Metrics and QoE assessment in P2PTV applications

René Serral-Gracià*, Marcelo Yannuzzi,
Eva Marin-Tordera, Anny Martínez Manzanilla and
Xavi Masip-Bruin

Advanced Network Architectures Lab (CRAAX),
BarcelonaTECH (UPC),
2nd floor, NEAPOLIS Building, Rambla Exposició,
61-69, 08800 Vilanova i la Geltrú, Spain
E-mail: rserral@ac.upc.edu
E-mail: yannuzzi@ac.upc.edu
E-mail: eva@ac.upc.edu
E-mail: annym@ac.upc.edu
E-mail: xmasip@ac.upc.edu

*Corresponding author

Abstract: Peer-to-peer (P2P) is a growing technology offering an affordable platform to deploy distributed services and streaming of multimedia content through peer-to-peer television (P2PTV). Nevertheless, to promote such technology it is necessary to provide a solid streaming quality assessment mechanism. In this context, legacy solutions tied to service level agreements are no longer suitable, as for P2P systems, the service consumers become an active part by assisting in the service delivery. To overcome this limitation, we present a generic multi-layer monitoring and management framework to assess the quality of service (QoS) and the quality of experience (QoE) of multimedia traffic in any P2PTV streaming application. We also demonstrate the usefulness of our solution, by analysing the performance of a real streaming application in a P2PTV environment.

Keywords: peer-to-peer television; P2PTV; quality of experience; QoE; quality assessment; quality of service; QoS; multi-layer metrics; video streaming; one-way delay; OWD; packet losses; online assessment.


Biographical notes: René Serral-Gracià received his degree in Computer Science from the Technical University of Catalunya (UPC), and PhD in the same area in the Department of Computer Architectures (DAC), at the same university in 2009. Currently, he is an Associate Professor at UPC, where his research is focused on topics such as inter-domain routing optimisation, software defined networks, overlay networks, and QoE assessment of multimedia traffic. His research has been performed in the past in European projects such as LONG, EuQoS, and is continued by actively participating in other FP7 projects such as ONE or TEFIS.

Marcelo Yannuzzi received a degree in Electrical Engineering from the University of the Republic, Uruguay, in 2001, and PhD in Computer Science from the Technical University of Catalonia (UPC), Spain, in 2007. He is the R&D Head of the Advanced Network Architectures Lab (CRAAX) at UPC, Spain (http://www.craax.upc.edu/), where he leads several research initiatives in close interaction with key European and US industries in the telecommunications market. He is the author of several scientific publications, and he is also co-founder of MATPOL Inc., a start-up company in the Silicon Valley in late 2012. His current R&D interests are in the areas of security, mobility, internet of things, overlay control and management, software defined networks, novel orchestration mechanisms, and networking-level operating systems.

Eva Marin-Tordera obtained her MS in Physics in 1993 and in Electronic Engineering in 1998 both from the Barcelona University; and PhD from the Technical University of Catalonia in 2007 where she works as an Assistant Professor. She has published many papers in national and international conferences. Her main interests focus on QoS provisioning, multi-layer networks and optical networks. She has been actively participating in the BONE and DICONET international projects and in the national project CATARO.
1 Introduction

Network performance assessment metrics were initially designed to comply with the requirements of services using the client/server paradigm. Particularly, they covered the need of service providers to monitor the quality of service (QoS) and the service level agreement (SLA) compliance of the traffic in the network. Such approach has the issue that it is not devised to take into consideration the particularities of the underlying traffic. Equally treating bulk data transfers than real-time multimedia streaming.

Then, with the modernisation of end user devices, e.g., smartphones, tablets, and so on. The usage of multimedia data in the internet has experienced an exponential growth. Thus, distinguishing bulk data transfers and multimedia flows becomes a relevant issue, especially when assessing their performance. This QoS assessment of multimedia traffic has been a well-studied topic in the past, e.g., in videoconferencing (Baldi and Ofek, 1998), or VoIP conversations (Serral-Gracià et al., 2008). Despite of this, most of the works in this area are restricted to objectively assess the network layer observed performance, neither considering specific application restrictions nor the end-users subjective perception of the service, the latter being the research focus in the area of quality of experience (QoE) assessment.

Due to the proliferation of peer-to-peer (P2P) file sharing applications, the adoption of P2P technology has experienced a massive adoption by home users. This caused a radical shift in the observed traffic profile given that now, users that were typical consumers, become passive servers of the particular service they are part of. Participants in such P2P systems are viewed as logical and functional equivalent, systems that transmit data among each other in an overlay network decoupled logically from the actual network topology. This differentiates the solution from the pure client-server (CS) protocols, where participants either consume or serve the content.

The side effect of such massive adoption of P2P systems, is that new protocols and applications have been devised. In particular, in this work, we bring to attention peer-to-peer television (P2PTV), a P2P technology used to broadcast video content to registered users. P2PTV has been adopted by several applications (e.g., Coolstreaming, PPlive, PPStream, SopCast, etc.) that aim at an affordable system to provide multimedia content to the users. The commercialisation of such services, stimulates the desire of both end-users and content providers to obtain feedback about the quality of the delivered service. Nevertheless, while the legacy performance metrics are suitable for objective quality assessment of network traffic, they are not applicable when used at higher layers, e.g., user perceived quality assessment, or when used to evaluate P2P overlay networks, where the content of the same multimedia flow can be simultaneously obtained from different peers.

In this regard, this paper is an extension of our contribution in Serral-Gracià et al. (2012). Particularly, we enhance the contribution of that work by providing a formal definition of the performance metrics to produce an accurate quality assessment of P2PTV traffic from the end-user perspective. We do this by adapting the classic metric definitions – one-way delay (OWD), packet loss ratio (PLR), and inter-packet delay variation (IPDV) – to consider multi-layer information in three complementary steps. First, we cope with the specific particularities of P2PTV overlays, later we extend these definitions to media streaming applications, to finally assess the user perceived quality of multimedia flows in this environment. To the best of our knowledge, this is the first attempt to such multi-layer formalisation approach of the performance metrics within a P2P environment.
On the other hand, further enhancing our previous work, we extend our monitoring and management framework (MMF) to assess some aspects of QoE on P2PTV streams. As we shall show, using the proposed multi-layer metrics allows MMF to seamlessly assess the perceived video quality in a broad range of scenarios. The proposed framework sets the basis to a complete management suite offering capabilities such as, debugging of P2PTV protocols, live reporting of perceived video quality both to the content providers and the end-users, assisting with billing and accounting processes based on quality assessment, or supporting the P2PTV protocol to better accommodate technologies such as adaptive bitrate resolution depending on the experienced video quality.

Then, as a proof of concept, we deploy MMF in the PlanetLab network with two different goals, first to analyse the correctness of MMF as a quality assessment platform, and second to evaluate the suitability of SopCast (2009) as a P2PTV broadcasting solution. Our results show that MMF can effectively assess the user perceived quality compared to the analysis of the video traces, the results also show that SopCast has a fairly good performance in PlanetLab, with acceptable video quality, but very high start-up times in most cases, which might prevent its deployment in some environments.

The rest of this paper is structured as follows, Section 2 reviews the related work, then, in Section 3, we present our formal multi-layer metric definitions and adaptations. After that, in Section 4 we outline the management framework to perform P2PTV application assessment. Later, we describe the tests and the testbed used to validate our proposal. In Section 6, we detail the experimental deployment of SopCast in the PlanetLab network, and finally, in Section 7 we draw some conclusions and outline our future work.

2 Related work

Since the explosive growth of multimedia content on the internet, content providers are concerned about the quality of the offered service. Initially, this concern was addressed by enforcing SLAs between network operators and content providers and by setting specific policies over multimedia traffic. As a consequence, several solutions appeared in order to monitor the offered QoS by the network, as well as, to manage these SLAs. For instance, Sommers et al. (2010) presented a method for actively assessing the SLA compliance of the monitored links. Other approaches, such as Serral-Gracià et al. (2010b), use passive traffic analysis to achieve a similar goal.

During the quick expansion of multimedia content streaming, it was clear that solely measuring the network performance was not sufficient to assess the user satisfaction (Greengrass et al., 2009), given that the subjective user perception might considerably differ from the measured network performance. Therefore, the focus shifted from the resource intensive network performance monitoring, to traffic analysis at the edges of the network. In particular, the monitoring was moved as close to the end-user premises as possible, because the closer to the end-user the more accurate the perceived quality estimation would be, e.g., the user perceived quality can be assessed through QoE techniques. In the particular case of video quality assessment, there are several solutions to assess the user perceived QoE of IPTV services, e.g., in Won et al. (2008), the authors discuss about the effects on the end-user quality of video streaming on their premises. In Yamagishi and Hayashi (2008), the authors propose a framework for parametric assessment of the QoE of IPTV services from a content provider perspective. All these proposals provide an objective measurement of end-user perceived QoE which, in general, it is quantified in the same scale as the mean opinion score (MOS) (ITU-T Recommendation G.107, 2005), that is, a value between 1 (for bad quality) to 5 (for perfect quality) of the received content. There are many methods to objectively estimate the MOS, and it is not the goal of this paper to perform a thorough description of the different approaches, further reading in this regard can be found in Serral-Gracià et al. (2010a) and references therein. Regardless of MOS estimation, in the video quality assessment area there are three different solutions for content quality measurement (Winkler, 2009):

1. **full-reference (FR)**
2. **reduced-reference (RR)**
3. **no-reference (NR).**

The FR method requires access to the original and received content to perform the evaluation. The RR uses an out-of-band signalling process between senders and receivers to transmit some relevant parameters about the delivered content, or even use in-band (e.g., watermarking) schemes to assist the computation of the perceived quality. Finally, the NR approach assesses the content quality level without any knowledge of the original material. As a proof of concept, the solution presented in this work is based on the FR approach, the most accurate of the three, to demonstrate the flexibility of our proposal. However, we leave as an important part of our future work to enhance its functionality to the NR model.

Most of the works in the area of QoE are centred around the classical CS paradigm, where all the assumptions and defined metrics for QoE assessment assume fixed and well-known source and destinations. However, in this paper we focus our analysis in P2PTV applications. In this area, there are several efforts to measure and assess the perceived user quality, e.g., in works such as Lu et al. (2008) analyse one important aspect of QoE, namely, the end-to-end blocking probability, by comparing the results obtained by SopCast with other regular IPTV applications. While in Silverston et al. (2009) perform an empirical study of real use of many different P2PTV alternatives, with an analysis of the differences among them. Another experimental study on P2P video streaming can be found in Liang et al. (2009) experimentally discuss about the profits of having complex P2P systems by evaluating different P2P algorithms.
Our work differs from those in two aspects, first we focus on the understanding and formal definition of the different metrics involved in the quality assessment, and second, we use this knowledge to build a MMF for performance evaluation of P2PTV applications.

Regarding our second contribution, i.e., the design of a full-fledged multi-layer MMF in a P2PTV overlay network, to the best of our knowledge currently there are no other efforts in this direction. However, with similar objectives, De Vera et al. (2008) propose a full video network monitoring suite, which is aimed at QoE estimation using PSQA (Mohamed and Rubino, 2002). Our work differs from theirs in the sense that we gather multi-layer information opposed to the frame level information extracted in that work. Moreover, our focus in P2PTV further differentiates our contributions. Continuing with quality assessment but in the P2P environment, also by the same authors, in Da Silva et al. (2008) they use PQSA in order to design a P2P protocol in order to distribute real-time video streams. In this case, our solution offers a more generic framework that is able to operate in different modes allowing to assess the end-user satisfaction in either existing and new P2PTV systems in a multi-layer fashion.

3 Multi-layer performance metrics

Having network performance metrics is useful, but insufficient in a P2P network. Network performance metrics are measured ‘end-to-end’ at the network layer for the packets belonging to the same TCP (or UDP) flow, where packets have the same source and destination. On the contrary, in a P2PTV context, a single source is broadcasting streamed multimedia traffic to several nodes, which in turn send the same content to other destinations based on the peering policies that form the overlay. As a consequence, the concept of ‘end-to-end’ in a P2PTV overlay does not refer to the flow’s source and destination, but rather to the content generator and the specific consumer. This shift in the grounds of the CS paradigm requires the performance metrics to be redefined in order to investigate the aspects that accurately capture the perceived quality in this new scenario.

Depending on the location and the mechanism used to compute the performance metrics, the assessment may capture different aspects. In particular, in Figure 1, we show the different alternatives to metric collection (left-hand side in the figure) where we detail the different operational objective assessment for all the layers on the TCP/IP stack. As expected each layer in the stack provides access to different information, hence at the TCP and IP layers we can obtain per packet information referring to particular flows and their source and destination hosts. Opposed to this in the P2P layer we have information related to the different peers connected to the node, its status, together with information on which peer is sending a particular piece of information. Finally, in the application layer the obtained information goes one step further, now we have access to application information, such as codec used to transmit the video, or status of the playback buffers.

In Figure 1, we also detail QoE related metrics (centre of the figure), where depending on the information provided by the operational performance metrics can deliver different accuracy as we detail later. Finally, the figure also depicts the subjective QoE assessment (right-hand side of the figure). In this work, we define the set of multi-layer performance metrics necessary to perform an accurate video quality assessment. To accomplish this, we first focus our interest on the definition of the P2P overlay metrics, which are specified based on the network and transport metrics already defined (e.g., in Almes et al., 1999a, 1999b; ITU-T Recommendation Y.1540, 2002; Demichelis and Chimento, 2002). Second, we focus on the application of specific information, where we use run-time information, such as buffer status or error concealment capabilities of multimedia codecs. This brings a closer, and more accurate, estimation of the delivered quality from an end-user perspective.

Figure 1 Gathered information on the different layers, (a) from objective low level network information (b) to subjective user perceived QoE (see online version for colours)
3.1 P2P overlay performance metrics

Prior to extending the performance metrics to suit P2PTV, we need to formalise several concepts related with P2PTV overlays, namely streamed media content \( M \), chunk \( c_i \), and overlay path \( P^S(c_i, D_j) \). After that we introduce the specific P2P metrics, i.e., P2P OWD, P2P chunk losses and P2P InterPacket chunk variation.

- **Streamed media content**: Streamed media content \( M \) in a P2P environment is the stream of multimedia information produced by a specific content generator that is received by a set of nodes belonging to the overlay. In a P2P overlay, there will be a single content generator per \( M \), but with an arbitrary number of final destinations.

Formally, streamed media content is defined as \( M = \{ S, D_1, ..., D_n, C \} \), where \( S \) (also defined as \( D_0 \)) is the content generator, \( D_j \) (with \( 0 < j \leq n \)) is a content consumer (destination), and \( C = \{ c_1, ..., c_m \} \) is the set of chunks composing the streamed content.

- **Chunk**: A chunk \( c_i \) is the minimum transmission unit in a P2P overlay. In this paper, we consider, without loss of generality, that a chunk \( c_i \) corresponds to a set of continuous video (or audio) frames whose sizes might vary depending on each implementation. In particular, a chunk is \( c_i = \{ f_i, ..., f_i \} \), with all the frames, and packets therein, following the same path in the overlay. Consequently, a chunk, once decoded by the application, has a given duration \( d \) in time, which we assume constant for each chunk.

- **Overlay path**: \( P^S(c_i, D_j) = \{ H_{1}^c, ..., H_{y}^c \} \) is the set of hops (nodes) \( H_{k}^c \), \( 0 < k \leq y \) that constitute the path from the source \( S \) to the destination \( D_j \), to distribute chunk \( c_i \) where \( H_{1}^c = \{ D_0, D_k \} \) is a hop between two nodes with active peering on the overlay.

3.1.1 Peer-to-peer one-way delay

**Definition 1**: Peer-to-peer one-way delay (P2POWD) is the time span passed from the content generation at \( S \) until its arrival to all its destinations \( D_j \). For a particular chunk \( c_i \), the P2POWD is:

\[
T^c = \left[ t_1^c, ..., t_s^c \right]
\] (1)

where \( t_j^c \) is the end-to-end delay for the overlay path \( P^S(c_i, D_j) \) for chunk \( c_i \).

Given that each chunk towards a destination can be sent from a different peer, then the P2POWD of a chunk towards a specific destination is composed by the partial delays of its overlay path. Namely:

\[
t_j^c = \sum_{k=1}^{n} OWD(H_k^c) + \rho_k
\] (2)

where \( H_k^c \) is the \( k \)th hop within the chosen path \( P^S(c_i, D_j) \) for chunk \( c_i \), and \( OWD(H_k^c) \) the one way delay caused by the hop. Finally, \( \rho_k \) is the \( k \)th node processing time, which is protocol dependent, and can be orders of magnitude higher than the actual transmission delay of the link.

3.1.2 P2P chunk loss

We distinguish two different types of degradation caused by losses in the network:

1. peer-to-peer chunk losses (P2PCL) when all the packets within a chunk are lost, causing video blocking at the receiver
2. peer-to-peer chunk degradation (P2PCD) when not all the packets belonging to a chunk are lost, causing content corruption and visual artefacts on the video reception.

**Definition 2**: We define chunk degradation \( L_D^c \) as the PLR of chunk \( c_i \) once received at \( D_j \).

Where \( 0 \leq L_D^c \leq 1 \), if \( L_D^c = 0 \), then the chunk is not degraded (good video quality), and if \( L_D^c = 1 \), then all the packets on the chunk are lost (bad video quality). To compute \( L_D^c \) we use expression 3.

\[
L_D^c = \frac{p_{D_j}^c}{|c_i|}
\] (3)

where \( p_{D_j}^c \) is the lost packets within chunk \( c_i \) at \( D_j \), over the total packets in the chunk.

This metric is the most critical from the end-user point of view in video streaming solutions. Embedding mechanisms in a P2PTV protocol to compute it can be useful in order to detect network congestion, and most especially to assess the quality of the service delivered to the user.

3.1.3 P2P burst degradation

Extending the definition of P2PCL, another important metric to assess the degradation of a multimedia stream is the loss burst (LB), which determines the length of consecutive chunks that experience degradation in the multimedia stream. This metric is particularly important given that end-users are more sensible to long periods with video disruption than to small glitches that can be easily ignored.
Definition 3: We define peer-to-peer loss burst (P2PLB) in a P2P overlay as the number of consecutive chunks with packet losses. Then,

\[ LB_{j,k}^{c_i,a_k} = \{c_a, ..., c_b\} \forall x / a \leq x \leq b \land \Delta_{j,k} > 0 \]  

Clearly, the length of the burst \(|LB_{j,k}^{c_i,a_k}|\) will directly impact in the quality of the streaming service. Hence, this metric is important to quantify the duration of continuous video disruptions.

### 3.1.4 P2P inter-chunk delay variation

There is no general consensus on the proper mechanism to compute IPDV (Demichelis and Chimento, 2002; ITU-T Recommendation Y.1541, 2005; Sommers et al., 2010), some alternatives propose to compute it by using OWD of consecutive packets, others propose to use any pair of packets in the stream, and others even to compare the OWD of a given packet with the averaged OWD of the rest of the stream. Regardless of these differences, IPDV is a critical metric in some environments, especially when there are small buffers at the destination nodes, e.g., interactive applications, where big variations in OWD lead to service degradation at the end-points caused by late arrival of packets. Opposed to this, in a P2PTV overlay with streamed traffic, such variations are less relevant given the big buffers present, yet its assessment is useful to assess the suitability of application layer buffer sizes. It is worth noticing that in a P2PTV overlay, smoothing the inter-chunk delay variation (ICDV) its complex in this environment because the chunks arrive from different peers simultaneously, potentially with different upstream bandwidth and congestion levels.

Definition 4: We define peer-to-peer inter chunk delay variation (P2PICDV) as the difference in P2POWD of consecutive chunks in the streamed content. It can be computed by:

\[ J_{c_i}^{c_i} = \begin{cases} T_{c_i} - T_{c_i+1}, & 1 < i \leq m \\ 0, & i = 1 \end{cases} \]  

where \( m \) is the number of chunks in \( M \), we subtract the P2POWD of a chunk \( c_i \) with the P2POWD of the previous one \( c_{i-1} \), for all the destinations \( D_j \) in the overlay. Similarly, we also define \( J_{c_i}^{D_j} \), the particular P2PICDV of \( c_i \) at destination \( D_j \).

### 3.2 P2PTV application metrics

So far we defined the P2P overlay performance metrics to report reliably and accurately values from the overlay perspective. Nonetheless, to assess the end-user perception, the metrics must be acquired as close as possible to the end-user to consider application specific information. To this end, we continue our discussion with the definition of the P2P application metrics, which are built upon the P2P overlay metrics, and extend them considering application behaviour, such as buffer sizes, or error concealment algorithms present in the multimedia codecs. In particular, we focus on the effects of the application on the final video quality.

#### 3.2.1 P2P application one-way delay (P2PAOWD)

It is worth noticing that besides the delays introduced by the intermediate peers on the overlay, the multimedia player at the destination also uses internal de-jitter buffers that adds-up to the final user perception. Therefore, the delay perceived by the end-user is determined by Expr. 6.

\[ w_j^i = t_j^i + b_j^i \]  

where \( b_j^i \) is the reception play-out buffer at the destination \( D_j \) at the instant of reception of chunk \( c_i \). Such buffer introduces an extra delay from the arrival of the packet until its processing and displaying to the end-user by the media player.

Analogously to Expr. 1 the P2PAOWD for a particular chunk is defined as:

\[ W_j^c = \{w_j^i, ..., w_j^m\} \]  

#### 3.2.2 P2P application chunk degradation (P2PAACL)

As we noticed previously, from the P2PAOWD perspective, the application overhead increases the delay of content delivery to the end-user, which is generally not desired. Opposed to this, the multimedia codecs present nowadays implement very advanced error masking and recovery techniques, which considerably ease the effects of packet losses within a chunk. These codec introduce a recovery threshold where it is possible to regenerate a signal similar to the original one. As a consequence, it is necessary to adapt the definition of chunk degradation issued on Definition 2 as follows:

\[ \delta < \Delta_{j,k} \leq 1 \]  

where \( \delta \) stands for the lower threshold where the error concealing algorithms present in multimedia codec can recover the original stream. It is out of the scope of this work to study actual values for \( \delta \). However, we leave it as a configurable parameter of our system.

#### 3.2.3 P2P application inter chunk delay variation (P2PAICDV)

As we already mentioned, the destination media players use a fairly big de-jitter buffer in order to overcome potential delay variations in the chunk delivery. Nevertheless, in the case of faulty peering, or even because of node churn, big P2PICDV are bound to happen in such overlays, incurring in buffers underflows that block the video playback. As far as the user is concerned, such events are effectively considered as packet losses, i.e., when \( J_{D_j}^{c_i} > b_{D_j}^i \), with
\( \mathcal{J}_{D_j} \) is the P2PAICDV of chunk \( c_i \) at node \( D_j \), and \( \bar{b}_{D_j}^{c_i} \) is the current buffer time. Hence, a multimedia stream will observe video blocking \( B_{D_j}^{c_i} \) caused by P2PAICDV as modelled by Expr. 9.

\[
B_{D_j}^{c_i} = \begin{cases} 
1, & \mathcal{J}_{D_j} > \bar{b}_{D_j}^{c_i} \\
0, & \mathcal{J}_{D_j} \leq \bar{b}_{D_j}^{c_i} 
\end{cases}
\]

where \( B_{D_j}^{c_i} \) states whether there is video blocking or not for chunk \( c_i \) at \( D_j \).

### 3.3 P2P overlay QoE metrics

After developing the classic performance metrics at the overlay and application layers, we now map them to the computation of the objective user aware QoE metrics (central box in Figure 1). We achieve this by separating the user-aware metric assessment into two different parts, the direct metrics and the indirect metrics.

In this context, we understand direct metrics as the metrics which directly affect the video quality, while indirect metrics refer to the ones that affect the service but do not impact on the video quality.

#### 3.3.1 Direct quality

As it is broadly accepted, video quality is the most relevant factor when assessing the end-user satisfaction of a given service. However, it is not within the scope of this work to propose other mechanisms to compute such end-user satisfaction. On the contrary, as a first approach we use the peak signal to noise ratio (PSNR) as a FR metric, while leaving as an important part of our future work to extend this assessment environment to NR metrics which will be much useful for real-time QoE assessment. Then, by using Klaue et al. (2003), we obtain the MOS for the given PSNR.

With this approach we obtain a quality value \( q_f \) per frame, which in the case of P2PTV is transmitted within a chunk. Hence, \( \mathcal{Q} = \{q_1, ..., q_n\} \) is the quality (MOS values) of all the frames belonging to the chunk. Given the generally small size of the chunks in multimedia streaming applications and the correction algorithms of the video codecs, a good estimator of the overall quality of chunk \( c_i \) from the end-user perspective is its expected value, \( \bar{\mathcal{Q}}^i = E[\mathcal{Q}^i] \). Then, a user will find that portion of the video is annoying when \( \bar{\mathcal{Q}}^i < \kappa \), where \( \kappa \) is the lower MOS threshold acceptable for the service, which in P2PTV normally is 3 (Takahashi et al., 2006). As a consequence, for the chunk duration an unsatisfaction period \( \mathcal{U}(c_i) \) will be defined following Expr. 10.

\[
\mathcal{U}(c_i) = \begin{cases} 
1, & \bar{\mathcal{Q}}^i < \kappa \\
0, & \text{otherwise}
\end{cases}
\]

From this expression, we can conclude that the total period of unsatisfaction observed by the user will be

\[
\mathcal{U} = d \sum_{i=1}^{n} \mathcal{U}(c_i)
\]

where \( d \) is the chunk duration defined previously, and \( n \) is the number of chunks in the media \( M \).

As already mentioned previously, beside the amount of unsatisfaction periods, their effect is more noticeable when their duration is longer. Hence, we define as the consecutive unsatisfaction periods as \( \mathcal{U}_l = \{u_1, ..., u_p\} \), where

\[
u_z = \{\mathcal{U}(c_y), ..., \mathcal{U}(c_x)\}
\]

\[\forall x \, (a \leq x \leq b \wedge \mathcal{U}(c_y) = 1) \text{ and } 1 \leq z \leq p.\]

#### 3.3.2 Indirect metrics

Often when designing multimedia streaming systems the focus of the developers relies on the video and audio quality. However, in order to provide a full and successful multimedia experience to the users, other important factors must be considered, e.g., usability, or system’s reaction time.

In this paper, we focus on two critical aspects which affect P2PTV services, namely, start-up times, and freshness of the received information.

**Start-up time.** The start-up time (ST) is the time span since the user asks for the content, i.e., connects to the P2PTV overlay, until it is reproduced to the end-user.

Formally, \( ST_j = w_j^p \), where \( w_j^p \) is the P2PAOWD defined before in Expr. 6. However, in practice, most P2PTV protocols have a warm-up time that overlaps with the start-up time, where the playback starts but initially incurs in several buffer underflows given the poor status of the buffers and the initial state of the peering on the overlay.

**Freshness.** The freshness determines the absolute age of the delivered content, i.e., the time since the content was generated until it is consumed by the end user. This metric is critical because many P2PTV applications are used in order to stream live content, namely, live sports or live TV shows, in these cases, users want the content as fresh as possible compared to other sources offering the same content, e.g., to celebrate a goal. However, when viewing stored content, such as movies, this metric might not be relevant from the end-user perspective.

### 4 Monitoring and management framework

After the metric definition, we present the MMF, a full-fledged quality assessment framework of P2PTV applications that integrates the multi-layer performance metrics defined in the previous section with an efficient monitoring infrastructure. The main goal of MMF is to seamlessly provide quality assessment features to new and already existing P2PTV solutions. To this end, MMF is
Metrics and QoE assessment in P2PTV applications

designed to perform the monitoring and management using two different approaches, integrated monitoring (IM) and decoupled monitoring (DM). IM is used when MMF is embedded into the system and is free to interoperate with it, while DM is devised as an external helper application, running in a separate process, to support content providers and developers in the evaluation of already existing P2PTV solutions.

Depending on the chosen monitoring approach, different information of the system will be available. In particular, when using IM the monitoring system has access to firsthand information about the internal status of the P2PTV application, e.g., video stream and P2P network status, and therefore, it can tweak the application behaviour to improve the overall end-user experience. On the contrary, with DM the system must use estimates and reverse engineering techniques to infer some of that information, without the possibility of directly affecting the application behaviour.

To illustrate the general structure of MMF, in Figure 2 we show a sample P2P network with the different components. To ease the exposition, the figure only details the internal MMF components of one P2PTV node, and the general blocks of another as an example, note however that all peers have exactly the same features. As it can be observed, the MMF has two different layers, the collection layer (CL) and the management layer (ML), CL is in charge of collecting and analysing the traffic in order to compute the different multi-layer metrics, while the ML is responsible for the management of the overlay node depending on the information gathered by the CL.

Depending on the requirements of the P2PTV platform we differentiate two assessment mechanisms:

1. online assessment
2. offline assessment - both marked in dashed lines in the figure.

In online assessment, MMF performs all the monitoring and management tasks in real-time distributed across all the nodes the overlay, while the offline assessment separates the monitoring and management into two different stages, first the monitoring, which continues to be distributed and performed online, and second the performance evaluation at the end of the P2PTV session, which is centralised and executed offline.

4.1 Layer description

MMF is composed by two different layers with two clearly separated roles, on the one hand there is the monitoring tasks that are performed by the CL and on the other hand the ML, responsible of assisting the P2PTV management tasks.

Figure 2  Sample P2PTV network, with the different components forming the MMF (see online version for colours)
4.1.1 Collection layer

The CL abstracts all the tasks related with the collection of multi-layer information within the node, e.g., it collects per chunk statistics in the P2P layer, while in the application layer it gathers specific per frame data.

The CL is composed by three different entities:

- **Overlay monitor (OM):** the OM collects statistics about the received information at the P2P layer. Specifically, it computes the timestamps and the loss ratios of the received chunks of video and audio packets, list of active peers serving each chunk, and required network resources for the streaming.

- **Application monitor (AM):** analogously to the OM, the AM collects information from the application perspective, e.g., buffer under/overflow status, used video codec, type of the dropped frames (I, P, or Bframes), presentation timestamp of the frames.

- **Collection manager (CM):** the CM gathers the information from the AM and the OM and sends it to the ML (when performing online assessment), or to the storage subsystem (while using offline assessment).

The details on how the information is gathered and processed depend on the used assessment mechanism and will be described later in this section.

4.1.2 Management layer

The ML is responsible of computing the metrics defined in Section 3 by merging the information obtained by the different CLs. This procedure is performed in four different steps, first step is to gather the information collected by the CM in the neighbouring peers, the second is to compute the different overlay and application metrics, the third is to detect the periods where the quality drops below predefined boundaries, and finally to take the necessary actions in order to minimise such unsatisfaction periods.

To implement the above steps, the ML is composed by two different entities.

- **The performance analyser:** is in charge of performing the first and second steps, thus gathering the neighbour information received from the CM and computing the requested multi-layer metrics, then it reports the obtained results to the decision module.

- **The decision module:** decides whether the user is experiencing periods of unsatisfaction (step three above) in case that the quality is below the preset limits, and taking the necessary actions if needed (step four).

In more detail, in the performance analyser each peer node can compute the multi-layer metrics by computing P2POWD and P2PAOWD using timestamps, computing the P2PCLR, e.g., by detecting errored chunks or missing frames at reception. Analogously to CL, the particular approach used in order to compute these metrics depends on the used assessment mechanism which will be detailed in Section 4.2.

In its case the decision maker will determine the tolerable performance metric boundaries at the different layers for the given service as configured by the P2PTV overlay administrators. A particular boundary definition of these constraints is issued later in Section 5.1. Finally, the decision module stores the computed metrics into a database, and in the case that the preset quality boundaries are not met, an alarm is raised and the system decides the actions to be taken. This database can be queried by different external entities, e.g., by a network management system – in the service provider premises – and different actions can be triggered, e.g., to update the resource reservation, or the traffic engineering policies to maximise the end-user satisfaction. It is not our goal to detail the internal workings of the action triggering system in order to assist in solving potential quality issues, thus in the rest of this paper we focus on the performance analyser and on the detection of unsatisfaction periods, leaving the action triggering as part of our future work.

4.2 Quality assessment mechanisms

Depending on the mechanism used to deploy MMF the capabilities and methods to collect the information will vary. Currently, MMF considers three different alternatives: IM with online data collection, DM with online data collection, and DM with offline data collection.

4.2.1 Integrated monitoring

When using IM the only supported data collection method is online with real-time computation of the quality metrics. To perform such operations we propose the following methodology. The video generator \( \hat{S} \) embeds within the P2PTV protocol the necessary information to compute the metrics, such as the emission timestamp for each frame (and chunk), and a per frame (and chunk) sequence numbers. Simultaneously, each peer computes the basic metrics described previously, e.g., P2POWD and P2PAOWD are computed at overlay and at application layers by timestamping the chunks and frames once received. In the case of P2PCLR it is computed by counting lost chunks or frames once they have expired1, the amount of losses per time unit can be computed by using the sequence numbers. Finally, computing the P2PICDV is straight-forward as it is obtained from the P2POWD. From this point, computing the perceived QoE can be done by using already existing alternatives, e.g., using the method described in De Vera et al. (2008).

Regarding the cost of this solution in terms of overhead bandwidth, all the information exchanged by the peers is limited to embed into the P2PTV protocol itself a timestamp and a sequence number per chunk and per frame from the source generator. Then the overhead ratio can be obtained by Expr. 12

\begin{equation}
\text{Overhead ratio} = \frac{\text{Size of timestamp and sequence number}}{\text{Size of actual data}}
\end{equation}
the analysis to the overlay metrics, leaving out of the study the more detailed application layer analysis. Independently of using online or offline monitoring, MMF uses the same methodology in order to collect metric information. First both OM and AM in $S$ will collect the transmitted information, this is performed by gathering information in the network interface, then identifying the different chunks and frames through deep-packet inspection, generating the identifiers, timestamping, and generating sequence numbers. Once the chunk is finished, the CM aggregates this information and generates a source chunk descriptor $C_D$ that holds all this information. Simultaneously, the rest of peers, perform the same operations with all the received video traffic from the other peers in the overlay generating a destination chunk descriptor $C_D$.

- **Online monitoring.** When using online monitoring the $C_D$ is sent by any peer when sending a particular chunk to another peer, this way, each peer can autonomously compute the requested metrics. The metrics are computed by matching all the chunks and frame identifiers from $C_D$ against $C_D$. If a particular identifier can be found in $C_D$ then the requested metrics are computed. However, if the identifier is not found then the chunk or frame is considered lost, and then, by using the sequence number it is possible to detect consecutive periods with frame drops in the network.

Even if no direct actions with the P2PTV protocol can be issued using this approach, it is still possible to take actions outside the protocol, e.g., by updating network resource reservation, or enforcing traffic engineering policies to maximise the end-user satisfaction, as we already discussed previously.

- **Offline monitoring.** Opposed to online monitoring, when performing offline monitoring each peer stores locally either $C_S$ or $C_D$, and once the video session is finished, all the information is gathered and post-processed offline, using the same methodology as previously in online monitoring.

### 5 Testing environment

In this section, we discuss the used P2PTV application and the testbed used to evaluate the proposed MMF. To this end, we deployed SopCast in the PlanetLab network, where we streamed a video session among 137 peers of the SopCast P2P overlay.

#### 5.1 SopCast

SopCast (2009) is a free but proprietary P2PTV application used to broadcast public TV channels. SopCast uses bit rates from 250 Kbps to 400 Kbps with a few channels as high as 800 Kbps, for the streaming, with the encoded video content

$$
\text{MMF needs to share some information among the peers in order to compute the necessary metrics. In particular we propose to transmit for each chunk and each frame the following information:}
$$

- **Identifier:** The identifier uniquely distinguishes the different chunks and frames in the stream, they are generated by applying a hash function $H$ to specific fields within the chunk or frame payload, the field selection is dependent on the P2PTV protocol, in Section 5.3 we detail the particular details of the identifier generation for the use case of SopCast.

- ** Generation and reception timestamps:** Each peer in the overlay, particularly $S$, timestamps all the chunks and the frames within the video stream in order to compute the different time related metrics.

- **Sequence number:** Analogously to IM, the sequence number will be used to compute the loss ratios incurred in the video transmission.

- **CRC:** Given that both chunks and frames may be composed by a number of IP packets, it could happen that a particular chunk, or frame, is only partially received.

For the frames MMF also stores the frame type, which will be useful in order to compute the QoE. It is important to notice that depending on the used P2PTV protocol some of the above information may not be available to MMF. Particularly, most of the time having first-hand application information is not feasible, in such a case, MMF will restrict

$$
\frac{\#c_i + (\sigma_c + \tau_c + n(\tau_f + \sigma_f))}{\#c_i},
$$

where $\#c_i$ is the size of the chunk, $\sigma_c$ and $\sigma_f$ are the sequence number sizes for the chunk and the frame respectively, which have a size of 4 bytes, and $\tau_c$ and $\tau_f$ are the timestamp sizes of the chunk and the frames, also 4 bytes, with $n$ frames contained within the chunk. Assuming an average frame size of 1,500 bytes and six frames per chunk the ratio of the overhead is 1.006, negligible compared with the original size.

### 4.2.2 Decoupled monitoring

In contrast to IM, when using DM the system cannot be integrated into the P2PTV protocol, hence MMF must perform the assessment through an external procedure, with the consequent increase in the system’s complexity. However, the advantage of this approach is that it is not necessary to perform any modification to the used P2PTV application.

With DM, MMF defines two different modes of operation, online, which, similar to the case using IM, monitors the quality metrics simultaneously to the video session, and offline, which collects detailed information of the video stream on each peer, to process them once the video session is finished.

MMF needs to share some information among the peers in order to compute the necessary metrics. In particular we propose to transmit for each chunk and each frame the following information:

- **Identifier:** The identifier uniquely distinguishes the different chunks and frames in the stream, they are generated by applying a hash function $H$ to specific fields within the chunk or frame payload, the field selection is dependent on the P2PTV protocol, in Section 5.3 we detail the particular details of the identifier generation for the use case of SopCast.

- **Generation and reception timestamps:** Each peer in the overlay, particularly $S$, timestamps all the chunks and the frames within the video stream in order to compute the different time related metrics.

- **Sequence number:** Analogously to IM, the sequence number will be used to compute the loss ratios incurred in the video transmission.

- **CRC:** Given that both chunks and frames may be composed by a number of IP packets, it could happen that a particular chunk, or frame, is only partially received.

For the frames MMF also stores the frame type, which will be useful in order to compute the QoE. It is important to notice that depending on the used P2PTV protocol some of the above information may not be available to MMF. Particularly, most of the time having first-hand application information is not feasible, in such a case, MMF will restrict

$$
\frac{\#c_i + (\sigma_c + \tau_c + n(\tau_f + \sigma_f))}{\#c_i},
$$

where $\#c_i$ is the size of the chunk, $\sigma_c$ and $\sigma_f$ are the sequence number sizes for the chunk and the frame respectively, which have a size of 4 bytes, and $\tau_c$ and $\tau_f$ are the timestamp sizes of the chunk and the frames, also 4 bytes, with $n$ frames contained within the chunk. Assuming an average frame size of 1,500 bytes and six frames per chunk the ratio of the overhead is 1.006, negligible compared with the original size.
divided into chunks and distributed to users through the SopCast P2P overlay.

The main advantages of using SopCast in a testing scenario compared to other P2PTV alternatives, is that SopCast, as far as we know, is one of the few solutions that allows end-users to broadcast their own content, which delivers an easily reproducible scenario for testing.

5.2 MMF deployment

Given that we do not have access to the source code of SopCast or to its specification, we deploy our MMF in the DM mode of operation, with offline assessment. This yields plenty of options to perform the quality assessment of the received video for all the receivers.

In the experiments, we define two different types of nodes in the P2P overlay network:

1. A content provider located in Delft University campus network, which acts as the source provider (S) in charge of broadcasting the entire video. S registers a TV channel to the SopCast network, broadcasting a documentary with a 480 × 384 resolution, a bitrate of 410 kbps, and a length of 300 seconds.

2. The second type of nodes are PlanetLab nodes that act as SopCast peers viewing the streamed TV channel. Each of the 136 PlanetLab nodes under consideration are running a SopCast Client (Linux version), with command line control; and tcpdump to enable passive monitoring on the traffic transmitted at the SopCast peers.

We setup the SopCast overlay with 136 PlanetLab nodes and the server. During the streaming period, we collected the traffic at the content generator and also at the 136 peers. In parallel, we also dumped the received video to each node’s local disk, in order to have a reference of ‘perfect knowledge’ against which we can compare the performance of MMF. Finally, the collected trace files from the 137 nodes are further processed and analysed offline.

In order to guarantee the accuracy of our measurements, we monitored during the tests the clock accuracy using NTP, and afterward for the delay study we removed the traces with dubious timestamps, leaving out a total of 130 traces for delay analysis, and the 136 for losses and delay variation, since for those latter the accuracy of the clock does not affect the results.

5.3 Reverse engineering SopCast

The DM offline quality assessment was performed by identifying SopCast video packets – we consider a video packet if it has a payload bigger than 1,000 bytes (Tang et al., 2009) – in all the traces, and offline running the performance analyser process to compute the performance metrics. The procedure required some reverse engineering of the SopCast protocol, given that it is a closed source application. After deeply studying the generated traffic we realised that SopCast recodes the received frames in each peer, this fact complicates substantially the frames and chunks identification and later the generation of the required identifiers by MMF.

After a closer inspection, we observed that the protocol keeps an internal sequence number that is sent within the header and it identifies the same video frame in all the nodes in the overlay. Therefore, our $H$ (as defined in Section 4.2) considers the following bytes of each video packet payload: video stream identifier (4 bytes), frame ID (4 bytes), sequence number (4 bytes). And following the recommendations found in Duffield and Grossglauser (2001) our hash key is generated by the following expression:

$$H(\phi(x)) = CRC(\phi(x)) \mod A$$  \hspace{1cm} (13)

where $\phi(x)$ is the byte stream of the fields selected for the hash, $CRC$ is a 32 bit cyclic redundancy check and $A$ is the size of the hash table, which is set to 16,979 for the chunks and the same for the frames within each chunk.

By matching the different descriptors we were able to compute the overlay quality metrics, while the user-perceived QoE was assessed by analysing the video traces delivered by the multimedia player, in the analysis we considered chunks of 10 Kbytes (about 200ms of video) (Tang et al., 2009). It is important to notice that with access to SopCast’s protocol specification, or in the case of using IM, we could have avoided such reverse engineering and accomplished the metric computation easier.

6 Use case: SopCast performance analysis

As a proof of concept, this section deploys MMF in a real testbed to perform a throughout study of SopCast as a P2PTV media streaming solution. The experiments are focused on two different aspects:

1. perceived video quality
2. perceived service quality.

It is not the goal of these experiments to evaluate the performance of SopCast per se, but rather to demonstrate a use case of MMF and how it can be seamlessly used independently of the underlying protocol. Therefore, all the results shown in this section cannot be generalised to SopCast but to our particular use in the PlanetLab testbed.

6.1 Perceived video quality analysis – direct metrics

One of the most relevant aspects of multimedia streaming from the end-user perspective is the perceived quality of the audio and video content. Hence, we first study the perceived unsatisfaction periods of the SopCast users in our testbed, to continue our analysis with the localisation of such unsatisfaction periods and the effects of chunk degradation caused from the overlay point of view.

In the analysis of the perceived quality we take as reference the quality of the received video at application layer, which is extracted by dumping the received video to
disk as we discussed previously. Using this value as reference our goal is now to assess the accuracy and features of MMF in this particular deployment.

6.1.1 User satisfaction

In order to evaluate both MMF and SopCast within the PlanetLab network, we first measure the overall MOS as perceived by the end-users over the whole test duration. To this end, in Figure 3, we plot the average MOS evolution and its confidence interval extracted from all the tests spanning the whole duration of the video session. We obtained these quality values from the application perspective by computing the PSNR of the received video packets and by using Klaue et al. (2003) we extracted the equivalent MOS for each video frame. Since such analysis would not be possible in a real deployment we use these values in the whole section as a reference of ‘perfect knowledge’ about the user perceived video quality, and use it in order to compare the obtained values by our MMF.

As it can be noted in the figure during the warm-up phase of the P2PTV protocol, i.e., the first seconds of the session, the perceived quality is noticeably lower than the one obtained once the protocol is fully operative, where the quality is consistently above 4 in average, which indicates good user perception of the service regarding video quality.

![Figure 3 Overall average MOS for all the tests with the confidence intervals](image)

<table>
<thead>
<tr>
<th>Table 1</th>
<th>Unsatisfaction ratio for all the experiments. With $\hat{Q} &lt; 3$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Unsatisfaction ratio</td>
</tr>
<tr>
<td>Overlay perspective</td>
<td>0</td>
</tr>
<tr>
<td>App. perspective</td>
<td>-0</td>
</tr>
</tbody>
</table>

Analysing the results in more detail, in Table 1 we compare the fraction user unsatisfaction periods observed in our tests from application and overlay perspectives. The total periods of unsatisfaction are obtained by computing the MOS as discussed in Section 3.3.1 for each chunk, each column in the unsatisfaction ratio states the 5th percentile, the average, the 95th percentile, and the maximum respectively. A chunk is considered to produce unsatisfaction when its MOS is below the threshold $\kappa$, i.e., $\hat{Q} < \kappa$ with $\kappa = 3$, while the ratio is computed by counting such chunks over the total chunks sent on each peer in the overlay as indicated by Expr. 14:

$$r = \sum_{i=1}^{m} U(c_i) \left/ \sum_{i=1}^{m} C \right.$$  (14)

As it can be observed in the table, in general from the reference application layer, there is a high satisfaction level among the users, given that on average 10% of the test duration suffered from video quality disruptions. It can also be observed, that 5% of the users have a very good multimedia experience, having a negligible amount of glitches, while there is another 5% who observe larger video disruptions, i.e., > 48% of the experiment duration. In the case of the overlay layer – and thus MMF – we observe in general the results tend to underestimate the effects of the disruptions, this effect is not present at application layer where we have information about the specific video frames. To further highlight these differences in the perceived quality, in Figure 4(a) we plot the cumulative density function (CDF) of the total fraction of unsatisfied periods (Y-axis) per test (X-axis), for convenience we normalise the tests to 1. The figure underlines that around 80% of the tests have unsatisfaction periods below 10%, while the other 20% imply that either the P2PTV protocol or the network used in the deployment have performance issues, given the larger periods of unsatisfaction for the users during the experiment.

<table>
<thead>
<tr>
<th>Table 2</th>
<th>Continuous duration of unsatisfaction periods for all the experiments</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$U_i$ (in seconds)</td>
</tr>
<tr>
<td>Overlay perspective</td>
<td>0.20</td>
</tr>
<tr>
<td>App. perspective</td>
<td>0.20</td>
</tr>
</tbody>
</table>

Note: With $\hat{Q} < 3$

To complement the previous analysis, we now assess the duration of the continuous periods with unsatisfaction. We show the obtained results in Table 2, where the values are computed by counting consecutive chunks whose quality is $\hat{Q} < \kappa$. It is worth noticing that the unsatisfaction duration is critical from the end-user point of view, since having short bursts of low quality might be observed as a small glitch on the video, while longer periods can be devastating from the end-user perspective, the longer is the period the worse is the perceived quality by the user. From the application perspective, the average continuous video disruption is higher than one second, which is fairly annoying from the end-user perspective. However, if we observe Figure 4(b) we notice that, analogously to the previous analysis, most of the time, i.e., 65% of the continuous periods, $U_i$ is below one second. In the figure,
the X-axis holds the normalised periods with unsatisfaction obtained from all the tests, while in the Y-axis we obtain its duration.

From Tables 1 and 2 and Figure 4, we can further notice that the satisfaction computed at the overlay layer tends to overestimate the quality perceived by the users – by delivering lower unsatisfaction ratios and shorter continuous unsatisfaction periods – because the overlay only considers chunk losses, delays, and delay variations, without having a more detailed perspective such as the used codecs, type of the dropped frames, and their actual perceived effects of chunk degradation.

These results show that, in PlanetLab, once the node converges to proper video quality it can cope with the necessary user demands not incurring in any more noticeable service disruptions. Nevertheless, as we notice in the following sections, the overlay continues to observe packet drops, which are successfully recovered by the video codecs, masking its effects to the final users, i.e., MOS higher than \( \kappa \).

### 6.1.2 P2P overlay chunk degradation

In a real deployment, it is not always possible to access application information. Besides, the user-perceived values, sometimes it can be useful for the overlay management to infer the overlay performance directly from the raw metrics such as chunk degradation.

As we discussed previously, a chunk is considered as degraded (or lost) when \( L_{\delta}^c \) is higher than a threshold \( \delta \), where \( \delta \) represents the sensitivity in the estimation of the delivered quality. The loss threshold \( \delta \) has a two-fold utility:

1. for assessing the compliance with different SLA, e.g., setting QoS management strategies, or defining billing policies depending on the delivered quality
2. setup \( \delta \) to align it with the resilience in error concealment of the audio and video codecs.

In Table 3, we represent the chunk losses at the overlay level for \( \delta \geq 0, 5, 10, 15 \) and 20% to analyse various quality levels. To ease the comprehension of the results they are aggregated for the whole set of nodes involved in the testing.

<table>
<thead>
<tr>
<th>( \delta )</th>
<th>w/ losses</th>
<th>Burst length (in chunks)</th>
<th>5th</th>
<th>Avg</th>
<th>50th</th>
<th>95th</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>0%</td>
<td>30.8%</td>
<td>2</td>
<td>3.9</td>
<td>3</td>
<td>9</td>
<td>31</td>
<td></td>
</tr>
<tr>
<td>5%</td>
<td>26.1%</td>
<td>2</td>
<td>3.6</td>
<td>3</td>
<td>8</td>
<td>25</td>
<td></td>
</tr>
<tr>
<td>10%</td>
<td>~18%</td>
<td>2</td>
<td>~3</td>
<td>2</td>
<td>6</td>
<td>14</td>
<td></td>
</tr>
<tr>
<td>15%</td>
<td>10.7%</td>
<td>2</td>
<td>2.5</td>
<td>2</td>
<td>4</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>20%</td>
<td>6.2%</td>
<td>2</td>
<td>2.3</td>
<td>2</td>
<td>4</td>
<td>9</td>
<td></td>
</tr>
</tbody>
</table>

The column labelled w/ losses in the table represents the percentage of chunks \( L_{\delta}^c > \delta \). The other columns hold the 5th, the average, the 50th, the 95th percentile and the maximum burst length respectively for all the 136 nodes in the experiment at the overlay level. As it can be noted, the results reveal that a fairly large amount of chunks suffer some kind of degradation. As we already mentioned, in this particular study, we only consider chunk degradation, without regarding its effects on user satisfaction, i.e., \( Q^c \). Henceforth, these results cannot be directly mapped to the ones in Section 6.1.1, namely, losses with low \( L_{\delta}^c \) might not degrade the MOS noticeably.

![Figure 4 Unsatisfaction results from overlay and application perspectives, (a) total unsatisfaction periods per test (b) continuous unsatisfaction periods](image-url)
the results are aligned with the intuition that bigger $\delta$ implies lower chunk disruption.

Figure 5  Clustering of unsatisfaction periods to the beginning of the experiment

In order to model the behaviour of the chunk degradation in a SopCast overlay, in Figure 6(a) we also plot the different normal distribution fits for each $\delta$. As it can be noted, for low values of $\delta$ the chunk degradation tightly follows a normal distribution. In particular, we observe that in 95% of the cases SopCast users will experience less than 60% of corrupted chunks for $\delta = 0$, while in the best case, i.e., $\delta = 20$, the experienced degradation for 95% of the users will be below 21%, with expected values as already presented in the first column of Table 3. It is important to highlight that most of these disruptions do not represent a noticeable video disruption as $Q(c_i) > \kappa$ due to the correction algorithms present on the video codecs.

As a second analysis, related with the overlay performance, in Table 3 we show the effects of $\delta$ on the burst length (i.e., consecutive chunks with $Q(c_i) < \kappa$), which ranges from 460 ms to 780 ms in average, for $\delta = 20\%$ and $0\%$, respectively. If consecutive chunks are degraded, the video/audio quality quickly drops quickly to unbearable limits, that is, we pass from a small glitch in the video to hard video disruption or even full blocking. With this rationale we analysed the traces and looked for bursty lost chunks. In Figure 6(b), we study the ratio of the burst duration for different $\delta$, for convenience we also show the isolated P2PCL (i.e., burst lengths of 1). In the results most of the losses affect only single or two consecutive chunks, which in general means that the quality degradation is temporary and can be quite easily corrected by the used multimedia codec. Nevertheless, some losses (which take from ~15% to ~5% of the total depending on $\delta$) are periods affecting at least three chunks (i.e., around 600 ms of disruption), sensibly reducing the video broadcasting quality, as they affect the end-user perception.

From our specific deployment, we can derive that SopCast experiences a fair amount of frame degradation periods. In our opinion, this can be caused by two different reasons:

1. poor implementation of the protocol
2. because of the underlying network (i.e., PlanetLab).

6.2 Perceived service quality analysis – indirect metrics

Besides perceived video quality, there are many other factors that can affect the quality perception by the users. In this section, we consider the start-up time, the content freshness and the experienced blocking in the video playback.

6.2.1 Start-up time

For each user, the streaming process in SopCast traverses two buffers in local memory: the SopCast buffer and the media player buffer, plus the different overhead caused by the intermediate peers. A SopCast Client downloads the media streaming content chunk by chunk from different peers, and reassembles them in order as video packets,
forming the streaming at its local memory. We understand
the start-up time in a SopCast overlay as the lapse of time
from the connection start in a SopCast TV channel, to the
time that the video is formed on the local streaming engine.
After this time, a media player can be launched and it can
playback the media from the SopCast buffer.

Since the chunks come from different peers, and each
node needs different amount of time to wait for the video to
be formed, the start-up time is a relevant measure to infer
the time an user must wait to get the contents. In Figure 7,
we plot the CDF of the obtained start-up times for each
node in our deployment.

The average value for the start-up time in the SopCast
buffer is 10.79 seconds. Then, adding the adjustable media
player threshold (between 1 and 60 seconds for typical
media players), a node needs to wait for at least
11.79 seconds in average to see the media being displayed
since it is requested. In the figure we can also observe that
85% of nodes has less than 20-second start-up time. Since
the curve follows a power-law like curve, the chance to
experience a large start-up times (such as more than
1 minute) is very rare.

6.2.2 Content freshness

Another relevant aspect to assess the user satisfaction is the
freshness of the content being broadcasted, i.e., how long it
takes since the content is generated until its reception at the
end-user. In practice, in a SopCast overlay, the P2PAOWD
tends to be very high given the overhead caused by the
different peers, and especially caused by SopCast and the
media player buffers. Each intermediate node has to wait for
the reception of the full chunk, buffer it, recode it (this is a
specific SopCast behaviour), and finally send it to the next
hop in the overlay network, which triggers the same
procedure once again. Moreover, not all nodes join the
overlay at the same time, which forces the content generator
to maintain a fairly large time window to broadcast to the
peers, yet increasing P2PAOWD once again.

Analysing P2PAOWD in more detail, in Figure 8 we
show the CDF of the Averaged P2PAOWD per node,
obtained from all the peers in the overlay. For clarification,
we also show the minimum and maximum boundaries of
P2PAOWD in dashed-lines.

The average P2PAOWD ranges from 154 to
1,139 seconds. One of the main reasons for this broad range
is caused by the start-up times and the big variance in the
buffering both in the intermediate peers as in the content
provider.

The final effect on the user satisfaction of the content
freshness is closely related to the content being broadcasted,
e.g., when broadcasting stored content, such as movies or
news programmes, having large start-up times is less
relevant to the users than when broadcasting live events,
such as live sports.

6.2.3 Video blocking

When a video stream experiences blocking there are two
different alternatives:

1. discard the chunks and try to continue the playback
2. block and wait for more information.

In general, media players implement a mixed approach,
where, if there are few late chunks they try to correct the
input without blocking the playback. Or, in the case of a
high amount of late chunks, the application just blocks
(most likely because it does not have any more information
to reproduce), starting the buffering procedure from scratch,
thus causing stalling on the video.

Differently from content freshness, video blocking is
related to P2PAICDV. As a consequence, avoiding the
blocking is normally performed through big de-jitter
buffers, in the particular case of SopCast, the present buffers
range from 1 to 60 s as inferred in Sentinelli et al. (2007).
Hence, SopCast can avoid blocking while the P2PAICDV is
smaller than the current buffer usage.

6.2.3 Video blocking

In Table 4, we detail the minimum, maximum and average
of blocking caused by late chunks. We also show the
amount of blocking caused by the warm-up phase (when
selecting the best peers). We do this second analysis in
order to evaluate whether the user is able to notice the difference between when the swarm is setup and when the system reaches the more stable steady state, which, as we studied before, affected the start-up time and chunk degradation noticeably.

Specifically, in Table 4, we detail the total amount of blocking in all the nodes regarding the start-up phase of the system. We also show the amount of blocking that is caused within the first 5, 15, 30 and 60 seconds – with respect to all the blocking present in the system.

<table>
<thead>
<tr>
<th>Table 4</th>
<th>Blocking ratio for all the tests: number of late chunks over the total received, and blocking occurred in the first seconds of the video streaming</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Blocking ratio</td>
</tr>
<tr>
<td></td>
<td>Min.</td>
</tr>
<tr>
<td>All tests</td>
<td>0%</td>
</tr>
<tr>
<td>First 5 sec.</td>
<td>0%</td>
</tr>
<tr>
<td>First 15 sec.</td>
<td>0%</td>
</tr>
<tr>
<td>First 30 sec.</td>
<td>0%</td>
</tr>
<tr>
<td>First 60 sec.</td>
<td>0%</td>
</tr>
</tbody>
</table>

It can be observed that a non-negligible amount of blocking is caused in the start-up phase, which has an average between 6.4% and 27.4% of the total blocking. This means that up to almost one third of the blocking occurs within the first minute of the video streaming. We also found out that in some tests (in particular in 2 out of 136) all the blocking happens within the first five seconds of the test.

### 7 Conclusions

In this paper, we presented MMF, a framework to perform P2PTV quality assessment by using a formalisation of multilayer performance metrics. The assessment is performed incrementally, first we defined the metrics for a P2P overlay, considering its specific constraints; second we extended these definitions to cope with application layer particularities; and finally we used those metrics to infer the end-user perceived P2PTV quality in aspects such as video degradation, start-up times, content freshness, and video blocking by using the proposed MMF.

The deployment of MMF has many potential applications, e.g., to integrate it in the P2PTV protocols for online quality assessment, to perform billing and accounting based on the media quality, to analyse P2PTV service deployment feasibility, to study end-user satisfaction, or to profile P2PTV protocol requirements. In this work, and as a proof of concept, we used the MMF and the multi-layer metrics to analyse the performance of SopCast in the PlanetLab research network. We have found that SopCast provides a scalable protocol, which with relatively low amount of resources can provide a good framework for live P2PTV broadcasting. However, we also noticed that SopCast has non-negligible start-up times, and that the content observes very high lags in its delivery. This can be an important issue when performing live streaming such as sport events (e.g., Football World Cup finals), where such a lag can be considered as unbearable by the customers.

After the analysis performed in this work, we left a number of open issues which require further development. In particular, we plan to validate and extend the proposed metrics with subjective QoE assessment with real users. We also plan to extend the QoE assessment metrics by providing non-reference estimation facilities embedded into the system. Such metrics could aid in improving the internal workings of P2PTV protocols in aspects such as peer selection, or bandwidth scheduling depending on the user perceived multimedia quality.

Another open issue left for our future work involves the integration of our MMF proposal into an existing P2PTV application, e.g., PeerCast, in order to compare the efficiency and accuracy of both IM and DM approaches and evaluate the advantages and shortcomings of each alternative. Finally, we plan to extend this work in order to provide a comparison of the impact on each layer of the metrics, especially with the goal of comparing QoS and QoE approaches to multimedia quality assessment.

### Acknowledgements

The authors would like to thank both Yue Lu and Fernando Kuipers from Delft Technical University for the access to the SopCast traces, along with many technical discussions about the performance assessment with the P2PTV system.

### References


