AN EVALUATION OF PIECE-PICKING ALGORITHMS FOR LAYERED CONTENT IN BITTORRENT-BASED PEER-TO-PEER SYSTEMS

Michael Eberhard, Hermann Hellwagner, Christian Timmerer

ITEC, MMC
Alpen-Adria-Universitaet Klagenfurt, Austria
firstname.lastname@itec.uni-klu.ac.at

Tibor Szkaliczki, Laszlo Szobonya

eLearning Department
MTA SZTAKI, Hungary
sztibor@sztaki.hu, szobonya@sztaki.hu

ABSTRACT

In this paper the performance of layered piece-picking algorithms for Bittorrent-based peer-to-peer systems is evaluated and compared to traditional single layer solutions. In a Bittorrent-based peer-to-peer system, the piece-picking algorithm needs to ensure that the pieces of an audiovisual content are received in time to ensure smooth playback of the content. For layered content provided in multiple qualities, the task of the algorithm becomes more complex, as it has to consider the layer of the pieces to be fetched in addition to the deadline. Thus, the goal of the layered piece-picking algorithm is to ensure that the best possible quality under the given network conditions is received while the playback is not disturbed. To illustrate the advantages of such a layered piece-picking algorithm, its performance is evaluated and compared to a single layer piece-picking algorithm. The results show that layered piece-picking algorithms can significantly improve the quality in terms of peak signal-to-noise ratio.

Index Terms— Peer-to-Peer, Layered Content, Piece-Picking, Evaluation

1. INTRODUCTION

The distribution of content using peer-to-peer (P2P) systems has been a popular research field in recent years, as P2P can help to significantly reduce the costs for providing the content to the users. Due to the fact that all users who consume content also help with its distribution, P2P systems provide good scalability. Thus, P2P can also help to provide content to a large user base with no need for an extensive distribution infrastructure. In Bittorrent-based systems, the content is split into pieces and the pieces can be independently downloaded from separate neighbor peers. Multiple evaluations have shown that Bittorrent-based P2P systems show a very good performance in terms of bandwidth usage, fairness, and download time (see, e.g., [1]).

Another requirement to content distribution that has become more important in recent years is that users consume the content on a variety of terminals like mobile phones, tablet computers, laptops, or HDTV-sets. Thus, it is not sufficient to provide the content in a single quality. To satisfy the varying user preferences and requirements of their terminals (e.g., resolution, network bandwidth), the content needs to be provided in a number of different qualities. Layered codecs provide a coding-efficient way to supply a number of different qualities. The main advantage of using layered codecs is that the sharing process is more efficient, since all users interested in the same content can at least share the base layer with each other (and enhancement layers with users interested in higher qualities). Additionally, layered codecs increase the flexibility when streaming content to the users, e.g., by starting the streaming with the base layer to reduce the start-up delay or by performing layer switches to avoid frame freezing when the network conditions change. Although the P2P system presented in this paper was designed to be codec independent, the system was implemented using Scalable Video Coding (SVC) codecs [2].

When streaming multimedia content live or as Video on Demand (VoD) in Bittorrent-based P2P systems, the piece-picking algorithm ensures that all pieces are downloaded in time for display. If layered content is streamed, the piece-picking algorithm also needs to provide the best possible quality under the given network conditions. Thus, the piece-picking algorithm needs to find the best trade-off between smooth playback (i.e., deadline awareness) and quality. In this paper, piece-picking algorithms for single and multi layer content are evaluated and compared.

The remainder of this paper is organized as follows. In Section 2, an overview of the related work is presented. Section 3 provides a more detailed introduction to the layered piece-picking problem. In Section 4, the evaluation of the layered piece-picking algorithm in comparison to the single layer approach is presented. Finally, Section 5 concludes this paper.

t-1	t	t+1	t+2	t+3	t+4	t+5	t+6	t+7
1.0	1.0	1.0	1.0	0.9	0.5	0.0	0.0	0.0

Fig. 1. Single Layer Sliding Window

2. RELATED WORK

The distribution of layered content in P2P systems has been a popular research field and has been addressed in a number of recent publications. Thus, there already exists a number of P2P systems supporting the distribution of layered content. LayerP2P [3] provides a very well defined solution for distributing layered live content in a P2P system. However, the work does not provide details how the pieces of the layered content are selected for download. Additionally, the P2P client has been implemented in order to support layered content and the used P2P protocol is not compatible with other existing clients. [4] describe a VoD P2P system with support for SVC. Although the scheduling problem for selecting the pieces of layered content is described in detail, the proposed zigzag-like scheduling can lead to frequent quality switches, which have been found to be more disturbing to users than watching a video at constant lower quality [5].

Other P2P systems supporting layered codecs include [6] and [7]. PALS [6] provides a receiver-based approach to download the best possible quality from the neighbor peers but does not specifically address the quality dimensions provided by SVC. [7] describes how support for layered codecs can be integrated into a tree-based P2P system. Although both papers describe detailed approaches for supporting layered content, the presented implementations cannot be easily integrated into existing P2P clients to enable sharing of content with an existing user base.

The work presented in this paper focuses on the integration of layered piece-picking algorithms into an already existing open source Bittorrent-based P2P system. Although this imposes requirements like the usage of fixed piece size and the need for backwards compatibility, it allows the integration of layered content distribution into an existing system that already has a large user base [8]. To solve the optimization problem imposed by the layered piece-picking process, we have already proposed, evaluated and compared different types of layered piece-picking algorithms in [9]. As the piece-picking problem is closely related to the knapsack problem, a number of algorithms for the knapsack problem have been investigated and an efficient piece-picking algorithm for layered content has been developed. The recommended algorithm also ensures that quality switches to higher layers are only performed if the new quality can be provided for many time slots to avoid frequent quality switches.

	t-1	t	t+1	t+2	t+3	t+4	t+5	t+6	t+7
EL2	1.0	1.0	1.0	0.7	0.3	0.0	0.0	0.0	0.0
EL1	1.0	1.0	1.0	1.0	0.8	0.5	0.0	0.0	0.0
BL	1.0	1.0	1.0	1.0	1.0	0.9	0.5	0.0	0.0

Fig. 2. Multi Layer Sliding Window

3. LAYERED PIECE PICKING

In Bittorrent-based P2P systems, the piece-picking algorithm is responsible for downloading the pieces in time for smooth display. When layered content is consumed, the algorithm also needs to maximize the consumable quality while still ensuring that the pieces are downloaded in time. The piece-picking algorithm works on all pieces within a sliding window, which contains the pieces for the current time slot and pieces for the near future. While the pieces within the sliding window are most important for playback and are downloaded first, the pieces with a deadline in the future and outside of the sliding window are downloaded using Bittorrent's rarest-first algorithm if there is still bandwidth available. An overview of the sliding windows used by the piece-picking algorithm for single layer and multi layer content is provided in Figures 1 and 2.

Figure 1 illustrates the situation for single layer content. Each cell represents a piece providing the content for a specific time slot. The values of the cells indicate the download status, i.e., 1.0 is the status of an already downloaded piece, while 0.0 indicates that the download has not started yet. The sliding window is illustrated by the rectangle around the cells. For single layer content, the earliest deadline-first piece-picking algorithm is utilized [8], which gives higher priority to pieces with earlier deadlines. Thus, the piece with the earliest deadline that is not currently being downloaded is selected next for downloaded, i.e., the piece with deadline $t + 5$ in Figure 1.

In Figure 2 the situation when consuming layered content with 3 layers is illustrated. In this case, every row of pieces represents one layer of the content. Thus, the lowest row represents the base layer (BL) and the higher rows represent the enhancement layers (EL1 and EL2). For layered content, the priority assignment is more complex, as the piece-picking algorithm has to consider the deadline as well as the layer. The algorithm has to try to maximize the quality while still ensuring that the playback is never stopped (i.e., download the pieces for all layers before the deadline) and frequent quality switches are avoided (which disturbs the viewing experience).

In the situation illustrated in Figure 2, the piece-picking algorithm could hence decide to download all base layer pieces first to ensure that the playback is never stopped. On the other hand the algorithm might decide to first maximize the quality for $t + 4$, i.e., to ensure that the high quality piece will be downloaded in time.

To solve the optimization problem imposed by the layered piece-picking process, we have already proposed, evaluated and compared different types of layered piece-picking algorithms in [9]. Based on these previous evaluation results, the greedy piece-picking algorithm was selected for the evaluation in this paper, as it has shown a better performance than other algorithms. The greedy algorithm calculates the priority for all pieces in the sliding window that have not been downloaded based on a utility function. For the calculation of the utility, the layer of the piece, its probability to be received in time and its urgency (i.e., the remaining time until the playback deadline) are taken into account. Once the utility for all the pieces is calculated, the pieces are selected for download according to their utility as long as sufficient download bandwidth is available. For more details regarding the greedy piece-picking algorithm and alternative piece-picking algorithms please refer to [9].

4. EVALUATION

In this section, the results of numerous tests comparing our layered piece-picking algorithm to the single layer solution currently used in our P2P system are presented. The tests were performed using the Oversim P2P simulation framework [10]. The simulation framework was extended to support the Bittorrent-based protocol utilized in our P2P system based on [8] and the greedy layered piece-picking algorithm was implemented.

The goal of this evaluation process was to determine the behaviour of the P2P system when distributing single layer and multi layer content. The metrics used for the evaluation of the algorithms in the different test scenarios were average streaming quality, the fraction of late frames, and the impact of churn on the streaming rate. Although the received streaming bitrate is another popular metric used for evaluations of P2P systems, we did not evaluate it for our single and multi layer comparison, as the results for both approaches are the same for the given architecture (fixed piece size for both implementations, the same amount of pieces is received for both approaches for the same network conditions). However, we acknowledge that the bitrate is a very good metric for P2P systems using a dynamic piece size, as the results can then vary significantly between single and multi layer implementations.

For the evaluations presented in this section, a one hour video was streamed by 1000 peers. The performance metrics presented in this section are calculated as an average of the results for all peers. The video has a size of 640x480, a frame rate of 25 frames per second, and was encoded with 4 quality layers using medium-grain scalability. The details of the video encoding are shown in Table 1. The video was split up into pieces each containing a group of pictures (GOP) of 64 frames for every layer. Thus, the overall video was split into 1440 time slots (each time slot containing 64 frames or approx. 2.5 seconds of content) and for every time slot 4 pieces were created, where every piece represents a quality layer. The single layer video was encoded with the properties of the highest layer of the multi layer video (see the last line of Table 1) using an Advanced Video Coding (AVC) encoder. However, as the single layer encoding has a better coding efficiency than the multi layer encoding (for the example sequence, the multi layer encoding requires approx. 7 % more bitrate to achieve the same PSNR), the peak signal-to-noise ratio (PSNR) of the single layer sequence is slightly higher than the PSNR of the highest layer of the multi layer sequence. For the single layer content the 64 frames in full quality are also contained in 4 pieces (approx. 16 frames per piece). Before the actual streaming measurements start, the peers enter a pre-buffering phase of 5 time slots, where the buffer is initially filled.

Table 1. Video Layer Structure

Bit Rate	Resolution	Quality	Frames/sec
512 kbps	640x480	basic	25
1024 kbps	640x480	low	25
1536 kbps	640x480	medium	25
2048 kbps	640x480	high	25

To test the performance of the single layer and multi layer solutions, a number of scenarios was defined. The overall bandwidth is limited by restricting the bandwidth of the seeding peers. In the scenarios where a churn rate is applied, an average churn rate of 10 % is utilized. This means that approx. 10 % of the seeding peers leave the swarm during the streaming and approx. 10 % of new peers join.

- Scenario 1: In this scenario, the seeders provide sufficient bandwidth for all 1000 peers to download all layers and no churn occurs.
- Scenario 2: Again, sufficient bandwidth for all peers to download all layers is provided, but a churn rate of 10 % is utilized.
- Scenario 3: In this scenario the upload bandwidth of the seeder peers is limited so that on average only approx. 3/4 of the piece requests can be served. The churn rate is set to 10 %.

Please note that we have not defined scenarios where the bandwidth provided by the seeders gets lower than 3/4 of the best quality bitrate. Although the performance advantage of the multi layer solution increases in lower bandwidth scenarios, it is not realistic to start a single layer streaming session when clearly less than the required bandwidth is available.

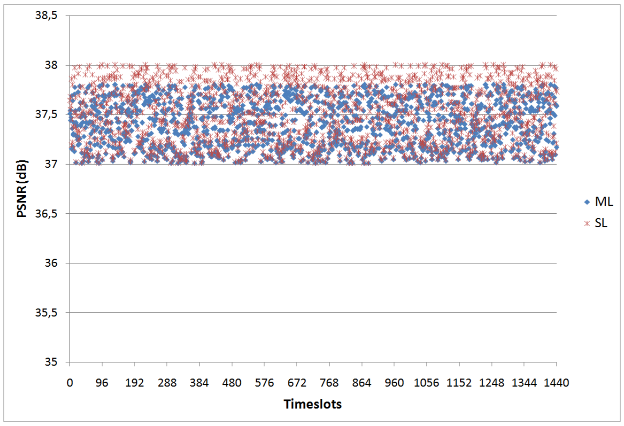

Fig. 3. Evaluation Scenario 1: PSNR

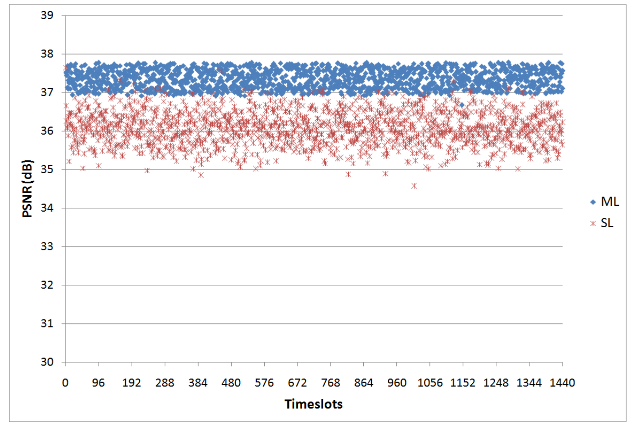

Fig. 4. Evaluation Scenario 2: PSNR

Thus, results for low bandwidth scenarios are not presented in detail in these evaluations, but the results for these scenarios are indicated in the following sections.

4.1. Received Video Quality Comparison

In this section, the received video quality in PSNR of the intensity component (Y-PSNR) is compared for the streaming of single layer and layered content. The PSNR is calculated as the average PSNR of the received pieces (e.g., the average PSNR of the 64 frames that are contained within a multi layer piece). If the frames are not received in time, the PSNR for these frames is determined by calculating the mean square error between the original frames and a black frame (no image can be shown for frames not received in time). The reason for selecting this approach instead of the error concealment technique to repeatedly show the last received frame is that a late piece leads to the late arrival of at least 64 frames for the multi layer approach and at least approx. 16 frames for the single layer approach. Thus, showing the same picture repeatedly is not sufficient to conceal the missing frames.

The results for the received PSNR based on Scenario 1 is presented in Figure 3. In the figure the average received PSNR of all 1000 peers over the 1440 time slots (each having a length of 2.5 seconds, 60 minutes in total) is presented. The received quality for the multi layer video is indicated by ML and for the single layer video by SL. The results show that the average received PSNR for the single layer approach is slightly higher than for the multi layer approach. This is due to the slight overhead in terms of coding efficiency of the multi layer implementation.

However, as a scenario with no bandwidth restriction and no churn does not realistically represent a real world system, the performance of the single and multi layer approach is also investigated for Scenarios 2 and 3. The results for these scenarios are illustrated in Figures 4 and 5.

The results for Scenario 2 in Figure 4 show that when a realistic churn rate is set, the multi layer approach already provides a slightly better average quality than the single layer approach (approx. 1.5 dB higher PSNR). The reason for the difference is that a missed piece for the multi layer implementation usually only leads to a decrease in quality of all frames for the time slot to a lower layer. This is due to the piece-picking algorithm selecting the lower layers with higher priority. Thus, if a piece is not received in time it belongs usually to a higher layer. However, for the single layer implementation the PSNR for a part of the frames for the time slot is significantly decreased if any of the pieces belonging to the time slot is not received in time. This is even more clearly visible in the results for Scenario 3 illustrated in Figure 5. When the bandwidth is limited so that not all of the pieces can be received in time, the average quality for the single layer approach decreases significantly. For the multi layer approach, the decrease in PSNR is not very drastic, as mostly only the highest enhancement layer piece cannot be downloaded in time.

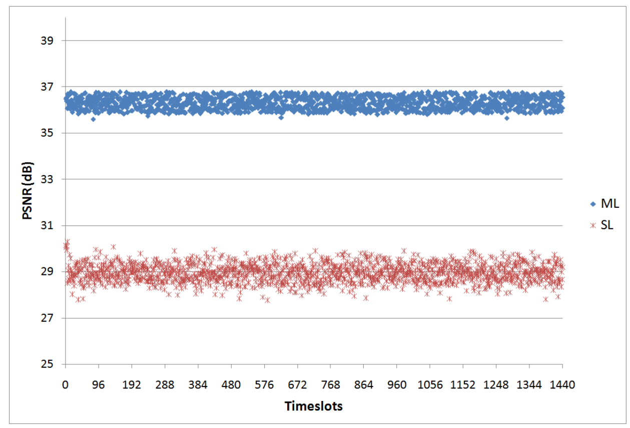

Fig. 5. Evaluation Scenario 3: PSNR

To better illustrate why the single layer approach decreases significantly if not all pieces can be received in time,

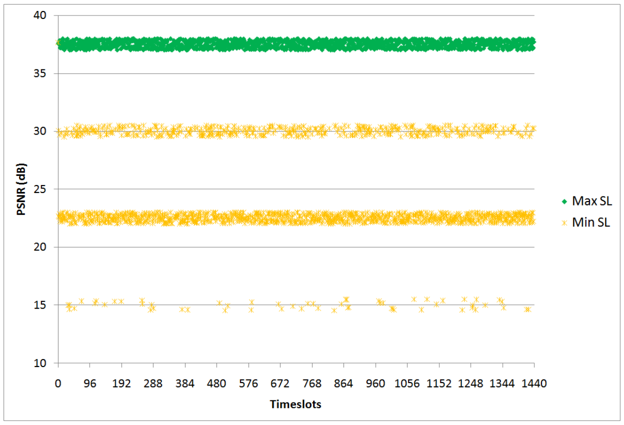

Fig. 6. Evaluation Scenario 2: Min/Max PSNR (SL)

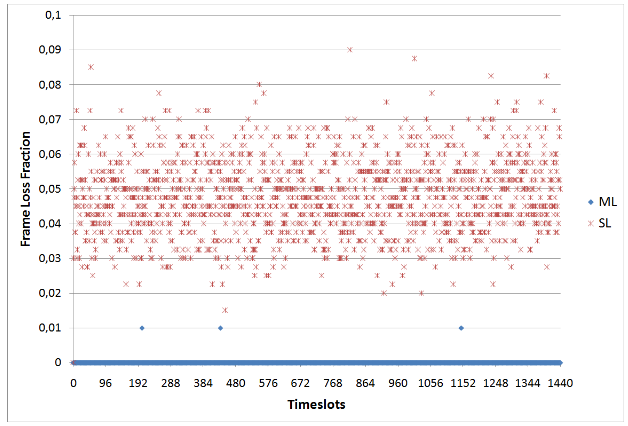

Fig. 7. Evaluation Scenario 2: Frame Loss

Table 2. Evaluation Scenario 2: Received Pieces/TS (SL)

# Pieces	%
4	82.16
3	16.63
2	1.17
1	0.04
0	0.00

the minimum and maximum received quality of the 1000 peers for all the time slots when applying the single layer approach is shown in Figure 6.

The results illustrate that always one of the 1000 peers received the best quality, but also at least one peer always did not. As for every piece that is not received in time, 16 frames are weighted with the PSNR of a black picture, the received quality is significantly decreased. It should be noted that Figure 6 shows only the minimum and the maximum results for all 1000 peers for every time slot. The percentage of peers receiving a specific number of pieces per time slot is additionally presented in Table 2.

The percentages in Table 2 show that in most cases the peers still receive all 4 or 3 pieces per time slot. However, although on average only 1.17 % of the 1000 peers receive only two pieces per time slot, it is still the most frequent minimum in Figure 6. This is an additional reason for the visible difference between the multi and single layer approaches, as even a single peer receiving only 2 pieces per time slot visibly decreases the average quality for all 1000 peers.

As mentioned in the introduction to this section, we do not present results for lower bandwidth scenarios in this paper. However, from the results presented in Figures 4 and 5 it should be clear that the performance advantage of the multi layer approaches increases when the overall bandwidth of the seeders decreases.

4.2. Fraction of Late Frames

Another important metric when streaming videos live or as VoD is the fraction of late frames, as late frames lead to frame skipping (live) or freezing (VoD) which significantly disturbs the viewing experience. Thus, the impact of using layered codecs on the number of frames arriving after their deadline is investigated in this section. An illustration of such a test run is provided in Figure 7.

In Figure 7 the y-axis represents that fraction of frames that were received after their deadline or not received at all, while the x-axis again represents the time slots. The results show that the churn rate applied in Scenario 2 leads to significantly higher frame loss for the single layer implementation. The difference is mainly due to the fact that any piece arriving after the deadline leads to frame losses for the single layer approach, while only a missed piece of the base layer leads to a frame loss for the multi layer approach. As the download of the base layer is prioritized by the layered piece-picking algorithm, a late arrival of a base layer piece is unlikely in high bandwidth scenarios. Given that the frame loss in lower bandwidth conditions significantly increases for the single layer implementation and does only slowly increase for the multi layer implementation, we do not present the frame loss results for lower bandwidth scenarios, as the results are easy to predict.

4.3. Churn

In this section, the impact of dynamic churn on the received streaming quality is investigated. Although a realistic churn rate has already been applied for the evaluations in Scenarios 2 and 3, the impact of a frequently changing churn rate has not been addressed so far.

For the following evaluations, the network conditions for the two relevant scenarios remain the same, while the churn rate is initially set to zero and increases every two minutes (48 time slots) until a churn rate of 50 % is reached at the end of

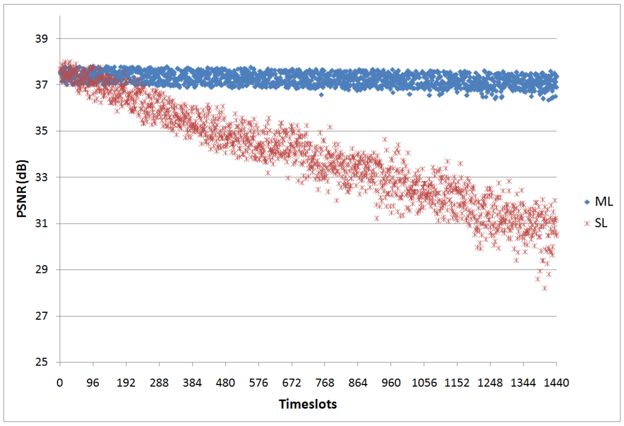

Fig. 8. Evaluation Scenario 2: Churn

the test run.

A first evaluation of the modified Scenario 2 is provided in Figure 8. As has already been shown in the evaluation results of Section 4.1, an increasing churn rate has a more significant impact on the visual quality for the single layer implementation. Although the received PSNR is in the beginning slightly higher for the single layer approach, it soon falls below the multi layer approach as the churn rate increases. The effect of the churn rate is also visible for the multi layer solution, but as long as at least the base layer piece is received, the decrease in received PSNR is rather limited. As the results for lower bandwidth scenarios are similar to the previous results (i.e., the received quality for the single layer approach is already significantly lower in the beginning), these results are not presented in detail.

5. CONCLUSION

In this paper, an evaluation of multi and single layer piece-picking algorithms has been presented. In Section 3 an introduction to the layered piece-picking problem was given. In Section 4, detailed evaluations for the multi and single layer approach considering the received quality, the frame loss and the behaviour related to peer churn were presented.

The evaluation results have shown that the layered piece-picking solution can very well compete with the single layer solution. Although the single layer solution provides slightly better performance in unrestricted bandwidth scenarios with no churn due to the better coding efficiency of single layer codecs, the multi layer solution performs clearly better in more realistic scenarios with limited bandwidth and/or churn. The evaluations presented in this paper have been performed using a simulation framework. In the future, we will evaluate the layered piece-picking algorithm in a real-world P2P system and perform detailed measurements to validate the simulation results.

6. ACKNOWLEDGMENT

The research leading to these results has received funding from the European Union's Seventh Framework Programme (P2P-Next) under grant agreement no. 216217. Additional support of the Hungarian Science and Technology Foundation (AT-2/07), the Austrian Agency for International Cooperation in Education and Research (HU-6/08), and the Hungarian National Science Fund and the National Office for Research and Technology (Grant No. OTKA 67651) are gratefully acknowledged.

7. REFERENCES

[1] A. R. Bharambe, C. Herley, and V. N. Padmanabhan, "Analyzing and improving a Bittorrent network's performance mechanisms," in *INFOCOM Proc.*, 2006, pp. 1–12.

[2] H. Schwarz, D. Marpe, and T. Wiegand, "Overview of the scalable video coding extension of the H.264/AVC standard," *IEEE Trans. on CSVT*, vol. 17, no. 9, pp. 1103–1120, 2007.

[3] Z. Liu et al., "LayerP2P: Using layered video chunks in P2P live streaming," *IEEE Trans. on MM*, vol. 11, no. 7, pp. 1340–1352, 2009.

[4] Y. Ding et al., "Peer-to-peer video-on-demand with scalable video coding," *Computer Communications*, vol. 33, no. 14, pp. 1589–1597, 2010.

[5] M. Zink et al., "Subjective impression of variations in layer encoded videos," in *Intl. WS on QoS*, 2003, pp. 137–154.

[6] R. Rejaie and A. Ortega, "Pals: Peer-to-peer adaptive layered streaming," in *NOSSDAV Proc.*, New York, NY, USA, 2003, pp. 153–161.

[7] P. Baccichet et al., "Low-delay peer-to-peer streaming using scalable video coding," in *Packet Video*, 2007, pp. 173–181.

[8] J. A. Pouwelse et al., "Tribler: a social-based peer-to-peer system," *Concurrency and Computation: Practice and Experience*, vol. 20, no. 2, pp. 127–138, 2008.

[9] M. Eberhard et al., "Knapsack problem-based piece-picking algorithms for layered content in peer-to-peer networks," in *AVSTP2P WS Proc.*, 2010, pp. 71–76.

[10] I. Baumgart, B. Heep, and S. Krause, "OverSim: A flexible overlay network simulation framework," in *IEEE GI Proc.*, May 2007, pp. 79–84.