
Main_Delay	0.1	0.1	0.7	0.1
Main_Degree	0.1	0.1	0.1	0.7

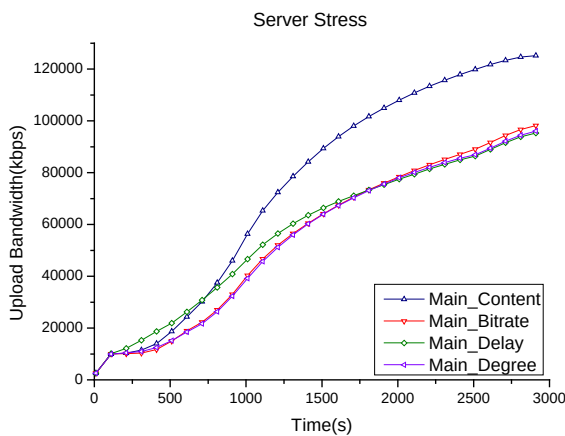


Figure 6. The average server stress from the systems with different parameters

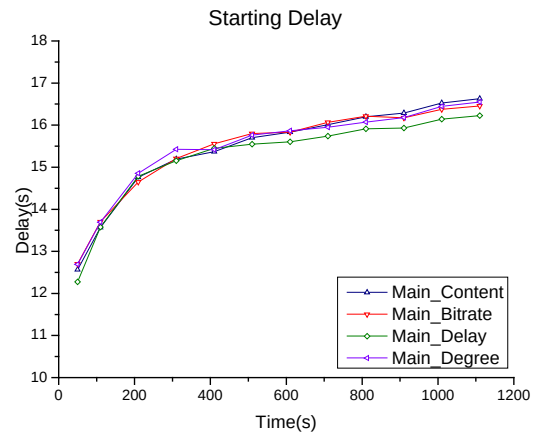


Figure 7. The starting delay from the systems with different parameters

From these figures, we found that the factors about last round bit rate and the degree are more important than the other factors. And the content similarity is not the most important one. Probably, although some neighbors have required content, their capability is too low to provide high quality service. Besides, if the weight of delay is large, the starting delay can obtain the best performance, which means that delay is an important factor to affect the user experience. Therefore, how to combination these different factors will affect the performance obviously and we will deeply investigate it in the future work.

E. Discussion

Figure 8 gives the average bandwidth utilization factor of all neighbors from different systems. We can see that after feedback model is used, the candidate neighbors is more useful than before, which means the resources of peers can be used more efficient in feedback model based system, and it is also a proof that why the server stress can be improved significantly. Besides, in the feedback control model based system, the occupation of the upload bandwidth is flatter than before, which can bring the nodes some benefit that some other applications can run stably on them synchronously.

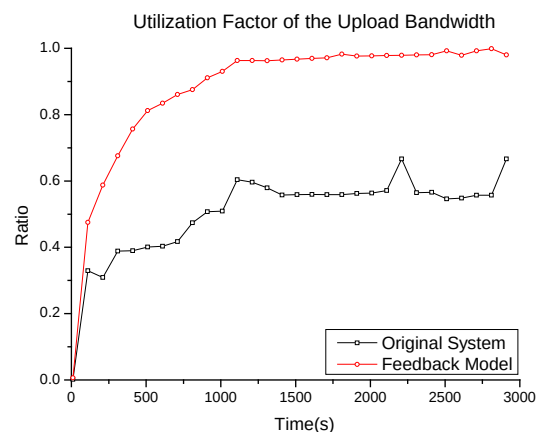


Figure 8. The comparison of the utilization factor from different systems

V. CONCLUSIONS

In this paper, a hierarchical VoD system is designed and a feedback control model is used to describe the interaction process between overlay and scheduling layers. In this model, a history service node list is reserved and several factors are considered to select the appropriate candidate neighbors for each peer. Based on this model, the peer is prone to find the neighbors that may bring benefit for its future scheduling. And the peer capability can be make the best use. To verify the efficiency of the feedback model, a popular program is simulated. The results show that after feedback model used, the server stress and starting delay are both achieve the satisfying performance. In this paper, we assume that all experiment parameters can be measured correctly. In the future work, we will research on the system performance when the noise cannot be avoided, and how to combine different factors in node selection step will also be investigated.

ACKNOWLEDGMENT

The work is supported in part of the National Science Foundation of China(60903218), 863 Program(2011AA01A102), Strategic Priority Research Program of the Chinese Academy of Sciences (XDA06010301) and the Innovation Project of Institute of Acoustics, Chinese Academy of Sciences(Y154211601).

REFERENCES

- [1] M. Zhang, Q. Zhang, and S. Yang, "Understanding the Power of Pull-Based Streaming Protocol: Can We Do Better?," IEEE Journal on Selected Areas in Communications, special issue on Advances in Peer-to-Peer Streaming Systems, Vol. 25, No. 8, December 2007.
- [2] L. Vu, I. Gupta, J. Liang and K. Nahrstedt, "Measurement and Modeling of a Large-scale Overlay for Multimedia Streaming," in the Fourth International Conference on Heterogeneous Networking for Quality, Reliability, Security and Robustness, 2007(QSHINE '07), No.3. <http://www.ppstream.com>
- [3] S. Xie, B. Li, G.Y. Keung, and X. Zhang, "Coolstreaming: Design, Theory, and Practice," IEEE Transactions on Multimedia, Volume 9 Issue 8, December 2007, pp.1661-167
- [5] N. Magharei, R. Rejaie, "PRIME Peer-to-Peer Receiver-driven Mesh-based Streaming," IEEE/ACM Transactions on Networking, 17, (4), pp. 1052-1065.
- [6] C. Dana, D.Li, D. Harrison and C.Chuah, "BASS: BitTorrent Assisted Streaming System for Video-on-demand", IEEE 7th Workshop on Multimedia Signal Processing, pp.1-4, 2005.
- [7] A. Bhattacharya, Z. Yang and D. Pan, "COCONET, Co-Operative Cache Driven Overlay Network for P2P Vod Streaming," Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, 2009, vol.22, Part 1, pp. 52-68.
- [8] X. Qiu, C. Wu., X. Lin, and F.C.M.Lau, "InstantLeap, Fast Neighbor Discovery in P2P VoD Streaming," in proceedings of the 18th international workshop on Network and operating systems support for digital audio and video, pp. 19-24, 2009.
- [9] S. Annapureddy, S. Guha, C. Gkantsidis, and etc., "Is High-Quality VoD Feasible using P2P Swarming?," in proceedings of the 16th international conference on World Wide Web, 2007, pp. 903-912.
- [10] T. Zhang, J. Lv, and X. Cheng, "Mediacoop-Hierarchical Lookup for P2P-VoD Services," international Conference on Parallel Processing(ICPP), pp. 486-493, 2009.
- [11] Y. Guo, K. Suh., J. Kurose, and D. Towsley, "P2Cast Peer-to-peer Patching Scheme for VoD Service," in Proceedings of the 12th international conference on World Wide Web, pp.301-309, 2003.
- [12] T. T. Do, K.A. Hua, and M.A. Tantaoui, "P2vod Providing Fault Tolerant Video-on-demand Streaming in Peer-to-peer Environment," IEEE International Conference on Communications, 2004, vol. 3, pp. 1467-1472.
- [13] Y. He, Y. Liu., "VOVO,VCR-Oriented Video-on-Demand in Large-Scale Peer-to-Peer Networks," IEEE Transactions on Parallel and Distributed Systems, April, 2009, Vol. 20, issue 4, pp. 528-539.
- [14] S. Annapureddy, C. Gkantsidis, and P. Rodriguez, "Providing Video-on-Demand using Peer-to-Peer Networks," In Proc. Internet Protocol TeleVision (IPTV) workshop in conjunction with WWW '06 (2006).
- [15] C. Huang, J. Li and K.W. Ross, "Ross: Can Internet Video-on-Demand be Profitable?," SIGCOMM, pp. 133-144, 2007.