Multimed Tools Appl
DOI 10.1007/s11042-015-2913-x

Program-driven approach to reduce latency during surfing periods in IPTV networks

Sajjad Zare [1] **· Akbar Ghaffarpour Rahbar** [2]

Received: 1 January 2015 / Revised: 9 July 2015 / Accepted: 24 August 2015

Abstract Nowadays, using delay sensitive services such as IPTV is rapidly growing. Unlike traditional TV that supports a limited number of channels due to the fixed radio frequency bandwidth, IPTV can support hundreds of channels through IP networks. However, finding a desired IPTV channel among hundreds of channels is a difficult and time consuming issue. To solve this problem, we propose two novel methods to reduce channel surfing period. The first method is channel number-based and is called the Program-Driven Channel Switching (PDCS) method. The second method is popularity-based and is called the Program-Driven with Weight (PDW) method. It is noted that the number of channel switches has a main effect on the channel surfing period. Our proposed methods are based on programs; i.e., program-driven methods. In these methods, instead of choosing channels, users select their desired programs by which they can reach the desired channels that play the programs. Note that a user likes to watch his/her desired program independent of the channel number. Simulation results show that the proposed methods can reduce the number of channel switches; thus reducing the latency for channel surfing period.

Keywords IPTV · Channel switching · Surfing period · Program-driven

1 Introduction

Nowadays, use of delay sensitive services (such as interactive gaming, IPTV, and Voice over IP) is growing. These services not only need huge bandwidth, but also require desired quality

✉ Sajjad Zare
sajjad.zare@pnu.ac.ir

✉ Akbar Ghaffarpour Rahbar
akbar_rahbar92@yahoo.com

[1] Department of Information Technology Engineering, Payame Noor University, Tehran, IRAN

[2] Computer Networks Research Lab, Sahand University of Technology, Sahand New Town, Tabriz, Iran

Published online: 06 October 2015

of service (QoS) [16]. One of the new technologies is IPTV that uses digital television technology which transmits TV content over IP-based network [17].

Unlike the available methods used for broadcasting TV channels (such as terrestrial or satellite), in which channels are limited to radio frequency bandwidth, IPTV operates over IP-based network and delivers digital TV services using Internet protocol over the computer network infrastructure. This enables IPTV to provide hundreds of TV channels for customers [13].

It is difficult to find the desired TV channel because of a large number of channels. In addition, there are some channel zapping delays that occurs during channel switching in IPTV. Therefore, one of the main challenges in IPTV experienced by users is slow reaction times when switching from one TV channel to the other [14].

There are three basic elements in creation of channel zapping time as: (a) Internet Group Management Protocol (IGMP) internal processing delay, (b) decoding delay, and (c) buffering delay. Moreover, there is an important factor; i.e., the number of channel switching times to achieve the desired channel. Number of channel switching times is multiplied by the prior three factors [1].

When a user switches to a channel, his Set-Top Box (STB) performs the process of sending IGMP multicastingmessage, leaving and joining messages to obtain the new channel, then waiting for the object video stream to come. The STB waits more for a decodable frame, which is called an Intra-codedframe (I-frame). Then, STB buffers some frames to avoid the unsmooth display caused by the delay jitter over the Internet [1].

The objective of this paper is to propose a novel method to reduce channel surfing period. As mentioned above, the number of channel switching times has the main effect on channel surfing period. Almost, all methods that try to reduce channel zapping time are channel-driven. This means that these methods try to select the prejoin channels based on popularity of channels. To reduce channel surfing period, our proposed method is based on programs; i.e., a program-driven method. This means that users do not select channels, instead they select programs. The key idea of the proposed method is that in IPTV with hundreds of channels, selection of a channel with the desired program is difficult and users like to select their programs. Our contribution is to propose a novel program-driven method to reduce channel surfing period in IPTV networks.

This paper is organized as follows. Section 2 reviews the methods used for reducing channel surfing period. We detail our proposed methods in Section 3. In Section 4, we evaluate our proposed methods followed by a conclusion in Section 5.

2 Related work

Channel surfing period is one of the major challenges in IPTV. Many methods have been proposed that try to reduce the channel surfing period. In general, these methods are based on three approaches [15]:

- *Content-based solutions* that modify the content being delivered to improve the response time during the channel change phase; for example during the video coding and processing phases [4, 8]. The content-based solutions take into account video coding schemes to reduce the channel zapping time in IPTV. One of the content-based solutions uses the H.264 scalable video coding scheme in order to reduce channel change time [11]. In this scheme, a base layer and enhancement layer of each channel are allocated to two separate multicast groups. In the preview mode at selection period, users access the base layers of different channels already stored in the buffer, so they can switch channels without delay. In the watching mode, they use both the base and enhancement layers of the selected channel to achieve high quality [11].

- *Network-assisted solutions* that operate on network level; for example upgrade the network infrastructure or utilize dedicated servers to service the clients' channel change requests immediately [2, 7]. In the context of network-based solutions, Joo et al. presented how to effectively reduce the network delay by adjusting the number of broadcasting channels to be serviced from closer routers. They have also discussed how to efficiently decrease video decoding delay by adding additional I-frames to normal video frames [9].
- *Client-based solutions* that operate at the user side and on the user equipment; for example, upgrade in STB [5, 10]. The client-based solutions typically require the clients to prejoin a selective set of channels concurrently during the channel switching process. When a user selects a channel, these methods try to pre-join to some channels and place them in a list. However, which channels are prejoined and with which order they should be placed in the list are two problems.

Two groups of researches have been proposed to identify the channels to be prejoined in client-based solutions. A popularity-based arrangement scheme presents channels in a list based on their popularities with viewers who can be expected to spend most of their times watching a limited number of channels [12]. The second strand of research involves adjacent channel pre-fetching schemes; for example, the Adjacent Groups Join-Leave method is applied to reduce the channel zapping time in IPTV [6]. The method presented in [6] sends an IGMP membership report message not only for the currently requested channels, but also for the adjacent channels. On the other hand, when a user selects a channel, he/she prejoins to the adjacent of the selected channel. Therefore, IGMP internal processing delay is eliminated for the adjacent of the selected channel and users can watch the adjacent channels without network delay because multicasting streams for the adjacent channels arrive at their home gateway [6].

The method in [13] identifies 'hot' channels, those likely to be watched in the near future by a target viewer. Then, the scheme maintains a list of hot channels, and a second list of cold channels (i.e., those channels that have less popularity). During channel navigation, the hot list is searched first. The hot channels are also prefetched in low resolution to support a preview mode. This combination usually reduces zapping time substantially, even if the network bandwidth is limited [13].

In [3], some information is gathered about users' behaviors, say which user, when watches which channel. In this method, there are two groups of information. One group is about users' behavior in selecting the channels collected for each user individually, and another group is about all users' behavior in selecting the channels which are public. Then, this information is used to predict the channels that users will select.

As mentioned before, there are some methods to select pre-join channels. However, the problem is how to order pre-join channels. In the non-weighted circular ordering (NWCO) technique, the IPTV channels are visited in a numerical order. In this state, the popular channels are likely to be scattered, and this imposes a long channel surfing period to the system. This means that a user must do many channel changes for switching from one popular channel to another popular channel. As a result, the user wastes a lot of time within channel surfing period to find his/her desired channel [1]. To solve this problem, the authors in [14] have proposed a solution called the frequency circular ordering (FCO) technique that orders the sessions based on their frequencies. On the other hand, in this method, all hot channels are placed in the right and all cold channels are placed in the right of the circle. Therefore according to Fig. 1b, accessing to some cold channels are easier than some hot channels using the FCO method. For example, to achieve channel 10 from channel 4, we have three channel switches and to achieve channel 17 from channel 4 we need two channel switches according to Fig. 1b.

To solve this problem, the authors in [12] have proposed a solution called the frequency interleaved ordering (FIO) technique that orders the sessions by distributing them evenly based on their access frequencies.

Figure 1a, b, and c illustrate the channel sequences of the numerical ordering, frequency circular ordering, and frequency interleaved ordering, respectively. The number in each cell represents the frequency priority and the numbers outside the cell are channel numbers. Number 1 represents the least popular channel and 17 represents the most popular channel. For example, if a user's channel selection behavior is: ch4►ch13►ch16, then number of channel switching in numerical ordering, frequency circular ordering and frequency interleaved ordering is 12, 4 and 2, respectively. For example, assume a user watches ch 4. To select ch 13 in numerical ordering, he/she first switches to ch 5, then 6, next 7,…, until ch 13 (see Fig. 1a). Then, to select ch 16, he/she needs three channel changes (i.e., ch 14, ch 15, ch16) to select ch 16.

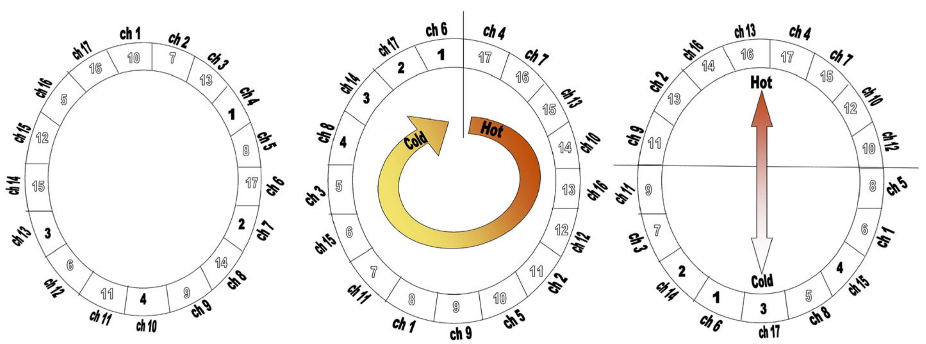
However, in frequency circular ordering, for selecting ch 13, he/she needs only two channel switches as ch 7 and ch 13 as shown in Fig. 1b. Then, to select ch. 16, he/she needs two channel changes (i.e., ch 10, ch16) to select ch 16.

3 The PDW and PDCS methods

As mentioned above, the number of channel switching times is the main factor in channel surfing period [1]. Since in IPTV there are hundreds of channels, selecting a channel with a desired program is difficult.

Define n_{switch} to be the number of channel switches and L_{surf} to be channel selection period. It is noted that there is a relation between L_{surf} and n_{switch} based on Eq. (1) from [1]

$$L_{surf} = n_{switch} \times \left(\delta_{wait} + T_{join} + T_{dbuf}\right). \tag{1}$$



(a) Numerical ordering (b) Frequency circular ordering (c) Frequency interleaved ordering

Fig. 1 **a** Numerical ordering **b** Frequency circular ordering **c** Frequency interleaved ordering

Here, parameter δ_{wait} represents the waiting time for the delivery of the earliest key-frame after selecting a channel, where its value varies in the interval $[0,T_{GOP}]$, where T_{GOP} is the Group Of Picture (GOP) duration. Parameter T_{join} is average time to join a broadcasting group (based on the Internet group management protocol), and T_{dbuf} is the minimum decoding and buffering latency.

As shown in Eq. (1), the summation of other delay factors is multiplied by n_{switch}, and therefore, this factor has main effect on the overall channel surfing period. To solve this problem, we propose a method to decrease number of channel switches.

Some methods use popularity to select a channel. These methods have some problems as:

a) In IPTV with hundreds of channels, there are many channels with high popularity, and therefore, the number of channel switching times becomes high.
b) If we want a program that is played in a channel with less popularity, the number of channel switching times becomes high.
c) With channel-driven, users may miss exciting programs that are broadcast on other channels.

Therefore, to solve these problems, we propose the selection of channels to be program-driven. In the proposed methods, we categorize TV programs in different groups such as (a) news, (b) movies, (c) sport programs, (d) health programs, (e) cooking programs, (f)cartoons and so on. Then, we insert them as buttons in TV remote controller or as a menu in TV.

Please note that if we categorize the channels, the range of the search will be low; therefore, the desired channel can be founded sooner. For example, if a user decides to watch a filmstrip, in channel driven methods, he must find his desired channel among hundreds of channels. Therefore, the number of channel switches becomes high. However, in our program driven methods, the user selects his desired program (in this example filmstrip) only among the channels (listed for him) that play the program, where this list is very smaller than the number of all channels. Then, the user only needs to surf among them to find his desired filmstrip. On the other hand, in program driven, the number of channel switches becomes lesser.

3.1 The PDW method

Our first program-driven method is called Program-Driven with Weight (PDW). In PDW, after prejoining channels based on program-driven, we select the prejoined channels based on their popularities.

We show channel surfing period in the FIO and PDW methods in Fig. 2. We assume that a user uses the up/down button in a TV control in surfing period. As is shown in Fig. 2, when the user request a channel under the FIO technique, the channels with high popularity are prejoined and only channels in the prejoined list are delivered to the user. But, in the PDW, when the user selects a group, the channels that play that group programs are pre-joined and the user can select one of them that he/she likes. For example, if a user decides to watch a football match, he/she selects the sport group, then all channels that play this sport are pre-joined and the user can select one of them. We can also subcategorize each group to have an effective system. For example, the sport group is categorized into (a) football, (b) volleyball, (c) wrestling, and so on.

It is noted that when the channels are pre-joined using the program-driven method, we can select the next channel in the list based on its channel number or its popularity. In the case of popularity, we firstly select the channel with the most popularity similar to FIO. In other words, to reach the desired channel, we apply the proposed method and construct pre-join channels firstly, and then apply the FIO technique to select the channels in the pre-joined list. We call this proposed method Program-Driven with Weight (PDW). For example, assume that there are 100 TV channels and 11 of them play movies. When a user decides to watch a movie, he/she presses the movie button in TV controller or selects the movie group in the TV menu. Then, the channels that play movies at that time are pre-joined as shown in Fig. 3. We can use frequency interleaved ordering to place the channel in the list.

3.2 The PDCS method

Our second program-driven method is called Program-Driven Channel Switching (PDCS). In this method, after prejoining channels based on program-driven, we select the prejoined channels based on their channel numbers. In other words, we select adjacent channels in the list in this method.

In the case of channel number, we firstly select the minimum channel number in the list. In other words, to reach the desired channel, we apply the proposed method and construct pre-join channels firstly, and then select the adjacent channels in the pre-joined list. For example, assume that there are 100 TV channels and 11 of them play movies. When a user decides to watch a movie, he/she presses the movie button in the TV controller or selects the movie group in the TV menu. Then, the channels that play movies at that time are pre-joined as shown in Fig. 4. We can use adjacent channel ordering to place the channel in the list.

As is mentioned in Section 2, in the Frequency Interleave Ordering (FIO) technique, the channels are ordered by distributing them based on their access frequencies. The frequency (or weight) of a channel for a given user is obtained based on the number of channel accesses by the user. When the user selects a channel, the frequency of that channel for the user is increased.

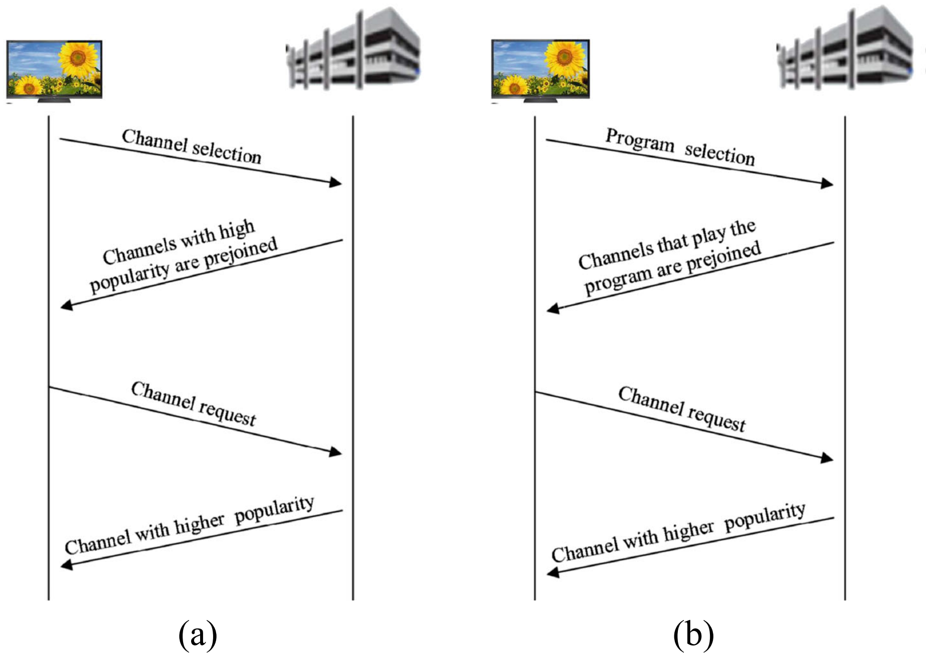


Fig. 2 Channels surfing under **a** FIO, **b** PDW

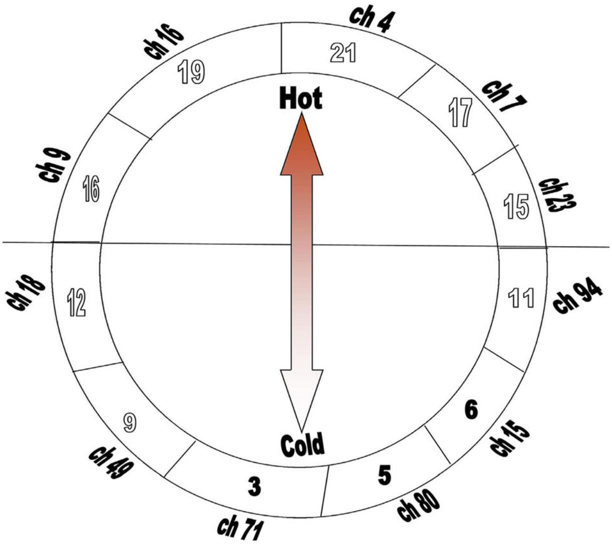


Fig. 3 Channels that play movie are pre-joined (interleave)

4 Performance evaluation

To evaluate the proposed methods, we product a software with Microsoft SQL server and C#. We assume that there are 100, 70, and 40 TV channels, where each channel has a popularity rate between 0 and 19. We categorize TV programs in different groups as (a) news, (b) movies,

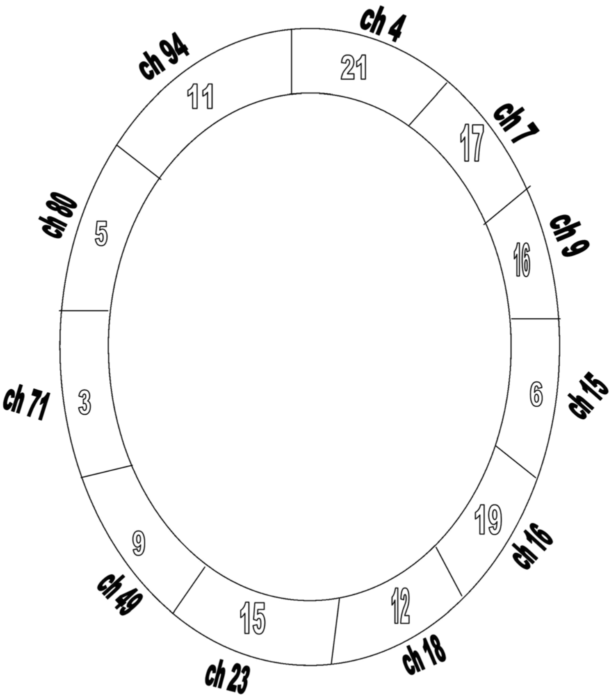


Fig. 4 Channels that play movie are pre-joined

(c) sport programs, (d) health programs, (e) cooking programs, (f) cartoons, (g) documentary and (h) entertainment.

Frequency of each channel and type of program that each channel is broadcast are generated by a random function in C#, but channel searching is performed by users.

In each experiment, we distribute different types of programs in each channel and channel popularity randomly. As mentioned before, the frequency or popularity (weight) of a channel is obtained based on the number of channel accesses by a user.

In the proposed methods, we focus on channel zapping time; therefore, we make attention to some parameters such as the number of channels and average session join latency. Other parameters such as data rate, jitter and so on are not considered. For example, we assume that data rate is enough and there is not jitter and packet loss in the system similar to [1].

In each experiment, it is firstly determined which channels play the desired program of a user. Then, the software determines how many channel switches are necessary to select the target channel. This issue is common in both channel-driven methods (popularity based) and our proposed methods.

In our simulations, we have repeated all experiments 15 times, and 95 % confidence intervals are found to be within at most 5 % of the mean values shown. We make the assumptions stated in Table 1.

It is noted that in the PDCS approach, when a user selects a group, and the channels of that group are pre-joined, we select the channels in an ascending order based on the channel number. However, the Program-Driven with Weight approach (PDW) chooses the pre-joined channels based on their popularities similar to FIO. For example in Fig. 3 (Section 3), channel 4 with popularity 21 is selected first, then channel 19 with popularity 5, then channel 7, and so on. Recall that 1 represents the least popular channel and 21 represents the most popular channel.

We assume that after making a request by a client, he/she waits for the received content to be displayed on the screen before switching to the next channel. In addition, in the evaluation, we assume that the favorite program of a user is broadcast on channel x, and then we compute how many channel switches are needed to reach that channel in each method.

We compare PDCS, Frequency Interleave Ordering (FIO), Frequency Circular Ordering (FCO) and PDW in the simulations. Figure 5a, b, and c show the number of channel switches when the number of TV channels are 40, 70 and 100, respectively. Define channel switching to be the number of channel changes to achieve the desired channel. As is shown in these figures, the PDW can provide the best results. It is noted that when the weight of a channel is high, the results of four methods, PDCS, FIO, FCO and PDW are close to each other since the channel with high weight is the first selection in these methods. However, for those channels that have not high weights, the results of the four methods are different. In this case, the PDW has the best results. Since the PDW method

Table 1 Assumptions

Parameter	T_{GoP}	T_{join}	T_{dbuf}
Value	1 s	0.1 s	250 ms
Description	The GOP duration is equal to 1 s	Average session join latency (based on the Internet group management protocol) is equal to 100 ms	The minimum decoding and buffering latency is equal to 250 ms

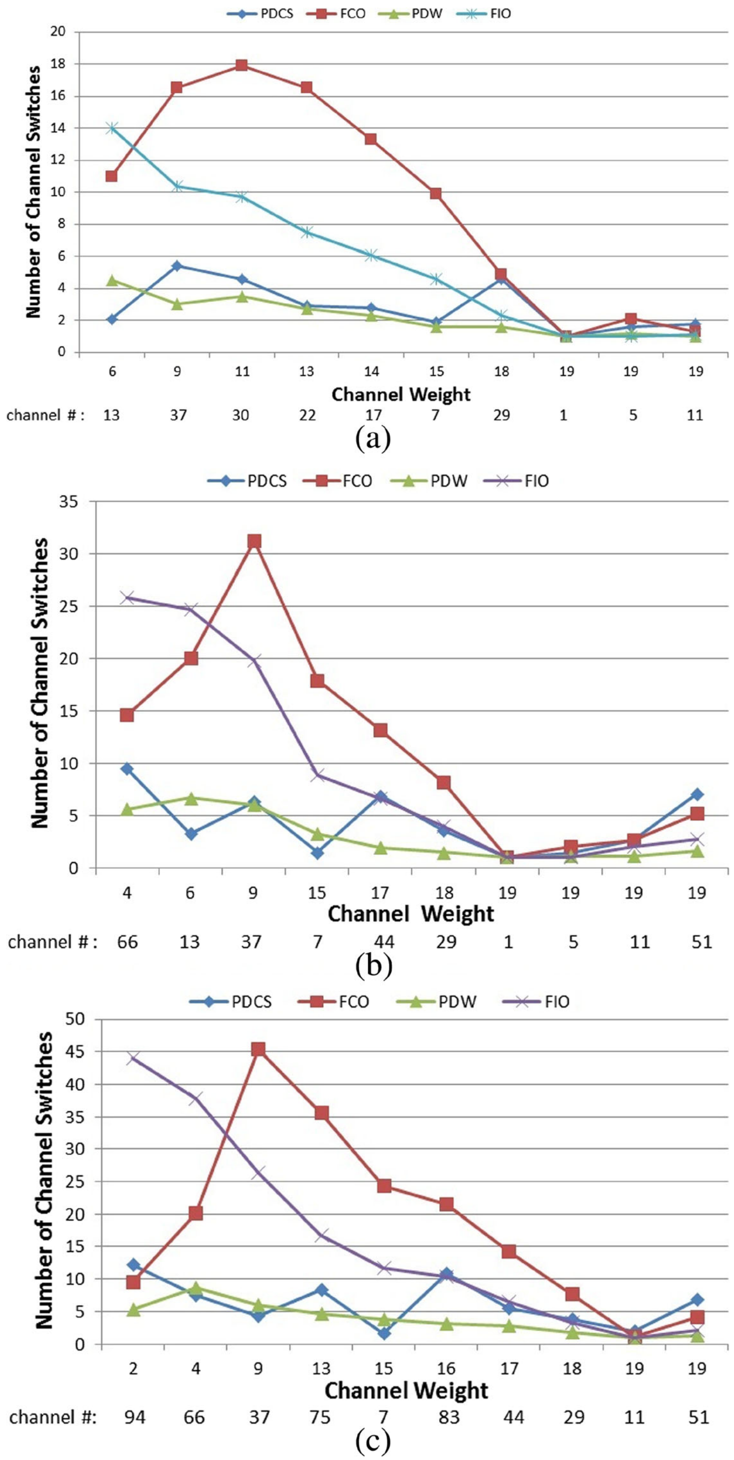


Fig. 5 The number of channel switches when the number of TV channel are **a** 40, **b** 70, and **c** 100

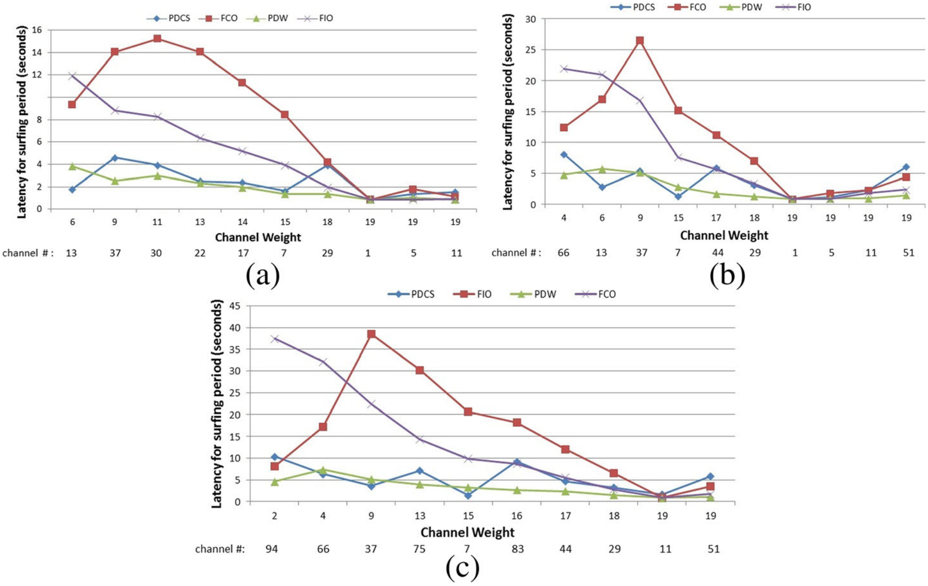


Fig. 6 Latency within surfing period when the number of TV channel are **a** 40, **b** 70, and **c** 100

eliminates the channels that are not in the goal channel group, the selection domain becomes small. It is noted that the FCO method has a maximum point. Since for example according to Fig. 1b, if the number of channels is 17, to select channel 4►13, we have two channel switches, but to select channel 4►15, we need 6 channel switches. This is because in this method, the pre-joined channels are circular and we select short track from the beginning. If channel position is less than half, we track right; otherwise we track the left of circle according to Fig. 1b.

As seen in the Fig. 5, the FCO diagram has a max point. This is because when we select a channel, the channels with high priority are buffered (pre-joined) in a circular manner as seen in Fig. 1b. Therefore, if the next selected channel is in semicircle on the right, then channels are selected from the right. If the next selected channel is in semicircle on the left, then channels are selected from the left. The channels with low priority are on the left side of circle and channels with high priority are on the right side of circle. Therefore, to reach these channels, we have a few channel switches, but channels with middle priority are on the middle of the circle; to reach them, we need more channel switches and this issue generates the max point.

To solve this problem, the FIO puts channels with high priority in the upper semicircular and channels with low priority in the lower semicircular as an interleave manner as seen in Fig 1c. Therefore in the FIO diagram, if channel priority is low, the number of channel switches is high and vice versa. Therefore, we have a descending diagram from low priority channels to high priority channels.

The PDCS diagram in Fig. 5 has a zigzag manner. This is because when we select a program, the channels that play that program are pre-joined and we select these channels regardless of the priority in the PDCS method. To solve this problem, we use priority to select channels in PDW, therefore this diagram experiences descending order from low to high priority channels.

Figure 6a, b and c show the latency for surfing period under different number of channels based on Eq. (1). As is shown, the PDW can provide the best performance results. This is because in this method we select the channel group firstly; therefore, other groups are eliminated and the selection domain becomes smaller. The results show that the number of channel switches to receive the desired channel has much effect on the latency within surfing period. In short, the proposed methods can reduce the number of channel switches.

We also provide subjective metric to evaluate the superiority of PDW over FCO and FIO. In this evaluation, first we make sample IPTV system using SQL and C# with three methods to select channels under (PDW, FCO and FIO). We have requested from 17 different people to work with the sample IPTV system with the three methods (PDW, FCO and FIO) to select channels. Each method is given a score between 1 (worst quality) and 5 (best quality) based on its user friendly and easy to working by all reviewers. Then, we calculate the mean opinion score (MOS) for each technique, which is the arithmetic mean among all individual scores. The MOS of the PDW, FCO and FIO techniques are computed 4.11, 1.72, and 2.05, respectively. This shows that PDW outperforms both FCO and FIO, even under the subjective evaluation.

5 Conclusion

In this paper, we have proposed novel methods to reduce the latency for surfing period in IPTV. Our proposed methods are based on programs; i.e., a program-driven method. In these methods, instead of choosing channels, users select their desired programs by which they can reach the channels that play the programs. Note that a user likes to watch his/her desired program independent of the channel number. Simulation results show that the proposed methods can reduce the number of channel switches; thus reducing the latency within channel surfing period.

Acknowledgments This research was financially supported by Payame Noor University (PNU), Iran.

References

1. Azgin A, Altunbasak Y (2013) Dynamic channel reordering to reduce latency during surfing periods in IPTV networks. IEEE Trans Broadcast 59(3):471–483
2. Bejerano Y, Koppol P (2009) Improving zap response time for IPTV. In: Proc. IEEE INFOCOM, Apr, pp 1971–1979
3. Beyragh AA, Rahbar AG (2014) IFCS: an intelligent fast channel switching in IPTVover PON based on human behavior prediction. Multimed Tools Appl 74:1049–1071
4. Boyce JM, Tourapis AM (2005) Fast efficient channel change [set-topbox applications]. In: Proc. IEEE Int. Conf. Consumer Electron. Jan, pp 1–2
5. Cho C, Han I, Jun Y, Lee H (2004) Improvement of channel zapping time in IPTV services using the adjacent groups join-leave method. In: Proc. IEEE Int. Conf. Adv. Commun. Technol., vol. 2. Feb, pp 971–975
6. Cho C, Han I, Jun Y, Lee H (2004) Improvement of channel zapping time in IPTV services using the adjacent groups join-leave method. Proceedings of the 6th International Conference on Advanced Communication Technology, vol.2, pp 971–975
7. Degrande N, Laevens K, Vleeschauwer DD, Sharpe R (2008) Increasing the user perceived quality for IPTV services. IEEE Commun Mag 46(2):94–100

8. Jennehag U, Zhang T, Pettersson S (2007) Improving transmission efficiency in H.264 based IPTV systems. IEEE Trans Broadcast 53(1):69–77
9. Joo H, Song H, Lee D, Lee I (2008) An effective IPTV channel control algorithm considering channel zapping time and network utilization. IEEE Trans Broadcast 54(2):208–216
10. Lee J, Lee G, Seok S, Chung B (2007) Advanced scheme to reduce IPTV channel zapping time. In: Managing next generation networks and services (Lecture Notes Computer Science, vol. 4773). Springer, Berlin, pp 235–243
11. Lee Y, Lee J, Kim I, Shin H (2008) Reducing IPTV channel switching time using H.264 scalable video coding. IEEE Trans Consum Electron 54(2):912–919
12. Lee E, Whang J, Oh U, Koh K, Bahn H (2009) A popular channel concentration scheme for efficient channel navigation in IPTV. IEEE Trans Consum Electron 55(4):1945–1949
13. Lee E, Ku JY, Bahn H (2014) An efficient hot channel identification scheme for IPTV channel navigation. IEEE Trans Consum Electron 60(1):124–129
14. Manjunath L, Mastani SA (2013) A novel approach for increasing channel navigation in IPTV based on user's channel selection interests. Int J Eng Res Appl (IJERA) 3(3):1331–1336
15. Siebert P, Caenegem TNMV, Wagner M (2009) Analysis and improvements of zapping times in IPTV systems. IEEE Trans Broadcast 55(2):407–418
16. Zare S, Ghaffarpour Rahbar A (2012) An FEC scheme combined with weighted scheduling to reduce multicast packet loss in IPTV over PON. J Netw Comput Appl 35(1):459–468
17. Zare S, Ghaffarpour Rahbar A (2014) Congestion control in IPTV over PON using Digital Fountain forward error correction. J Netw Comput Appl 37(1):240–252

Sajjad Zare received his technician degree in Computer Software from Imam Khomeini International University, Qazvin, Iran in 2006 and B.Sc. degree in Computer Software engineering from Zanjan University, Zanjan, Iran in 2008. Mr. Zare received his M.Sc. degree in Computer Networks from Sahand University of Technology, Sahand New Town, Tabriz, Iran in 2010. At present he is a lecturer in Information Technology engineering group at Payame Noor University in Qazvin, Iran. His main research interests are optical networks, scheduling, wireless networks, IPTV, and VoIP.

Dr. Rahbar received the B.Sc. and M.Sc. degrees in computer hardware and computer architecture from the Iran University of Science and Technology, Tehran, Iran, in 1992 and 1995, respectively, and the Ph.D. degree in computer science from the University of Ottawa, Ottawa, ON, Canada, in 2006. He is currently an Associate Professor with the Electrical Engineering Department, Sahand University of Technology, Sahand New Town, Tabriz, Iran. He is the director of the Computer Networks Research Laboratory, Sahand University, Iran. Dr. Rahbar is a senior member of the IEEE. Dr. Rahbar is currently on the Editorial Board of the Wiley Transactions on Emerging Telecommunications Technologies Journal and the Journal of Convergence Information Technology. He is editor-in-chief of Journal of Nonlinear Systems in Electrical Engineering. His current research interests include optical networks, optical packet switching, scheduling, PON, IPTV, VANET, network modeling, analysis and performance evaluation, the results of which can be found in over 100 technical papers including two books, six book chapters, 45 journal papers, 50 referred conference papers, and 12 non-referred technical papers and talks (see http://ee.sut.ac.ir/showcvdetail.aspx?id=13).