High-Quality P2P Video Streaming System Considering the Cooperation of Constitution Information and Delivery Status

Yohei Okamoto†, Yosuke Tanigawa† and Hideki Tode†
†Graduate School of Engineering, Osaka Prefecture University
1-1 Gakuen-cho, Naka-ku, Sakai, Osaka, 599–8531, JAPAN
Email: {okamoto@com., tanigawa@, tode@}cs.osakafu-u.ac.jp

***Abstract*—Recently, video streaming services using P2P(Peer-to-Peer) have attracted attention to solve the problem of load concentration on servers and reduce a lot of latency. Many P2P streaming systems, like Coolstreaming, however, have a complicated approach to control playback timings severely. And this leads to less churn resiliency and less adaptability to fluctuation of network traffic. Therefore, we focus on a simple and robust approach to realize "pseudo" streaming with high quality, which is based on BitTorrent. In the existing methods with the simple approach, peers download pieces just closer to playback timings to decrease the playback discontinuity. In contrast, in this paper, we propose a new P2P video streaming system considering the cooperation of three metrics; video structure, playback timings, and pieces dispersion on network. In this system, users vary three piece selections to suit the delivery status. Specifically, users preferentially download pieces which affect the video quality, are closer to playback timings, and improve the delivery efficiency. Moreover, we show the effectiveness of the proposed method by computer simulation.**

I. Introduction

In recent years, video delivery (e.g., progressive download like YouTube [1] and video streaming) becomes prevalent widely with the diffusion of broadband network and the advance of PC performance. There are a lot of researches targeted on the video streaming delivery. Specifically, P2P(Peer-to-Peer) video streamings have attracted attention [2][3] to solve load concentration problems on servers and reduce a lot of latency. P2P video streaming systems are roughly classified into two types [4]; tree-based type, such as CoopNet [5] and Peercast [6], and mesh-based type, such as Coolstreaming [7] and Chaimsaw [8].

Coolstreaming [7] is one of typical P2P video streaming distribution systems and used widely in current video streaming with the high performance of distribution. However, Coolstreaming takes a severe approach to playback timings, and has a complicated method to adjust the timings. Moreover, in principle, Coolstreaming tends to have a low tolerance for churns.

Our approach realizes a simpler and more robust "pseudo" streaming service with high quality. Specifically, we construct the high-quality streaming system based on a simple BitTorrent which can manage the index information easily and is very widespread in P2P. Based on a similar concept, there are several existing methods [9][10] that adapt BitTorrent to video streaming. In these methods, peers select pieces just closer to the playback timings preferentially. As a result, the playback discontinuity decreases. However, these methods do not consider the constitution of video structure. To realize a high-quality video streaming, it is required to consider the video structure, playback timings, and pieces dispersion on network cooperatively.

Therefore, in this paper, we propose a new P2P video streaming system. In the proposed method, peers preferentially receive high-priority pieces considering video structure information in H.264/AVC to decrease video image degradations by the piece losses. In addition, to decrease playback discontinuity, peers receive pieces closer to the playback timings by limiting the range of piece selection. The range is also decided autonomously and asynchronously based on video structure, which differs from the existing methods. On the other hand, when peer requests concentrate on rare pieces of specific peer, download efficiency decreases. Therefore, peers also receive rare pieces of network in cooperation with the above preferential policies about video structure and playback timings. Finally, we conduct the performance evaluation of the proposed method with computer simulation and show the effectiveness.

The rest of this paper is organized as follows. We give a short introduction about the background technologies of video compression technique and P2P (BitTorrent), and review several related works in Section II. Section III describes the proposed method for P2P video streaming. In Section IV, we conduct the performance evaluation with computer simulation. Finally, we conclude this paper in Section V.

II. Background Technologies and Related Works

A. Background Technologies

1) H.264/AVC: H.264/AVC [11][12] is one of video compression techniques with inter-frame prediction. In this prediction, current frame is predicted from the neighboring frames because the neighboring frames are similar with each other. Current frame can be predicted from the previous frame (forward prediction), the following frame (backward prediction), and the both of the frames (bidirectional prediction). The frame encoded with only forward prediction is called Predicted Frame (P-Frame), and the frame encoded with one of forward, backward and bidirectional predictions is called Bi-directional Predicted Frame (B-Frame). The frame encoded without inter-frame prediction is called Intra-coded Frame (I-Frame). Moreover, the group of frames with dependence for prediction is called GOP (Group Of Picture). In H.264/AVC, a specific frame loss affects not only the corresponding frame but also the predicted frames, and hence, error propagation occurs.

2) P2P Network: P2P [13] is a network model where member nodes, called "peer", communicate with each other in equal parts. In client-server model, when the request loads increase, servers become overloaded. On the other hand, in P2P model, each peer can operate as both of a client and a server according to the network situation. Therefore, even

978-1-4244-8790-5/11/$26.00 ©2011 IEEE

when the requests of the data delivery increase, peers can communicate based on the autonomous distributed processing.

P2P can be classified into Pure P2P and Hybrid P2P based on the management of information that associates each data with the corresponding key. In Pure P2P like Winny [14] and Gnutella [15], each peer maintains the information in a distributed manner. Pure P2P needs no management servers for the information, hence it has large scalability in terms of handling this kind of management information. However, each peer needs more loads for data searching. On the other hand, in Hybrid P2P like Napster [16] and BitTorrent [17], one or more specific management servers manage the information, and each peer concentrates on transferring content data. Thus, it is easy to manage the information.

3) BitTorrent: BitTorrent [17] is a file delivery application based on Hybrid P2P. In BitTorrent, an entire content file is divided into multiple small data called "piece", and peers transfer the pieces with each other.

In BitTorrent, a group of peers downloading the same content file are called "swarm" and construct one P2P network. In a swarm, there is a management server called "tracker" to manage several information about all peers. Each peer requests the tracker to search information about peers and informs the tracker about their download situation. Peers receive a list of other connectable peers from the tracker. Peers select several peers from the list to connect, and then, they download pieces to exchange each other.

The download order of pieces is very important for delivery efficiency. In BitTorrent, "rarest-first" is utilized as piece selection algorithm. In this algorithm, peers manage the status on piece acquisition about neighboring peers in the same swarm. Based on the status, peers preferentially download the rarest pieces which fewest peers have acquired. Rarest-first algorithm aims at dispersing as many pieces as possible on swarm. As a result, concentrated requests for a specific peer that has rare pieces can be prevented.

However, rarest-first algorithm considers only pieces' scarcity. Hence, when applying it to P2P video streaming directly, peers download pieces of video data randomly not considering the playback orders, and cannot receive the pieces close to the deadlines for playback preferentially. Therefore, rarest-first algorithm is unsuitable for video streaming.

B. Related Works

Recently, P2P video streaming has been popularly studied. Specifically, Vlavianos et.al. [9] proposed the method to decrease playback discontinuity. Oechsner et.al. [10] proposed to provide each user with suitable video considering their network bandwidths. In these methods, BitTorrent is applied as a P2P system because it can disperse pieces quickly on each user in addition to very practical and widely spread P2P system. As mentioned in Sec. II-A3, BitTorrent is not available directly for video streaming. However, BitTorrent has been modified to adapt to video streaming.

BiToS (Enhancing BitTorrent for Supporting Streaming Applications) [9] adapted BitTorrent to video streaming by improving piece selection algorithm in BitTorrent. Figure 1 shows the overview of piece selection algorithm in BiToS. In BiToS, based on the playback timings, unreceived pieces are classified into either a High-Priority Set or a Low-Priority Set. The High-Priority Set is fixed-length set composed of the pieces that are close to the playback timings and are needed to receive quickly. The Low-Priority Set is composed of the other unreceived pieces. Peers select pieces from the High-Priority Set with the probability P or from the Low-Priority Set with the probability $1 - P$. Then, based on rarest-first algorithm, peers select a piece in the selected set. When peers receive the piece in the High-Priority Set, they shift the closest piece in the Low-Priority Set to the High-Priority Set.

In this method, the pieces in each set are selected based on rarest-first algorithm. Hence, the pieces that are close to the playback timings but have been already received by many peers are seldom selected. Moreover, when the number of peers increases, it is difficult to decide the probability P properly.

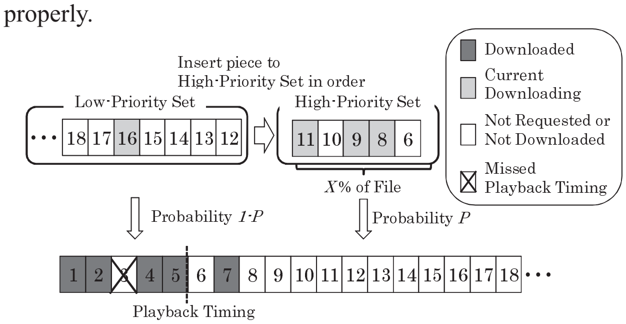

Fig. 1. Piece Selection Mechanism in BiToS.

Reference [10] proposed video streaming with the suitable quality for users' bandwidths supposing H.264/SVC. The video codec H.264/SVC [18] is the extended edition of H.264/AVC. H.264/AVC supposes that users receive all data to play the video. Hence, the video image falls into heavy deterioration even if only a part of data are lost. On the other hand, in H.264/SVC, the video data are classified into a base layer and some extended layers. The base layer contains essential data for playback while the extended layer contains additional data for higher quality. Even if the data in extended layers are lost, the data in base layer prevents extreme video image degradation.

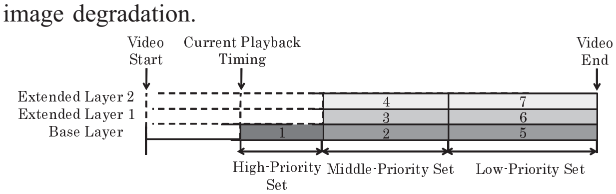

Fig. 2. Piece Selection Mechanism in [10].

Figure 2 shows the overview of piece selection algorithm in [10]. Unreceived pieces are classified into High-Priority Set, Middle-Priority Set or Low-Priority Set based on their playback timings. At first, to decrease a buffering time, peers select pieces from base layer of the High-Priority Set in order of playback timings. Next, peers select pieces from base layer of the Middle-Priority Set based on rarest-first algorithm, and then, pieces in extended layers of the Middle-Priority Sets are selected based on rarest-first algorithm. After that, peers select pieces from the Low-Priority Sets in the same manner.

The method in [10] has been presented supposing H.264/SVC with layered concept. If the method is adopted to general H.264/AVC, it has similar performance to BiToS. However, it is also insufficient to realize high-quality delivery because the pieces in the same set cannot be differentiated.

Moreover, it is difficult to decide the suitable set interval in various network environment.

III. Proposed Methods

In this paper, we improve piece selection algorithm in BitTorrent in order to realize high-quality video streaming system using BitTorrent.

At first, we introduce the concept of "Round" in order to confine the range to select pieces. Peers select pieces only in specific Rounds. By sliding the Round considering their own acquisition situation, peers can receive only pieces close to the playback timings. Then, as the way to decide candidate pieces to be selected, we introduce three piece selection algorithms considering error propagations caused by piece losses, playback timings, and piece dispersion on network. Among unreceived pieces in their own Round, peers select a candidate piece per each selection algorithm. Finally, peers decide the selected piece based on pre-determined probabilities.

A. Autonomous and Asynchronous Round Concept

In video streaming, peers need to select pieces closer to the playback timings preferentially. However, BitTorrent adopts rarest-first as piece selection algorithm and cannot confine the range to select piece. Thus, peers may select pieces far from the playback timings even though nearer pieces have not downloaded yet. Although [9], [10] improved this problem as mentioned in Sec. II-B, these methods cannot give the best performance especially in a bad network situation. Therefore, these methods still have insufficient support from the viewpoint of best video quality.

Therefore, in this paper, we introduce "Round" to each peer, which confines the range of piece selection in order to select only pieces close to the playback timings. Figure 3 shows the example of Round.

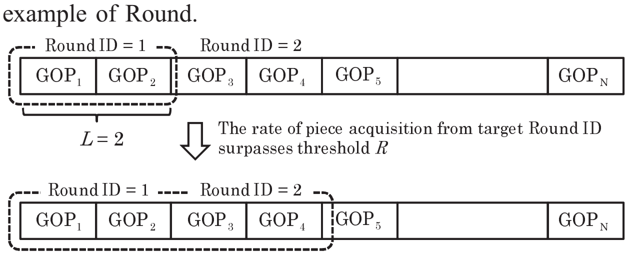

Fig. 3. Introduction of Round.

Each Round is composed of a certain number (L) of GOPs, "RoundID" is set to each Round. Each peer manages "MyRoundID" to decide the Rounds from which the peer should select the piece to receive. Peers can select pieces only from Rounds whose RoundID is equal to or smaller than MyRoundID. MyRoundID is automatically incremented by one when the rate of received pieces within MyRoundID exceeds a threshold R. Thus, note that each peer updates MyRoundID autonomously and asynchronously. Moreover, when a playback timing exceeds the range of Round with current MyRoundID, MyRoundID is also incremented even if the rate of piece acquisition from the Round does not surpass the threshold R.

The following shows the procedure of piece selection introduced by Round.

1) A peer sets MyRoundID at an initial value.
2) The peer selects pieces to download from the Rounds whose RoundID is equal to or smaller than MyRoundID.
3) The peer increments MyRoundID when the rate of received pieces within MyRoundID exceeds a threshold R.
4) The peer increments MyRoundID when a playback timing exceeds the range of Round with MyRoundID.
5) return to 2)

B. Piece Selection

Here, we propose the following three piece selection algorithms to decide candidate piece that should be received.

- Candidate piece selection for *importance* considering video structure information
- Candidate piece selection for *urgency* considering the playback timing
- Candidate piece selection for *scarcity* considering the piece dispersion on network

Each peer selects one candidate piece per piece selection algorithm. Then, the peer decides the piece to receive based on the process in Sec. III-B4.

1) Candidate Piece Selection for Importance: This selection algorithm depends on error propagation caused by frame losses. As mentioned in Sec. II-A, H.264/AVC has dependent relationship between frames because of inter-frame prediction coding. A loss of specific frame causes impairment propagation to all frames that depends on the frame. The number of frames affected by impairment propagation differs according to the frame type. A specific frame which affects more frames by impairment propagation is important frame, and should be selected preferentially.

In this selection algorithm, to prevent the losses of important frames, we define the number of frames affected by the loss of the focusing frame as the number of impairment propagation. Figure 4 shows video frame structure compressed by H.264/AVC, and the number of impairment propagations.

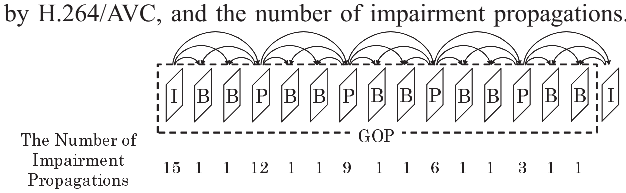

Fig. 4. The Number of Impairment Propagations based on H.264/AVC.

The number of frames in a GOP is 15. I frame is referred by all the other frames in GOP, hence the number of impairment propagation is 15. On the other hand, B frame has the difference from I frame or P frame, and is not referred by other frames. Thus, the number of impairment propagation of B frame is 1.

Each peer selects the pieces with the largest number of impairment propagation as candidate pieces for importance. This can decrease impairment propagation to the minimum.

2) Candidate Piece Selection for Urgency: This piece selection algorithm selects a frame based on the playback timing of piece. The piece which will be played in near future has high urgency and should be selected preferentially.

Therefore, in this piece selection algorithm, peers select the pieces closest to the playback timings as candidate piece for urgency.

3) Candidate Piece Selection for Scarcity: This piece selection algorithm is based on the number of peers which have already received the corresponding piece in P2P network.

When the requests of a rare piece concentrate on a specific peer, the upload speed of the peer decreases, and this affects on the delivery efficiency. Hence, each piece should be dispersed on network as soon as possible.

Therefore, in this selection algorithm, we use rarest-first algorithm in BitTorrent. For every unreceived pieces, each peer obtains the number of peers that already have the piece. Then, each peer selects the pieces, which the fewest peers have received, as candidate pieces for scarcity.

4) Final Piece Selection: Among three candidate pieces selected by the above three algorithms, each peer decides a final piece to receive with pre-determined probability. The probability to finally select the corresponding solution from among three pieces obtained by importance, urgency, and scarcity is defined as $P_{importance}$, $P_{urgency}$, and $P_{scarcity}$, respectively. These probabilities are set preliminarily and have the following relationship.

$$P_{importance} + P_{urgency} + P_{scarcity} = 1 \quad (1)$$

Based on these probabilities, each peer decides the piece to be received from three candidate pieces.

By piece selection for importance, peers select the piece whose loss affects significantly to the quality preferentially. By piece selection for urgency, peers select the piece whose playback timing is near preferentially. Moreover, by piece selection for scarcity, peers select other pieces that improves the delivery efficiency in P2P network preferentially. By balancing three piece selection algorithms appropriately, peers achieve efficient pieces' selection for both of themselves and entire network.

IV. Performance Evaluation

We evaluate the performance of the proposed method by computer simulation. We use the simulator of our own making programmed with Java. We use the simulator of our own making because we can make the simulator flexibly and make additional functions easily.

A. Simulation Scenario

In this scenario, there are 30 peers and 1 origin that sends a video data originally. At the beginning of simulation, all 30 peers join a P2P swarm (flash crowd), and start download. Each peer has the bandwidth of 2Mbps and the transmission delay between peers is 3msec.

Through fundamental experiment, we verified that the capacity ratio of I, P, and B frame data in H.264 using quality of 2.5Mbps is the following.

$$Iframe : Pframe : Bframe = 4 : 2 : 1 \quad (2)$$

In this simulation, the ratio of the numbers of I, P, and B frame is also set to $Iframe : Pframe : Bframe = 4 : 2 : 1$. Specifically, I, P, and B frames consist of four, two, and one pieces, respectively. The total number of frames per one GOP is 15 and the numbers of I, P, and B frames are 1, 4, and 10, respectively. Video data has the length of 300 seconds, which was encoded using quality of 2.5 Mbps, and the number of piece is 13200 (600GOPs). Therefore, the number of pieces in each frame is $Iframe = 2400$, $Pframe = 4800$, $Bframe = 6000$, respectively.

Table I and table II show the parameters in the proposed method and in BiToS [9], respectively.

TABLE I
SETTINGS IN THE PROPOSED METHOD.

The number of GOPs in a Round, L	10
The threshold of acquisition rate in Round change, R	0.7
The probability of each piece selection algorithm ($P_{importance}$, $P_{urgency}$, $P_{scarcity}$)	$\frac{1}{3}$

TABLE II
SETTINGS IN BITOS [9].

The size percentage of High-Priority Set in file, X [%]	10
The selection probability from High-Priority Set, P	0.8

B. Evaluation Index

In order to evaluate the performance of the proposed method, we use the following evaluation indexes.

- Continuity Index (CI)
- Continuity Index of Frame ($CIframe$)

Continuity Index (CI) is used as one of index values for evaluation in [7]. CI shows the continuity of playback, and is calculated by the following formula.

$$CI = \frac{NumOfPiece_{Deadline}}{NumOfPiece_{Total}} \quad (3)$$

Here, $NumOfPiece_{Deadline}$ is the number of pieces that a peer could receive by the playback deadlines. $NumOfPiece_{Total}$ is the total number of pieces in video data.

Continuity Index of Frame ($CIframe$) shows the continuity of playback considering the frame loss caused by the corresponding piece loss and the impairment propagation. Specifically, if any pieces of a frame are lost, the frame and the other frames, affected by the impairment propagation, are treated as losses. $CIframe$ is calculated by the following formula.

$$CIframe = \frac{NumOfFrame_{DeadlinewithNoLosses}}{NumOfFrame_{Total}} \quad (4)$$

Here, $NumOfFrame_{DeadlinewithNoLosses}$ is the number of frames that a peer could receive by the playback deadlines without any losses. $NumOfFrame_{Total}$ is the total number of frames in video data.

C. Experimental Results

In simulation experiments, we evaluate the proposed method compared to standard BitTorrent [17] and BiToS [9]. As described in Sec. II-B, [10] supposes H.264/SVC with layered concept. Hence, we exclude the comparison with [10]. Figure 5 shows average CI and $CIframe$ for each method. In this figure, "normal", "BiToS", and "proposal" show the piece selection in standard BitTorrent [17], BiToS [9], and the proposed method, respectively. From this figure, we can see that the CI and the $CIframe$ in the proposed method are higher than those in the existing methods. Here, in the proposed method, peers can play video with less frame errors by specific pieces' losses.

Figure 6 shows average CI and $CIframe$ for proposed method against the number of GOPs in a Round, L. From this figure, we can see that CI and $CIframe$ keep significantly high value when L is small. however, lower value as L is over 100. This is a consequence that the range of piece acquisition

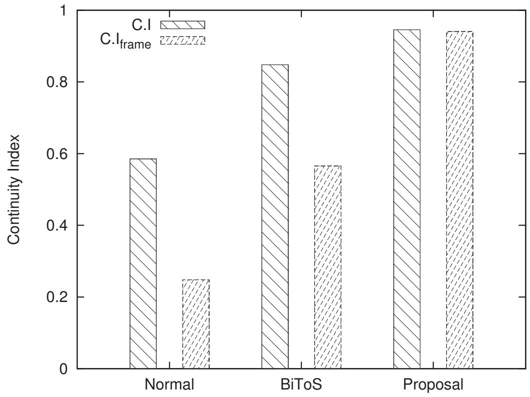

Fig. 5. CI and $CIframe$ in Each Method.

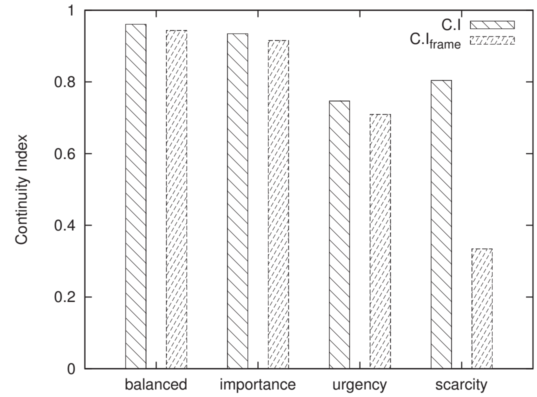

Fig. 7. CI and $CIframe$ against the probability in each piece selection.

TABLE III
SETTINGS IN FIG. 7.

	$P_{importance}$	$P_{urgency}$	$P_{scarcity}$
balanced	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{3}$
importance	1	0	0
urgency	0	1	0
scarcity	0	0	1

is too wide and peers cannot receive many pieces by the deadline timings. For this result, in this network situation, we see that the best value of L is over 5 but less than 10.

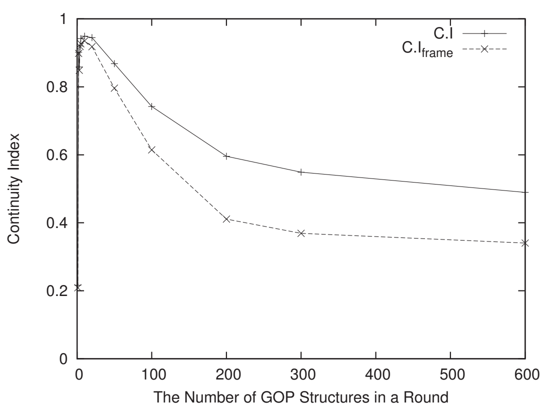

Fig. 6. CI and $CIframe$ against the Number of GOPs in a Round.

Figure 7 shows average CI and $CIframe$ for proposed method against the probabilities in each piece selection. Table III shows the probabilities in each piece selection.

From this figure, we can see that CI and $CIframe$ in "balanced" are higher than the other methods. CI and $CIframe$ in "importance" are relatively high because peers receive several I frames or P frames preferentially, and this provides peers with quite large range of piece acquisition. However, in "urgency", the CI and the $CIframe$ decrease because the range of piece acquisition is too limited and pieces are not dispersed on network. Moreover, in "scarcity", $CIframe$ is low in particular, because the method does not consider video structure and cannot receive I and P frames preferentially. Thus, we can see that peers need to receive pieces in a well-balanced way considering video structure, playback timings, and piece dispersion on network.

V. Conclusion

In this paper, we proposed high-quality video streaming system using BitTorrent to reduce latency and decrease video image degradation. In performance evaluation through the computer simulation, we have shown that the proposed method can keep the playback continuity compared to existing methods. Future work includes the adaption to various network conditions.

References

[1] Youtube, "URL: http://www.youtube.com/" .

[2] Y. Guo, K. Suh, J. Kurose, and D. Towsley, "A peer-to-peer on-demand streaming service and its performance evaluation," in Proc. IEEE Int. Conf. on Multimedia and Expo (ICME' 03), vol.2, pp.649–652, July 2003.

[3] D. Xu, M. Hefeeda, S. Hambrusch, and B. Bhargava, "On peer-to-peer media streaming," in Proc. 22nd Int. Conf. on Distributed Computing Systems (ICDCS' 02), vol. 1, pp.363–371, July 2002.

[4] J. Liu, et al., "Opportunities and Challenges of Peer-to-Peer Internet Video Broadcast," in Proc. IEEE Special Issue on Recent Advances in Distributed Multimedia Communications, 2007.

[5] V. Padmanabhan, et al., "Distributing streaming media content using cooperative networking," in Proc. the 12th international workshop on Network and Operating Systems Support for Digital Audio and Video, pp. 177–186, May 2002.

[6] Peercast p2p radio, "URL: http://www.peercast.org/".

[7] X. Zhang, J. Liu, B. Li, and T. P. Yum, "Coolstreaming/donet: A data-driven overlay network for peer-to-peer live media streaming," in Proc. IEEE INFOCOM, Miami, March 2005.

[8] V. Pai, et al., "Chainsaw: Eliminating trees from overlay multicast," Lecture Notes in Computer Science, vol. 3640, p. 127, Nov. 2005.

[9] A. Vlavianos, M. Iliofotou, and M. Faloutsos, "BiToS: Enhancing bittorrent for supporting streaming applications," in Proc. IEEE INFOCOM' 06, Apr. 2006.

[10] S Oechsner, T. Zinner, J. Prokopetz, and T. HoBfeld, "Supporting Scalable Video Codecs in a P2P Video-onDemand Streaming System," Proc, The 21st ITC Specialist Seminar, March 2010.

[11] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264 / AVC Video Coding Standard," IEEE Transactions on Circuits And Systems for Video Technology, July 2003.

[12] Marpe D, Wiegand T, and Sullivan GJ, "The H.264/MPEG4 advanced video coding standard and its applications," IEEE Communications Magazine, vol. 44, no. 8, pp.134–143, 2006.

[13] Lua, E.-K., Crowcroft, J., Pias, M., Sharma, R., and Lim, S., "A Survey and Comparison of Peer-to-Peer Overlay Network Schemes," IEEE Communications Surveys and Tutorials 7, pp. 72–93, 2005.

[14] Winny, "URL: http://www.geocities.co.jp/SiliconValley/2949".

[15] Gnutella, "URL: http://gnutella.wego.com".

[16] Napster, "URL: http://www.napster.jp/".

[17] BitTorrent, "URL: http://www.bittorrent.com".

[18] H. Schwarz, D. Marpe, and T. Wiegand, "Overview of the Scalable Video Coding Extension of the H.264/AVC Standard," IEEE Trans. CAS-VT, vol. 17, no. 9, pp. 1103-1120, Sept. 2007.