Providing Fast Channel Switching in P2P IPTV Systems

Daniel A. G. Manzato and Nelson L. S. da Fonseca

***Abstract*—In spite of the increasing deployment of IPTV services, various functionalities still need to be improved. One of the main challenges is a reduction in startup delays, especially in channel switchings, a problem which is quite relevant in P2P IPTV systems due to bandwidth limitations, as well as the employment of buffers and overlay structures. This paper presents and compares four novel schemes for providing fast channel switching that reduce the occurrence of latency. The results suggest that, in general, the proposed schemes are all capable of performing 68% of all channel switchings instantaneously on average. Moreover, the *Fast 2* scheme, which presented the best performance, reduces the overall stream quality received less than 19%, on average.**

***Index Terms*—IPTV, fast channel switching, peer to peer networking, peer to peer computing, multimedia streaming, multimedia communication, multimedia systems, digital multimedia broadcasting, quality of service.**

I. INTRODUCTION

IN spite of the increasing deployment of IPTV services, various functionalities still need to be improved [1], [2]. One of the main issues is a reduction in startup delays, especially in channel switchings. In P2P IPTV systems, due to bandwidth limitations, only a portion of the broadcast content is transmitted to the users. Moreover, although buffers and overlay structures are employed to ameliorate the problems arising from network bandwidth fluctuations and connection failure, these cause playback latency and degrade the usability of the system, especially when fast navigation through multiple channels is desired.

The PPLive, a typical P2P IPTV system [3], needs a certain amount of time to store tens of seconds of video frames in buffers before playback [1]. New peers can spend as long as 10 to 15 seconds before they can join a P2P overlay [4], and it can take another 10 to 15 seconds to launch the media player and store the video frames in the buffers. Since the users of traditional television expect to be able to switch channels quickly [1], P2P IPTV services must reduce startup delays to just a few seconds [1].

This paper presents and compares four novel schemes for providing fast channel switching in P2P IPTV systems that reduce the occurrence of latencies caused by buffers and overlay structures. The novelty of these schemes is the employment of multiple description coding (MDC) [5]–[7] and multiple distribution trees. Multiple description coding generates sub-streams with a minimum quality for each channel. These are transmitted through multiple distribution trees. By combining a set of sub-streams, it is possible to provide users with the option of multiple channels with minimum quality instead of a single channel with full quality, but the reduced quality is still sufficient to verify the programs being broadcast. When the user requests a channel that is received in background, the switching of channels is immediate. This approach yields better network resource utilization and facilitates the adjustment of stream qualities according to user viewing state, as well as the adjustment to different situations of bandwidth contention.

To increase the probability of immediate switching, the proposed schemes define strategies for the selection of the content most likely to be requested. The set of sub-streams selected for transmission will have either various descriptions of the same channel or single descriptions of multiple channels. In previous work [8], [9], the benefits of this approach were shown. Although other proposals in the literature have adopted the transmission of redundant streams with minimum quality [10]–[12], to the best of the authors' knowledge, the proposed schemes are the first to employ multiple description coding and multiple distribution trees for this purpose.

The effectiveness of the proposed schemes was evaluated via simulation using various scenarios employing realistic values for network parameters, different arrival patterns and different sizes of distribution trees. The proposed schemes are all capable of performing 68% of all channel switchings instantaneously on average. Moreover, the *Fast 2* scheme, which presented the best performance, reduces the overall stream quality received less than 19%, on average.

The remainder of this paper is organized as follows. In Section II, related work is presented. In Section III, an example of an architecture for P2P IPTV in which the proposed channel switching schemes can be employed is discussed. In Section IV, the proposed schemes are introduced, and in Section V, simulation experiments are described. In Section VI, the schemes are evaluated, and in Section VII, conclusions are drawn.

II. RELATED WORK

Several schemes have been proposed for channel switching in IPTV systems, all designed to reduce different components of the total switching delay. According to the literature [10], [13], [14], these schemes can be classified into four categories. One of them is formed by schemes that try to reduce the network delay [14]–[16], which is the period of time the set-top box needs to leave the current multicast group and join the multicast group of the newly requested channel. Although Ramos et al. [13] consider the network delay to be irrelevant

Manuscript received February 28, 2012; revised July 14, 2012.

The authors are with the State University of Campinas, Institute of Computing, Av Albert Einstein 1251, 13083-852, Campinas, SP, Brazil (e-mail: {dmanzato, nfonseca}@ic.unicamp.br).

Digital Object Identifier 10.1109/JSAC.2013.SUP.0513029

to the total delay, given that in commercial IPTV it is usually shorter than 100-200 ms, this delay is quite significant in P2P IPTV, since it can reach values as long as 10-15 seconds [4].

In [16], a channel switching scheme was proposed for P2P mesh systems [3], [17]. In this scheme, peers maintain information about contact neighbors in the overlay of the watched channel, as well as about contact neighbors in the overlays of the channels most likely to be requested. Although results show that this scheme performs 58% of the switchings locally, without any need of contacting the server, they do not show the real impact on startup delays. Moreover, a major drawback of the scheme is that it is specific for P2P mesh systems, which naturally involve longer startup delays than do P2P multiple tree systems [2].

Another category comprises schemes employing certain video coding techniques to reduce the synchronization delay [18]–[20], which is the period of time the set-top box needs to wait for a reference frame to start decoding the video stream. The synchronization delay takes, on average, half the group of pictures (GOP) duration [13], and typically lasts from 500-2000 ms [18]. This period of time is more significant in commercial IPTV than it is in P2P IPTV, considering the total delay of both types of systems.

A third category is composed by schemes employing proxy servers with boost streams [13], using either unicast [21], [22] or multicast [12], [23], [24]. Such schemes seek to reduce both the synchronization and buffering delays by furnishing a secondary stream with a higher bit rate or a larger number of reference frames concurrent with the main one. When the main stream is ready to be reproduced, the secondary stream is dismissed. The buffering delay varies depending on the amount of protection required by the system, with typical values of 1-2 seconds in commercial IPTV [13], and of 10-15 seconds in P2P IPTV [4]. Thus, this delay is quite significant in the total delay of both types of systems.

The scheme proposed in [12] is similar to ours; using additional multicast groups, it delivers streams with minimum quality for each channel. However, while in the present paper multiple description coding and multiple distribution trees are used to enable pre-joining of multiple channels, in [12] the coding scheme is MPEG-4 and only a single additional multicast group per channel is employed as a means of providing boost streams. The results of the two approaches differ mainly in the following: while the scheme in [12] *reduces* the playback latencies of *all* channel switchings *to half*, the schemes in the present paper *nullify* the playback latencies of *a significant portion* of the channel switchings (67.74% of all switchings). One disadvantage of the scheme proposed in [12] is the 50% I/O overhead on the servers for any channel with at least one user surfing. Moreover, all channels need to be recoded in order to obtain the streams with minimum quality. The schemes proposed in the present paper do not impose any such overheads, since they employ multiple description coding.

The fourth category comprises pre-join schemes [10], [11], [13], [25]–[29] which try to predict the next channels that will be requested by the user and then transmit them redundantly in advance in order to make them available by the time of the next channel switching. This approach reduces the total delay by addressing all the above mentioned delays (network, synchronization and buffering). The schemes proposed in the present paper fall into this category.

TABLE I

CATEGORIES OF CHANNEL SWITCHING SCHEMES AVAILABLE AND COMPONENTS OF THE TOTAL SWITCHING DELAY ADDRESSED, INCLUDING SIGNIFICANCE IN P2P AND COMMERCIAL IPTV.

Categories of Channel Switching Schemes	**Network Delay**	**Synchron. Delay**	**Buffering Delay**
1. Reduction of Network Delay	×		
2. Video Coding Techniques		×	
3. Proxies with Boost Streams		×	×
4. Pre-join Schemes	×	×	×
Significance in P2P IPTV	○ ○ ○	○○	○ ○ ○
Significance in Commercial IPTV	○	○ ○ ○	○ ○ ○

To address the disadvantage of the existing pre-join schemes which assume that the user's bandwidth will be large enough to receive simultaneously multiple streams in full quality, the pre-join schemes proposed in [10], [11] employed an extension of the H.264/AVC video coding, called scalable video coding (SVC), to generate low quality streams. The scheme in [10] extended the idea of [11] by pre-joining a small number of channels with minimum quality during watching periods, in addition to the watched channel in full quality. During surfing periods, only channels with minimum quality are pre-joined, as in [11]. Although this approach has similarities to ours including the adoption of low quality streams and two viewing states, our proposal differs from that in [10] by the use of multiple description coding; moreover, it seeks to reduce as much as possible the occurrence of delayed channel switching events while maintaining as much as possible the stream quality, so that the user's bandwidth can be better utilized. Furthermore, our proposal employs three different strategies for selecting the redundant channels to be transmitted in advance (adjacent, popular and old channels), whereas that in [10] considers the buttons most recently pushed by the user in a similar fashion as in [27]. A further difference is that the schemes proposed in the present paper employ a gradual transition from one viewing state to the other, which is facilitated by the use of multiple description coding and results in enhanced stream quality.

Table I summarizes the categories of channel switching schemes available and the components of the total switching delay addressed, including their significance in P2P and commercial IPTV.

III. P2P IPTV ARCHITECTURE

There are two main approaches for video distribution in P2P networks: trees [30]–[32] and mesh [3], [17]. The tree approach, based on multicasting at the application layer, has a distribution structure always ready for transmission, thus avoiding the overhead of transmission scheduling and, as a consequence, reducing the startup delay. The mesh approach employs connections on demand, thus avoiding the cost of maintaining active distribution structures. However, it adds overhead for content dissemination, as well as for transmission scheduling. Since one of the main challenges of P2P IPTV systems is to reduce startup delays, especially in channel

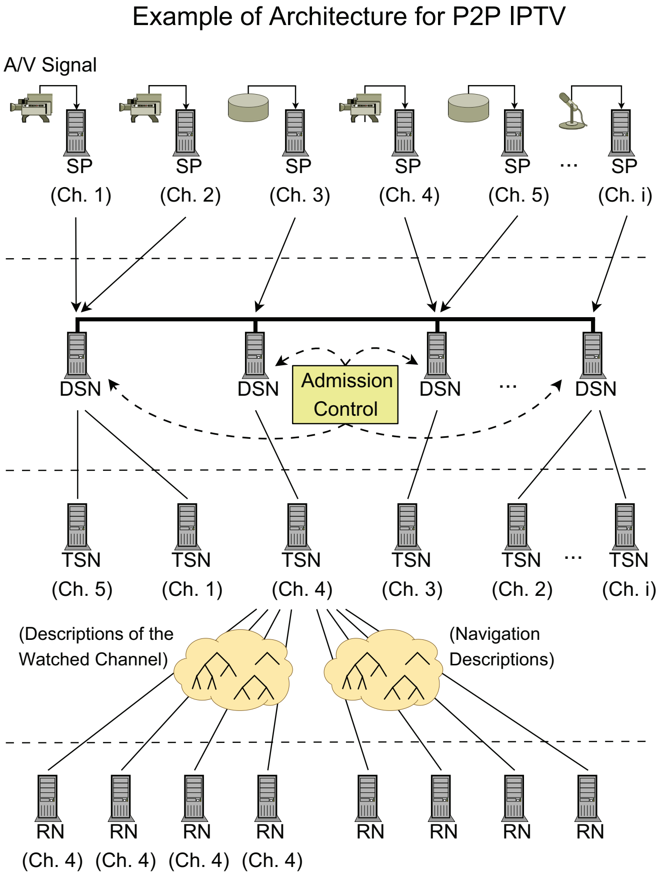

Fig. 1. Example of an architecture for P2P IPTV which employs multiple distribution trees [2].

switchings, the tree approach is considered to be the most interesting [2].

Fig. 1 provides an example of an architecture for P2P IPTV which employs multiple distribution trees [2]. Components called Dedicated Super Nodes (DSNs) are responsible for propagating streams from all channels and making them available to users. The streams are generated by components called Stream Providers (SPs) and transmitted directly to DSNs, with each DSN admitting one or more SPs. Since streams are encoded in multiple description coding, all DSNs make available all descriptions of all existing channels.

Peers are classified into two categories according to their capacities: Temporary Super Nodes (TSNs) and Regular Nodes (RNs). The TSNs connect directly to DSNs and admit RNs, cooperating with the system in the tasks of tree management and content distribution. Each RN is admitted under a TSN through multiple distribution trees, with each tree used for the transmission of a description of a specific channel. As each TSN provides only a single channel in full quality, the channel selection of an RN will determine its admittance under a specific TSN. In addition to serving this channel in full quality, a TSN also forwards navigation descriptions of other channels, thus enabling fast navigation to RNs.

IV. PROPOSED CHANNEL SWITCHING SCHEMES

This section introduces the proposed channel switching schemes.

A. Background

The idea behind the proposed schemes is that users do not need to receive streams in full quality to verify the content being broadcast on a channel. For the identification of this content, only minimum quality is sufficient [8], [9]. Since such streams have low bandwidth demands, multiple channels can be transmitted simultaneously, and media players can keep multiple buffers continuously filled for immediate playback of these channels. It is thus possible to provide instantaneous channel switching to the content stored in the buffers. The set of channels stored can vary depending on the user, as well as on the bandwidth of the streams [8], [9].

The proposed schemes rely on only two specific characteristics of video streaming: multiple distribution trees and multiple description coding. The employment of multiple distribution trees in P2P IPTV systems is common and preferable to mesh topologies [2], [33]. For channel switching, this approach helps to reduce startup delays by avoiding the overhead of transmission scheduling. Multiple description coding is employed to divide a single channel into multiple sub-streams (or descriptions) with minimum quality, which are then transmitted via multiple distribution trees. There are various ways of implementing multiple description coding [5], [34] and the proposed schemes are not affected by a specific implementation. For instance, in platforms for which a video codec is already available, a technique based only on pre-/post-processing and the use of legacy coders [34] could be used. Besides significantly reducing the development time, this technique demands minimal infrastructure change [34]. Moreover, since P2P IPTV systems usually employ multicasting at the application layer, no infrastructure change will be required to implement multiple distribution trees. In this way, the proposed schemes can be easily adopted for P2P IPTV.

Multiple description coding presents several advantages. First, all the descriptions have the same importance; as a consequence, *any one* of them can provide the stream with minimum quality for the channel, and the loss of a subset of the descriptions does not compromise the reassembly of the original stream, as would be the case with layered coding. Another advantage is that the final quality of the reassembled stream is proportional to the number of descriptions received. This feature allows incremental improvement of the stream with minimum quality as additional descriptions are received. In P2P IPTV, this flexibility is very important in providing adaptations for bandwidth heterogeneity. Finally, this approach eliminates the need for recoding the original video to generate streams with minimum quality, thus avoiding additional overhead of I/O and processing at the stream providers [8].

The employment of the proposed schemes in P2P IPTV requires only the availability of the streams with minimum quality to the connected users. This means that a user must be able to be admitted into at least one distribution tree for each channel, as well as into all distribution trees for the watched channel when she/he is not surfing [2]. In a P2P IPTV environment, the management overhead of employing multiple trees is distributed among peers, as in the architecture proposed in [2], which distributes the overall load by employing multiple temporary servers (TSNs), which are themselves

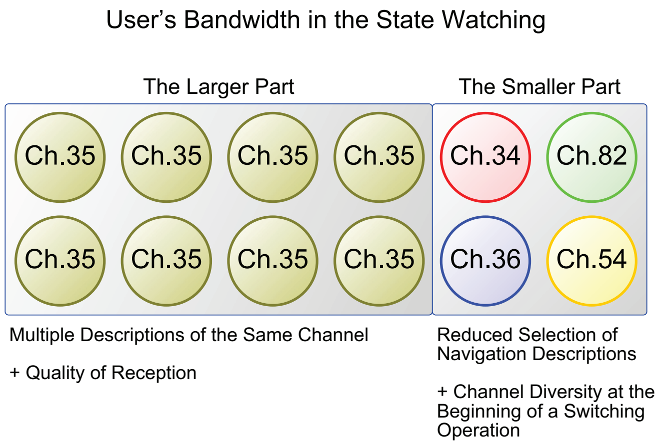

Fig. 2. Utilization of the user's bandwidth in the state *watching*.

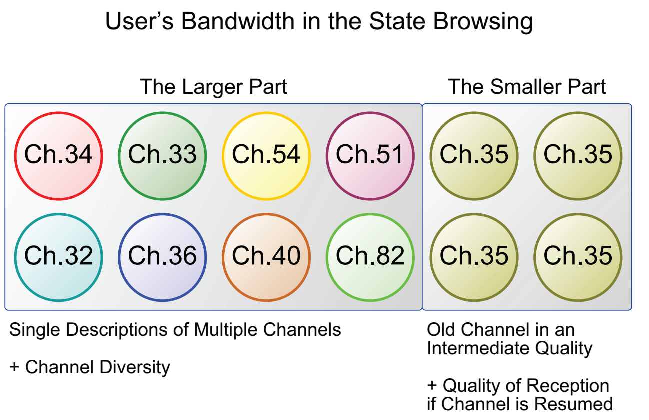

Fig. 3. Utilization of the user's bandwidth in the state *browsing*.

peers.

The availability of the descriptions of all channels, offered via multiple distribution trees, makes it possible to select and transmit a subset of them to the user. It is important to note, however, that the aggregate bandwidth of this selection cannot exceed the user's incoming bandwidth. Depending on whether the user is surfing or not, the selection received will vary. Therefore, the proposed schemes define two viewing states: *watching* and *browsing*, illustrated in Figs. 2 and 3, respectively, for the bandwidth of a single user. In the state *watching*, the user's bandwidth is employed primarily for the reception of the descriptions of the watched channel, while in the state *browsing* the bandwidth is used primarily for the reception of descriptions of different channels, i.e., streams with minimum quality from other channels.

In order to decrease the probability of startup delays at the beginning of a channel switching operation, in the state *watching* part of the user's bandwidth is reserved for the reception of a reduced selection of streams with minimum quality. These streams consist of the channels most likely to be requested in that state. There is a clear trade-off between providing higher quality of reception for a watched channel and providing access to more channels to increase the probability of instantaneous channel switching when this operation is initiated. In the state *browsing*, part of the user's bandwidth is also reserved for partially maintaining the reception of the old channel. Thus, if the user returns to the old channel at the end of a channel switching operation, this channel will already be available and will have an intermediate quality, higher than that provided during navigation, but lower than that obtained in the state *watching*. According to [35], the probability of a user returning to the old channel after terminating a channel switching operation is about 17%, since many channel switchings occur during advertisements, when users check what is being broadcast on other channels.

It is also known that 56–60% of channel switchings involve sequential channels (linear), with 69–72% of them upward and 28–31% of them downward [35], [36]. This information suggests that in both states part of the user's bandwidth reserved for navigation should be for channels which are adjacent to the watched one, with the relevant upward/downward proportions preserved. Switchings to non-sequential channels (non-linear) correspond to 40–44%, which suggests that another part of the user's bandwidth reserved for navigation should be for popular channels, since these have a high probability of being selected. Considering these patterns, three different strategies for the selection of navigation descriptions have been proposed: old channel, popular channels and adjacent channels. These strategies should always be used together by the proposed schemes for a better prediction of the channels most likely to be requested by a given user.

Whenever a new channel is requested, if at least one description of this channel is available in the buffers of the media player, the change is immediate, because the content is ready to be played. If, however, no description of the requested channel is available, latency is introduced due to the time required for admission into the new distribution tree, as well as to the time used for synchronizing and buffering the video frames. The proposed schemes fall into the fourth category discussed in Section II (pre-join schemes), addressing all three of these components of the total switching delay: network, synchronization and buffering.

Since the probability of channel switching is higher in the state *browsing*, channel diversity is given priority with the transmission of single descriptions of multiple channels, whereas in the state *watching*, the quality of reception is given priority with the transmission of multiple descriptions of the same channel. The proposed schemes are thus designed to address not only usability, but also the quality of service of the system.

Next, the schemes themselves are introduced.

B. *Linear Scheme*

Fig. 4 illustrates the state transitions for the *Linear*, *Fast* and *Slow* schemes. When a user is in the state *watching* and starts a channel switching operation, most of the incoming bandwidth is released by dismissing a subset of the descriptions received for the watched channel. This bandwidth is used to receive more navigation descriptions of other channels. Such an operation is implemented by the removal of the user from one set of distribution trees and her/his admission into another, with the state changed to *browsing*.

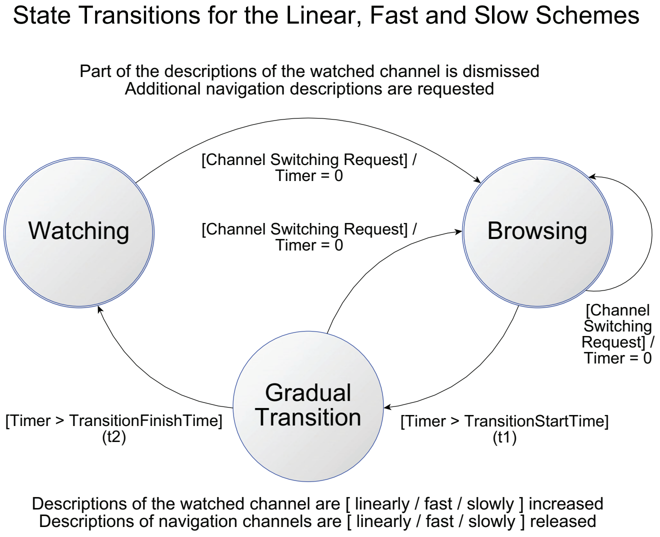

Fig. 4. State transitions for the *Linear*, *Fast* and *Slow* schemes.

Each time a user enters the state *browsing* or requests a new channel change, an individual timer is started, which counts the time elapsed since the last channel change. During an initial period, i.e., the timer value is in the interval of [0:$TransitionStartTime$) seconds, no change is made in the descriptions provided to the user. During the following period, i.e., while the timer value is in the interval of [$TransitionStartTime$:$TransitionFinishTime$], a gradual transition back to the state *watching* takes place. If the timer reaches the value of $TransitionFinishTime$ seconds, the state is changed back to *watching*; otherwise, the user remains in the state *browsing*. Both parameters $TransitionStartTime$ and $TransitionFinishTime$ are subject to configuration.

In the *Linear* scheme [8], the gradual state transition consists in increasing linearly with time the number of descriptions received for the watched channel until the quality of reception reaches the maximum value (which is that of the state *watching*). This is accomplished by periodic update events every $TransitionUpdatePeriodicity$ seconds during the interval of [$TransitionStartTime$:$TransitionFinishTime$]. The parameter $TransitionUpdatePeriodicity$ can also be configured. Let T be the number of update events in the interval ($T = \lfloor(TransitionFinishTime - TransitionStartTime)/TransitionUpdatePeriodicity\rfloor$), t the discrete number of the current update event ($0 <= t <= T$), and M the maximum number of descriptions in the state *watching*. The number of descriptions received (n) for the watched channel at the current discrete time (t) is given by:

$$n(t) = 1 + \lfloor (t/T) * (M-1) \rfloor \tag{1}$$

The number of descriptions varies in the set $[1..M]$, since at least one description of the watched channel must be provided to the user at any given time. To avoid the aggregate bandwidth exceeding the user's incoming bandwidth limitations, the *Linear* scheme gradually releases the descriptions of the non-watched channels. This operation is also implemented by the user's departure from one set of distribution trees and acceptance into another. If, during the state transition, the user requests another channel, then the additional descriptions received for the watched channel are released, and new navigation descriptions are requested. Since the update of the old channel only occurs when the user leaves the state *watching*, no change related to this strategy occurs at this moment. At the end of the transition, the user is already receiving all the descriptions of the new watched channel, all the descriptions of the old channel have been released, and the state is changed back to *watching*.

When releasing the descriptions of the non-watched channels, $n-1$ descriptions of the old channel are first gradually dismissed, so that at least one description remains to ensure immediate switching in case the user decides to return to that channel. The scheme then gradually dismisses the navigation descriptions with a low probability of being requested until the number of navigation descriptions is equivalent to what is established for the state *watching*. At this time, the last description of the old channel is dismissed, thus completing the transition.

C. *Fast Scheme*

In the *Fast* scheme [9], the transition from the state *watching* to the state *browsing* occurs in the same way as in the *Linear* scheme, with the user initiating a channel switching operation. As with the *Linear* scheme, the transition from the state *browsing* to the state *watching* is a gradual one over the same interval of [$TransitionStartTime$:$TransitionFinishTime$] seconds. However, in the *Fast* scheme, the number of descriptions received for the watched channel increases at a logarithmic rate during the gradual transition, with the steepness of this logarithmic rate defined by the factor C. The number of descriptions received (n) for the watched channel at the current discrete time (t) is given by:

$$n(t) = 1 + \left\lfloor \left(\frac{\ln(C*t+1)}{\ln(C*T+1)} \right) * (M-1) \right\rfloor \tag{2}$$

As in the *Linear* scheme, the descriptions of the non-watched channels are released gradually, following the same order. The value of C varies between 0.5 and 2000, giving rise to two versions of the scheme. In the *Fast 1* scheme, the value of C is defined so that an increase of 80% in the descriptions is completed during the first half of the interval. In the *Fast 2* scheme, the value of C is defined so that an increase of 80% in the descriptions is completed during the first 10% of the interval.

Both of these *Fast* schemes are designed to improve the stream quality in the state *browsing* in comparison to the *Linear* scheme, while maintaining as much as possible its reduction of the number of delayed channel switching events. This is done by advancing the release of the navigation descriptions during the gradual state transition, thus allowing more room to accommodate additional descriptions of the watched channel.

D. *Slow Scheme*

In the *Slow* scheme [9], the transition from the state *watching* to the state *browsing* is also similar to what happens in the *Linear* scheme, with the transition from the state *browsing* to the state *watching* taking place gradually during the same interval of $[TransitionStartTime{:}TransitionFinishTime]$ seconds. Nevertheless, during the gradual transition in the *Slow* scheme, the number of descriptions received for the watched channel increases at a rate which is the inverse of the logarithmic function adopted for the *Fast* scheme. The number of descriptions received (n) for the watched channel at the current discrete time (t) is given by:

$$n(t) = 1 + \left\lfloor \left(1 - \left(\frac{\ln(C*(T-t)+1)}{\ln(C*T+1)}\right)\right) * (M-1) \right\rfloor \tag{3}$$

The descriptions of the non-watched channels are also released gradually, following the same order of the *Linear* scheme. Two versions are also available, with the value of C varying between 0.5 and 2000. In the *Slow 1* scheme, the value of C is defined so that an increase of 20% in the descriptions is completed during the first half of the interval, whereas in the *Slow 2* scheme, the value of C is defined so that an increase of 20% in the descriptions is completed during the first 90% of the interval.

Both of these *Slow* schemes are designed to further reduce the number of delayed channel switching events in comparison to the *Linear* scheme, while maintaining as much as possible its stream quality in the state *browsing*. To accomplish this, these schemes postpone the release of the navigation descriptions during the gradual state transition, so that there is a higher chance of at least one description of the requested channel being available when channel switching occurs.

E. Immediate Scheme

Fig. 5 illustrates the state transitions for the *Immediate* scheme [9]. This scheme does not employ a gradual transition. When the user is in the state *browsing* and the initial interval of $[0{:}TransitionStartTime)$ seconds has elapsed, the additional navigation descriptions are released all at once, including the descriptions of the old channel. Moreover, all the additional descriptions of the watched channel are requested at the same time, and the state is immediately changed back to *watching*. The transition from the state *watching* to the state *browsing* occurs in the same way as in the previous schemes, and the parameters $TransitionFinishTime$ and $TransitionUpdatePeriodicity$ are not included in the scheme.

The number of descriptions requested (n) for the watched channel at the time $t = TransitionStartTime$ is given by:

$$n(t) = M \tag{4}$$

The $TransitionStartTime$ parameter varies in the set: [5, 30, 60] seconds, giving rise to three versions of the scheme. In the *Immediate 1* scheme, the immediate transition occurs at the same time that the gradual transition starts in the other schemes ($TransitionStartTime = 5$). In the *Immediate 2* scheme, the immediate transition occurs at a time corresponding to the middle of the gradual transition time in the other schemes ($TransitionStartTime = 30$). In the *Immediate 3* scheme, the immediate transition occurs at the same time as the gradual transition ends in the other schemes ($TransitionStartTime = 60$).

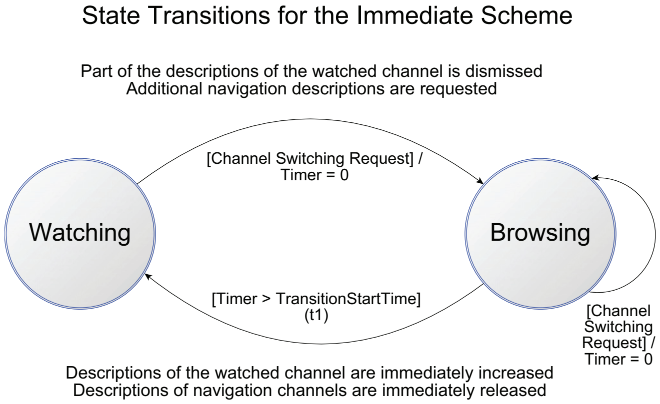

Fig. 5. State transitions for the *Immediate* scheme.

While the aim of the *Immediate 1* scheme is similar to that of the *Fast* schemes, the aim of the *Immediate 3* scheme is similar to that of the *Slow* schemes. The *Immediate 2* scheme represents an intermediate trade-off between them. Moreover, all of these *Immediate* schemes balance the benefits of a gradual state transition, as well as establishing upper bounds for advancing or postponing the release of the navigation descriptions.

V. SIMULATION EXPERIMENTS

A simulator for P2P IPTV systems was developed to assess the effectiveness of the proposed schemes, since to the best of the authors' knowledge, none of the existing simulators implementing the fundamental operations of IPTV services is publicly available. Although a tool was developed in [36] for the generation of synthetic traffic (constructed from statistical models derived from real traces), it was not available for use at the time of the development of this paper, so the simulator developed had to implement the fundamental operations of IPTV services from the statistical models introduced in [35] and [36]. The simulator was validated considering the different characteristics of the synthetic traffic generated, the correctness of processing for each event type, and the resulting effects on the system, such as the total number of users at a time, daily usage patterns, rates of arrival, departure and channel switching, channel popularity and transition probabilities, as well as the distribution of the existing connection classes. Although the models from [35] and [36] were derived from a commercial IPTV system, they were used because there were no equivalent models for P2P IPTV available at the time of the development of this paper; moreover, the traffic for commercial IPTV is more demanding than that for P2P IPTV, typically with a higher rate of channel switching requests, and its use yields to evaluation of the P2P IPTV system under high demands.

The arrival of new peers was modeled by a non-stationary process consisting of sequences of piecewise-stationary Poisson arrival processes, each lasting 15 minutes [37], [38]. The rates of these sequences varied from 5 to 2000 arrivals per minute, and they were defined to reflect the total number of users of a real system per stationary period [35]. As a result, daily patterns of real systems were reproduced, with two main peaks around 3PM and 10PM, and a smaller one around 8AM [35]. The session duration followed a Lognormal distribution, with the parameters $\mu = 6.351$ and $\sigma = 2.01$ [37], for an average period of 4320 seconds (1.2 hours) [35]. Channel holding times were modeled using a histogram built from a trace collected in an operational IPTV system [35] (mean and median of 14.8 minutes and 8 seconds, respectively). A seven-day period was simulated, generating more than 62 million channel switching events.

The number of channels used was 105 [35]. Channel popularities and transition probabilities were derived from information and data obtained from [35] and [36]. The viewing time threshold was set at 60 seconds [35]. To simulate heterogeneous access profiles, connection classes were defined according to the statistics of the Brazilian Internet access pattern [39], with classes of ADSL 1 Mbps, ADSL 2 Mbps, ADSL 4 Mbps, ADSL 8 Mbps, HFC 500 Kbps, HFC 3 Mbps, HFC 6 Mbps, and HFC 12 Mbps. Stream bandwidth values varied in the set: 300 Kbps, 1024 Kbps and 2048 Kbps [1]. The number of distribution trees varied in the set: 4, 8, 16 and 32 [30], [40]. Startup delays (period of time necessary for admission into the trees and for buffering video frames) were modeled by a uniform distribution in the interval $[5-20]$ seconds [1], [4].

Since most of the channel switching events occur about 4 seconds after the last change [35], the default value of $TransitionStartTime$ was set at 5 seconds. Following the viewing time threshold adopted in [35], the value of $TransitionFinishTime$ was set at 60 seconds. The value of $TransitionUpdatePeriodicity$ was set at 1 second. In the state *watching*, the fraction of the user's bandwidth reserved for navigation was set at 50%, with the remainder employed to receive the descriptions of the watched channel. In the state *browsing*, the fraction of the user's bandwidth reserved for receiving the descriptions of the old channel was set at 50%, with the remainder employed to receive navigation descriptions. In both states, the fraction of the navigation bandwidth used for adjacent channels was set at 56%, with the remaining bandwidth employed for popular channels (70% for upward channels and 30% for downward ones). Before being set at 56%, the fraction of the navigation bandwidth used for adjacent channels was varied from 40% to 60%, having produced similar results for the same traffic demand. Indeed, this fraction value may require further fine-tuning in operational IPTV systems.

The following metrics were obtained in the simulations:

- *Delayed/immediate channel switching events*, i.e., percentage of channel switching events with/without latency;
- *Stream quality*, i.e., mean number of descriptions received for the watched channel by the peers during their sessions.

The metrics are disjunct and complementary to each other, i.e., while the number of navigation descriptions received by a peer affects the delayed/immediate channel switching events, the number of descriptions received for the watched channel impacts the stream quality. Since the user's bandwidth is limited, the two metrics are important to balance the numbers for each set of descriptions in the two viewing states. Moreover, the stream quality is closely related to user-perceived quality, since the final quality of the reassembled stream is proportional to the number of descriptions received for the watched channel (one of the properties of multiple description coding) [5].

VI. EVALUATION OF THE SCHEMES

In this section, the proposed channel switching schemes are evaluated. This evaluation is conducted in two steps. In the first, we evaluate the benefits of employing a channel switching scheme in a P2P IPTV system. For that, we compare a P2P IPTV system with the *Linear* scheme to one system without the use of any scheme to support channel switching (subsection VI-A). The *Linear* scheme was chosen for comparison not only because it features a gradual state transition, but also because this transition is linear. We also investigate here the effectiveness of the three strategies proposed for the selection of navigation descriptions: old channel, popular channels and adjacent channels. After showing the benefits of adopting a scheme for the support of channel switching in P2P IPTV, we compare the performance of the *Linear* scheme to that of the other proposed schemes, *Fast*, *Slow* and *Immediate* (subsection VI-B). This comparison was designed to identify which of the proposed schemes would lead to the best system performance.

In both steps, the stream bandwidth of existing channels and the number of distribution trees employed were varied to investigate the performance of the schemes in different situations of bandwidth contention. The scenario with the greatest bandwidth contention (worst case scenario) is that with the largest stream bandwidth and the lowest number of trees, since the bandwidth of each description is proportional to the total stream bandwidth and inversely proportional to the number of trees. Conversely, the scenario with the least bandwidth contention (best case scenario) is that with the smallest stream bandwidth and the highest number of trees. The scenario with a stream bandwidth of 1024 Kbps and 16 distribution trees is used here as an average case scenario, since it has intermediate values for both variables.

A. Linear Scheme

In this subsection, the *Linear* scheme is compared to a system with no scheme. Table II shows the configurations for the different scenarios considered.

1) Delayed and Immediate Channel Switching Events: Fig. 6 shows the percentages of channel switching events with latency, for each scenario considered in this subsection. When no scheme is employed, no reduction in the number of delayed events is observed. The use of the *Linear* scheme led to varying reductions in latency, which are greatest in those scenarios with the least bandwidth contention. This is due to the fact that the number of navigation channels that each user can receive is determined by the bandwidth of each description, and the more navigation descriptions received, the higher is the probability of a channel requested being available for playback.

In the best case (scenario S300T32, with a stream bandwidth of 300 Kbps and 32 distribution trees), the use of the *Linear* scheme led to a reduction of 89.94% in the number

TABLE II
SCENARIOS CONSIDERED, VARYING STREAM BANDWIDTH (STR.), NUMBER OF TREES (TRS.) AND EMPLOYMENT OF THE LINEAR CHANNEL SWITCHING SCHEME (SCH).

Scenario	Str.	Trs.	Sch.	Scenario	Str.	Trs.	Sch.
S300T4	300 Kbps	4	Off	S1024T16	1024 Kbps	16	Off
			On				On
S300T8		8	Off	S1024T32		32	Off
			On				On
S300T16		16	Off	S2048T4	2048 Kbps	4	Off
			On				On
S300T32		32	Off	S2048T8		8	Off
			On				On
S1024T4	1024 Kbps	4	Off	S2048T16		16	Off
			On				On
S1024T8		8	Off	S2048T32		32	Off
			On				On

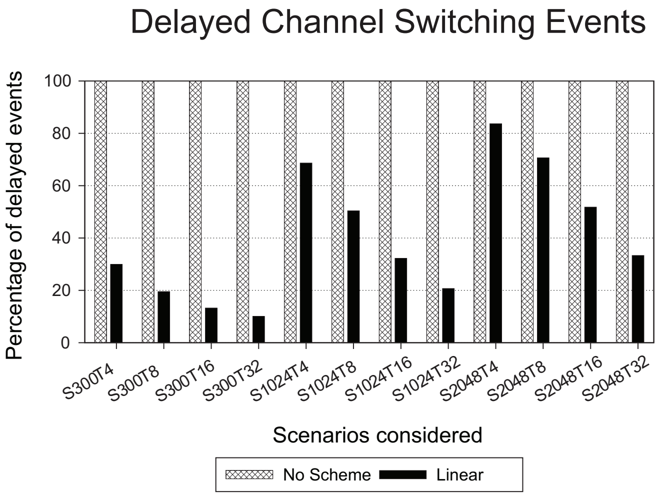

Fig. 6. Percentages of channel switching events with latency, for each scenario considered.

of delayed events. In the worst case (scenario S2048T4, with a stream bandwidth of 2048 Kbps and 4 distribution trees), the reduction was only 16.31%. In the average case (scenario S1024T16, with a stream bandwidth of 1024 Kbps and 16 distribution trees), a reduction of 67.74% was obtained.

Fig. 7 describes the percentages of channel switching events without latency, separated by the three strategies for the selection of navigation descriptions: old channel, popular channels and adjacent channels. Only the scenarios that employ the *Linear* scheme are shown, since no immediate events exist in the scenarios employing no scheme. The percentages of immediate events per strategy are relative to the total number of events for each scenario.

Fig. 7 shows that in the scenarios with greatest bandwidth contention (S1024T4, S2048T4, S2048T8 and S2048T16) the adjacent channels strategy produces higher percentages of immediate events than does the popular channels strategy. This is due to the fact that the adjacent channels strategy requires less bandwidth than does the popular channels strategy, because it has a good likelihood of having hits simply by ensuring that the channels immediately above and below the current one are available for the linear switchings, whereas the popular channels strategy must make a higher number of channels available for the non-linear switchings. In general, the greater the bandwidth contention, the lower is the hit probability of the popular channels strategy, whereas with the adjacent channels strategy this number increases.

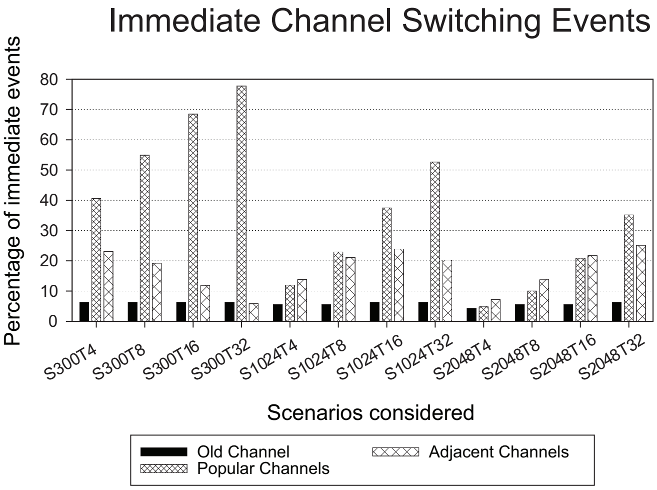

Fig. 7. Percentages of channel switching events without latency, separated by strategy of the proposed schemes, for each scenario considered.

An analysis of the old channel strategy shows that the percentages of immediate events are fairly constant. This is due to the fact that the descriptions made available by this strategy are descriptions of the *same* channel. The hit probability of this strategy is related to the chances of the last channel watched being reselected in a switching operation. Thus, the number of descriptions available affects only the quality of reception when the old channel is resumed, with no effect on the chances of immediate switching, since only a single description is sufficient for that. As a consequence, the variation in bandwidth contention imposed by the scenarios considered has no effect on the hit probability of this strategy.

2) Stream Quality: Fig. 8 describes the mean numbers of descriptions received for the watched channel by the peers during their sessions, for each scenario considered in this subsection. In the case of this metric, scenarios with different numbers of trees produce different maximum numbers of descriptions. When no scheme is employed, the number of descriptions received is close to the maximum value in the scenarios with a small stream bandwidth (300 Kbps). In the scenarios with a large stream bandwidth (1024 and 2048 Kbps), however, there is a slight decrease in that number, since the increase in the bandwidth contention means that there are more users who are unable to receive all the descriptions of the channels. This trend, however, is independent of the adoption of a scheme.

The employment of the *Linear* scheme leads to a further reduction in the number of descriptions received for the watched channel, because part of the user's bandwidth is used for the reception of navigation descriptions. This metric thus reveals the cost of using the scheme, although this cost varies according to the bandwidth contention. In general, the greater the contention, the greater is the reduction in the stream quality.

This reduction does not vary, however, as a function of the number of trees used, being almost constant for all the

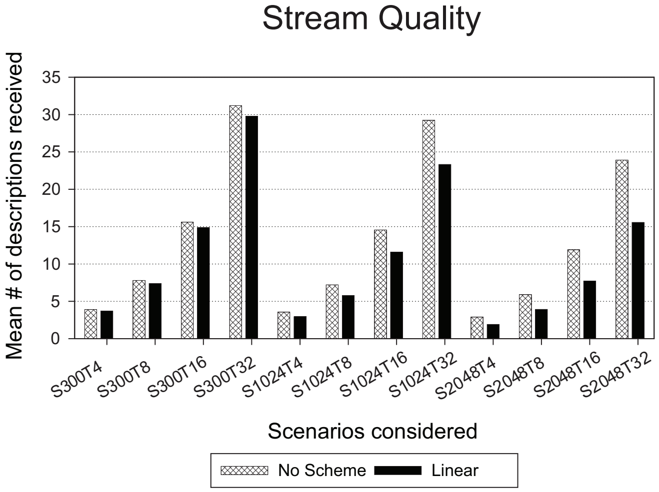

Fig. 8. Mean numbers of descriptions received for the watched channel by the peers during their sessions, for each scenario considered.

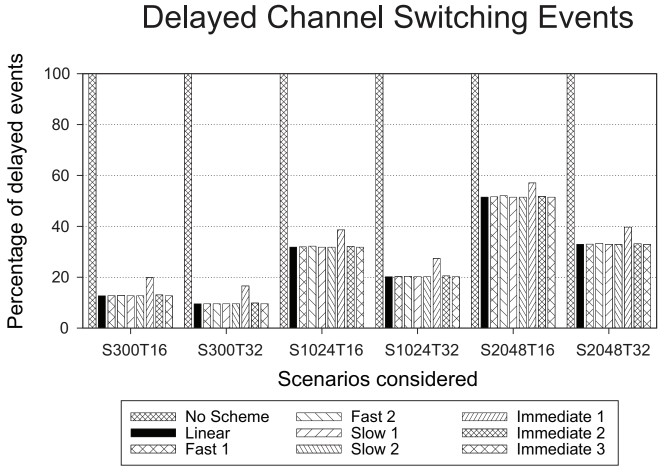

Fig. 9. Percentages of channel switching events with latency, for each scenario and scheme considered.

TABLE III

SCENARIOS CONSIDERED, VARYING STREAM BANDWIDTH (STR.) AND NUMBER OF TREES (TRS.); AND SCHEMES EVALUATED, VARYING LOGARITHMIC FACTOR (C) AND TIME OUT FOR STATE TRANSITION (t).

(a) Scenarios considered.

Scenario	Str.	Trs.
S300T16	300 Kbps	16
S300T32		32
S1024T16	1024 Kbps	16
S1024T32		32
S2048T16	2048 Kbps	16
S2048T32		32

(b) Schemes evaluated.

Scheme	Variation
No Scheme	N/A
Linear	N/A
Fast 1	$C = 0.5$
Fast 2	$C = 2000$
Slow 1	$C = 0.5$
Slow 2	$C = 2000$
Immediate 1	$t = 5s$
Immediate 2	$t = 30s$
Immediate 3	$t = 60s$

scenarios with the same stream size. In the best case (scenarios with a stream bandwidth of 300 Kbps), the cost of using the scheme provides an average reduction of 4.88%, whereas in the worst case (scenarios with a stream bandwidth of 2048 Kbps), the cost undergoes an average reduction of 34.33%. In the average case (scenarios with a stream bandwidth of 1024 Kbps), the cost reduces 19.48%.

B. Fast, Slow and Immediate Schemes

In this subsection, the *Fast*, *Slow* and *Immediate* schemes are evaluated. Table III illustrates the configurations for the different scenarios considered, as well as the variations of the schemes evaluated. Due to the number of different schemes evaluated, we show only the results for scenarios with 16 and 32 distribution trees; as a consequence, the scenario S2048T16 (with a stream bandwidth of 2048 Kbps and 16 distribution trees) is now the one with the greatest bandwidth contention (worst case scenario).

1) Delayed Channel Switching Events: Fig. 9 shows the percentages of channel switching events with latency, for each scenario and scheme considered in this subsection. The results are in agreement with those in Fig. 6: when no channel switching scheme is employed, no reduction in the number of delayed events is achieved; whenever a scheme is employed, the reduction obtained is inversely proportional to the bandwidth contention. In comparison to the *Linear*, all the schemes except the *Immediate 1* led to a similar reduction in the number of delayed events.

The reduction from the use of the *Immediate 1* scheme is 7% smaller than that of the other schemes. This is due to the fact that most of the navigation descriptions are dropped simultaneously 5 seconds after the last channel change, without taking full advantage of the state *browsing*. Although most channel changes do occur in the first 5 seconds after the last change, some take place later, which is why both the *Immediate 2* and *Immediate 3* schemes provide a better performance than does the *Immediate 1*. Even without the gradual state transition, the *Immediate 2* and *Immediate 3* schemes wait 30 and 60 seconds after the last channel change, respectively, to drop the navigation descriptions.

The effectiveness of these schemes for the three strategies for the selection of navigation descriptions (old channel, popular channels and adjacent channels) was the same as that of the *Linear* scheme (Fig. 7).

The *Slow 1*, *Slow 2*, *Immediate 3*, and (to a certain extent) the *Immediate 2* schemes were unable to reduce the number of delayed events any more than the *Linear* scheme (reduction of approximately 0.04%). Since postponing the release of the navigation descriptions has a negative impact on the stream quality, these schemes are less effective than the *Linear* one.

2) Stream Quality: Figs. 10, 11 and 12 present the mean numbers of descriptions received for the watched channel by the peers during their sessions, for each scenario and scheme considered in this subsection. Figs. 10 and 11 show this metric for the states *browsing* and *watching*, respectively, while Fig. 12 shows the overall results for the two states. The results in Fig. 12 are in agreement with those in Fig. 8: fewer descriptions are received when a channel switching scheme is employed than when no such scheme is employed due to the cost of using the schemes. Moreover, the greater the bandwidth contention, the greater is the reduction in the stream quality.

The three versions of the *Immediate* scheme produced the worst stream qualities, as can be seen in Figs. 10 and 11. For the scenario S1024T16 in the state *browsing*, for example,

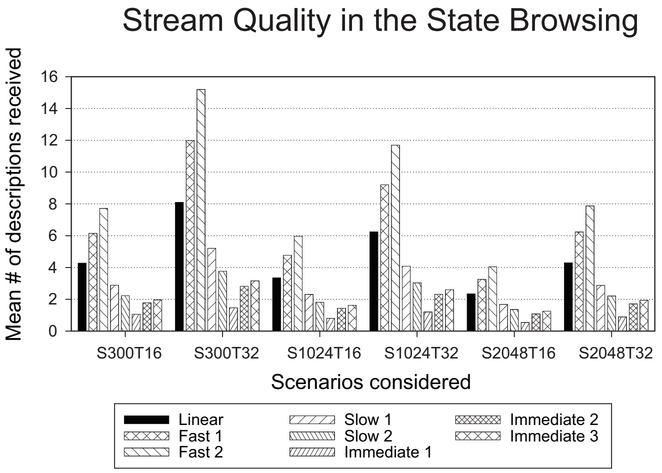

Fig. 10. Mean numbers of descriptions received for the watched channel by the peers during their sessions in the state *browsing*, for each scenario and scheme considered.

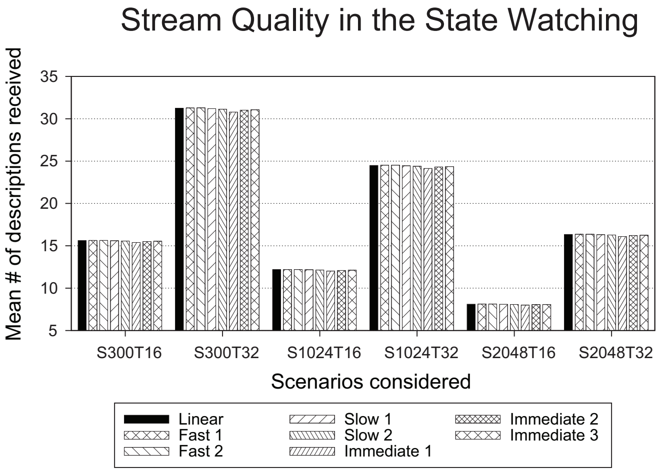

Fig. 11. Mean numbers of descriptions received for the watched channel by the peers during their sessions in the state *watching*, for each scenario and scheme considered.

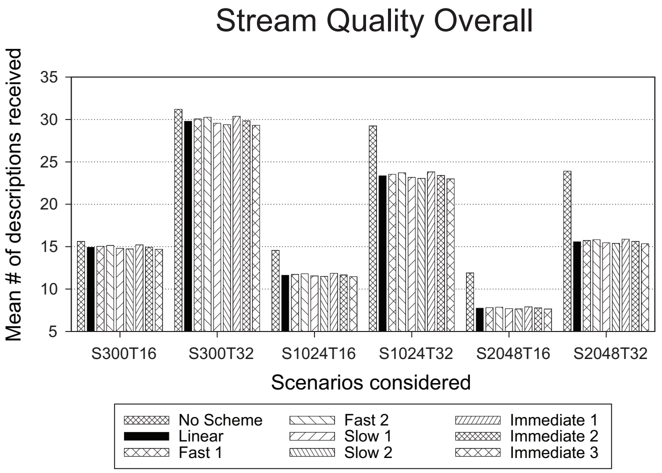

Fig. 12. Mean numbers of descriptions received for the watched channel by the peers during their sessions in the two states, for each scenario and scheme considered.

the stream qualities of the *Immediate 1*, *Immediate 2* and *Immediate 3* schemes were 76.05%, 56.89% and 51.5% lower than that of the *Linear* scheme, respectively. The only reason the *Immediate 2* and *Immediate 3* schemes produced slightly better stream qualities than the *Immediate 1* scheme was the postponement of the transition to the state *watching*, which enabled the old channel strategy to be used more often while the state was still in *browsing*. Since these three schemes produced the worst stream qualities and did not reduce the number of delayed channel switching events any more than the *Linear* scheme, they provide no special benefits. Moreover, these results suggest that the gradual state transition employed by the other schemes does help in reducing the occurrence of changes with latency, as well as improving the stream quality received.

The second worst group of schemes in relation to stream quality consists of the *Slow 1* and *Slow 2* schemes. To illustrate this, let us consider the scenario S1024T16 in the state *browsing*. In this case, they produced stream qualities 31.14% and 45.81% lower than that produced by the *Linear* scheme, respectively. Since these two schemes yielded such low stream qualities and were unable to reduce the number of delayed channel switching events any more than the *Linear* scheme, they have also been shown to provide no special benefits.

In the state *watching*, the *Fast 1* and *Fast 2* schemes produced stream qualities equivalent to those produced by the *Linear* scheme. However, in the state *browsing*, the stream qualities achieved were considerably better. In the scenario S1024T16, for example, the *Fast 1* and *Fast 2* schemes produced stream qualities 42.81% and 78.74% higher than that of the *Linear* scheme, respectively. The *Fast 1* and *Fast 2* schemes were thus both successful in improving the stream quality in the state *browsing* in comparison to the *Linear* scheme, while maintaining basically the same reduction in the number of delayed channel switching events. The *Fast 2* scheme, however, surpassed the *Fast 1* scheme in all scenarios, proving to be the best scheme evaluated here. The improvement of 78.74% over the *Linear* scheme was in the average case (scenario S1024T16, with a stream bandwidth of 1024 Kbps and 16 distribution trees). In the best case (scenario S300T32, with a stream bandwidth of 300 Kbps and 32 distribution trees), the *Fast 2* scheme led to an improvement of 87.65%, and even in the worst case (scenario S2048T16, with a stream bandwidth of 2048 Kbps and 16 distribution trees), the improvement was of 72.65%. Thus, the *Fast 2* scheme was just as competent as the *Linear* scheme in reducing the number of delayed channel switching events (68%, on average), doing so with an improvement of 78.74% (on average) in the stream quality during the channel switching operations (in the state *browsing*).

Since the probability of a peer being in the state *browsing* is considerably lower than being in the state *watching*, the great variations observed in Fig. 10 are smothered by the rare variations seen in Fig. 11, resulting in the overall metric given by Fig. 12. Although in this subsection the stream quality was analyzed separately in relation to viewing state, in actual use the overall metric is what matters when evaluating the cost of employing the channel switching schemes. Thus, the *Fast 2* scheme results in an average reduction of 3% in the best case (scenarios with a stream bandwidth of 300 Kbps), whereas in the worst case (scenarios with a stream bandwidth of 2048 Kbps), the average reduction is 33.9%. In the average case (scenarios with a stream bandwidth of 1024 Kbps), the average reduction is 18.85%. Given that these costs are lower than those for the *Linear* scheme, presented in subsection VI-A, the *Fast 2* has proved to be the most efficient scheme to be adopted for channel switching in P2P IPTV systems.

VII. CONCLUSION

In this paper, four novel schemes for fast channel switching in P2P IPTV systems were presented and compared, all designed to reduce the occurrence of latencies caused by buffers and overlay structures, while maintaining the stream quality as much as possible. Multiple description coding and multiple distribution trees were employed to generate substreams with minimum quality for each channel, providing either a greater channel diversity or a higher quality of reception. Moreover, three different strategies were defined for the selection of navigation descriptions, namely old channel, popular channels and adjacent channels. The popular channels strategy addresses the non-linear changes, while the adjacent channels strategy deals with the linear ones. The old channel strategy, on the other hand, not only addresses the changes back to the last watched channel, but also improves the quality of service by ensuring that the old channel has an intermediate quality when resumed. When the bandwidth contention is great, the adjacent channels strategy is more efficient.

The benefits of the adoption of a channel switching scheme have been highlighted by comparing the use of the *Linear* scheme to a system which does not employ any scheme. The performance of the other proposed schemes (*Fast*, *Slow* and *Immediate*) was then compared to that of the *Linear* scheme. The results suggest that the *Fast 2* is the most efficient scheme to be adopted for channel switching in P2P IPTV systems.

As future work, a performance evaluation study of all of the proposed schemes in a commercial IPTV setting, with stream bandwidths varying from 4 Mbps (SDTV) to around 20 Mbps (HDTV), will be conducted. Moreover, studies on the Visual Quality Experience and Perceived Latency will be conducted to evaluate user experience in various different situations of bandwidth contention, complementing the metrics analyzed in the present work.

REFERENCES

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

[2] D. A. G. Manzato and N. L. S. da Fonseca, "Peer-to-peer IPTV services," in *Proc. 2nd IEEE Workshop Enabling Future Service-Oriented Internet*, Nov. 2008.

[3] "Pplive," Mar. 2011 [Online]. Available: http://www.pplive.com/en/index.html

[4] J. Liu, S. G. Rao, B. Li, and H. Zhang, "Opportunities and challenges of peer-to-peer Internet video broadcast," *Proc. IEEE*, vol. 96, no. 1, pp. 11–24, 2008.

[5] V. K. Goyal, "Multiple description coding: Compression meets the network," *IEEE Signal Process. Mag.*, vol. 18, no. 5, pp. 74–93, Sep. 2001.

[6] P. A. Chou, H. J. Wang, and V. N. Padmanabhan, "Layered multiple description coding," in *Proc. Packet Video Workshop*, Apr. 2003.

[7] J. G. Apostolopoulos, T. Wong, S. J. Wee, and D. Tan, "On multiple description streaming with content delivery networks," in *Proc. IEEE INFOCOM*, June 2002, vol. 3, pp. 1736–1745.

[8] D. A. G. Manzato and N. L. S. da Fonseca, "A channel switching scheme for IPTV systems," in *Proc. 53rd IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2010, pp. 1–6.

[9] ——, "A comparison of channel switching schemes for IPTV systems," in *Proc. IEEE Int. Conf. Commun. (ICC)*, June 2011, pp. 1–6.

[10] C. Y. Lee, C. K. Hong, and K. Y. Lee, "Reducing channel zapping time in IPTV based on user's channel selection behaviors," *IEEE Trans. Broadcast.*, vol. 56, no. 3, pp. 321–330, Sep. 2010.

[11] Y. Lee, J. Lee, I. Kim, and H. Shin, "Reducing IPTV channel switching time using H.264 scalable video coding," *IEEE Trans. Consum. Electron.*, vol. 54, no. 2, pp. 912–919, May 2008.

[12] D. Banodkar, K. K. Ramakrishnan, S. Kalyanaraman, A. Gerber, and O. Spatscheck, "Multicast instant channel change in IPTV systems," in *Proc. IEEE 3rd Int. Conf. COMmun. Syst. softWAre MiddlewaRE (COMSWARE)*, Jan. 2008, pp. 370–379.

[13] F. M. V. Ramos, J. Crowcroft, R. J. Gibbens, P. Rodriguez, and I. H. White, "Reducing channel change delay in IPTV by predictive pre-joining of TV channels," in *Signal Processing: Image Communication*, 2011 [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0923596511000257

[14] M. Sarni, B. Hilt, and P. Lorenz, "A novel channel switching scenario in multicast IPTV networks," in *Proc. IEEE 5th Int. Conf. Netw. Services (ICNS)*, 2009, pp. 396–401.

[15] J. Kim, H. Yun, M. Kang, T. Kim, and J. Yoo, "Performance evaluation of channel zapping protocol in broadcasting services over hybrid WDM-PON," in *Proc. 7th Int. Conf. Advanced Commun. Technol. (ICACT)*, Feb. 2005, vol. 2, pp. 1152–1155.

[16] A.-M. Kermarrec, E. L. Merrer, Y. Liu, and G. Simon, "Surfing peer-to-peer IPTV: Distributed channel switching," in *Proc. 15th Int. Euro-Par Conf. Parallel Process. (Euro-Par)*, Aug. 2009, pp. 574–586.

[17] X. Zhang, J. Liu, B. Li, and T. S. P. Yum, "Donet/coolstreaming: A data-driven overlay network for live media streaming," in *Proc. IEEE INFOCOM*, Mar. 2005, vol. 3, pp. 2102–2111.

[18] J. M. Boyce and A. M. Tourapis, "Fast efficient channel change," in *Proc. Int. Conf. Consum. Electron. (ICCE)*, Jan. 2005, pp. 1–2.

[19] U. Jennehag, T. Zhang, and S. Pettersson, "Improving transmission efficiency in H.264 based IPTV systems," *IEEE Trans. Broadcast.*, vol. 53, no. 1, pp. 69–78, Mar. 2007.

[20] U. Jennehag and S. Pettersson, "On synchronization frames for channel switching in a GOP-based IPTV environment," in *Proc. 5th IEEE Consum. Commun. Netw. Conf. (CCNC)*, Jan. 2008, pp. 638–642.

[21] D. E. Smith, "IP TV bandwidth demand: Multicast and channel surfing," in *Proc. 26th IEEE Int. Conf. Comput. Commun. (INFOCOM)*, May 2007, pp. 2546–2550.

[22] N. Degrande, K. Laevens, D. D. Vleeschauwer, and R. Sharpe, "Increasing the user perceived quality for IPTV services," *IEEE Commun. Mag.*, vol. 46, no. 2, pp. 94–100, Feb. 2008.

[23] S. K. Mandal and M. MBuru, "Intelligent pre-fetching to reduce channel switching delay in IPTV systems," Department of Computer Science, Texas A & M University, Tech. Rep., 2008 [Online]. Available: http://students.cs.tamu.edu/skmandal/research/channelswitching.pdf

[24] Y. Bejerano and P. V. Koppol, "Improving zap response time for IPTV," in *Proc. 28th IEEE Int. Conf. Comput. Commun. (INFOCOM)*, Apr. 2009, pp. 1971–1979.

[25] C. Cho, I. Han, Y. Jun, and H. Lee, "Improvement of channel zapping time in IPTV services using the adjacent groups join-leave method," in *Proc. 6th Int. Conf. Advanced Commun. Technol.*, 2004, vol. 2, pp. 971–975.

[26] J. Lee, G. Lee, S. Seok, and B. Chung, "Advanced scheme to reduce IPTV channel zapping time," in *Proc. 10th Asia-Pacific Netw. Operations Manage. Symp. Managing Next Generation Netw. Services (APNOMS)*, 2007, pp. 235–243.

[27] Y. Kim, J. K. Park, H. J. Choi, S. Lee, H. Park, J. Kim, Z. Lee, and K. Ko, "Reducing IPTV channel zapping time based on viewer's surfing behavior and preference," in *Proc. IEEE Int. Symp. Broadband Multimedia Syst. Broadcast. (BMSB)*, Mar. 2008, pp. 1–6.

[28] F. M. V. Ramos, J. Crowcroft, R. J. Gibbens, P. Rodriguez, and I. H. White, "Channel smurfing: Minimising channel switching delay in IPTV distribution networks," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, July 2010, pp. 1327–1332.

[29] U. Oh, S. Lim, and H. Bahn, "Channel reordering and prefetching schemes for efficient IPTV channel navigation," *IEEE Trans. Consum. Electron.*, vol. 56, no. 2, pp. 483–487, May 2010.

[30] V. N. Padmanabhan, H. J. Wang, and P. A. Chou, "Resilient peer-to-peer streaming," in *Proc. 11th IEEE Int. Conf. Netw. Protocols (ICNP)*, Nov. 2003, pp. 16–27.

[31] M. Castro, P. Druschel, A.-M. Kermarrec, A. Nandi, A. Rowstron, and A. Singh, "Splitstream: High-bandwidth multicast in a cooperative environment," in *Proc. 19th ACM Symp. Operating Syst. Principles (SOSP)*, Oct. 2003, pp. 298–313.

[32] V. Venkataraman, P. Francis, and J. Calandrino, "Chunkyspread: Multitree unstructured peer to peer multicast," in *Proc. 5th Int. Workshop Peer-to-Peer Syst. (IPTPS)*, Feb. 2006.

[33] D. A. G. Manzato and N. L. S. da Fonseca, "An incentive mechanism for peer-to-peer networks with media streaming," in *Proc. 49th IEEE Global Telecommun. Conf. (GLOBECOM)*, Nov. 2006, pp. 1–6.

[34] C. Greco, M. Cagnazzo, and B. Pesquet-Popescu, "H.264-based multiple description coding using motion compensated temporal interpolation," in *Proc. IEEE Int. Workshop Multimedia Signal Process. (MMSP)*, Oct. 2010, pp. 239–244.

[35] M. Cha, P. Rodriguez, J. Crowcroft, S. Moon, and X. Amatriain, "Watching television over an IP network," in *Proc. 8th ACM SIGCOMM Conf. Internet Meas. (IMC)*, Oct. 2008, pp. 71–84.

[36] T. Qiu, Z. Ge, S. Lee, J. Wang, J. Xu, and Q. Zhao, "Modeling user activities in a large IPTV system," in *Proc. 9th ACM SIGCOMM Conf. Internet Meas. (IMC)*, Nov. 2009, pp. 430–441.

[37] E. Veloso, V. Almeida, W. Meira, A. Bestavros, and S. Jin, "A hierarchical characterization of a live streaming media workload," in *Proc. 2nd ACM SIGCOMM Workshop Internet Meas. (IMW)*, 2002, pp. 117–130.

[38] K. Sripanidkulchai, B. Maggs, and H. Zhang, "An analysis of live streaming workloads on the Internet," in *Proc. 4th ACM SIGCOMM Conf. Internet Meas. (IMC)*, 2004, pp. 41–54.

[39] "Teleco," Mar. 2011 [Online]. Available: http://www.teleco.com.br/

[40] V. N. Padmanabhan, H. J. Wang, and P. A. Chou, "Supporting heterogeneity and congestion control in peer-to-peer multicast streaming," in *Proc. 3rd Int. Workshop Peer-to-Peer Syst. (IPTPS)*, Feb. 2004, pp. 54–63.

Daniel Antonio Garcia Manzato is a Ph.D. candidate in computer science at the State University of Campinas, Brazil. He received his B.S. degree in computer science from the University of Sao Paulo in 1998, and his M.S. degree in computer science from the State University of Campinas in 2006. He has worked for more than 11 years as a software engineer and project manager at companies such as UOL, a leading online media portal and Internet access provider in Brazil. His main research interests are peer-to-peer networks, IPTV, video on demand services, and incentive mechanisms for cooperation.

Nelson Luis Saldanha da Fonseca received his Electrical Engineer (1984) and M.Sc. in computer science (1987) degrees from The Pontificial Catholic University at Rio de Janeiro, Brazil, and the M.Sc. (1993) and Ph.D. (1994) degrees in computer engineering from The University of Southern California. He is a Full Professor at the Institute of Computing of The University of Campinas, Campinas, Brazil. He has published 300+ papers and supervised 50+ graduate students. As a Visiting Professor, he lectured at the University of Trento, the University of Pisa, and the University of Basque Country. He has also held Lecturer positions at The Pontificial Catholic University at Rio de Janeiro and worked in the Computer Communications group at the IBM Rio Scientific Center. He is the recipient of the ComSoc Latin America Service award (2011). He received the Medal of the Chancelor of the University of Pisa in 2007. He is the recipient of the 2003 State University of Campinas Zeferino Vaz award for academic productivity, the Elsevier Computer Network Journal Editor of Year 2001 award, the 1994 University of Southern California International Book award, and the Brazilian Computer Society First Thesis and Dissertations award. His graduate students have received over 10 awards in Latin America Thesis contests.

He is past EiC of IEEE COMMUNICATIONS SURVEYS AND TUTORIALS, past EiC of the ComSoc Electronic Newsletter, and past Editor of the Global Communications Newsletter. He is Senior Editor for IEEE COMMUNICATIONS SURVEYS AND TUTORIALS and *IEEE Communications Magazine*. He is a member of the editorial board of *Computer Networks*, *Peer-to-Peer Networking and Applications*, the *Journal of Internet Services and Applications*, and the *International Journal of Communication Systems*. He served on the editorial board of the IEEE TRANSACTIONS ON MULTIMEDIA. He founded the IEEE Latin America Conference on Communications (LATINCOM) and the Latin America Conference on Cloud Computing and Communications (LATINCLOUD). He has been technical chair of over 10 ComSoc conferences.

Nelson is an active IEEE ComSoc Volunteer. Currently, He is ComSoc Vice Chair for Member Relations. He has served as: Member-at-Large on the ComSoc Board of Governors, Director of the Latin America Region, Director of on-line Services, Chair of the Multimedia Communications technical committee, and Chair of the Communications Systems Integration and Modeling technical committee.