A Scheduling Strategy for P2P-TV Systems using Scalable Video Coding

R. Fortuna, L. A. Grieco, G. Boggia, and P. Camarda,
DEE - Politecnico di Bari - Via E. Orabona, 4 - 70125 Bari (Italy)
e-mail: {r.fortuna,a.grieco,g.boggia,camarda}@poliba.it

***Abstract*—The adoption of scalable encoders can improve P2P-TV performance in heterogeneous scenarios, where peers with different capabilities coexist. In fact, each peer could dynamically adapt the resolution of the audio/video signals to its computation capabilities and available communication resources. Despite these clear benefits, casting a scalable encoder in a P2P-TV system requires sophisticated scheduling strategies in order to maximize user satisfaction. In this paper, we propose a novel chunk scheduler for layered video designed on the basis of control theory. It is able to significantly limit base chunk losses at the expenses of enhancement ones, thus maximizing overall video quality. Simulation results, conducted using a real-encoded video sequence and variable size chunks, have clearly shown the effectiveness of the algorithm, able to reach near-optimal performances.**

I. Introduction

In the last years P2P systems have become, for video broadcasters, the most efficient way for delivering multimedia contents without expensive infrastructures [1]. The distributed P2P approach, already known for file sharing, is nowadays extended to live streaming applications. In such systems, nodes (*peers*) interested to a given content form an overlay communication infrastructure to disseminate and share data [2]. As in P2P applications for file sharing [3], also in P2P-TV systems, multimedia contents are divided, at the source, into *chunks* and distributed over the overlay with a gossip-based approach [2]. The main difference with respect to file sharing applications is that, in order to preserve the real-time nature of live broadcasting, time constraints have to be imposed on chunk delivery [4]. Usually, chunks are not decoded as soon as they are received by peers, but a playout delay is inserted between the creation of a chunk at the source and its decoding process. In literature, the playout delay has been often set equal for all peers in order to provide a synchronized visualization to all users [5] [6]. When a chunk is received after the playout delay, it is useless and cannot be forwarded to other peers, thus avoiding bandwidth wastage.

Playout Delay is a key parameter in P2P-TV systems because it strongly influences the Quality of Experience of the users [7]. In fact, a small playout delay improves the timeliness information delivered at the users, but at the expense of an increased chunk loss ratio[6]. On the opposite, a large playout delay ensures that a high quota of chunks is received within the deadline, reducing the video distortion at the expense of system timeliness. To avoid forwarding expired chunks, each peer implicitly defines a transmission window made up of the most recent chunks, i.e., those that do not violate the constraint on playout delay. The problem of deciding which chunk of the transmission window has to be transmitted first and which ones can be further delayed is a major task handled by the so called *chunk scheduler* [8]. Loss probability for each chunk notably depends on chunk scheduler because a chunk loss substantially corresponds to a chunk arrival delay greater than the playout delay. Given the importance of such a problem, so far many scheduling algorithms in P2P-TV systems have been proposed in literature. Anyway, these contributions (e.g., [5], [8], [9]), do not take into account that the most powerful video codes, such as H.264 Advanced Video Coding with Scalable Video Coding (H.264/ACV SVC) [10], adopt a quality adaptive encoding to represent the video using two or more layers with different priorities. The base layer represents a very low quality version of the video and can be transmitted even in presence of a limited available bandwidth. Enhancement layers, instead, can be added or striped according to the network status to change video quality. In a layered video, composed of base and enhancement layers, the base layer is more important than the enhancement ones because it allows the fluent visualization of reconstructed video at low quality. Enhancement layers increase video quality proportionally to the amount of received enhancement data and need the corresponding base layer to be correctly decoded [10]. Scalable video is particularly useful in heterogeneous scenarios, where the peers of the overlay have different upload bandwidth or capabilities. Peers connected to high-capacity peers will be able to download a video sequence at higher quality with respect to those with a narrower available bandwidth and/or processing capabilities [11]. The problem of scheduling layered video in P2PTV systems has been faced in previous works, most of them under restrictive hypothesis. In [12] a scheduler has been proposed to maximize the video quality by prioritizing the most important chunks. This strategy is particularly suited for push-based, tree-structured overlays. The algorithm conceived in [13] minimizes the base layer losses, but it assumes equally-sized chunks and base layer rate equal to the enhancement one. This model of video is rather ideal and can be approximated only by FGS scalability. The scheduling strategies proposed in [14] and [11] are able to save base layer losses at the expense of the enhancement one, but, they have not been evaluated in terms of reconstructed video quality. Authors in [15] propose an optimal scheduling strategy to minimize the overall video distortion, but the approach is strongly related to the usage of Multiple Description (MD) coding, which is less efficient compared with layered video [16]. The most remarkable work on layered video in live P2P-TV has been done by authors in [17]. They designed and implemented a new scheduler, integrated with a tit-for-tat strategy able to adapt experienced video quality to the selfishness of the peer. This strategy, however, requires to know the bit rate of each video layer, which could be not available in real settings. In fact, only the average rate of each layer can be known in advance. In this paper we leverage on control theoretic arguments to design a novel chunk scheduler for layered video in P2P-TV systems. The target of the proposed scheduler is to strongly limit the losses occurrences for base layer chunks, at the expense of the enhancement ones, thus improving the overall quality of the delivered video. The scheduling scheme has been designed without any assumptions on the overlay topology nor on the encoding standard. To demonstrate its validity in a realistic case study, we have adopted the last version of P2P-TV-sim NAPAWINE simulator, available on-line at [18]. A mesh-based overlay has been considered, preferred to the tree-based structure by most of the commercial P2P systems [19], [20] due to its robustness against churning [4]. Furthermore, a real H264/AVC encoding scheme with variable bit rate and temporal scalability has been employed. This choice is motivated by the fact that temporal scalability, based on hierarchical frame B prediction [10], grants the highest compression efficiency [21]. The scheduler evaluation has been done taking into account base losses percentage, enhancement losses percentage, and overall video quality evaluated in terms of the Peak Signal to Noise Ratio (PSNR) [22]. The proposed scheduler has been compared with respect to: the Random Useful chunk scheduler [9] and an optimal chunk-scheduler that prioritize the base layer, designed under the ideal hypothesis of knowing in advance the available bandwidth at each peer and the properties of the layered video. Simulation results clearly show that the algorithm we propose can outperform the basic Random Useful scheme, reaching performances very close to the ideal case. The rest of the paper is organized as follows: Sec. II describes the innovative scheduling scheme we propose. Sec. III presents simulation results. Finally, the last Section draws conclusions and forecasts future research.

II. Closed Loop Scheduling Strategy

The proposed chunk scheduling algorithm has been designed using control theoretic arguments. The basic idea is to strongly limit base layer chunk losses, which severely impair user satisfaction [10]. In fact, loosing a base layer chunk can impair the fluency of the video and compromise the possibility to decode future enhancement data. With our approach, enhancement layers can be downloaded if and only if available bandwidth is greater than base video rate. The strength of the proposed algorithm lies in its capability to correctly allocate network resources, without knowing in advance neither available bandwidth nor base video rate. In particular, it allows each peer to autonomously choose the amount of base layer to retrieve from the overlay based on its decoding base layer buffer level. To this aim, each peer tries to keep the decoding buffer size at a constant reference level Q_R. The difference between the actual buffer level Q and the reference one Q_R is fed into a discrete-time regulator [23], that computes the amount of the required base layer. In this way, the base layer reception rate is implicitly forced to track the source base rate. The proposed idea is general because does not require any assumption on the network topology nor on the scalable encoding scheme.

A. Control Law

Herein, we design the scheduler control law starting from the general model of the buffer dynamics at the i-th peer. We will assume that control actions are taken at discrete time instants, so that the variables of interest will be observed at discrete time steps. Let $q_i(n)$ be the queue level associated with the receiver decoding buffer of the i-th peer at the n-th sampling instant expressed in chunks. In the discrete-time domain, its dynamics can be modelled by the following linear model:

$$q_i(n+1) = q_i(n) - d_i(n) + u_i(n) + \Delta u_i(n), \qquad (1)$$

where, with reference to the n-th sampling interval, $d_i(n) \geq 0$ is the number of chunks consumed by decoder; $u_i(n) + \Delta u_i(n) \geq 0$ is the number of base chunks the peer is able to download. Note that the latter term is made by two components: the first one, $u_i(n)$, accounts for the number of base chunks that the proposed scheduler is willing to download, whereas $\Delta u_i(n)$ models a network induced time-varying disturbance that limits the actual number of base layer chunks that a peer can download. That limitation can be due either to the scarce bandwidth availability (bandwidth bottleneck problem), or to the absence of needed base chunks in the peer's neighborhood (content bottleneck problem [24]). When the value of $u_i(n)$ computed by the regulator is equal to zero, meaning that no more base chunks are required by the control action, enhancement chunks can be also downloaded. In these circumstances, in fact, the peer randomly chooses a chunk to download among the ones offered by the sender and not jet held by the peer. To design the controller we use the standard $\mathcal{Z}$-transform technique [23]; that is, transfer functions are used to model and analyse the input-output dynamics of the controlled system. In particular, our scheduling algorithm is based on the feedback control loop pictured in Fig. 1. In such a scheme, the reference queue level Q_R is compared against the actual queue level $q_i(n)$ and the resulting error $e_i(n) = Q_R - q_i(n)$ is fed into a Proportional Integral (PI) regulator with transfer function $G_c(z)$ to compute $u_i(n)$. We have chosen a PI controller because it is able to filter out the continuous component of the disturbance $\Delta u(n)$ [25], [23], even if it is not known. It is worth to note that no delay is present in the feedback branch because the control algorithm has been designed and implemented on the receiver side, allowing the receiver to instantaneously know $q_i(n)$ values. The transfer function of the PI controller, computed as the

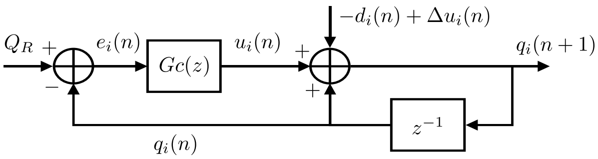

Fig. 1. Block Diagram of the controlled system.

$\mathcal{Z}$-transform of its discrete-time pulse response [23], is

$$G_C(z) = K \cdot \left(1 + \frac{1}{T}\frac{z}{z-1}\right), \tag{2}$$

where K and T are non negative constants. To tune the PI regulator, the following theorem has been exploited.

Theorem 1: The system shown in Fig. 1 is asymptotically stable if and only if the following inequalities are satisfied:

$$\begin{cases} 0 < K < 2 \\ \frac{K}{T} > 0 \\ 4 - 2K - \frac{K}{T} > 0 \end{cases} \tag{3}$$

Proof: The demonstration can be easily derived by applying the Jury criterion [23] to the characteristic polynomial $p(z) = T(z-1)^2 + K(T+1)z - KT$ of the system in Fig. 1.

It is worth noting that the signal $d_i(n)$ represents the number of base layer chunks decoded at the i-th peer during the n-th time step. As a consequence, assuming a one-to-one frame to chunk mapping, and a constant inter-chunk time, $d_i(n)$ can be modeled as a delayed step function $d_i(n) = d \cdot 1\,(\,n - M)$, with width d and playout delay equal to M time steps. Therefore, its $\mathcal{Z}$-transform is $D_i(z) = d\frac{z^{M+1}}{z-1}$.

Now, in absence of disturbance $\Delta u_i(n)$, it is easy to derive the steady state buffer level $q_i(\infty)$ using the final value theorem [23]. In fact, by equating the control loop in Fig. 1 and considering that the reference signal Q_R is a time step with $\mathcal{Z}$-transformate equal to $Q_R \cdot z/(z-1)$, we can easily derive the asymptotic queue length:

$$\begin{aligned} q_i(\infty) &= \lim_{n\to\infty} q_i(n) = \lim_{z\to 1}(z-1)Q_i(z) = \\ &= \lim_{z\to 1}(z-1)\frac{\frac{Q_R z}{z-1}G_c(z) - D(z)}{z-1+G_c(z)} = Q_R\,. \end{aligned} \tag{4}$$

III. Performance evaluation

The proposed chunk scheduler has been tested in a heterogeneous mesh-based P2P overlay, through the means of the P2P-TVsim NAPAWINE simulator, available on line [18].

The algorithm has been compared with respect to the Random Useful (RU) scheduler and an ideal optimal content-aware scheduler, prioritizing base layer delivery. When RU scheduler is employed, the receiving peer randomly selects an useful chunk among the ones offered by the sender peer, without distinction between base and enhancement layers [9]. The optimal scheduler has been designed by ideally assuming that the base layer download rate of each peer is at least equal to the base layer source rate in all time instants. To explain how this constraint has been imposed, the following basic notation has been considered:

- R_b: average base source rate;
- R_e: average enhancement source rate;
- $Up_{j,i}(t)$: upload bandwidth from the j-th to the i-th peer at the time instant t;
- $BW_i(t) = \sum_j Up_{j,i}(t)$: aggregate download bandwidth of the i-th peer;
- $Pb_i(t)$: probability to select a base chunk from those offered by a generic peer in the neighbourhood of node i at the time instant t;
- $Pe_i(t)$: probability to select an enhancement chunk from those offered by a generic peer in the neighbourhood of node i at the time instant t;

The ideal scheduler works as the random one if the available bandwidth $BW_i(t)$ is larger then the full quality video rate R_b+R_e (in this case we do not need to assign a higher priority to the base layer).

When the available bandwidth $BW_i(t)$ is smaller that the aggregate source rate (R_b+R_e), the ideal scheduler sets $Pb_i(t)$ and $Pe_i(t)$ as follows:

$$Pb_i(t) = \min[R_b/BW_i(t), 1], \tag{5}$$

$$Pe_i(t) = 1 - Pb_i(t). \tag{6}$$

Eq. (6) grants a quota of available bandwidth equal to the base rate to download the base layer. Eq. (6) assigns the remaining quota of bandwidth $BW_i(t) - R_b$. From eqs. (6) and (6), it clearly turns out that the ideal scheduler requires to know in advance several information that are not usually available in real P2P scenarios. As a consequence, it has to be considered as a theoretic performance bound. On the contrary, as explained in the previous section II, our scheduler does not requires any prior knowledge about the P2P system.

A. Simulation Scenario

The simulated P2P scenario consists in a hybrid push/pull system, in which each transmitting peer advertises available chunks to a randomly selected neighbour using a proper signalling message. On the other side, after a peer receives such an advertisement, it select the chunk to download. So that, in our system, the chunk selection is effectively done by the receiver. The overlay topology chosen for all the simulations is the General Random Graph, i.e. every peer is connected to g peers randomly chosen in the overlay network, where g is the degree of the overlay graph; here g has been set equals to 20. The number of peers is 1000, and the number of streamed chunks is always 2000, for a video sequence lasting 80 s. Playout delay has been set equal to 10 s (we have also tried several other playout delay values obtaining very similar results not reported here for lack of space). The simulation scenario is heterogeneous, i.e., peers composing overlay have different upload bandwidth. Average available bandwidth is 1.3 Mbps, and peers can be divided into 4 classes according to their upload bandwidth: 15% of peers are first class peers, with 5 Mbps upload bandwidth; 30% of peers have upload bandwidth of 1 Mbps; 35% have a 0.65 Mbps band; the other 20% are free riders not contributing to video delivery.

To highlight the effect of content-aware schedulers and their effect on video quality, a real-encoded VBR video will be used. The video sequence chosen for simulations is the 352x240 videoclip *Pink*, by Aerosmith. The sequence has been encoded using H264/AVC SVC temporal scalability [10], based on hierarchical frame B prediction. According to the chosen encoding scheme, four types of frames are available, that, in order of importance, are: IDR, P, B, b. A 29 frames long open-GOP structure has been used, following the sequence IDR 7xP B b b. In our simulations, the encoded stream has been divided into 2 layers: base layer made up of IDR and P pictures, and enhancement layer composed of B and b pictures. Original sequence temporal resolution is 25 fps. The chosen base layer allows to reconstruct a sub-sampled version of original video sequence at a rate equals to 6.25 fps. A one-chunk to one-frame mapping has been chosen. For video encoding and decoding, *x264* libraries and *ffmpeg* codec [26] have been used.

B. Simulation results

Simulation results are reported comparing proposed algorithm with RU scheme and the designed optimal scheduler.

Simulations have been run by varying both the source video rate (from 1 Mbps to 2.2 Mbps) and the setting point Q_R (from 10 to 60). Obviously, Q_R cannot exceed the maximum number of base chunks that can be present in the decoding buffer, i.e., the maximum number of base chunks which are generated in a time interval equals to Playout Delay. In our case, given the GOP structure and the temporal video resolution, the maximum number of enqueued base chunks is 68. For each simulation, we measured: (i) the mean chunk loss percentage, averaged over all peers; (ii) the mean base and enhancement loss percentage, computed taking into account, respectively, the number of base and enhancement chunks present in the sequence; (iii) the average PSNR, computed over 50 randomly selected peers. During decoding process, an error concealment strategy that replaces each lost frame with the previously decoded one has been used, because current commercial decoders do not implement error concealment when whole frames get lost [27] [26]. Fig. 2 shows the total number of lost chunks as a function of the video rate. Note that that the chunk loss ratio increases with the load for all considered algorithms. An important point to remark is that the random scheduler provides the smallest chunk loss ratio. Nevertheless, in order to compare the performance of the considered algorithms with respect to the service provided to users, we need to distinguish among base and enhancement loss ratios. Fig. 3, reporting only the number of lost base chunks, demonstrates that the random scheduler exhibits a base layer chunk loss ratio one order of magnitude larger than the ideal one. The performance of our scheduler instead ranges from that of the random one to that of the ideal one as Q_R moves from 10 to 60. The reason is that, increasing Q_R, a higher amount of base chunks is buffered at client side, with a higher probability to have already received also oldest chunks, which are close to their deadline. This effect is clearly demonstrated in Fig. 4, showing base layer buffer levels of a randomly selected peer.

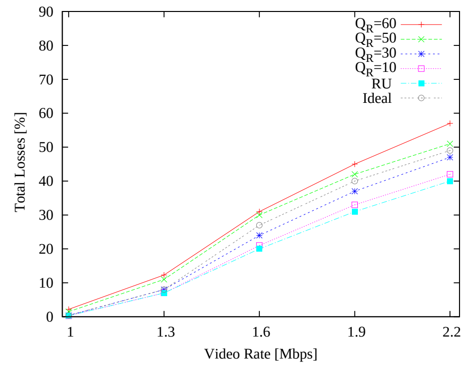

Fig. 2. Average Chunk Loss Percentage.

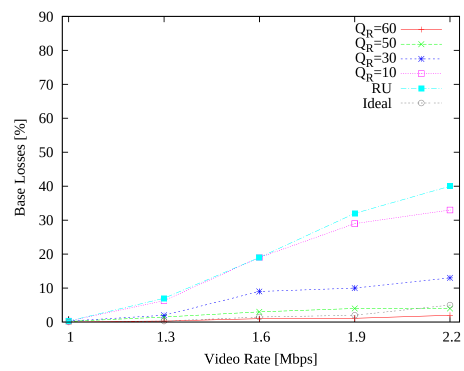

Fig. 3. Average base Loss Percentage.

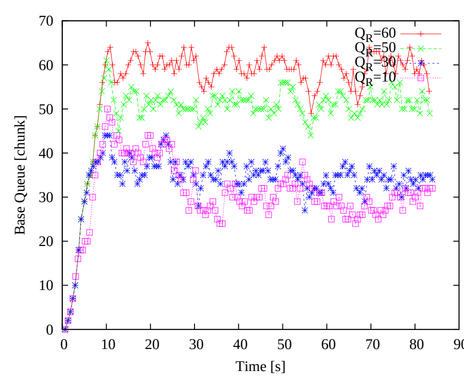

Fig. 4. Base decoding buffer level for a randomly chosen peer.

Regarding the enhancement layer, Fig. 5 shows that the random scheduler experiences the lowest enhancement chunk loss ratio (very smaller than in the ideal case). This obviously is a not efficient behaviour because to maximize the system performance the scheduler should give more priority to the base chunks and not to the enhancement ones. Also in this case, the behaviour of our scheduler is equivalent to the random one using a low Q_R and to the ideal one considering a high Q_R.

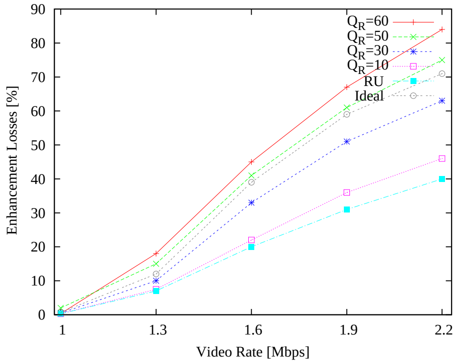

Fig. 5. Average Enhancement Loss Percentage.

To conclude, Fig. 6 pictures the measured PSNR and clearly highlights that the performance of the random scheduler are very poor (up to 15 dB smaller than the ideal one), despite it provides the lowest chunk loss ratio. Furthermore, also in this case our scheduler can achieve near-to-optimal performance using a high value of Q_R.

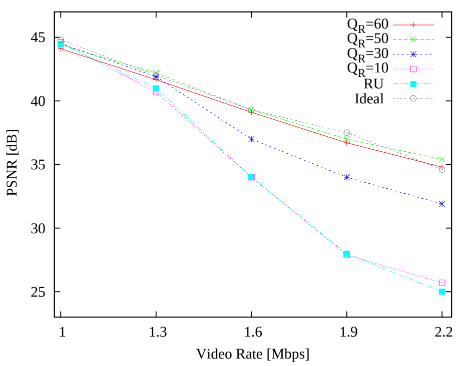

Fig. 6. Average PSNR.

IV. Conclusion

A novel scheduling algorithm for P2P-TV systems using layered video has been proposed in this work, designed using a control theoretic approach. Simulation results have clearly shown it is able to preserve relevant video data contained in base layer and to concentrate losses over enhancement chunks, reaching near-to-optimal performance. Future works will investigate its validity, in comparison with approaches recently proposed in literature, also when more than one enhancement layer is used and considering several kinds of overlay topology. Furthermore, enhanced versions of the scheduler will be conceived to take into account the inter-dependency among frames, thus improving the overall quality of reconstructed video.

References

[1] D. Braun, J. Buford, R. Fish, A. Gelman, A. Kaplan, R. Khandelwal, S. Narayanan, E. Shim, and H. Yu, "Up2p: a peer-to-peer overlay architecture for ubiquitous communications and networking," *Communications Magazine, IEEE*, vol. 46, no. 12, pp. 32–39, Dec 2008.

[2] J. Li, "Peer-to-Peer multimedia applications," in *In Proc. of ACM international conference on Multimedia*, Santa Barbara, CA, USA, 2006.

[3] M. R. Ceballos and J. L. Gorricho, "P2p file sharing analysis for a better performance," in *In Proc. of the 28th International Conference on Software Engineering*. ACM, 2006, pp. 941–944.

[4] J. Liu, S. G. Rao, B. Li, and H. Zhang, "Opportunities and Challenges of Peer-to-Peer Internet Video Broadcast," in *In Proc. of Special Issue on Recent Advances in Distributed Multimedia Communications*, 2008.

[5] E. Leonardi, M. Mellia, M. Meo, and A. P. Couto da Silva, "Bandwidth-Aware Scheduling Strategy for P2P-TV Systems," in *In Proc. of International Conference on Peer-to-Peer Computing*, Sep. 2008.

[6] R. J. Lobb, A. P. C. da Silva, E. Leonardi, M. Mellia, and M. Meo, "Adaptive overlay topology for mesh-based p2p-tv systems," in *NOSSDAV '09: In Proc. of International workshop on Network and operating systems support for digital audio and video*. ACM, 2009, pp. 31–36.

[7] N. M. Zapater and G. Bressan, "A Proposed Approach for Quality of Experience Assurance of IPTV," in *In Proc. of International Conference on the Digital Society*. IEEE Computer Society, 2007, pp. 25–30.

[8] L. Abeni, C. Kiraly, and R. L. Cigno, "On the optimal scheduling of streaming applications in unstructured meshes," in *In Proc. of International Federation for Information Processing Networking Conference*, 2009, pp. 117–130.

[9] T. Bonald, L. Massoulié, F. Mathieu, D. Perino, and A. Twigg, "Epidemic live streaming: optimal performance trade-offs," in *In Proc. of ACM SIGMETRICS*, 2008, pp. 325–336.

[10] H. Schwarz, D. Marpe, and T. Wiegand, "Overview of the Scalable H.264/MPEG4-AVC Extension," in *ICIP06*, 2006, pp. 161–164.

[11] M. Zhang, C. Chen, Y. Xiong, Q. Zhang, and S. Yang, "Optimizing the throughput of data-driven based steaming in heterogeneous overlay network," in *In Proc. of ACM Multimedia Modeling 2007*, Jan. 2007.

[12] P. Baccichet, T. Schierl, T. Wieg, and B. Girod, "Low-delay Peer-to-Peer Streaming using Scalable Video Coding," in *In Proc. of International Conference on Packet Video*, 2007.

[13] R. K. Rajendran and D. Rubenstein, "Optimizing the Quality of Scalable Video Streams on P2P Networks," *Computer Networks: The International Journal of Computer and Telecommunications Networking*, vol. 50, no. 15, pp. 2641–2658, Oct. 2006.

[14] X. Xiao, Y. Shi, and Y. Gao, "On Optimal Scheduling for Layered Video Streaming in Heterogeneous Peer-to-Peer Networks," in *In Proc. of ACM international conference on Multimedia*, 2008, pp. 785–788.

[15] L. Zhou, X. Wang, Y. Li, B. Zheng, and B. Geller, "Optimal Scheduling for Multiple Description Video Streams in Wireless Multihop Networks," *IEEE Communications Letters*, vol. 13, no. 7, pp. 534–536, Jul 2009.

[16] F. Fitzek, B. Can, R. Prasard, and M. Katz, "Overhead and Quality Measurements for Multiple Description Coding for Video Services," in *In Proc. of International Symposium on Wireless Personal Multimedia Communications*, Sep. 2004.

[17] Z. Liu, Y. Shen, K. W. Ross, S. S. Panwar, and Y. Wang, "LayerP2P: Using Layered Video Chunks in P2P Live Streaming," *Multimedia, IEEE Transactions on,*, vol. 11, no. 7, pp. 1340–1352., Aug 2009.

[18] P2ptvsim. [Online]. Available: http://www.napa-wine.eu/cgi-bin/twiki/view/Public/P2PTVSim

[19] S. Spoto, R. Gaeta, M. Grangetto, and M. Sereno, "Analysis of PPLive through active and passive measurements," in *In Proc. of International Parallel & Distributed Processing Symposium*. IEEE, 2009, pp. 1–7.

[20] J. Moreira, R. Antonello, S. F. L. Fernandes, C. A. Kamienski, and D. Sadok, "A step towards understanding Joost IPTV," in *In Proc. of Network Operations and Management Symposium*, 2008, pp. 911–914.

[21] T. Schierl, C. Hellge, S. Mirta, K. Grüneberg, and T. Wiegand, "Using h.264/avc-based scalable video coding (svc) for real time streaming in wireless ip networks," in *ISCAS*, 2007, pp. 3455–3458.

[22] R. Dusselman and X. D. Xiao, "Existing and Emerging Image Quality Metrics," in *In Proc. of Canadian Conference on Electrical and Computer Engineering (CCECE)*, 2005, pp. 1892–2899.

[23] K. J. Astrom and B. Wittenmark, *Computer controlled systems: theory and design*, 3rd ed. Prentice Hall, Englewood Cliffs, 1995.

[24] N. Magharei and R. Rejaie, "PRIME: peer-to-peer receiver-driven mesh-based streaming," *IEEE/ACM Trans. Netw.*, vol. 17, no. 4, 2009.

[25] G. Boggia, P. Camarda, L. A. Grieco, and S. Mascolo, "Feedback-based control for providing real-time services with the 802.11e MAC," *IEEE/ACM Trans. Netw.*, vol. 15, no. 2, pp. 323–333, Apr. 2007.

[26] FFMPEG codec. [Online]. Available: http://ffmpeg.org/

[27] JM reference software for H264/AVC. [Online]. Available: http://iphome.hhi.de/suehring/tml/