Assessing the impact of signaling on the QoE of push-based P2P-TV diffusion algorithms

Dario Rossi, Paolo Veglia
Telecom-ParisTech, Paris, France, name.surname@telecom-paristech.fr

***Abstract*—Internet video and peer-to-peer television (P2P-TV) are attracting more and more users: chances are that P2P-TV is going to be the next Internet killer application. In recent years, valuable effort has been devoted to the problems of chunk-scheduling and overlay management in P2P-TV systems. However, many interesting P2P-TV proposals have been evaluated in an idealistic environment: in this work, we instead study them by taking special care in defining realistic conditions for their evaluation. In particular we analyze the impact that signaling errors can have on a push-based P2P-TV overlay by means of simulation. Results are expressed in terms of both user-centric and system-centric indexes: our main finding is that push P2P-TV systems are deeply affected by even very rare signaling errors, which are often overlooked without justification.**

I. INTRODUCTION

Internet users habits are changing, and consequently the shape of Internet traffic is changing as well. The primate of Peer-to-Peer (P2P) file sharing, which for a long time was accounted to as constituting the bulk of Internet traffic, is now being challenged by video, which is indicated as the primary source of this shift [1]. The amount of video data is currently rising faster than any other type of service, and is estimated that all form of video (TV, Video on Demand, Internet, and P2P) will account for over 90% percent of Internet exchanges [1] in the next few years. However, as P2P is still growing, we can expect P2P-TV to account for a significant fraction of global Internet traffic.

In the last years, a number of different proposals have targeted mesh-based P2P streaming [2]–[6]. At the same time, such proposals have typically been studied in isolation, possibly focusing on very specific aspects of the system (notably, chunk scheduling policies), in possibly highly ideal settings (e.g., overlay-only studies, homogeneous settings, synchronous time-lines, perfect neighborhood knowledge, etc.). Thus, work remains to be done, especially in terms of a thorough comparison of the different algorithms under a common, more realistic, framework. The first aim of this work is thus not to propose any new algorithm, but rather to compare existing ones so to understand how the performance of these system declines under more realistic settings. We develop a custom simulator, taking special care to scenario realism, in which real network characteristics are enforced such as latencies, heterogeneity in peer capacity distribution, loss rates and so on. In particular we focus on the effect that a non-optimal signaling (due to packet delay or loss), can have on the performance of a push mesh.

We find that, signaling errors can significantly degrade the achievable performance, thus suggesting that optimal performance bounds, shown in previous theoretical work assuming perfect knowledge of the neighborhood buffer maps, can be hard to reach in practice. As such, we believe that signaling should not be neglected in future studies aiming at a realistic assessment of the quality provided by P2P-TV services.

II. RELATED WORK

P2P-TV is a relatively long studied subject, which has been the focus of many interesting work that we overview in this section. P2P-TV studies started from seminal work on single [2], [7] and multiple [8], [9] trees architectures, where video GOPs (possibly encoded using multiple descriptors) are pushed from the source down the trees. To overcome the inherent limitation of tree architectures, inspired by BitTorrent, the design of latest generation P2P-TV has moved toward chunk-based diffusion architectures featuring partly meshed architectures [3]–[8], on which we focus on in this work.

In mesh-based architectures, video chunks are either *pulled* or *pushed* on the overlay: in a pull system, receiver peers initiate the exchange, asking to some peers for the content they need; in push systems, instead, sender peers decide to whom send content. Interestingly, under particular hypothesis and settings, a given class of scheduling algorithms in push-systems, on which we focus on the following, has been proven [6] to achieve rate-optimality (and delay-optimality up to an additive constant term).

As far as the above systems are concerned, a further classification of the research work can be made. On the one hand, we have full blown systems [3], [10] that are often evaluated by means of middle to large scale deployments of real prototypes. On the other hand, there is valuable work [4]–[6] that instead adopts a possibly highly idealized view of the system and of the network models. Both approaches can help in understanding P2P-TV system: the former by giving realistic performance results and the latter by allowing to gather solid theoretic foundations for specific algorithms design choices.

This work aims at reducing the gap between the two approaches above, by performing a first realistic comparison of the last classes of work [4]–[6].

III. SYSTEM DESCRIPTION

We carry on the comparison with a custom chunk-level event-based simulator, which we make available to the community in [11], that takes into account several components,

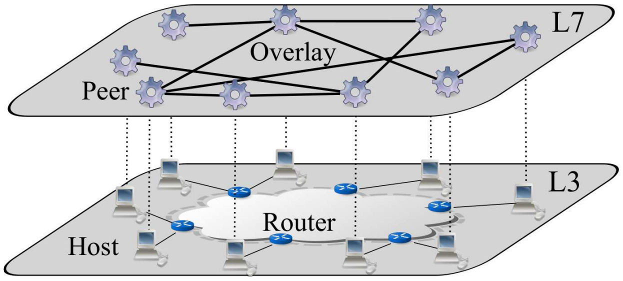

Fig. 1. Sketch of the evaluation scenario: overview of L3 and L7 components under study.

TABLE I
BREAKDOWN OF HOSTS INTO CLASSES.

Class	Ratio	BW_D	$\overline{\text{BW}_U}$	$\overline{t_{TX}}$
I	10%	∞	5.0 Mbps	20 ms
II	40%	∞	1.0 Mbps	100 ms
III	40%	∞	0.5 Mbps	200 ms
IV	10%	∞	0 Mbps	∞

which are visually presented in Fig. 1. From an high-level point of view, our testing environment consists of two layers: namely, the underlying physical L3 network and the logical L7 overlay, which are coupled by different models of their possible interactions.

From the L3 point of view, at the edge of the architecture we have end hosts, which are physically interconnected to the L3 network by access links that are modeled as a capacity–delay pair. Hosts are attached to edge routers, which constitute the entry point of P2P-TV traffic in the network. From the L7 viewpoint, hosts run P2P-TV applications, which we express in terms of the algorithms (e.g., chunk scheduling, peer selection, topology management) they implement, and of the overlay graph resulting by those algorithms. Finally, we model L7/L3 interaction by taking into account that, in the real world, different sources of error can slip in at any point of the process: specifically we consider the signaling aspects of the dissemination mechanism.

A. L3 Network

With L3 components we indicate objects in the physical world, such as (i) hosts and (ii) routers, that are interconnected by a (iii) network.

Hosts are machines running P2P applications instances, and are characterized by a physical interface to the L3 network. We consider $N_H = 2000$ hosts divided in different classes with different upload bandwidth BW_U according to table I. In each class i, the up-link capacity of each peer p is set to $\nu \cdot \overline{\text{BW}_U(i)}$ where ν is a random variable uniformly distributed in $[0.9, 1.1]$ (i.e., the actual up-link of each peer deviates at most 10% from the average for that class). Download bandwidth BW_D is set to ∞ as we do not enforce inbound traffic limitation. Corresponding chunk transmission time, with a fixed chunk size of 100 kbit, is indicated with $\overline{t_{TX}}$. Each host is bound randomly to one of the $N_R = 100$ routers, so that in average $N_H/N_R = 20$ hosts are attached per router.

As depicted in Fig. 1, routers are placed at the edge of the *network* and act as access points forming a logical full mesh at L3 level. Each router monitors the amount of traffic that it has to handle, discriminating traffic between *remote* (i.e., the traffic that it injects further down toward the core) and *local* (i.e., the traffic that is reflected toward other access links insisting on the same router). Routers directly yield a simple measure of traffic locality (which is independent from the network topology, from the Autonomous System (AS) level topology, from the router-to-AS mapping policy, etc). We denote the percentage of proximity traffic as $P\% = \text{local}/(\text{local} + \text{remote})$.

As far as the L3 network is taken into account, we consider the access link to be the bottleneck, with no queuing happening within the network core: as such, the network simply models the delay of the end-to-end path. In this case, the network topology is well represented by a *static* end-to-end latency matrix, where the latency essentially represents the propagation delay along links of the end-to-end path. We use a subset of the latency matrix provided by the Meridian project [12], where end-to-end delays are derived from real measurement performed among a large number of Internet hosts.

B. L7 Overlay

L7 overlay consists of *peers*, which are instances of L7 P2P-TV applications running on L3 hosts.

Each peer establishes and maintain several logical connections to other peers which form the overlay topology: we denote with $N(p)$ the set of peers in the neighborhood of p. To optimize system performance, peers need to perform topology management: i.e., they rearrange the overlay in order to exploit population heterogeneity, so to globally optimize the topology based on local decisions. Intuitively, if peers having higher capacity are located near the source, they can serve their neighbors quicker. Similarly, if such peers also serve a larger number of neighbors, this reduces the depth of the tree (i.e., the instantaneous tree followed by each chunk, which differs from chunk to chunk). In both cases, the above topology management decisions are beneficial to the whole system. In this work, we make use of a topology management process that runs continuously and adapts an initial random topology basing its decisions on peer capacity (see [13] for more details).

For the sake of simplicity, in this paper we do not consider churn (i.e., peers arrival or departure). While this choice may seem strange at first sight, especially given our attention to the realism of the scenario, we nevertheless believe it to be a reasonable one. Indeed, there is a large difference between P2P-TV and file-sharing user behavior: in fact, in file-sharing systems, users are not using the same content in a lively interactive fashion, which actually breaks correlation. As opposite, as shown in [14] (in the case of IP-TV applications) TV users are interested in programs with a given start time, equal for all users: this correlates peer arrivals, which means flash-crowds during the very first few minutes of the program. Most of the users stay then for the whole duration of the program, with few negative spikes during commercial breaks. Thus, considering

TABLE II
CHUNK SCHEDULER POLICIES

Scheduler		Description		
ru/r	[6]	Random useful chunk	/	Random peer
lu/r	[6]	Latest useful chunk	/	Random peer
lu/la	[15]	Latest useful chunk	/	Latency-aware peer
lu/ba	[4]	Latest useful chunk	/	Bandwidth-aware peer
lu/pa	[5]	Latest useful chunk	/	Power-aware peer

a stable P2P-TV population equals to consider periods (during a movie, or a sport event), where the peer population can be reasonably assumed to be roughly stationary.

Finally, as in mesh-push systems chunks are not received in play-out order, peers need to have a buffer-map B that represents the chunks received and stored into the peer memory (which can thus be uploaded towards other peers). Given a peer p, we indicate with $B(p)$ its buffer-map, and denote by $c \in B(p)$ the fact that peer p has received chunk c. The size of the buffer map $B(p)$ determines P2P-TV performance as in the following trade-off: large buffer maps reduce the chunk loss probability, but increase the time lag with respect to the source chunk generation time; conversely, small buffer maps reduce the play-out delay with respect to the source at the price of an increased chunk loss probability (as chunks that arrive later than the play-out delay are no longer useful and thus can be considered as lost).

C. L7 Scheduler

The ultimate goal of any P2P-TV system is to give to each peer a continuous stream of data: as such, peers must avoid having gaps in the buffer-map positions that are closer to the play-out deadline. The video exchange process is handled by a chunk scheduler, which acts whenever a peer can use the host upload bandwidth. In push systems, any peer p runs a scheduler that has to choose: (i) a chunk from its buffer map $B(p)$ and (ii) a destination peer among its neighbors $N(p)$.

Scheduling algorithms can be divided in two classes depending on the order in which the chunk/peer selection is made: in this work, we focus on algorithms that first choose the chunk to send and then the destination peer. We consider the chunk scheduling algorithms proposed in [4]–[6], [15] which we summarize in Tab. II. Loosely following [6], we denote each algorithm as *c/p* where *c* and *p* stand for *chunk* and *peer* selection algorithm respectively.

The simplest scheduler is the work-conserving ru/r, that selects a *random* chunk $c \in B(p)$ which it sends to a random *useful* peer $p' \in N(p)$, i.e., a peer that misses that chunk $c \notin B(p')$. We then consider a series of schedulers that select the *latest* chunk of their buffer-map, which then they send to a useful peer selected according to either a lu/r random strategy [6] or a *network-aware* criterion $lu/\{la, ba, pa\}$. As far as network-aware strategies are concerned, we consider a latency-aware lu/la strategy [15], a bandwidth-aware lu/ba strategy [4], and a power-aware lu/pa strategy [5] (i.e., based on the ratio of bandwidth and latency). Selection is performed by (i) ranking peers in the neighborhood using their properties (e.g., low latency, high bandwidth or power) and (ii) selecting them *probabilistically* (i.e., not in strict order), with a probability that decreases with increasing ranking.

Intuitively, lu/r aims at keeping the play-out delay from the source as low as possible by diffusing the most recent chunk at their disposal (i.e., the latest in their buffer-map $B(p)$). We consider ru/r for reference purposes, and lu/r as it is proven to be optimal in ideal homogeneous settings [6]. Network-aware schedulers [4], [5], [15] $lu/\{la, ba, pa\}$ are instead expected to enhance performance beyond lu/r, especially in case of heterogeneous realistic scenarios: in more details, lu/la aims at locally confining the traffic by proximity peer selection, lu/ba aims at reducing the chunk diffusion time by preferring peers with higher upload capacities and lu/pa aims at combining both benefits.

D. L3/L7 Interaction

A potential factor affecting the scheduling decisions is represented by errors that possibly affect the control information exchanged by peers. For example, control information can be *lost at L3*: in case of gossiping algorithms using UDP, such information would not be retransmitted, distorting thus the vision that each peer has on the system status. Under a slightly different perspective, P2P-TV system may wish to limit the amount of signaling traffic they inject on L3, by reducing the refresh rate of control information exchange: yet, we point out that information that is not *timely disseminated* at L7, may also induce a inconsistent view of the system state. Indeed, between two consecutive exchanges of information between any two peers, inconsistency can easily arise.

Considering mesh-push P2P-TV systems, such errors translate into out-of-date knowledge concerning their neighbors' buffer maps: in this case, a peer may decide to schedule the transmission of a chunk even if the destination has already received that chunk, resulting in an unnecessary chunk transmission (i.e., a chunk collision)

In order to assess the impact of signaling without being bound to specific signaling algorithms (nor to their settings), we resort to a high-level abstraction, and model errors due to packet loss or out-of-date system knowledge as error on the buffer-maps.

IV. SIMULATION RESULTS

Prior to address the description of the results, let us illustrate the general simulations settings. For each parameter under investigation, simulations are averaged over 6 repetitions: namely, we consider three instances of two different types of random overlay graph[1].

Each overlay consists of $N_H = 2000$ peers, of which we simulate a lifetime of 150 seconds, during which 1500 chunks of video stream are disseminated in the overlay. We consider a single source node, that streams video at an average rate of 1 Mbps, and consider 100 kbit fixed-size chunks (i.e., 10 new chunks are generated in each second). Statistics are

---

[1] The two overlays differ in the number of out-link per node, d_i, which can be fixed, $d_i = 10$, or modeled by a Poisson process with mean $\lambda_d = 10$.

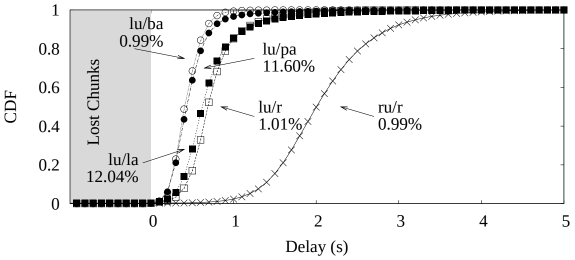

Fig. 2. Cumulative chunk delay distribution for different schedulers (ru/r and $lu/\{r, la, ba, pa\}$).

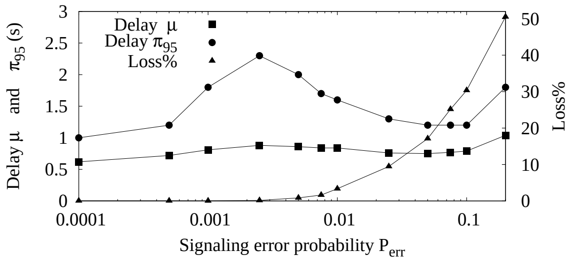

Fig. 3. Delay and chunk losses as a function of signaling errors

collected starting from 500th chunk, in order to avoid the initial transient. We consider that buffer maps store 50 chunks, which correspond to a play-out delay of 5 seconds.

A. Scheduling comparison

Curves in Fig. 2 show the cumulative distribution function (CDF) of chunk delays perceived by each peer (i.e., the temporal interval elapsed from the generation of the chunk at the source and its arrival to a given peer). Each curve represents a different scheduler, and we indicate lost chunks (i.e., chunks that arrived later than the play-out deadline) as chunk with negative delay (i.e., falling into the gray shaded zone.). The picture further reports the traffic-locality $P\%$ percentage along each curve. Recall that $P\%$ represents the fraction of chunks that do not traverse the core network (i.e., the destination host is attached to the same router of the sender host), and is thus an indication of network friendliness.

All schedulers share a limited fraction of lost chunk (which is very close to 0%), but they differ in the chunk delay and locality $P\%$ measures. Considering lu/r and lu/la, both strategies select the latest chunk and send it to peers which do not own it: lu/r selects a the destination peer at random, while lu/la proportionally prefers closer neighbors. Clearly, locality improves when lu/la latency-aware peer selection is performed with respect to lu/r. At the same time, notice that lu/r and lu/la are very close in terms of delay, despite lu/la preference of low latency neighbors. This can be explained with the fact that the propagation delay has a less prominent impact with respect to transmission delay, especially considering that chunks possibly travel multiple hops on low-capacity access links.

Consider indeed that the average propagation delay between any two peers is 35 ms, whereas from Tab. I we have that the average chunk upload times range from 20 ms for class-I peers to 200 ms for class-III peers. This entails that, at each hop, the transmission delay likely plays the most important role in determining the chunk delay performance: thus, merely choosing a peer which is closer in terms of the propagation delay does not allow to improve the overall system chunk delay performance.

Finally, the lu/ba and lu/pa schedulers achieves the best delay performance. Consider that both lu/ba and lu/pa assign scores according to the destination upload bandwidth, with the power-aware lu/pa scheme taking into account the propagation latency as well. Results confirm that uploading chunks to high-upload peers which can in turn diffuse them fast is beneficial to the whole system [4]. Moreover, we gather confirmation of the fact that explicitly taking into account peer latency improves locality $P\%$ but does not further ameliorate delay performance. In the following we focus on lu/pa since it shows both good delay characteristics and traffic locality.

B. Signaling impact

Here we investigate the effect of signaling errors on the system performance. Up to now, we have evaluated network-aware P2P-TV systems performance by assuming that peers have a perfect instantaneous knowledge of their neighbor state, situation which is unlikely to exist. In fact, due to overhead considerations, having a perfect signaling system would be a rather unrealistic assumption.

We model the impact of low signaling rates (or signaling messages losses at L3) as a degradation of the quality of system state knowledge in the distributed P2P system. In more detail, we model imprecision of system state knowledge as “usefulness” errors: in other words, with a given probability P_{err} a peer p can take a scheduling decision of chunk c toward p' which he believes to be useful (i.e., $c \notin B(p')$) despite it is not (i.e., $c \in B(p')$), which generates a collision.

Fig. 3 shows the mean μ and 95th percentile π_{95} delay, along with the chunk loss statistics as a function of the signaling error probability P_{err}. Notice that while the mean delay is roughly unaffected by signaling error probability P_{err}, a counter-intuitive phenomenon characterizes the π_{95} measure. Indeed, the 95th delay percentile increases until $\mathrm{P}_{err} = 1/400$, and afterward starts decreasing: this behavior is strongly correlated to the chunk loss rate, which starts rising roughly at $\mathrm{P}_{err} = 1/400$. What happens is that for increasing P_{err}, peers indeed receive chunks with higher delay, which in turns raises the probability that chunks arrive beyond the play-out delay (i.e., delay larger than 5s), and are thus marked as lost: as lost chunk are not accounted in the delay curve, the decreasing part of the peak is thus an artifact due to the play-out deadline.

To better assess the quality of video signal delivered to users, we evaluate the value of peak signal-to-noise ratio (PSNR). We consider the standard Soccer sequence (H624 format, CIF resolution, 300 frames @30Hz), and record for each peer the list of lost chunks. We then make use of Evalvid

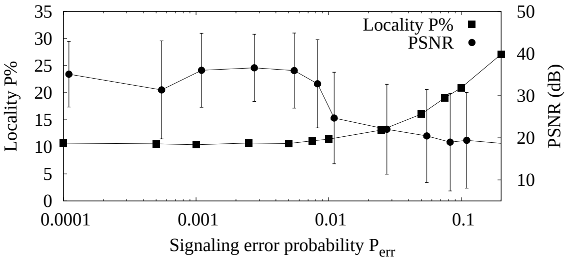

Fig. 4. PSNR and traffic locality as a function of signaling errors

[16] to evaluate video quality, by feeding the tool with the video sequences where we take into account the chunk loss pattern for each peer (notice that we have to loop the Soccer sequence, which lasts 10 seconds, for the whole simulation duration).

As PSNR evaluation is very time consuming and due to the size of our system, we resort to stratified sampling: specifically, we rank peers according to the amount of losses and select a 10-peers sample (corresponding to different loss amounts) out of the total $N_H = 2000$ peer population. Right y-axis of Fig. 4 reports the PSNR averaged over the 10-peers sample (bars report the standard deviation over the sample), which due to stratification is however representative of the whole population. It can be seen that PSNR drops significantly as soon as losses occur in the system: notice further that, since a PSNR$<$24 dB is generally considered as an indicator of extremely bad video quality, this suggest that buffer-map errors should be kept below 1/100. In our evaluation, as buffer map holds 50 chunks and a new chunks is generated every 100ms, this suggest that the signaling rate should be about as high as the chunk generation rate (which still do not entirely avoid signaling error due to loss or L3 latency).

Traffic locality exhibits a non-straightforward behavior as well: indeed, it can be seen from Fig. 4 (left y-axis) that locality increases as buffer-map errors increase. This can be explained by considering that the lu/pa scheduler preferentially selects nearby high-capacity peers. When the error probability is low, these peers will be fed first, but then, as peers rarely fail in estimating the usefulness of their decisions, other lower-capacity higher-latency peers get successfully serviced during the remaining upload slots. Conversely, when error probability is high, the scheduler will keep on sending chunks to close high-capacity neighbors, despite they likely already have received that chunk from other peers (notice that we forbid peers sending the same chunk to the same peer multiple times).

Finally, to analyze the impact of peer heterogeneity, we studied the performance break-down into classes which, for reason of space, we can only report qualitatively. We highlight that QoE is unfair with respect to classes: especially poorer peers, which are placed in the farthermost positions of the chunk diffusion trees, are more exposed to the diffusion inefficiency caused by signaling errors, and experience higher performance degradation in terms of both delay and loss.

V. CONCLUSIONS

In this work, we compare different state-of-art network-aware P2P-TV systems: i.e., systems whose main algorithms (such as chunk selection and topology management) are based, not only on content availability and overlay topology, but also on informed decisions concerning the status of the network (such as host capacity, path latency, etc.). By performing a thorough simulation campaign, we aim at comparing these algorithms and understanding in which measure signaling errors can affect their performance on user QoE: we conclude that even small error enforcement on signaling can significantly degrade the performance of an overlay and, in particular, degradation impacts more on peers with poor characteristics.

These results propel new research perspectives: in our future work, we aim at more closely addressing this issue, by further investigating QoE performance versus signaling algorithms (piggybacking, difference encoding, update frequency, etc.) and their overhead. Another interesting aspect we would like to address is the impact that measurement errors on other peers properties (such as access capacity and path latency) can have on performance of P2P-TV overlays.

VI. ACKNOWLEDGMENT

This work was supported by the European Commission under the FP7 STREP Project Network-Aware P2P-TV Application over Wise Networks (NAPA-WINE).

REFERENCES

[1] "Cisco visual networking index: Forecast and methodology, 2008-2013," 2009.

[2] D. Tran, K. Hua, and T. Do, "A peer-to-peer architecture for media streaming," *IEEE Journal on Selected Areas in Communications*, 2004.

[3] N. Magharei and R. Rejaie, "Prime: Peer-to-peer receiver-driven mesh-based streaming," *IEEE/ACM Transactions on Networking*, 2009.

[4] A. P. C. D. Silva, E. Leonardi, M. Mellia, and M. Meo, "A Bandwidth-Aware Scheduling Strategy for P2P-TV Systems," *IEEE P2P'08*.

[5] D. Ren, Y. Li, and S. Chan, "On Reducing Mesh Delay for Peer-to-Peer Live Streaming," in *IEEE INFOCOM'08*.

[6] T. Bonald, L. Massoulié, F. Mathieu, D. Perino, and A. Twigg, "Epidemic live streaming: optimal performance trade-offs," in *ACM SIGMETRICS'08*.

[7] R. Sherwood, S. Lee, and B. Bhattacharjee, "Cooperative peer groups in NICE," *Computer Networks*, vol. 50, no. 4, pp. 523–544, 2006.

[8] M. Castro, P. Druschel, A. Kermarrec, A. Nandi, A. Rowstron, and A. Singh, "SplitStream: high-bandwidth multicast in cooperative environments," in *SOSP'03*.

[9] R. Rejaie and A. Ortega, "PALS: peer-to-peer adaptive layered streaming," in *NOSSDAV'03*.

[10] F. Pianese, D. Perino, J. Keller, and E. Biersack, "PULSE: an adaptive, incentive-based, unstructured P2P live streaming system," *IEEE Transactions on Multimedia*, vol. 9, no. 8, pp. 1645–1660, 2007.

[11] http://www.napa-wine.eu/cgi-bin/twiki/view/Public/P2PTVSim.

[12] "Meridian project." http://www.cs.cornell.edu/People/egs/meridian/.

[13] R. J. Lobb, A. P. Couto da Silva, E. Leonardi, M. Mellia, and M. Meo, "Adaptive overlay topology for mesh-based P2P-TV systems," *NOSSDAV'09*.

[14] M. Cha, P. Rodriguez, S. Moon, and J. Crowcroft, "On next-generation telco-managed P2P TV architectures," in *IPTPS'08*.

[15] R. Bindal, W. Chan, J. Medved, G. Suwala, T. Bates, and a. Zhang, "Improving Traffic Locality in BitTorrent via Biased Neighbor Selection," in *ICDCS'06*.

[16] J. Klaue, B. Rathke, and A. Wolisz, "Evalvid–A framework for video transmission and quality evaluation," *Computer Performance*.