Hindawi Publishing Corporation
International Journal of Digital Multimedia Broadcasting
Volume 2010, Article ID 826351, 11 pages
doi:10.1155/2010/826351

Research Article

A Hybrid Approach to Assess the Network Awareness of P2P-TV Applications

Dario Rossi and Paolo Veglia

INFRES Department, TELECOM ParisTech, 46 rue Barrault, Paris, France

Correspondence should be addressed to Dario Rossi, dario.rossi@enst.fr

Received 24 June 2009; Accepted 30 November 2009

Academic Editor: Dave Marples

In this work, we develop a general methodology to assess the level of network awareness and friendliness of P2P-TV applications. The methodology is based on a combination of active and passive measurement techniques and can be applied to any P2P-TV system since it is designed to work considering such systems as a black-boxes. As an interesting case study, we then apply this methodology to PPLive, one of the most popular P2P-TV systems nowadays. Focusing on the video content distribution, we consider several per-path and per-peer metrics, investigating which of them mostly biases PPLive download preferences. Furthermore, in order to refine the picture of PPLive peer selection policy, we not only study the impact of different metrics in isolation, but also assess the joint impact of different metrics at the same time. Our main finding is that PPLive seems mainly bandwidth greedy, but does not show any preference toward peer proximity based on RTT delay; at the same time, our results also suggest that this choice alone may provide a nonnegligible level of geographical clustering among peers as a beneficial side effect.

1. Introduction and Motivations

We agree with [1] that "*the Obama inauguration marks a historic day in US politics and a remarkable day for the popularity of Internet streaming video. We look forward to watching more great things to come.*" Live streaming of video content over the Internet is finally hitting the masses: it is not hype that P2P-TV systems are candidates for becoming the next Internet killer applications, as testified by the growing success of several commercial systems, among which PPLive [2] is perhaps the most popular.

Yet, despite valuable research already investigated such commercial applications [3–15] still little information is available about their internal algorithms, which are proprietary and closed. Specifically, a major concern is that, unless such systems are considerate of the underlying network, the very same potentialities of P2P-TV may constitute a worry for network carriers, especially in the case of HD streaming. Indeed, notice that while download rate is limited by the stream rate, the upload rate may grow much larger depending on the number of served peers [3]. Furthermore, a P2P application confining most of the generated traffic within the Autonomous System (AS) could noticeably reduce operators' transit costs on peering links [5].

In the last years, a significant research effort in P2P networking has been devoted to techniques for achieving a better mapping between the overlay topology and the underlying physical network [16–18]. This can be obtained by biasing the peer selection process, so to choose overlay neighbors based on their proximity on the underlying IP network, that is, in other words to make the overlay "aware" and "friendly" toward the underlay. Such network-aware techniques can be either enforced at the overlay level alone (by inferring peer proximity based on latency measurement, coordinates systems [16], Autonomous System information [17], etc.) or even by allowing ISPs to participate in this process [18]. At the same time, we point out that most of these techniques have been developed in the more general context of P2P applications and have only more recently been cast to P4P-TV [19–24] which can be explained by the fact that other classes P2P applications, such as file-sharing, can be more easily engineered since they lack the need to preserve the temporal coherence of the fetched content.

Although a number of relevant works [3, 4, 7, 10–13] recently targeted the study of popular Internet P2P-TV applications, still the question remains whether such systems are aware of (and behave friendly with respect to) the underlying IP network, which is precisely the aim of this work. From a high level perspective, we can define two categories of metrics that pertain to network awareness: *pathwise metrics* (such as RTT delay, available bandwidth, packet loss, etc.) are determined by the conditions on the end-to-end path between two peers in the overlay; conversely, *peerwise metrics* (such as Autonomous Systems, geographical location, /16 IP prefix, access capacity, etc.) only depend on properties of a single peer. Our methodology (i) exploits *active techniques to artificially enforce pathwise metrics* in a controlled testbed, while (ii) adopts a *passive technique to infer preference of peerwise metrics* from observation of the traffic, gathered from multiple active probes, in uncontrolled live experiments. We point out that we implemented a slightly modified version of the methodology presented in this work in [14], which is available as an open source software tool at [15].

We apply this methodology to the study of PPLive where the combination of both techniques allows us to draw conclusions that are otherwise precluded using either methodology alone. By means of active measurement, we find that PPLive is mainly bandwidth greedy, but does not show any preference concerning, for example, IP distance or RTT delay. As ISP friendly peer selection and, more generally, network awareness in P2P system has only recently become a topic of interest, it would have rather been surprising if PPLive already explicitly considered this. Yet, by means of passive measurement, we further observe that bandwidth preference alone may provide a nonnegligible level of geographical clustering among peers as a beneficial side effect. While this is an interesting and positive finding, the overall results suggest that supplementary effort is needed in order to further increase the level of network friendliness—which, as PPLive is continuously evolving [25], may very well happen in the near future.

Summarizing, the main contributions of this paper are as follows.

(i) First, we propose a black-box methodology, based on a combination of active and passive measurement techniques, to assess the level of network awareness and friendliness of currently deployed Internet P2P-TV applications.

(ii) Second, we apply our methodology to the analysis of PPLive finding that geolocalization, while not explicitly enforced by the application, actually arises as beneficial side effect of bandwidth preference.

(iii) Third, the methodology allows to investigate the relative preference of different metrics as well; in the case of PPLive, we find that the peer selection process is continuously updated, with a relative preference among pathwise properties that depends on the actual magnitude of the metrics.

The reminder of this paper is organized as follows. After having overviewed related effort in Section 2, we describe the methodology and dataset in Section 3. We then apply the methodology to the analysis of PPLive reporting experimental results of active and passive techniques in Sections 4 and 5, respectively. Finally, Section 6 concludes the paper.

2. Related work

Recently, live streaming P2P-TV systems attracted the attention of the research community: as a result, a number of relevant works exist that exploit measurement techniques to study the behavior of such systems [3–13].

Many valuable work considers a single system, which is analyzed by active crawling as in the case of PPLive [3], CoolStreaming [4], and UUSee [5]. Other works instead focus on very specific aspects of a P2P streaming system, for example, node degree of popular versus unpopular channels [6] and node stability [7], while quality of service is of concern in [8, 9]. Finally, work also exists that attempts at comparing similarities and differences of the above systems as [10], which analyzes the traffic pattern generated by these applications from both a network as well as from a transport layer perspectives. Only very recently, work started to appear that focuses more closely on the study of P2P-TV network awareness [11–13]. Yet, although these works share some similarity in their aim with ours, as detailed in the following the adopted approaches are rather different.

For instance, authors in [11] exploit the application-layer logs of UUSee: interestingly, among their finding they observe a clustering phenomenon among peers in the same ISP, similar to the one pointed out in this paper. At the same time, we stress that our work differs from [11] concerning two important points. A first difference arises in the methodology employed, which in the case of [11] limits the applicability of the effort, indeed, authors not only have knowledge of the P2P-TV system inner workings, but also base their analysis on application-layer logs, which requires thus to instrument the application under analysis and is thus clearly not applicable in case of closed source proprietary systems. A second difference arises instead in the analysis of the results, and on the conclusions that can be gathered. The authors in [11] state that UUSee actually biases the selection of its neighbor peers via both (i) bottleneck bandwidth inference as well as (ii) RTT delay measurement. The authors further observe that a significant fraction of neighbor peers falls into the same ISP, despite that UUSee does not explicitly take into consideration ISP membership. These observations allows them to conclude that the reason behind the clustering is that “as connections between peers in the same ISPs have generally higher throughput and smaller delay than those across ISPs, they are more inclined to be chosen as active connections.” Interestingly, in our case we show that a similar geographical clustering can be observed in PPLive, which we also show *not* to be sensitive to RTT preference. To the best of our knowledge, this clustering effect solely induced by bandwidth preference has not been observed by other researchers to date.

A first step toward a methodology that can be applied to any system as a black-box is undertaken by both [12] (which, however, limitedly exploits an active measurement technique) and in our earlier work [13] (which instead only considers a passive measurement technique). In more detail, authors in [12], set up an active testbed to investigate the congestion control algorithms of different P2P-TV applications. Using active probes, authors enforce pathwise properties (such as artificial bandwidth limitations, packet loss, and delay) and examine P2P-TV reaction to adverse network conditions. Inspired by [12], we refine their setup, by (i) considering a more controlled setup and by (ii) jointly applying different impairments, so to assess the relative importance of multiple pathwise properties.

However, not all metrics potentially exploited by the overlay for neighbors selection and chunk scheduling can be artificially enforced via active measurement techniques—as, for instance, is the case of peer geographical and AS locations. In [13] we thus investigated such peerwise properties by means of a large pan-European measurement campaign, exploiting a purely passive measurement approach. In this work, we exploit the same dataset considered in [13] by performing a correlation-based analysis inspired by Principal Component Analysis (PCA) technique. Thus, although the dataset used in this work is the same as in [13], the analysis technique differs from the simpler one adopted in our earlier work [13], where we quantitatively weighted the relative amount of bytes exchanged with a peer that exhibits a specific peerwise property as an indication of P2P-TV system preference toward that property.

In this work, we define a methodology that jointly exploits active and passive measurements, encompassing thus and going beyond [12, 13]; by applying the methodology to the analysis of PPLive as a case study, we show that only the combination of both methodologies allows us to draw conclusions—such as the geographical clustering effect induced by bandwidth preference—that are otherwise precluded using either methodology alone.

3. Methodology

As previously outlined, our aim is to define a methodology able to tell whether a P2P-TV system has some level of knowledge of the underlying IP network, and whether it exploits this knowledge to bias the selection of the overlay neighborhood—especially for what concerns the chunk download preference. From a high level perspective, we can define two categories of metrics that pertain to network awareness.

(i) *Pathwise metrics*, such as IP path length (hops), loss rate, RTT delay, and available bandwidth,, are determined by the conditions on the end-to-end path between two peers in the overlay.

(ii) *Peerwise metrics*, such as Autonomous Systems, geographical location, /16 or /24 IP prefix, access capacity, instead only depend on properties of a single peer.

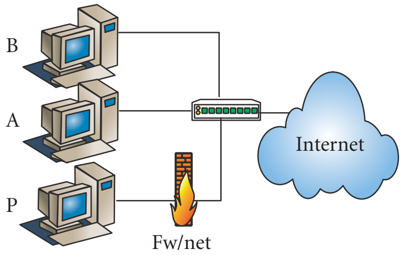


Figure 1: Pathwise metrics: active testbed setup.

In this work, we use two separate sets of experiments to assess the awareness of a P2P-TV system with respect to metrics falling in either of the two above categories.

(i) On the one hand, we exploit an *active measurement* technique to enforce controlled artificial conditions (such as path length, delay, loss and bottleneck bandwidth) on a specific network path in an Internet-scale testbed.

(ii) On the other hand, we adopt a live *passive measurement* approach, where we perform contemporary live measurement of unmodified peers from multiple vantage points, in order to investigate properties (such as AS or geographical location) belonging to real overlay peers.

3.1. Pathwise: Controlled Testbed. For preference related to path metrics, we setup a controlled testbed to enforce artificial network conditions as in [12], from which our approach differs for two main reasons. First, we decide to *completely* control the path metrics. This means that, unlike [12] where impairments are additionally enforced beyond the actual network conditions, we know precisely the conditions of the different peers involved in the experiments. Second, we not only test the impact of each metric in isolation, but also investigate their combined effect as well.

The configuration used for all active experiments is shown in Figure 1. We use three modern desktop PCs equipped with dual-core processor running native installations of Windows XP and of the P2P-TV application, which in our case is PPLive 2.4. Two machines A and B are connected to a network switch through their 100 Mbps Ethernet interfaces. Traffic is observed at the probe PC P, which is connected to the switch through a machine, referred to as *Fw/Net* in the picture, running a linux kernel 2.6 and acting as a bridge, firewall, and network emulator. Notice that a large population of users, as well as the primary source of the video itself, is reachable through the Internet.

At start-up, all machines A, B, and P run undisturbed clients for 5 minutes. During this start-up period (where we verify that play-out starts and that the clients are visually synchronized within a 1-2 seconds range), P naturally receives most of the traffic from Internet hosts. Then, at time $F_{\text{on}} = 5$ minutes, firewall rules are established at *Fw/Net* to block traffic coming from the Internet toward P, which can thus only receive traffic coming from either A or B. In this

Table 1: Peerwise metrics: multiple live measurement setup. Host IDs are internal to the site.

Host	Site	CC	AS	Access	Nat	FW
1–4	BME	HU	AS1	High-bw	—	—
5			ASx	DSL 6/0.512	—	—
1–9	PoliTO	IT	AS2	High-bw	—	—
10			ASx	DSL 4/0.384	—	—
11–12			ASx	DSL 8/0.384	*Y*	—
1–4	MT	HU	AS3	High-bw	—	—
1–3	FFT	FR	AS5	High-bw	—	—
1–4	ENST	FR	AS4	High-bw	—	*Y*
5			ASx	DSL 22/1.8	*Y*	—
1–5	UniTN	IT	AS2	High-bw	—	—
6–7				High-bw	*Y*	—
8			ASx	DSL 2.5/0.384	*Y*	*Y*
1–8	WUT	PL	AS6	High-bw	—	—
9			ASx	CATV 6/0.512	—	—

case, hosts A and B will still receive the video from the remote Internet peers, but our probe P will be forced to receive the totality of the video from A and B.

We then introduce, starting at $R_{\text{on}} = 10$ minutes, artificial network emulation rules (such as packet loss, RTT delay, bottleneck bandwidth limitation, etc.) on the path that joins our probe P to the hosts A and B from which he is receiving the video content. We point out that our aim is to understand how the system biases its peer selection during *normal* operation; we verify that probe P is correctly receiving the video stream. When a path metric X is considered in isolation, we artificially worsen network conditions (e.g., increase packet loss rate, delay, etc.) solely on the path from machine A to P, by properly configuring the queueing discipline on *Fw/Net*. Rules on path from B to P are instead enforced only to investigate the relative importance of different metrics (e.g., delay on A → P and loss on B → P). Finally, artificial network conditions are turned off at time $R_{\text{off}} = 20$ m and firewall limitations are removed at $F_{\text{off}} = 25$ m.

The above setup allows us to focus on the *breakdown* of the bit-rate received at P between its contributors (i.e., A, B, and Internet hosts), and to express in a visually simple way the network awareness of the P2P-TV application. Consider indeed the period when artificial network emulation rules are applied on the path from A to P, for instance, clearly, in absence of bias with respect to a given metric X, we expect the breakdown of the traffic received at P to be unaffected from variations of X. Conversely, a varying breakdown will reflect system awareness to X, with the extent of the breakdown variation as rough indication of the system sensitivity to X.

3.2. Peerwise: Multiple Live Measurement. The analysis of preference related to peer properties is instead based on a large testbed, setup in the context of the NAPAWINE project [26]. In this case, we mine the data gathered from the experiment in order to infer additional information concerning the P2P-TV system bias on peerwise properties.

The main idea is to use a *correlation-based* analysis of any given peerwise metric X and the amount of bytes exchanged between contributor peers. As peerwise metrics X, we will consider the Autonomous System (AS) and geographical Country (CC) properties. In this case, our aim is to test whether two peers that belong to the same AS/CC exchange more data than faraway peers, gauging the importance of X in the peer selection process. Deferring further details on the quantification of this bias, along with considerations concerning the fallacies of correlation-based analysis to Section 5, we now briefly describe the Internet-scale experiment setup used to gather the passive measurement dataset.

More precisely, partners run PPLive clients on PCs connected either to the institution LAN, or to home networks having cable/DSL access. In more detail, as summarized in Table 1, the setup involves a total of 46 peers, including 37 PCs from 7 different industrial/academic sites, and 7 home PCs. Probes are distributed over four European countries and connected to 6 different Autonomous Systems, while home PCs are connected to 7 other ASs and ISPs. Therefore, the setup is representative of a significant number of different network environments. Several 1-hour long experiments were performed during April 2008, when partners watched the same channel at the same time and collected packet-level traces. Since P2P-TV applications are mostly popular in Asian countries [3], we tuned PPLive to CCTV-1, a popular channel, during China peak hours. Nominal stream rate is 384 kbps, Windows Media 9 Encoder is used, and the perceived video quality is similar for all partners. We point out that during the experiments, several of our peers act as amplifiers [3]; that is, they exhibit an upload/download ratio significantly larger than 1, which further justifies potential ISPs worries concerning P2P-TV systems.

Results reported in this paper refer to 44 hours of video, during which our probes exchanged about 70 GBytes of data in 157 million packets with nearly 1 million external peers.

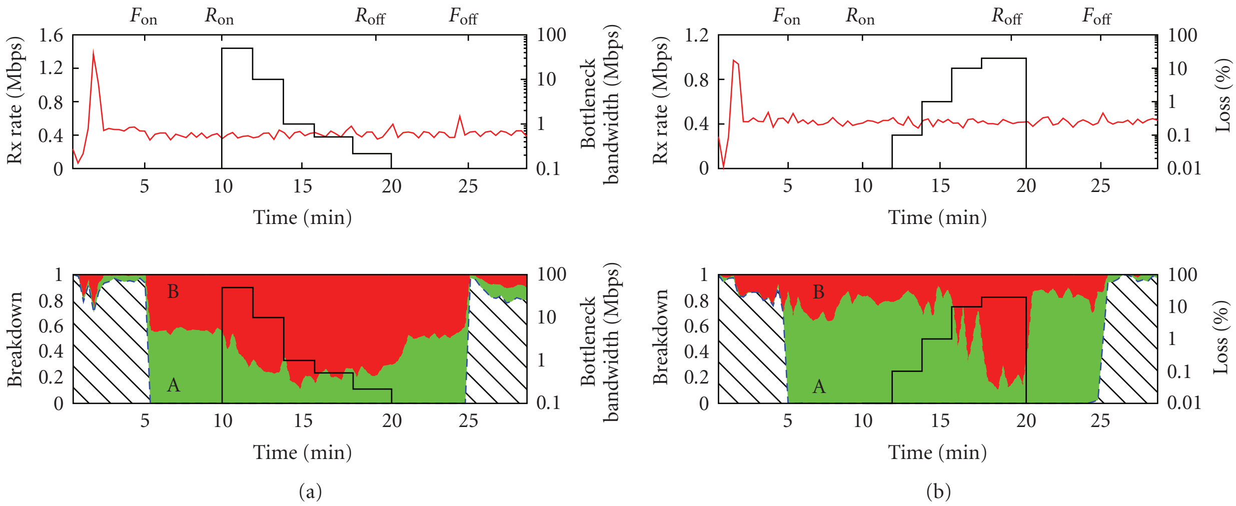


FIGURE 2: Pathwise metrics: PPLive aggregated received rate at P (top) and its breakdown between A, B and Internet hosts (bottom) for varying bottleneck bandwidth (a) and packet loss (b). Profiles of the bottleneck bandwidth and packet loss impairments are reported as solid black lines directly in the plot and refer to the right y-axis. PPLive shows a great sensitivity towards path capacity variations while it reacts only to very high loss rates.

We stress once more that, despite the dataset reported in Table 1 is a subset of the one considered in [13], nevertheless the analysis of the dataset carried on in Section 5 differs from our earlier work [13].

4. Experimental Results: Pathwise Metric

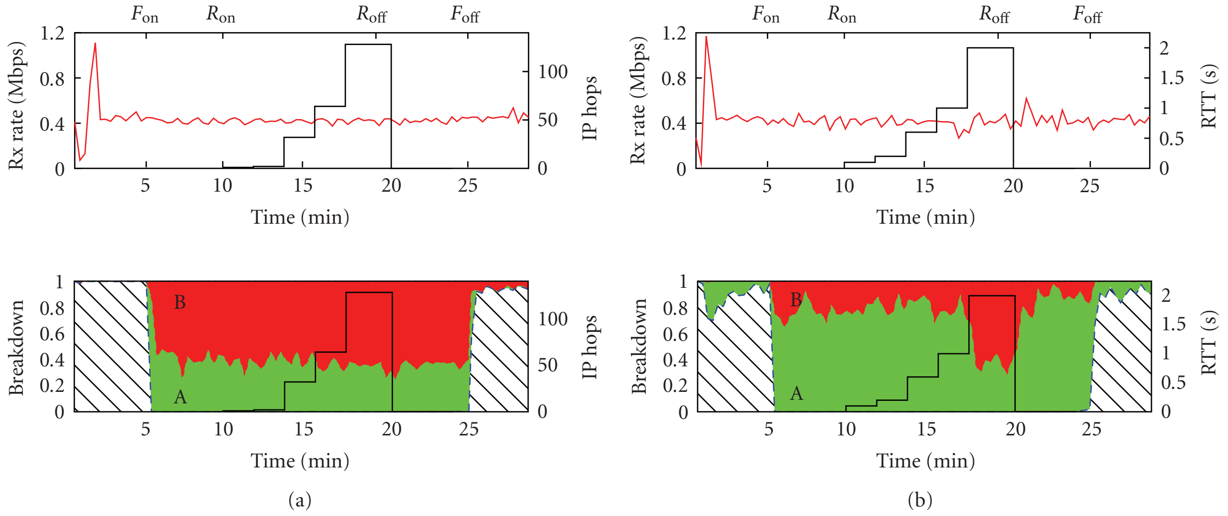
Let us start by inspecting pathwise preference, by means of the controlled testbed depicted early in Figure 1. Initially, we study pathwise metrics in isolation, enforcing either (i) decreasing bottleneck bandwidth, (ii) increasing packet loss rate, (iii) increasing RTT delay, (iv) increasing IP hop count on the path from host A to the probe P.

We then inspect the relative importance of the above metrics in the peer selection process by jointly considering different metric combinations, applying one condition (e.g., bottleneck bandwidth) on the path from the host A to the probe P, and a different condition (e.g., packet loss rate or RTT delay) on the B → P path.

4.1. Bottleneck Bandwidth. Results of the first experiment are reported in Figure 2(a). Time of the experiment runs on the x-axis, while the firewall start and end times are reported on the top axis as a reference. A decreasing bandwidth profile is enforced starting at R_{on} by means of a token bucket filter, with steps of $C = \{50, 10, 1, 0.5, 0.25\}$ Mbps every 2 minutes, as shown by the thick black line. Values of the bottleneck bandwidth are reported on the right y-axis, and the bottleneck is removed at R_{off}. The time evolution of the aggregated *received* rate at P is reported in the top portion of the plot, averaged over 20 seconds intervals. It can be seen that, after an initial start-up phase $t < F_{on}$ in which the incoming rate peaks up to 1.2 Mbps, the aggregated received throughput at P is steady around 400 Kbps, which account for both signaling and video traffic. Moreover, notice that the aggregate rate is undisturbed during the *whole* experiment, hinting to the fact that traffic shaping did not perturbed the perceived quality of service.

Bottom plot of the figure report the *breakdown* of the traffic incoming at P with respect to the different hosts that sent the traffic to P: host A is depicted at the bottom with light (green) color, host B with dark (red) color, and the remaining Internet hosts with a dashed pattern. It is easy to gather that, before firewall rules are in place $t < F_{on}$, more than 80% of the incoming traffic is received through Internet hosts. As soon as firewall rules start at $t = F_{on}$, P is forced to receive traffic exclusively from hosts A and B; since during $F_{on} < t < R_{on}$, no bottleneck bandwidth is enforced yet, the traffic splits roughly equally between A and B, as the network condition and play-out time of hosts A and B are alike. Then, as soon as a 50 Mbps bottleneck bandwidth kicks in at R_{on}, PPLive *immediately* starts preferring the unconstrained host B, which then provides the most significant portion of the traffic to P.

This observation is important as it means that (i) PPLive is extremely sensitive to the bandwidth and (ii) it may overreact or perform faulty bandwidth estimations. This behavior might be due to the token bucket shaper used to enforce bandwidth limitations, causing strange arrival patterns that mingle the bandwidth estimation algorithm. Moreover, packet time stamping may also bias the results (e.g., by poor timing due to clock drift, clock skews due to NTP synchronization, etc.). More likely, since packets of the same chunk are sent out in bursts [3], implementation of hardware card and drivers may play a very important role as well, especially due to *interrupt coalescing*. This feature, aimed at avoiding the overkill of raising an IRQ signal for every packet received, makes the cards wait during a short time-window for the arrival of other packets prior to notify the reception to the upper layers. Since packet time-stamping



(a)

(b)

Figure 3: Pathwise metrics: PPLive aggregated received rate at P (top) and its breakdown between A, B and Internet hosts (bottom) for varying IP hop distance (a) and RTT delay (b). Profiles of the IP hop distance and RTT delay impairments are reported as solid black lines directly in the plot and refer to the right y-axis. Here, PPLive shows no awareness of IP distance and a mild sensitiveness towards Round Trip Time.

is then performed by the OS, this means that interrupt coalescing has a possibly very nasty impact on the bandwidth estimation as well, because two consecutive packets received by the card during the same interrupt coalescing window will then be sent to the upper layer almost at the same time. As a consequence, bandwidth and capacity estimation tools that are based on packet pair or packet dispersion will rather measure the PCI bus speed more than bottleneck in the network.

Finally, bottleneck limitations are removed at R_{off}, which partly removes the breakdown bias toward host B; this holds until the firewall limitations are removed as well at F_{off}, after which contributors are again to be found mainly among Internet hosts.

4.2. Packet Loss. We then conduct similar separate experiments for the other considered metrics, enforcing a single impairment at any time. Results for the packet loss rate experiments are reported in Figure 2(b) using the same similar visual presentation (i.e., aggregated rate, breakdown, and packet loss profile).

We notice again that the incoming traffic rate at P is steady for any packet loss rate; indeed, since only the path A → P is impaired, P has still the possibility to receive the rest of the video from B. In this experiment, starting at time R_{on}, we increased packet loss rate experienced by host A using the profile $L = \{0.01, 0.1, 1, 10, 20\}\%$.

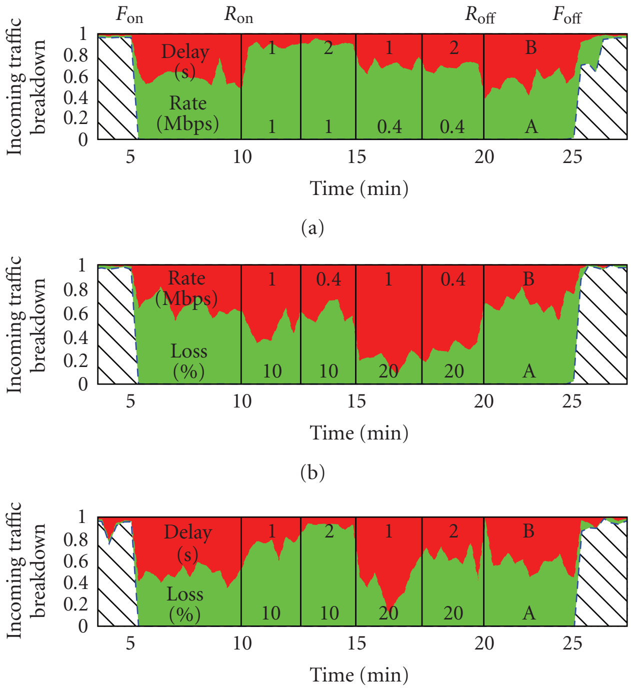
As the picture clearly shows, PPLive is not very sensitive to packet loss; noticeable changes in the breakdown happen only when packet loss percentage *exceeds* 10%—indeed when loss rate is 10%, still more than half of the data is downloaded from the impaired host A. If we couple this observation to the fact that, despite packet loss increases considerably, the sent traffic rate (measured at A and not shown in the picture) does not increases proportionally, we can conclude that PPLive seems to use an effective FEC techniques, as already observed in [12].

4.3. IP Distance. We then test whether PPLive is aware of the host proximity, which we measure in terms of the number of IP routers that packets cross on their path across the network. Applications using raw UDP sockets can infer IP proximity by means of the Time To Live (TTL) field of IP packet header, which is initially set to an OS-dependant value (namely, 60 or 64 for BSD and Linux, 128 for Windows) and then decremented by one unit at each hop in the network. We artificially increase the number of hops on the path from A → P by subtracting the number of additional hops $H = \{1, 2, 32, 64, 100\}$ from the IP TTL header field at Fw/Net.

As before, a change in the hop profile is enforced every two minutes, and the results are shown in Figure 3(a). As there is no noticeable change in the breakdown irrespectively of the additional hops values, we can conclude that PPLive is either unaware the IP hop distance between two hosts, or that its inner algorithms do not rely on this piece of information.

4.4. RTT Delay. Finally, we verify whether PPLive does instead take latency measurement into account. We increase the delay on the A → P path so that the Round Trip Time (RTT) equals $\mathrm{RTT} = \{0.1, 0.2, 0.5, 1, 2\}$ s, and report the results in Figure 3(b).

The plot clearly asserts that PPLive is not very sensitive to the delay, as the breakdown does not show any noticeable change until the round trip delay grows very large ($\mathrm{RTT} > 1$ s). We can thus conclude that PPLive peers do not bias their download policy in terms of nodes proximity, neither in terms of IP hop distance, nor in terms of delay. In other words, PPLive does not seem to implement proximity techniques such as [16–22].



(a)

(b)

(c)

Figure 4: Pathwise metrics: combined impairments, considering RTT delay versus bottleneck bandwidth (a), bottleneck bandwidth vresus packet loss (b), RTT delay versus packet loss (c). Relative importance of impairments depends on their actual magnitude.

Yet, another very important remark can be gathered from the picture. Indeed, when RTT > 1 s, the breakdown drastically drops; this suggest that PPLive *does* actually measure RTT, which is however not used afterward for the proximity-based video contributors selection. This behavior can be explained by recalling that one of the main aims of scheduling in P2P-TV streaming is to reduce as possible the play-out delay of the whole system. Therefore, despite that useful video content may be available at high RTT peers, such peers are preferentially discarded as they may not *timely* contribute to the video content delivery, and as such they would increase the whole system play-out delay. In other words, it seems that PPLive, streaming to keep a low end-to-end play-out delay, uses RTT as a "sanity-check" and disregard such peers on purpose to avoid the system pollution.

4.5. Combined Pathwise Metrics. In order to further refine the knowledge concerning PPLive network awareness, we investigate how PPLive reacts to different combination of impairments, so to sketch a relative order of importance of the above metrics. As we show that PPLive is not sensitive to IP hop count, we now limitedly consider packet loss, RTT delay and bottleneck bandwidth limitations, applying an impairment X on the A → P path, and another impairment Y on B → P at the same time.

For the sake of readability, we consider only a couple of values for each metric (i.e., $X_{hi} > X_{lo}$ and $Y_{hi} > Y_{lo}$) and investigate PPLive behavior on the four different operational points resulting from their combination $(X, Y) \in (\{X_{hi}, X_{lo}\} \times \{Y_{hi}, Y_{lo}\})$. Results are shown in Figure 4, which reports for the sake of readability the value of the impairment applied to a specific path directly on the plot. In this case, to avoid cluttering the pictures, we no longer report the aggregated received rate but limitedly depict its breakdown.

4.5.1. Delay versus Rate. Top plot of Figure 4 reports the case in which we apply a delay RTT = $\{1, 2\}$ s on the B → P path and enforce a rate limitation of BW = $\{0.4, 1\}$ Mbps on A → P. It can be seen that preference goes toward bandwidth limited host and is mainly driven by the bottleneck bandwidth; indeed, almost all content is downloaded from A when the rate limit is set to 1 Mbps, irrespectively on the delay toward B. Once the bottleneck rate drops to BW = 0.4 Mbps, the number of video chunks downloaded from B slightly increases (even though the rate limit would allow almost the whole content to be downloaded from A), but no noticeable effect of RTT variation is shown.

4.5.2. Rate versus Loss. Middle plot of Figure 4 refers to a rate limitation BW = $\{0.4, 1\}$ Mbps on B → P and loss rate $L = \{10, 20\}\%$ on A → P. In this case, contrary to the previous experiment, we see that both metrics have an impact in determining the breakdown. When $L = 10\%$ breakdown is roughly equal for A and B, with a slight bias toward A when bandwidth toward B drops at 0.4 Mbps. When losses instead grow to $L = 20\%$, the bandwidth limited host is always preferred, though the actual bandwidth limit still slightly influences the breakdown value.

4.5.3. Delay versus Loss. Finally, bottom plot of Figure 4 refers to a delay enforcement of RTT = $\{1, 2\}$ s on B → P and loss rate $L = \{10, 20\}\%$ on A → P. Again, both metrics play an important role in determining the breakdown, depending on the impairment level. As expected, loss $L = 10\%$ does not constitute a significant impairment, as such lossy path is preferred toward high RTT path, when RTT = 1 s, peer A is almost completely ignored. Behavior changes completely as soon as losses grow to $L = 20\%$, in which case breakdown favors completely B when RTT = 0.5 s and is more fairly split when RTT = 1 s.

The above observations allow us to conclude that the *relative* preference of pathwise metrics is weighted on the ground on the actual magnitude of the impairment. In principle, we point out that by exploring a wider number of (X, Y) impairment couples, it should be possible to get an even finer picture of the relative preference, but this falls outside the scope of this paper.

5. Experimental Results: Peerwise metric

In this section, we refine the picture of PPLive awareness by mining data gathered in the multiple-vantage point testbed. Following the methodology defined in [3], we extract from our traces about 16500 peers that contribute by providing video content. We focus on peers' Autonomous System (AS) and geographical location, which we represent by Country Code (CC) information. For each probe peer x in the testbed, we analyze the CC and AS properties of all its contributors peers y, gathered by *whois* and open IP databases queries. We point out that contributors y in this case may be either probes taking part in the experiment, or external peer of real users. As such, we no longer control the properties related to their path, and we need to infer them from packet level traces in case of need.

As core tool in this case, we use a correlation-based analysis, inspired by Principal Component Analysis (PCA) technique. While PCA is often used for dimensionality reduction—that is, to transform a set of correlated variables into a smaller subset of uncorrelated variables, called principal components—in our case our aim is to gauge the extent of the correlation, so to show the existence of a dependence (if any) between these variables.

More precisely, let us consider a set of N contributor peers $p_1 \cdots p_N$ observed during an experiment. By denoting with $X_i = X(p_i)$ the value of property X for peer p_i and similarly with $Y_i = Y(p_i)$ the value of property Y for the same peer, we measure the correlation between X and Y over the whole experiment as

$$\rho(X, Y) = \frac{\operatorname{cov}(X, Y)}{\sigma_X \sigma_Y} = \frac{E[(X - \mu_X)(Y - \mu_Y)]}{\sigma_X \sigma_Y}, \tag{1}$$

where μ_X and μ_Y are the means of X and Y over all samples, σ_X and σ_Y are the sample standard deviations of X and Y, respectively. Usually, (1) is referred to as the Pearson product-moment correlation coefficient which, dropping the sum bounds for the sake of readability, can be rewritten as

$$\begin{aligned}\rho(X, Y) &= \frac{\sum X_i Y_i - nE[X]E[Y]}{(N-1)\sigma_X \sigma_Y} \\ &= \frac{N \sum X_i Y_i - \sum X_i \sum Y_i}{\sqrt{N \sum X_i^2 - (\sum X_i)^2}\sqrt{N \sum Y_i^2 - (\sum Y_i)^2}}.\end{aligned} \tag{2}$$

5.1. Autonomous System and Geolocation. We are interested in assessing if PPLive is AS- and CC-aware, and whether its video scheduling policy exploits such information; that is, if in other words our PPLive probes tend to download video content from contributors falling in the same AS or CC. By mining the experimental data, we find that, despite only 1.3%(1.4)% of peers fall in the same AS(CC) of the probe, about the 12.8%(13.1%) of bytes are downloaded from them to further quantify this evident degree of geolocalization among contributors, we evaluate the coefficient of correlation ρ between the amount of bytes RX received from any given contributor x and the fact that this contributor belongs to the same AS or CC. Considering all probes x in the experiments, and by using the indicator function $I(x, y) = 1$ when both x and y belong to the same AS or CC, we obtain $\rho(\text{RX}, \text{AS}) = 0.21$ and $\rho(\text{RX}, \text{CC}) = 0.17$, respectively, which accounts for modest (though not negligible) correlation.

This is however surprising, since the controlled testbed early suggested that PPLive peers are greedy in terms of bandwidth, but that are not sensitive otherwise to the fact that contributors are "close" in underlay terms (e.g., IP distance or latency).

5.2. Bandwidth. We are therefore interested in assessing if (and to what extent) the geolocation can be a (rather desirable) *side effect* of PPLive bandwidth sensitivity. Therefore, we further evaluate the bandwidth (BW) between probes and contributor by measuring the throughput of chunks, which are typically sent out in packet bursts, to further investigate the existence of correlation between peerwise metrics.

5.2.1. Bandwidth Estimation Techniques. Since we are unaware of the technique actually employed by PPLive to measure the available bandwidth, we adopt a hands-on approach; we estimate BW using multiple techniques and require an agreement of our observations over *all* techniques. Considering only the downstream traffic direction of a contributing peer toward one of our testbed probes, we evaluate the BW over *windows* of fixed length. We express the window length in terms of either (i) a number of consecutive packets N or (ii) a temporal duration ΔT. Let us denote by t_i and B_i, respectively, the arrival time and size of the ith packet downloaded by probe x from contributor y during the current observation window. In case of fixed-length packet trains, we estimate the bandwidth BW_N over the current window as the amount of bytes carried by the train of consecutive N packets as

$$\text{BW}_N = \sum_{i=1}^{N} B_i/(t_N - t_1). \tag{3}$$

In case of fixed-duration trains, we estimate the bandwidth as

$$\text{BW}_{\Delta T} = \sum_{i=1}^{N(\Delta T)} B_i/\Delta T, \tag{4}$$

where $N(\Delta T)$ is the number of packets received during ΔT.

As far as the window length N and duration ΔT are concerned, we point out that their choice is made complex not only by the fact that we are unaware of the chunk size and chunk start time, but also from the fact that the estimation can be severely influenced by factors such as interrupt coalescing (which however affects both our estimates and PPLive methodology as well). We argue that choosing large values of N and ΔT would yield less noisy results, but due to chunk scheduling policy, it may introduce a *bias* in the result. Intuitively, counting the number of packets over large time windows equals to count the number of chunk exchanged, rather than their actual transmission throughput. Using large windows, peers that more actively contributed to the transmission will thus appear as *both* preferred and high-bandwidth, introducing thus an artificial correlation between the two terms. To avoid this bias, shorter windows should be preferable. At the same time, too small values of N and ΔT should be avoided, indeed, as packets are sent out in bursts, it may happen that interrupt

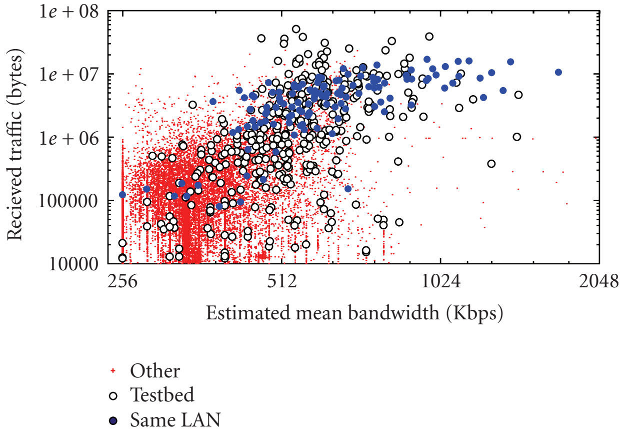


Figure 5: Scatter plot of received traffic versus estimated mean bandwidth (log-log scale).

coalescing (which we verified to be present in out traces) can squeeze the packet arrival pattern and increase the estimated BW.

In reason of the above observations, we select values of $N = \{5, 10, 20\}$ packets and $\Delta T = \{50, 100, 250\}$ ms. For every peer pair, we then construct a series of several BW samples gathered during the whole experiment, of which we then compute the mean and 99th percentile (p_{99}) values; we argue that both statistics are relevant, since the mean value is indicative of instantaneous network conditions, whereas p_{99} may be more representative of peer y access capacity. For lack of space, we will only report a subset of results, selecting $N = 10$ packets and $\Delta T = 100$ ms, since we verified that the same conclusions hold using the other parameters as well.

Scatter plot of the amount of received traffic versus mean bandwidth $\mathrm{BW}_{\Delta T}$ is depicted in Figure 5, using $\Delta T =$ 100 ms and log-log scale. Dark points represent exchanges between any two probes in the same LAN, white points are used for probes belonging to our testbed but belonging to different institutions, whereas any other contributor is represented with a small dot. First, hosts within our testbed, and especially hosts within the same LAN, achieve higher rates with respect to Internet hosts. Moreover, the estimated $\mathrm{BW}_{\Delta T}$ values are sound and consistent with our expectation.

5.2.2. Correlation-Based Analysis. Another interesting observation to gather from Figure 5 is that host achieving higher data rates also tend to contribute more data, and that this behavior is consistent across all three host groups.

To further quantify this behavior, we evaluate the coefficient of correlation $\rho(\mathrm{RX}, \mathrm{BW})$ between the amount RX of bytes received by a given probe and the bandwidth BW toward that contributor. For comparison purposes, we also evaluate the coefficient of correlation between the estimated bandwidth BW between two peers and the fact that they belong to the same AS or CC (using the indicator function as before). Though we are aware of the fallacies of correlation based analysis, we point out that we do not seek to *prove* directional cause-effect relationship between variables, but that we rather compare the magnitude of the correlation and relatively weight their impact.

Table 2: Correlation $\rho(X, Y)$ between bandwidth (BW), received bytes (RX) and peer localization information (CC, AS) for different groups of contributor peers (LAN, Internet, all).

	Y :	$\mathrm{BW}_{\Delta T}$		BW_N	
X :		mean	p_{99}	mean	p_{99}
AS		0.28	0.44	0.27	0.29
CC		0.27	0.42	0.26	0.26
RX	all	0.43	0.54	0.45	0.53
	LAN	0.62	0.75	0.51	0.61
	!LAN	0.33	0.40	0.37	0.45

Results are reported in Table 2 for different bandwidth estimation methods. In case of $\rho(\mathrm{RX}, \mathrm{BW})$ we also consider different peers subsets, namely, peers falling in the same LAN, peers that do not belong to the same LAN and all the peers altogether. As expected, we observe that irrespectively of the BW evaluation method considered, there is medium correlation between received bytes and bandwidth $\rho(\mathrm{RX}, \mathrm{BW})$, which is clearly stronger for peers belonging to the same LAN. Moreover, notice that this correlation is stronger with respect to $\rho(\mathrm{RX}, \mathrm{AS})$ or $\rho(\mathrm{RX}, \mathrm{CC})$ reported earlier, even when all peers are considered. Also, notice that the correlation between bandwidth and geolocation $\rho(\mathrm{BW}, \mathrm{AS})$ (i.e., the fact that high bandwidth contributors can be found within the same AS) is of the same order of magnitude of $\rho(\mathrm{RX}, \mathrm{AS})$ (i.e., the fact that video content is downloaded from contributors within the same AS). Overall, these observations suggest that, even outside the LAN environment, peers are primarily looking for bandwidth and that the early noticed geolocalization may be a beneficial side effect of the enforced bandwidth preference *alone*. Finally, we point out that part of the correlation might be explained by means of (i) mutual dependency between AS and BW, as well as (ii) additional hidden factors which causes both AS and bandwidth preference. At the same time, such hidden factors (e.g., preference based on IP/16 address similarity) are likely to play only an additional role beside the one played by the bandwidth, to which we shown early PPLive being extremely sensitive to.

6. Conclusions

This work proposed a methodology, based on the joint use of active and passive measurement technique for the analysis of the network awareness of currently deployed Internet P2P-TV system. The technique has been designed so to consider P2P systems as a black-box and as such can be applied to future systems as well. As a case study, we applied the methodology to the analysis of PPLive a very popular system nowadays, gathering interesting results, that we briefly summarize here.

First of all, by means of active testbed methodology, we find PPLive to be extremely sensitive to bandwidth, only mildly sensitive to losses and mostly unaware of IP distance, expressed in terms of either delay or IP hop count, which is in agreement with [12]. Refining further this picture, we find that actually the peer selection process is continuously updated, with a relative preference among pathwise properties that depends on the actual magnitude of the impairment.

Interestingly, by the correlation analysis of peerwise preference gathered through the passive technique, we find that the very same bandwidth sensitivity of PPLive seems to induce a desirable side effect, namely, a moderate geoclusterization of peers within the same AS and CC. Yet, it seems that PPLive does not, for the time being, explicitly enforce AS-awareness, which remains thus a new exciting challenge for the next steps of its evolution.

Acknowledgments

This work was funded by EU under the FP7 Collaborative Project "Network-Aware P2P-TV Applications over Wise-Networks" (NAPAWINE). The authors wish to thank all the project partners for having taken part to the multiple live measurement experiment.

References

[1] C. Labovitz and S. Iekel-Johnson, "The Great Obama Traffic Flood," January 2009, http://asert.arbornetworks.com/2009/01/the-great-obama-traffic-flood.

[2] PPLive, http://www.pptv.com.

[3] X. Hei, C. Liang, J. Liang, Y. Liu, and K. W. Ross, "A measurement study of a large-scale P2P IPTV system," *IEEE Transactions on Multimedia*, vol. 9, no. 8, pp. 1672–1687, 2007.

[4] B. Li, S. Xie, Y. Qu, et al., "Inside the new Coolstreaming: principles, measurements and performance implications," in *Proceedings of the Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '08)*, pp. 1705–1713, Phoenix, Ariz, USA, April 2008.

[5] C. Wu, B. Li, and S. Zhao, "Multi-channel live P2P streaming: refocusing on servers," in *Proceedings of the Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '08)*, pp. 2029–2037, Phoenix, Ariz, USA, April 2008.

[6] L. Vu, I. Gupta, J. Liang, and K. Nahrstedt, "Measurement of a large-scale overlay for multimedia streaming," in *Proceedings of the 16th International Symposium on High Performance Distributed Computing (HPDC '07)*, pp. 241–242, Monterey, Calif, USA, June 2007.

[7] F. Wang, J. Liu, and Y. Xiong, "Stable peers: existence, importance, and application in peer-to-peer live video streaming," in *Proceedings of the Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '08)*, pp. 2038–2046, Phoenix, Ariz, USA, April 2008.

[8] X. Hei, Y. Liu, and K. W. Ross, "Inferring network-wide quality in P2P live streaming systems," *IEEE Journal on Selected Areas in Communications*, vol. 25, no. 9, pp. 1640–1654, 2007.

[9] S. Agarwal, J. P. Singh, A. Mavlankar, P. Baccichet, and B. Girod, "Performance and quality-of-service analysis of a live P2P video multicast session on the Internet," in *Proceedings of the IEEE International Workshop on Quality of Service (IWQoS '08)*, pp. 11–19, Enschede, The Netherlands, June 2008.

[10] T. Silverston and O. Fourmaux, "Measuring P2P IPTV systems," in *Proceedings of the International Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV '07)*, June 2007.

[11] C. Wu, B. Li, and S. Zhao, "Exploring large-scale peer-to-peer live streaming topologies," *ACM Transactions on Multimedia Computing, Communications and Applications*, vol. 4, no. 3, 2008.

[12] E. Alessandria, M. Gallo, E. Leonardi, M. Mellia, and M. Meo, "P2P-TV systems under adverse network conditions: a measurement study," in *Proceedings of the Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '09)*, pp. 100–108, Rio de Janeiro, Brazil, April 2009.

[13] D. Ciullo, M. A. Garcia, A. Horvath, et al., "Network awareness of P2P live streaming applications," in *Proceedings of the IEEE International Symposium on Parallel and Distributed Processing (IPDPS '09)*, Rome, Italy, May 2009.

[14] D. Rossi, E. Sottile, S. Valenti, and P. Veglia, "Gauging the network friendliness of P2P applications," in *Proceedings of the ACM SIGCOMM, Demo Session*, Barcelona, Spain, 2009.

[15] http://www.infres.enst.fr/~drossi/P2PGauge.

[16] F. Dabek, R. Cox, F. Kaashoek, and R. Morris, "Vivaldi: a decentralized network coordinate system," in *Proceedings of the ACM SIGCOMM*, Portland, Ore, USA, 2004.

[17] R. Bindal, P. Cao, W. Chan, et al., "Improving traffic locality in BitTorrent via biased neighbor selection," in *Proceedings of the International Conference on Distributed Computing Systems (ICDCS '06)*, 2006.

[18] V. Aggarwal, A. Feldmann, and C. Scheideler, "Can ISPS and P2P users cooperate for improved performance?" *ACM SIGCOMM Computer Communication Review*, vol. 37, no. 3, pp. 29–40, 2007.

[19] V. N. Padmanabhan, H. J. Wang, P. A. Chou, and K. Sripanidkulchai, "Distributing streaming media content using cooperative networking," in *Proceedings of the International Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV '02)*, pp. 177–186, Miami Beach, Fla, USA, May 2002.

[20] R. Rejaie and A. Ortega, "PALS: peer-to-peer adaptive layered streaming," in *Proceedings of the International Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV '03)*, pp. 153–161, Monterey, Calif, USA, June 2003.

[21] M. Hefeeda, A. Habib, B. Botev, D. Xu, and B. Bhargava, "PROMISE: peer-to-peer media streaming using collectCast," in *Proceedings of the ACM International Multimedia Conference and Exhibition (MM '03)*, pp. 45–54, Berkeley, Calif, USA, November 2003.

[22] D. Ren, Y.-T. H. Li, and S.-H. G. Chan, "On reducing mesh delay for peer-to-peer live streaming," in *Proceedings of the Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '08)*, pp. 1732–1740, Phoenix, Ariz, USA, April 2008.

[23] R. Iqbal, B. Hariri, and S. Shirmohammadi, "Modeling and evaluation of overlay generation problem for peer-assisted video adaptation and streaming," in *Proceedings of the International Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV '08)*, pp. 87–92, Braunschweig, Germany, 2008.

[24] D. Purandare and R. Guha, "An alliance based peering scheme for P2P live media streaming," *IEEE Transactions on Multimedia*, vol. 9, no. 8, pp. 1633–1644, 2007.

[25] G. Huang, “Experiences with PPLive,” in *Proceedings of the ACM SIGCOMM P2P Streaming and IP-TV Workshop (P2P-TV ’07)*, Kyoto, Japan, August 2007, Keynote Speach.

[26] “Network-Aware P2P-TV Application over Wise Networks,” http://www.napa-wine.eu/cgi-bin/twiki/view/Public.