A Survey of Channel Switching Schemes for IPTV

Daniel A. G. Manzato and Nelson L. S. da Fonseca, State University of Campinas

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 switching, a problem that is quite relevant in IPTV systems due to bandwidth limitations, as well as the employment of buffers and distribution structures. The problem is essentially more serious in P2P IPTV systems, since the dynamic nature of P2P networks requires larger buffers and the exchange of more signaling messages to establish the overlay structures. Since users expect to be able to switch channels quickly, IPTV services must reduce startup delays to just a few seconds or even milliseconds. The aim of this article is to provide a survey of existing channel switching schemes for IPTV systems.

INTRODUCTION

IPTV services have emerged as an alternative delivery system for traditional cable and broadcast systems. In commercial IPTV, the infrastructure is typically owned by telecommunication providers, and the size of the network limits the population served. In peer-to-peer (P2P) IPTV, the Internet provides the infrastructure, which enables a global population to have access to services [1].

In spite of the increasing deployment of IPTV services, various functionalities still need to be improved [1]. One of the main issues is the length of startup delays, especially during channel switching. The problem does not exist in traditional analog television, because channel switching involves changing the frequency to access the content of another channel that is already available. The problem arose with traditional digital television, since the digital encoding of video streams imposes certain minor yet noticeable synchronization delays prior to the decoding of the video frames. In IPTV systems, however, the problem is aggravated. Due to bandwidth limitations, only a portion of the content is actually transmitted to users. As a result, when channels are switched, the user needs to leave one distribution structure and join another so that she/he can start receiving the content of the newly requested channel. Moreover, although the problems arising from network bandwidth fluctuations and connection failure can be ameliorated by buffers, their use leads to playback latency and degrades the usability of the system, especially for fast navigation through multiple channels [2].

In P2P IPTV systems, startup delays are specifically problematic due to the dynamic nature of these networks. Since peers can suddenly become unreachable or overloaded, or simply depart without prior notification, larger buffers are required to cope with peer churn. Moreover, joining and leaving distribution structures are expensive operations, since they usually require the exchange of various signaling messages to establish the new overlay structure, given that the use of multicast at the application layer means that neighbor peers may have to be encountered. New peers may have to wait up to 10–15 s before they can join a P2P overlay, and it can take another 10–15 s to launch the media player and store the video frames in the buffers [1]. These startup delays are less of a problem in commercial IPTV systems.

Since the users of traditional television expect to be able to switch channels quickly [1], IPTV services must reduce startup delays to just a few seconds or even milliseconds. This article provides a brief survey of some channel switching schemes available. It differs from previous papers [3, 4] by including schemes and discussing aspects that are specific to P2P IPTV. To the best of the authors' knowledge, this is the first survey of channel switching schemes to consider P2P systems. First, IPTV architectures are explained, and the various components of switching delay are discussed; finally, a selection of channel switching schemes is presented.

IPTV ARCHITECTURES

The two types of IPTV systems are discussed here. In P2P networks, there are two main approaches to video distribution: tree and mesh [1]. The tree approach, based on multicasting on the application layer, has a distribution structure always ready for transmission, thus avoiding the overhead of transmission scheduling and, as a consequence, reducing startup delay. The mesh approach employs connections on demand, thus

 0163-6804/13/$25.00

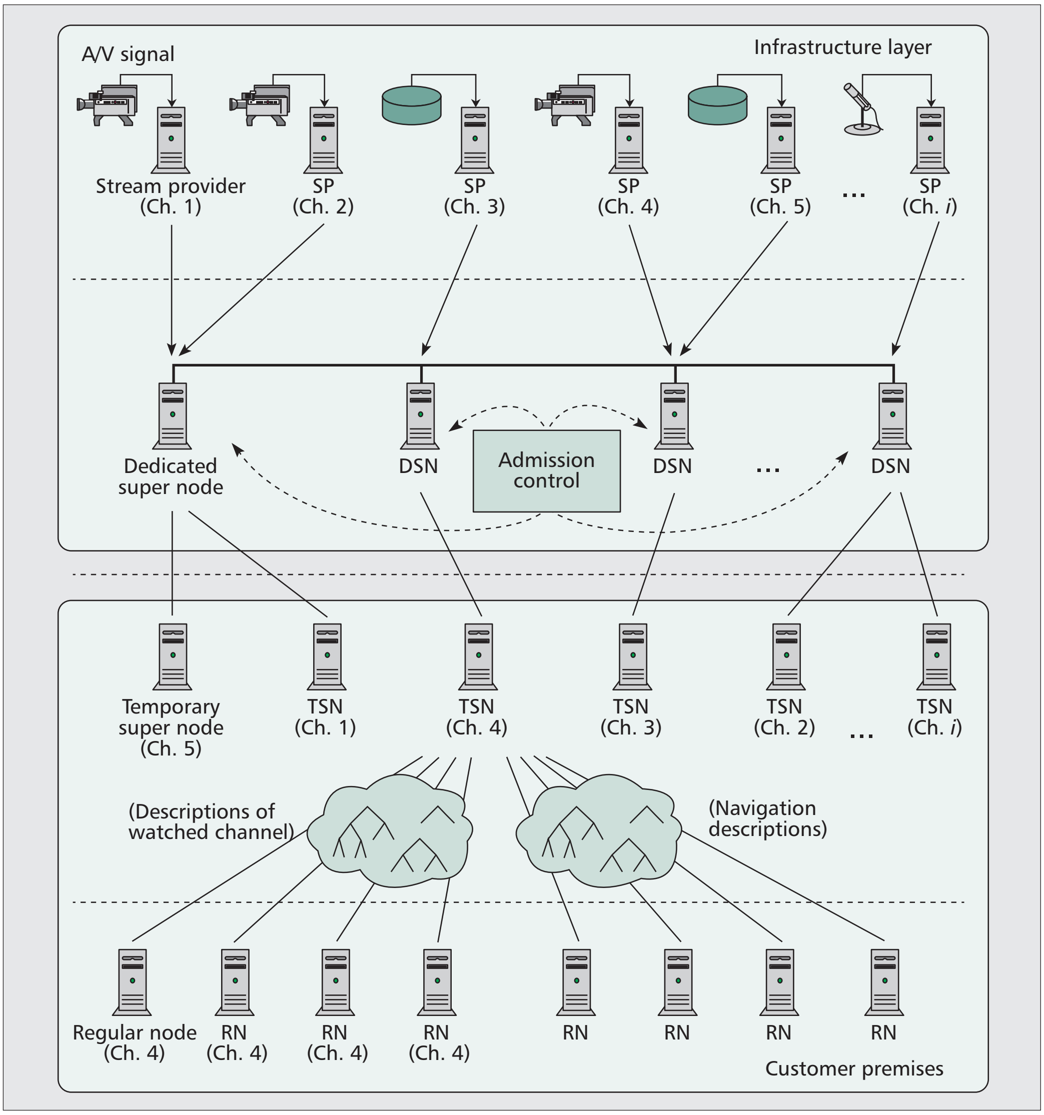

Figure 1. *Example of P2P IPTV architecture with multiple distribution trees.*

In P2P networks, there are two main approaches to video distribution: tree and mesh. The tree approach, based on multicasting on the application layer, has a distribution structure always ready for transmission. The mesh approach employs connections on demand.

avoiding the cost of maintaining an active distribution structure, although it accrues overhead for content dissemination as well as for transmission scheduling. Since one of the main challenges of IPTV systems is the reduction of startup delays, especially for channel switching, the tree approach is generally preferred [1].

An example of P2P IPTV architecture employing multiple distribution trees is presented in Fig. 1. In this example, stream providers (SPs) produce TV content from different channels and generate multiple substreams (or descriptions) for each channel by using multiple description coding (MDC). SPs are connected to other components, called dedicated super nodes (DSNs), which constitute the infrastructure layer of the architecture. DSNs are responsible for propagating and making available all the descriptions of all channels, as well as coordinating the admission of new peers into the system. Moreover, they compensate for the deficit in bandwidth introduced by peers with limited capacity.

Peers are classified into two categories, according to their capacity: temporary super nodes (TSNs) and regular nodes (RNs). TSNs have greater capacity, and cooperate in the tasks of tree management and content distribution; they are admitted by DSNs. The lower-capacity RNs, on the other hand, are admitted by TSNs via multiple distribution trees, with each tree used for the transmission of a specific description. As each TSN provides only a single channel in full quality, the channel selection of an RN will determine its admission under a specific TSN. In addition to serving a single channel in full quality, a TSN also forwards navigation descriptions of other channels, enabling fast navigation to RNs.

A typical commercial IPTV architecture is illustrated in Fig. 2. Most of these architectures are based on a well provisioned private network infrastructure with IP multicast. Streams of different channels are forwarded by TV head-ends for delivery to regional components, called digital subscriber line access multiplexers (DSLAMs), via IP multicast trees. Each DSLAM receives streams of all existing channels and aggregates the traffic from hundreds or thousands of users. However, users' bandwidth limits the transmission to a single channel per set-top box (STB). Each available channel in the system has one distinct multicast tree per DSLAM. In the user premises, the connection is made by a home gateway, which divides the traffic for TV, telephone, and Internet access. Each enabled TV is connected to the home gateway by a separate STB.

Components of Total Switching Delay

Whenever an IPTV user decides to check what is being broadcast on other channels, she/he initiates a channel switching operation. Such an operation consists of a series of requests to view other channels, with the user watching each channel chosen for a limited period of time. The operation terminates when the user settles into a channel for an extended period of time.

Whenever a user requests a channel switching, the system configuration needs to be changed by the STB or computer software to permit the delivery of the new TV content, which consists in changing the membership of the affiliated multicast group so that the user can start receiving packets from the stream of the new group and stop receiving packets from the stream of the current group. A Join message is first sent to the new multicast group, along with a Leave message to the current group. Depending on the system, multiple messages may be required for the establishment of a distribution structure. In commercial IPTV systems, a common protocol used is the Internet Group Management Protocol (IGMP), and a single pair of Join/Leave messages usually suffices. In P2P IPTV systems that employ multiple trees [1], on the other hand, the user is admitted into a couple of trees for each channel, demanding admittance to and departure from several groups each time a channel is switched. This scenario can be even worse in P2P IPTV systems that employ mesh [5], since peers may have to find new neighbors before sending the Join/Leave messages because active distribution structures are not available for immediate use.

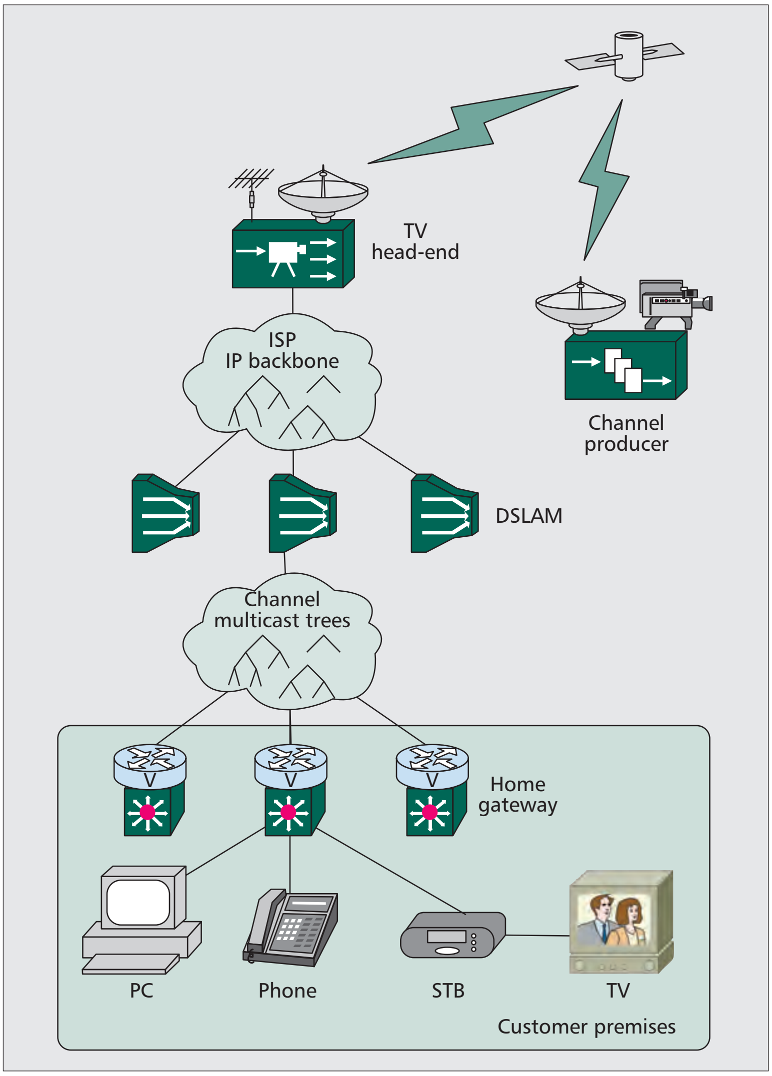

Figure 2. *Typical commercial IPTV architecture.*

The *network delay* includes not only the network latency for the exchange of signaling messages, but also the processing time at each node to set up the corresponding network state. In commercial IPTV, for example, the DSLAM and routers need time to register the requesting STB in a new multicast group, while in P2P IPTV, a group of peers will have to admit the user into their trees/mesh so that video packets can be forwarded through the new unicast connections established. This latter situation leads to longer network delays [2]. Indeed, in P2P IPTV, this network delay can be as long as 10–15 s [1], and is one of the major causes of the total switching delay. In commercial IPTV, on the other hand, the network delay is usually shorter than 100–200 ms, thus contributing very little to the total switching delay [3].

Other delays are also involved in channel switching. Even after the distribution structure is established for the new requested channel and a peer starts receiving video packets, there is further delay until the end system receives a reference frame and starts the decoding process. Such delay is called *synchronization delay*. Various coding schemes can be used, such as MPEG-2 and H.264/MPEG-4. In these schemes, the high correlation existing between video frames is exported into two types of frames, intra-coded (I) and inter-coded (P and B). I-frames can be as long as 3 to 10 times larger than the equivalent quality P- and B-frames [6], but P- and B-frames can only be used after an I-frame has arrived.

The synchronization delay typically has a duration of 500–2000 ms [6], which is, on average, half the duration of the entire sequence of one I-frame and the following P-/B-frames, called group of pictures (GOP) [3]. Although this delay is estimated to be equivalent for both P2P and commercial IPTV, with the exact value depending on the coding scheme used, its relevance is quite different for the two types of system because of its effect on the total switching delay. For commercial IPTV, for example, a typical switching delay value would be 1–2 s [3], and the synchronization delay can double this. For

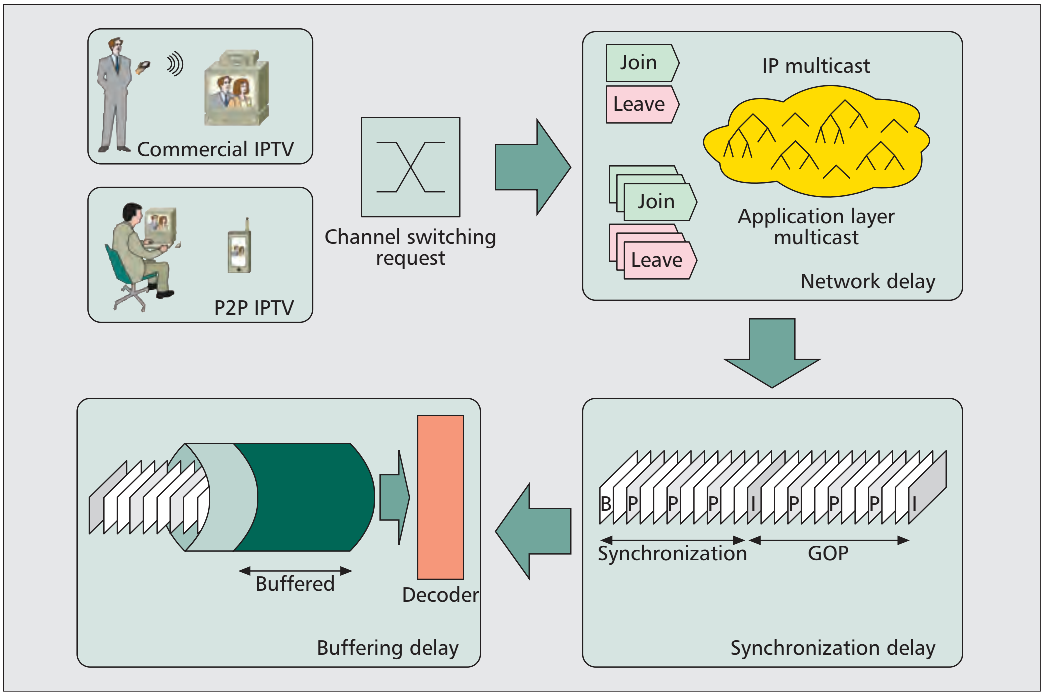

Figure 3. *Delay components involved in a channel switching request.*

Each available channel in the system has one distinct multicast tree per DSLAM. In the user premises, the connection is made by a home gateway, which divides the traffic for TV, telephone and Internet access. Each enabled TV is connected to the home gateway by a separate STB.

P2P IPTV, however, the total switching delay can be as long as 20–30 s [1], with the synchronization delay increasing only about 10 percent.

Further delay arises from buffering video frames after the arrival of the first I-frame, which is called *buffering delay*. This buffering is designed to help overcome the problems caused by the unavailability of content. There are several things that could cause such unavailability of data, including network jitter, packet reordering, and especially peer churn, given that for this reason peers can suddenly become unreachable. The buffering delay is quite significant for both P2P and commercial IPTV systems. In P2P IPTV systems, this delay can be as long as 10–15 s [1], whereas in commercial IPTV systems, typical values are 1–2 s [3].

Figure 3 shows the three components of delay involved in a channel switching request.

Selection of Channel Switching Schemes

Several schemes for reducing the various components of the total switching delay in IPTV systems have been proposed. They differ according to the components addressed, and fall into four main groups. Two of these groups comprise schemes that try to reduce a single delay component, the first one acting on the network delay [5, 7] and the second one on the synchronization delay [6]. The third group is composed by schemes that counteract both the synchronization and buffering delays [8, 9], which is done by the employment of proxy servers with boost streams [3]. The fourth group is formed by pre-join schemes, which aim to reduce the total delay by counteracting all delay components [2, 3, 10–15]. It involves transmitting in advance those channels most likely to be requested at the time of the next channel switching, since these are generally more effective.

The next two schemes described try to reduce the network delay. In P2P mesh networks, peers maintain information about neighbors on the overlay of the watched channel, as well as about neighbors on the overlays of those channels most likely to be switched to. This selection involves channels that exhibit videos of the same genre as that of the watched one, since on average 76 percent of channel switchingss occur to watch videos of the same genre. Neighbors are requested to provide information on peers watching all other channels who are at most a certain distance apart on the mesh network. Although such a switching scheme [5] allows roughly 70 percent of channel switchingss to be faster due to neighbors' support, one major drawback is that it is specific to P2P mesh systems, which usually imply longer startup delays than those experienced in P2P multiple tree systems [1].

For commercial IPTV networks that employ multicast, the IGMP Join message to the multicast group of the channel being requested is sent previous to the IGMP Leave message to the multicast group of the watched channel. This switching scheme [7], however, does not necessarily reduce startup delays, probably due to the short time interval (20–200 ms) between these two messages.

One channel switching scheme to decrease the synchronization delay was proposed in [6]. This scheme exploits a secondary sequence of low-quality I-frames in the main stream. Since these I-frames are considerably smaller than the regular I-frames of the main stream, they can be transmitted more frequently. Since the decoder at the destination can use any I-frame that arrives first, whether regular or of low quality, All the pre-join schemes discussed assume that the user bandwidth is large enough to simultaneously receive multiple streams in full quality, although this is not realistic. Other schemes do, however, consider the more realistic use of low-quality streams.

the frequency of regular I-frames can be reduced, thus compensating for the additional bandwidth required by the secondary substream. The use of low-quality I-frames results in a short transitional period of lower video quality. Nevertheless, this visual impact is considered low, since the human visual system takes some time to adjust to a new visual scene.

The next two schemes employ proxy servers with boost streams to reduce both the synchronization and buffering delays. The scheme in [8] uses additional multicast groups to deliver streams with a minimum quality for each channel. These streams are generated by recoding the original MPEG-4 stream with a lower bit rate, thus demanding roughly 50 percent of the original bandwidth. A user request for a new channel leads to admission into both multicast groups of the requested channel, one carrying the minimum quality stream and the other the full quality stream. Both synchronization and buffering delays are reduced, since the minimum quality stream demands less bandwidth and carries encoded I-frames only. One disadvantage of this scheme is the 50 percent I/O overhead imposed on servers for each channel with at least one user surfing. Moreover, all channels must be encoded with both minimum and regular quality.

Another variant of this type of scheme is the transmission of minimum quality streams to provide previews of other channels [9]. In this case, streams of the most previewed channels are prefetched, and their key MPEG frames (I and P) are buffered. The set of low-quality channels to be transmitted redundantly is chosen based on statistics sent periodically by the STBs. One disadvantage of this scheme is that it assumes that all users in a certain locality are likely to have a common subset of preferred channels, which is not always true. Other disadvantages are the extra processing overhead incurred by those channels previewed but never requested, and the overhead for recoding all the previewed channels.

All the schemes described next are of the pre-join type. The various pre-join schemes were designed to reduce the time necessary to hook up with a new channel in general. Most of them involve some sort of prediction about which new channels are likely to be visited in surfing or selected for watching. One way of reducing all delay components is the use of a home gateway as a local IGMP proxy, ensuring that at each channel switching the user is admitted into the multicast groups of the channels adjacent to the requested one so that switchings to those channels will be performed immediately if actually requested [10]. Adjacent channels are identified by following the navigation pattern of users, and these are registered in local tables at the IGMP proxy. One disadvantage of this scheme is the lack of support for switching to non-adjacent channels (nonlinear switchings).

An extension of the previous scheme is available to provide support for nonlinear switchings. In this scheme, expected random channels are pre-joined in addition to the adjacent ones [11]. The list of channels expected to be requested is provided by a rating server, which compiles relevant statistics from the channel switching information received from all STBs. The algorithm for compiling the expected channel list from the channel statistics, however, was not provided in [11]. In this scheme, no home gateway acting as IGMP proxy is employed, since the STBs themselves are responsible for joining and leaving the redundant channels. This approach has some drawbacks, such as the need to change the STB software and the impossibility of two or more STBs being installed in the same residence to share the redundant channels through a single home gateway.

Another approach is to select the redundant channels based on users' channel surfing behavior and program preferences [12]. In this case, surfing behavior is extracted from the pattern of use of the remote controller, which includes up/down channels (by pressing the up/down buttons), previous channel (by pressing the toggle button), next preset channel (by pressing the preset favorite button), and random channel (by pressing the number buttons). Information about program preference is obtained from the personalized recommendation system for the electronic program guide (EPG) often installed in STBs [12]. This scheme produces a slightly higher hit ratio and lower bandwidth consumption than do those schemes proposed in [10, 11].

In the scheme in [13], adjacent channels are pre-joined only during surfing periods in order to reduce bandwidth consumption. As a consequence, the first request of a channel switching operation cannot be served immediately, since the streams of the redundant channels are not immediately available. Nevertheless, this scheme can still reduce on average 45 percent of the number of channel switching requests with delay. Although the average bandwidth consumption of this scheme is low, the impact of channel switching operations on other network applications due to a bursty demand for bandwidth has not been assessed.

Other strategies are also available for advanced selection of the redundant channels. In the scheme in [3], selections based on popular channels and user behavior can also be pre-joined, but again during surfing periods only. However, the possibilities of these options have not outperformed the original scheme.

All the pre-join schemes discussed assume that the user bandwidth is large enough to simultaneously receive multiple streams in full quality, although this is not realistic. Other schemes do, however, consider the more realistic use of low-quality streams. One option is the scheme in [14], which employs an extension of H.264/AVC video coding, called scalable video coding (SVC), to generate layered coding streams, thus enabling pre-joining of multiple channels with minimum quality. During surfing periods, only the base layers are used, while during watching periods, both base and enhancement layers are used to deliver a single channel in full quality. The selection of the redundant channels to be transmitted during surfing periods is based on two different strategies: adjacent channels and recently watched ones. Although this approach decreases bandwidth consumption due to the use of SVC, the first request of a channel switching operation cannot be immediate, since the streams of the redundant channels are not immediately available.

In an attempt to address the first request of a channel switching operation, the scheme in [15] proposes the pre-join of 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 [14]. The most efficient number of channels to be pre-joined, during both surfing and watching periods, can be determined on the basis of channel switching delays and bandwidth requirements. Later work demonstrated this idea of minimum quality streams during watching periods with a method for the selection of the redundant channels based on the buttons pushed recently and the individual channel preferences of users, as proposed in [12].

The pre-join schemes recently proposed in [2] are designed for P2P IPTV systems. They employ MDC to generate substreams with minimum quality for each channel, which are transmitted to users via multiple distribution trees. By combining a set of substreams, these schemes can pre-join multiple channels with minimum quality. Moreover, a single channel in full quality is received with multiple descriptions of the same stream. One advantage of MDC is that the final quality of the reassembled stream is proportional to the number of descriptions received. As a result, the schemes in [2] do not impose any overhead for recoding the streams to obtain the minimum quality ones, different from the schemes in [8, 9]. Besides that, while the schemes in [14, 15] can use only the base layers from the H.264/AVC to provide the streams with minimum quality, the schemes in [2] can use any MDC description, since all are equally important. An additional advantage of the schemes using MDC is that 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. Thus, the employment of MDC and multiple distribution trees leads to a reduction in network utilization. Furthermore, it facilitates the adjustment of stream quality according to the viewing state of the user as well as the bandwidth limitations.

These schemes define the viewing states of watching and browsing. In the watching state, the user bandwidth is employed primarily for the reception of the descriptions of the watched channel, while in the browsing state 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 enable the first request of a channel switching operation to be addressed, the schemes in [2] reserve, in the watching state, part of the user bandwidth for the reception of a reduced selection of minimum quality streams. Moreover, in the browsing state, part of the user bandwidth is also reserved for maintaining partial 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 with intermediate quality until full quality is restored. The utilization of the user bandwidth in each viewing state can be seen in Fig. 4. Three strategies are employed for selecting the redundant channels to be transmitted in advance: old channel, adjacent channels (as in [10, 13]), and popular channels (as in [3, 11]). These schemes ensure better usage of the user bandwidth and can provide either higher reception quality or greater channel diversity. Results show that they are capable of performing an average of 68 percent of all channel switchings instantaneously with only an average reduction of 19 percent of the overall stream quality.

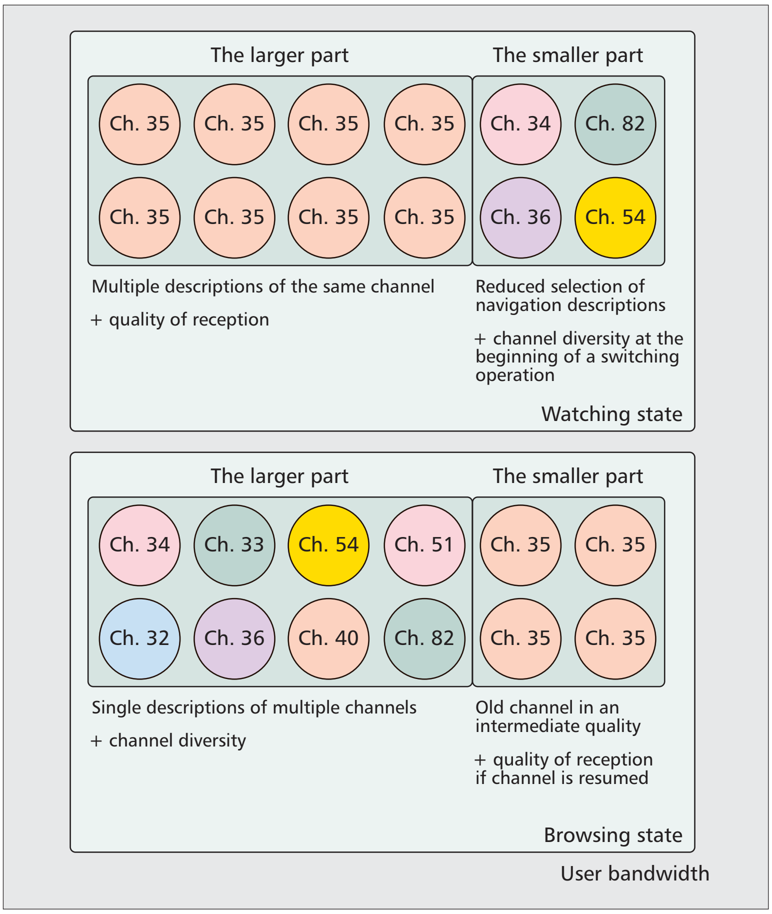

Figure 4. *Utilization of the user bandwidth in each viewing state for the pre-join schemes proposed in [2].*

Table 1 summarizes the main characteristics for all the surveyed schemes.

Conclusion

This article has provided a brief survey of various channel switching schemes for IPTV systems. The problem of reducing startup delays, especially during channel switchings, has received a lot of attention and presents a major challenge in IPTV systems. Although users expect to be able to switch channels quickly, such functionality is not easily provided in IPTV systems. One of the reasons is that due to bandwidth limitations, only a portion of the broadcast content can be transmitted to the users. As a result, each time the user switches channels she/he must leave a distribution structure and

Schemes	Delay component	IPTV architecture	Video coding	First switching req. addressed	Streams with min. quality	Multiple view. states	Strategies for prediction of next channels
Chen *et al.* [5]	Network	P2P	MPEG/H.264	Yes	No	No	Same genre as watched
Sarni *et al.* [7]	Network	Commercial	MPEG/H.264	Yes	No	No	—
Boyce *et al.* [6]	Synchronization	Commercial + P2P	MPEG/H.264	Yes	Yes	No	—
Banodkar *et al.* [8]	Synchronization + buffering	Commercial	MPEG/H.264	Yes	Yes	No	—
Mandal *et al.* [9]	Synchronization + buffering	Commercial	MPEG/H.264	Yes	Yes	No	Set-top box statistics/user locality/most previewed
Cho *et al.* [10]	All	Commercial	MPEG/H.264	Yes	No	No	Adjacent
Lee 2007 *et al.* [11]	All	Commercial	MPEG/H.264	Yes	No	No	Adjacent/expected list
Kim *et al.* [12]	All	Commercial	MPEG/H.264	Yes	No	No	Surfing patterns/channel preferences
Ramos 2010 *et al.* [13]	All	Commercial	MPEG/H.264	No	No	Yes	Adjacent
Ramos 2011 *et al.* [3]	All	Commercial	MPEG/H.264	No	No	Yes	Adjacent/popular/user behavior
Lee 2008 *et al.* [14]	All	Commercial	SVC (H.264/AVC)	No	Yes	Yes	Adjacent/recently watched
Lee 2010 *et al.* [15]	All	Commercial	SVC (H.264/AVC)	Yes	Yes	Yes	Surfing patterns/channel preferences
Manzato *et al.* [2]	All	P2P	MDC	Yes	Yes	Yes	Adjacent/popular/old channel

Table 1. *Channel switching schemes and main characteristics.*

join another, which causes network delay. Another reason is the need for buffering, especially in P2P IPTV systems, so that the problems arising from network bandwidth fluctuations and connection failures can be avoided. A third reason is the digital encoding of video streams, which induces the synchronization delay.

While some of the schemes surveyed try to reduce a single delay component, either that of the network or of the synchronization, others act simultaneously on both the synchronization and buffering delays. The pre-join schemes are among the most effective approaches, since they aim at reducing the total delay by acting on all delay components. Moreover, although the number of existing schemes for P2P IPTV is still limited, those surveyed do act on the most significant delay components for this type of architecture: the network and buffering delays.

One interesting property of channel switching schemes is the existence of multiple viewing states, since the user bandwidth can be exploited differently depending on the user state. For instance, a user initiating a channel switching operation is likely to make a couple of further requests before starting to watch another channel; thus, greater channel diversity can be offered instead of a high stream quality in the surfing state. Moreover, some schemes are designed to pre-join the redundant channels during the surfing state only, which can save some bandwidth.

Another interesting property is the existence of minimum quality streams. Regardless of the coding scheme employed, this option makes better utilization of user bandwidth. For instance, in a pre-join scheme, since the number of redundant channels is limited by the user bandwidth, greater channel diversity can be offered by using these low quality streams. The coding scheme is another important property for obtaining the minimum quality streams. Having to encode the video streams with both minimum and regular qualities incurs in extra overheads, besides not being at all practical.

Finally, among all the existing strategies for predicting the next channels that will be requested, the adjacent one seems to be the most efficient, given that roughly 60 percent of all channel switchings involve sequential channels [2]. Moreover, the combination of multiple strategies can be profitable in the design of a channel switching scheme.

References

[1] D. A. G. Manzato and N. L. S. da Fonseca, "Peer-to-Peer IPTV Services," *Proc. 2nd IEEE Wksp. Enabling the Future Service-Oriented Internet*, Nov. 2008.

[2] —, "Providing Fast Channel Switching in P2P IPTV Systems," *IEEE JSAC*, 2012 Special Issue on Emerging Technologies in Communications (2012 JSAC-SI-ET), Dec. 2012.

[3] F. M. V. Ramos *et al.*, "Reducing Channel Change Delay in IPTV by Predictive Pre-joining of TV Channels," *Signal Processing: Image Commun.*, vol. 26, no. 7, 2011, http://www.sciencedirect.com/science/article/pii/S0923596511000257, pp. 400–12.

[4] M. Z. Ahmad *et al.*, "Prediction-Based Channel Zapping Latency Reduction Techniques for IPTV Systems — A Survey," *Proc. Int'l. Conf. Emerging Technologies '09*, Oct. 2009, pp. 466–70.

[5] Y. Chen *et al.*, "OAZE: A Network-Friendly Distributed Zapping System for Peer-to-Peer IPTV," *Computer Networks*, vol. 56, no. 1, Jan. 2012, pp. 365–77, http://dx.doi.org/10.1016/j.comnet.2011.09.002.
[6] J. M. Boyce and A. M. Tourapis, "Fast Efficient Channel Change," *Proc. 2005 Int'l Conf. Consumer Electronics*, Jan. 2005, pp. 1–2.
[7] M. Sarni, B. Hilt, and P. Lorenz, "A Novel Channel Switching Scenario in Multicast IPTV Networks," *Proc. 5th Int'l. Conf. Networking and Services*, Washington, DC, 2009, pp. 396–401.
[8] D. Banodkar *et al.*, "Multicast Instant Channel Change in IPTV Systems," *Proc. COMSWARE '08*, IEEE, Jan. 2008, pp. 370–79.
[9] S. K. Mandal and M. MBuru, "Intelligent Pre-Fetching to Reduce Channel Switching Delay in IPTV Systems," Dept. of Comp. Sci., Texas A & M University, tech. rep., 2008, http://students.cs.tamu.edu/skmandal/research/channel-switching.pdf.
[10] C. Cho *et al.*, "Improvement of Channel Zapping Time in IPTV Services Using the Adjacent Groups Join-Leave Method," *Proc. 6th Int'l. Conf. Advanced Commun. Tech.*, vol. 2, 2004, pp. 971–75.
[11] J. Lee *et al.*, "Advanced Scheme to Reduce IPTV Channel Zapping Time," *Proc. APNOMS '07*, 2007, pp. 235–43.
[12] Y. Kim et al., "Reducing IPTV Channel Zapping Time Based on Viewer's Surfing Behavior and Preference," *Proc. 2008 IEEE Int'l. Symp. Broadband Multimedia Systems and Broadcasting*, Mar. 2008, pp. 1–6.
[13] F. M. V. Ramos *et al.*, "Channel Smurfing: Minimising Channel Switching Delay in IPTV Distribution Networks," *Proc. 2010 IEEE Int'l. Conf. Multimedia and Expo*, July 2010, pp. 1327–32.
[14] Y. Lee *et al.*, "Reducing IPTV Channel Switching Time Using H.264 Scalable Video Coding," *IEEE Trans. Consumer Electronics*, vol. 54, no. 2, May 2008, pp. 912–19.
[15] 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. Broadcasting*, vol. 56, no. 3, Sept. 2010, pp. 321–30.

Biographies

DANIEL A. G. MANZATO (dmanzato@ic.unicamp.br) 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 in Brazil.

NELSON L. S. DA FONSECA (nfonseca@ic.unicamp.br) received his Ph.D. degree from the University of Southern California in 1994. He is a full professor at the Institute of Computing, State University of Campinas, Brazil. Currently, he is ComSoc Vice President Member Relations, and has served as Director of Latin America Region and Director of On-Line Services. He is former Editor-in-Chief of *IEEE Communications Surveys and Tutorials*. He received the ComSoc Joseph Lo Cicero Award for Exemplary Service to Publications.