Reducing Channel Zapping Time in IPTV Based on User's Channel Selection Behaviors

Chae Young Lee, *Member, IEEE*, Chang Ki Hong and Kang Yong Lee*
Department of Industrial Engineering, KAIST, 373-1 Kusung Dong, Taejon, Korea
*ETRI, 138 Gajeongno, Taejon, Korea

***Abstract*— Channel zapping time is a crucial issue in Internet Protocol Television (IPTV) Quality of Experience (QoE) performance. One way to reduce channel zapping time is a predictive tuning method, which reduces channel zapping time by prejoining channels that are likely to be selected next, in addition to the currently watched channel. This paper presents an improved predictive tuning method that satisfies required channel zapping time with minimized bandwidth usage. Unlike existing methods, the proposed method determines the most efficient number of prejoining channels in a surfing period and in a viewing period differently by estimating expected channel zapping time and expected bandwidth usage with a Semi-Markov Process. We also propose a prejoining channel selection method according to the combined button and channel preference. Mathematical model and simulation show that the proposed method achieves the required channel zapping time with less bandwidth consumption and a more stable network than previous works.**

***Index Terms*—Channel selection behavior, channel zapping time, predictive tuning, IPTV.**

I. Introduction

The rapid diffusion of high speed internet and the fast advance of broadband networking technology have been breaking down the walls between telecommunication and broadcasting. IPTV (Internet Protocol Television) is one of the key applications in the telecommunication market which gives an opportunity for telephone companies to benefit from video delivery over IP networks. Currently, there are active trials and commercial deployments across the world, including in North America, Europe, and Asia/Pacific [1].

However, the nature of the IPTV mechanism limits its quality of service (QoS), especially in the aspect of channel zapping time. Channel zapping time is defined as the time difference between the user asking for a channel change by pressing some buttons on the remote control and the display of the first frame of the requested channel on the TV screen [2]. Compared with the conventional analog TV that broadcasts all channels, it multicasts only the currently watched channel stream to each Set Top Box (STB) at any moment through the IP network.

According to the QoE requirements of DSL Forum, the zapping time should be limited to a maximum of 2 seconds [3] and ITU-T FG IPTV is also considering it as one of the QoE metrics. Channel zapping time usually takes up to 1 second for MPEG-2 and as long as 2 seconds for H.264/MPEG-4 AVC, forcing the viewer to wait for the picture [4]. However, a model where Mean Opinion Score (MOS) depends on the channel zapping time on a logarithmic scale shows that the channel zapping time should be less than 0.43 seconds for an MOS score of 3.5 [5].

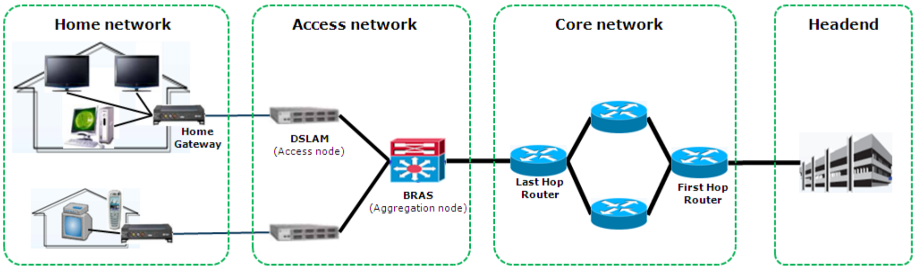
Channel zapping time consists of a) Internet Group Management Protocol (IGMP) internal processing delay, b) decoding delay, and c) buffering delay. If the user switches the channel, the STB has to perform the process of sending IGMP multicasting tree, leaving and joining messages to obtain a new channel. After waiting for the object video stream to come, STB waits more for a decodable frame, which is called an Intra-coded frame (I-frame). Then STB buffers some frames to avoid the unsmooth display caused by the delay jitter over the Internet.

Many research efforts have been undertaken to reduce channel zapping time. Methods adding tune-in servers at the end of the core network have been proposed [6], [7]. However, high-cost dedicated servers which provide high rate unicast streaming must be added into core networks, and STBs should also shift precisely between those temporary unicast streams and final multicasting streams to achieve this purpose [4][8].

Some methods that improve the video encoding or RTP protocols have been proposed [1], [9]-[15]. These may still be unsatisfactory due to the mandatory decoding delay, which is limited by the video coding standard, and buffering delay [4].

One of the approaches to reduce the channel zapping time is for the STB to join the adjacent channels of the current channel in advance [16]. If the user switches to an adjacent channel, the user can watch the selected adjacent channel without channel zapping time because the stream of the adjacent channel has already been sent to the STB. This method can be expanded to join more channels highly likely to be selected next [1][4][17]-[21]. This kind of method is called 'prejoining method' or 'predictive tuning.'

However, predictive tuning consumes additional bandwidth



for prejoining channels, which can cause congestion in the access network. To reduce bandwidth consumption, a method of prejoining channels only while a user is surfing channels is proposed [4] and [19]. While watching a channel (called 'viewing period'), the STB only receives the currently watched channel. When the user starts to switch the channel, the STB joins more channels in addition to the currently watched channel. This period is called the 'surfing period.' When the user decides to watch the channel, the STB leaves the extra channels. This can reduce average bandwidth usage. However, the first switch of each surfing always has to suffer from a very large channel zapping time, which makes the required channel zapping time unattainable.

Another aspect of predictive tuning is the accuracy of prediction for the next channel. Many researches consider channel popularity, personal channel preference, and behaviors in operating the remote control [1][4][17]-[21]. However, the user's button preference, which refers to which buttons on the remote controller a user uses most frequently, hasn't been considered. In predictive tuning, the button preference is more relevant than the channel preference.

In this paper, we present an improved predictive tuning method that satisfies average channel zapping time with minimized bandwidth consumption. The proposed method has two different aims: 1) determining the most efficient number of prejoining channels based on the user's channel selection behaviors and 2) selecting prejoining channels with a consideration of the button preference. To satisfy required channel zapping time with less bandwidth consumption, the proposed method prejoins a small number of channels during a viewing period and prejoins more channels during a surfing period. A Semi-Markov Process (SMP) is used to analyze the user's channel selection behavior. The optimal number of prejoining channels is obtained from expected channel zapping time and bandwidth usage calculated from the proposed model. The proposed method also takes advantage of H.264 Scalable Video Coding (SVC,) which was proposed by Lee, et al. [19].

The rest of this paper is organized as follows. We introduce the details of IPTV service system, predictive tuning, and H.264/SVC in Section II. Then, we develop a mathematical model to describe the viewer channel selection behaviors in section III. In Section IV, we describe the improved predictive tuning method, and Section V combines and derives a detailed analysis of the channel zapping time and the bandwidth usage. Numerical results, including the simulation to verify it, are in Section VI. We also present a simple algorithm to apply the proposed method to the practical IPTV service system in section VII. Finally, concluding remarks are given in Section VIII.

II. Background

A. IPTV Service System

A typical IPTV system consists of four elements, as shown in Figure 1. The video headend is responsible for encoding content received through satellite, terrestrial, or fiber networks into MPEG-2 or MPEG-4 formats. The content encoded with MPEG-2 takes a bit rate of 3~8 Mbps for standard definition (SD) quality, and 15~18 Mbps for high definition (HD) quality, while the content encoded with H.264/MPEG-4 AVC takes a bit rate of 1.5~8 Mbps for SD quality, and 8~12 Mbps for HD quality [3]. The content, which is encapsulated into IP packets, is sent to the core network using IP multicast or IP unicast. The core network groups the encoded video streams into their respective channels.

The access network contains the aggregation node and the access node. The aggregation node, which is generally called broadband access server (BRAS), is responsible for maintaining user policy management, such as authentication and subscription details. At the end of the access network, the access node, such as a digital subscriber line access multiplexer (DSLAM,) provides various internet access technologies to end users. It can be the 20 Mbps promise of asymmetric digital subscriber line 2+ (ADSL2+), the 50 Mbps capability of very high digital subscriber line (VDSL2), or the 100 Mbps potential of fiber to the x (FTTx) [22].

The home network distributes the data, voice, and IPTV traffic in subscribing homes with the home gateway. Each home can have two or more TV sets and other equipment using an IP network, such as VoIP phones or laptop computers. If each set shows one channel, then the home network should support at least two HDTV channels, which takes a bit rate of 36 Mbps in the case of MPEG-2 simultaneously. Therefore, the IPTV service system should be designed to manage the bandwidth of the channels effectively because each channel necessitates high bandwidth [1][8][23].

B. Predictive Tuning

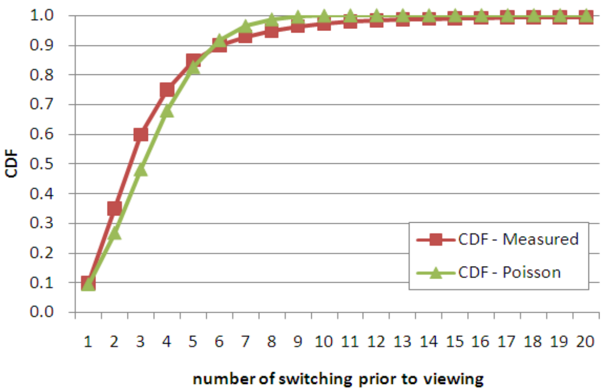
Cho, et al. [16] presented a method of reducing channel zapping time by sending the adjacent channels of the current one in advance. Predictive tuning has two beneficial characteristics: First, it is a pure client technology and can be implemented without changes to the IPTV system and in combination with other solutions to fast channel change. Second, overall channel zapping time can be reduced to nearly zero by decoding data streams of prejoining channels or buffering key frames, which are called ‘I-frames’, of prejoining channels. The second advantage of the predictive tuning is remarkable, because when a user switches to a prejoining channel, the user can watch the scene immediately - at least the still cut of the channel.

Various predictive tuning methods are also proposed, focusing on how to select prejoining channels. Instead of adjacent channels, a predictive tuning based on channel popularity, which is aggregated from the rate server, has been proposed [17]. Other researchers [18] have proposed a method that selects the candidate channels by considering the user’s remote control behavior and personal channel preference. They consider that viewers tend to keep pushing the same button to change channels, and the STB can match a channel number, called an ‘expected channel,’ with any pushed button. There is also a method of prejoining channels that have been recently watched and are numerically adjacent to the current channel [19].

One of the weak points in predictive tuning is that it consumes additional bandwidth. Basically, predictive tuning consumes significant bandwidth because each channel necessitates high bandwidth. If an STB prejoins two channels in parallel to the current channel, it takes a bit rate of 36 Mbps for HD quality in the case of H.264/MPEG-4 AVC. If there are two or more STBs in a home, it will cause lack of bandwidth of the access network. To reduce bandwidth consumption, the authors proposed the method of prejoining channels only while a user is surfing channels [4], [19]. While watching a channel, called a ‘viewing period,’ the STB receives the currently watched channel only. When the user starts to switch the channel, the STB joins more channels in addition to the currently watched channel. This period is called the ‘surfing period.’ When the user decides to watch a channel, the STB leaves the extra channels. This can reduce average bandwidth usage, but the first switch of each surfing period always has to suffer from a very large channel zapping time. This affects the average channel zapping time significantly.

C. *H.264 Scalable Video Coding*

Another approach to reduce the bandwidth usage of predictive tuning is using H.264/Scalable Video Coding (SVC). H.264/SVC is an extension of H.264/AVC (Advanced Video Coding) which achieves a high degree of spatio-temporal and quality scalability. An SVC bit-stream consists of a base layer and one or more enhancement layers. The base layer is an H.264/AVC bit stream, ensuring backward compatibility for legacy decoders and providing low quality of video with low resolution, low frame rate (temporal resolution), or low picture fidelity (PSNR). The enhancement layers include the information of the frames with higher resolution, frame rate, or



PSNR, but it cannot be decoded without a base layer. Low-end devices, which have low resolution, low network capacity or low computing power, can take only the base layer and high-end devices can take both the base layer and enhancement layers.

While the high quality of video is served with the base layer and the enhancement layers of the currently watched channel, the STB can prejoin the base layers of the candidate channels, because base layers consume less bandwidth. Lee et al. show that acceptable video quality for surfing is achieved with the base layer, which consumes 10 times less bandwidth than an overall SVC stream [19]. Therefore, the STB can prejoin 10 channels with the bandwidth usage of a high quality channel.

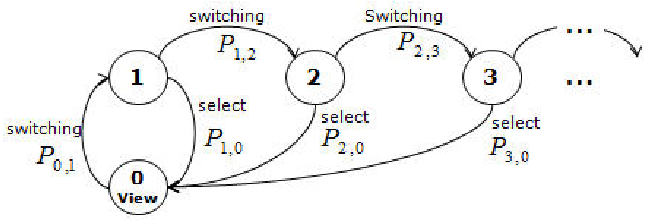
III. Behavior Model

There are two types of viewer behaviors in channel selection: (a) how many channels users switch and how long they spend switching before they enter a ‘viewing period,’ and (b) which channels they choose while switching. The first one can be called ‘channel switching behavior,’ and second one ‘channel surfing behavior.’

A. *Channel Switching Behavior*

Fortunately, there is a measurement for this behavior based on real IPTV service. In [24], the traces from a large-scale commercial IPTV are used to study how users select channels in the real-world. The study shows that users generally switch 4 channels on average before they decide on a channel, and 10% of users watch the first channel they switch to, while 10% of users switch more than 6 times prior to viewing. Figure 2 shows that the cumulative distribution function (CDF) of the number of changes prior to viewing a channel, which approximates to CDF of Poisson distribution. Therefore, the number of switches prior to viewing can be assumed to follow Poisson distribution.

A Semi-Markov Process (SMP) was used to analyze this behavior. It works as follows: Each state indicates the number of switches, the transition to the next state indicates ‘switching’ and the transition to the state ‘0’ indicates entering ‘viewing



period.' While a user is watching a channel, the user is in the state '0' and when the user switches the channel, the user transits to state '1.' If the user decides to watch the channel, the user goes back to the state '0.' Instead, if the user switches again, the user transits to state '2.' This model is illustrated in Fig 3.

Based on the assumption, the transition Probability ($P_{i,j}$) can be calculated from the Poisson distribution. If a user switches once prior to viewing, it means that the user transits from state '0' to state '1,' and then the user transits from state '1' to state '0'. Therefore, the probability of switching once prior to viewing becomes

$$\Pr\{\text{switching once prior to viewing}\} = \frac{e^{-\lambda}\lambda^1}{1!} = P_{0,1} \cdot P_{1,0}$$

We can expend this equation to

$$\Pr\{\text{switching } k \text{ times prior to viewing}\}$$
$$= \prod_{i=1}^{k} P_{i-1,i} \cdot P_{k,0} = \frac{e^{-\lambda}\lambda^k}{k!} \quad ,(k \ge 2)$$

The sum of the outgoing probabilities from each state should be 1.0. Hence

$$P_{0,1} = 1,$$
$$P_{i,0} = 1 - P_{i,i+1}$$

Which gives

$$P_{k-1,k} = 1 - \frac{e^{-\lambda}\lambda^k}{k!} / \prod_{i=1}^{k-1} P_{i-1,i} \quad ,(k \ge 2) \qquad (1)$$

We assume that the number of the state is limited, which is reasonable in the aspect of human behavior. In [24], the number of switching is limited to 100.

B. *Channel Surfing Behavior*

Users generally use a remote control that has various buttons: up/down, toggle, preset, and numeric buttons. There are two types of preferences: (a) button preference and (b) channel preference. Button preference refers to which buttons on the remote control a user uses more frequently and channel preference refers to how often a user watches a channel.

Sometimes a user switches to the objective channel, which follows the user's channel preference; sometimes the user just surfs channels to find an interesting channel, which follows the user's button preference. However, even when the user has the objective channel, the user should select a button to push with the button preference. Therefore, it can be assumed that the user selects the next channel as follows: first, users decide which button to push. If they decide to push the 'up', 'down,' or 'toggle' button, the next channel will be determined by the currently or previously watched channel. If they decide to push numeric buttons, they should choose the next channel by themselves.

IV. Proposed Method

A. *Determining the Number of Prejoining Channels*

Prejoining during a viewing period has two aspects. First, it affects the average channel zapping time more than prejoining during a surfing period because the first switching of every surfing is influenced by it. Second, it consumes bandwidth significantly because the duration of a viewing period is much longer than that of a surfing period. Therefore, we need to find the optimal number of prejoining channels in the 'viewing period' and the 'surfing period.'

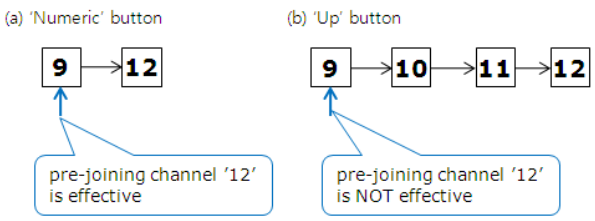
From the proposed Semi-Markov Model (SMP), the expected channel zapping time and expected bandwidth usage can be calculated. The number of prejoining channels in a viewing period and the number of a surfing period are assigned as variables. By adjusting those variables, the variation of the expected bandwidth usage and expected channel zapping time can be observed.

The detailed analysis for the expected channel zapping time and expected bandwidth usage is described in Section 5, and a simple algorithm to apply the result in the practical STB is described in Section 7.

B. *Selecting Prejoining Channels*

A remote control has various buttons: up/down, toggle, preset, and numeric buttons. Therefore, a user can use various ways to switch to the objective channel. For example, if a user is watching channel '9,' and the user decides to switch to channel '12,' which is the most preferred channel of the user, the user can push the 'up' button three times or use numeric buttons by pushing '12' directly as shown in Figure 4.

In the predictive tuning, the most important channel is not the objective channel but the next channel, because STB reselects prejoining channels whenever a user switches the channel. That implies that button preference affects more than channel preference. In the example shown in figure 4-(b), if the STB prejoins channel '12,' it is of no use because he will switch to channel '10' first by pushing the 'up' button.



Therefore, the channel preference and the button preference can be combined to the probability of a channel that a user will switch to the channel next at a particular moment. Prejoining channels are selected in descending order of the probability.

If there are N channels and K types of buttons, Let η_k denote button preference of button k (1,…K), ρ_j denote channel preference of channel j (1,…N) and η_1 indicate the button preference of numeric buttons. Then, we can define the combined probability ω_j of each channel j at a particular moment as follows

$$\omega_j = \eta_1 \rho_j + \sum_{k=2}^{K} \eta_k \beta_k^j \quad ,1 \le j \le N, \tag{2}$$

$$\beta_k^j \begin{cases} 1, & \text{if channel } j \text{ corresponds to button } k \\ 0, & \text{otherwise} \end{cases}$$

β_k^j is a corresponding function between buttons and channels. Except numeric buttons, corresponding channel to a button depends on the current channel, the previous channel, or user configuration. Thus, the STB can obtain the correct value of β_k^j in each switch.

However, ω_j can't be used to calculate the expected channel zapping time in the proposed SMP model, because ω_j is not for a steady state, but a particular moment. Instead of ω_j, We use an approximate value in the steady state for the proposed SMP model. We assume that there are K-1 more channels, which corresponds to buttons except numeric buttons. The approximate probability Π_j of channel j is obtained as follows

$$\Pi_j = \begin{cases} \eta_1 \rho_j, & 1 \le j \le N \\ \eta_{j-N+1}, & \text{N+1} \le j \le N+K-1 \end{cases} \tag{3}$$

V. Analysis

A. Expected Channel Zapping Time

The expected channel zapping time is only related to the number of transitions between the states, not to the time spent in the states. Therefore, expected channel zapping time can be obtained from the steady-state probability of embedded discrete time Markov chain (DTMC) and transition probability.

In the nature of predictive tuning, if a user switches to the channel which is not prejoined, he has to take on compulsory channel zapping time to finish the process of exchanging IGMP messages, stream waiting and decoding. We call this 'Full Delay' and denote it with F. If the user switches to the prejoined channel, there would be no channel zapping time.

If we define C_i as the set of prejoining channels in state i, the expected delay in state i can be presented as follows

$$E[D_i] = \sum_{j \in C_i} \Pi_j \cdot 0 + (1 - \sum_{j \in C_i} \Pi_j) \cdot F \tag{4}$$

To calculate expected channel zapping time, we should consider when the channel zapping time occurs. 'Switch' means the transition from i to i+1, which makes channel zapping time. Thus, the proportion of $E[D_i]$ in the expected channel zapping time can be obtained from the steady-state probability of the transition from i to i+1, which can be presented as follows: $\pi_i P_{i,i+1}$

If it is assumed that the number of state is limited to $m+1$, The steady-state probability of embedded DTMC can be obtained from those equations

$$\pi_0 = \pi_1 P_{1,0} + \pi_2 P_{2,0} + ... + \pi_m P_{m,0}$$
$$\pi_1 = \pi_0 P_{0,1},\ \pi_2 = \pi_1 P_{1,2},\ ....\ ,\ \pi_m = \pi_{m-1} P_{m-1,m}$$
$$\sum_{i=0}^{m} \pi_i = 1$$

which give

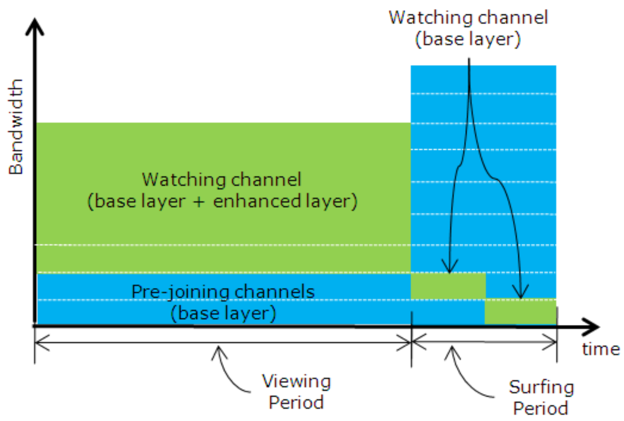
$$\pi_0 = \frac{1}{P_{0,1} + \sum_{i=0}^{m-1} \prod_{j=0}^{i} P_{j,j+1}}, \qquad \pi_i = \pi_{i-1} P_{i-1,i}, \quad (\text{i} \ge 1) \tag{5}$$

The probability that the transition from i to i+1 occurs in the steady-state is $\pi_i P_{i,i+1} = \pi_{i+1}$. Therefore, the expected channel zapping time is obtained as follows

$$E[D] = \sum_{i=0}^{m} \pi_{i+1} E[D_i] \tag{6}$$

B. Expected Bandwidth Usage

Using SVC in IPTV, each channel is split into a base layer and enhancement layers, which are allocated to different multicast groups [19]. In a viewing period, the STB receives both layers of watching channel to produce full-quality video and the base layers of prejoining channels. But in a surfing period, the STB receives only the base layers of watching



channel and prejoining channels as shown in Figure 11.

If we define n_i as the number of prejoining channels in the state *i*, the bandwidth usage in each state can be obtained as follows

$$BW_i \begin{cases} (n_i+1)BW_{base} + BW_{enh}, & i=0 \\ (n_i+1)BW_{base}, & i \neq 0 \end{cases} \quad (7)$$

If we define μ_i as the amount of the time spent in the state *i*, the steady-state probability of SMP is obtained from the steady-state probability of embedded DTMC and the time spent in the states.

$$P_j = \frac{\pi_j \mu_j}{\sum_{i=0}^{m} \pi_i \mu_i}$$

The expected bandwidth usage can be calculated through it

$$E[BW] = \sum_{i=0}^{m} P_i \cdot BW_i \ .$$

VI. Numerical Result

In this section, the analysis on the channel zapping time and bandwidth usage of various predictive tuning methods is presented. In Section VI.A, the performance evaluation about the policy for the number of prejoining channel is provided. It shows that our analytical results closely match the simulation results. In Section VI.B, the performance of the proposed prejoining channel selection method is compared with previous methods in terms of the average channel zapping time.

To make more precise comparison, we compare the policies of the number of prejoining channels in the same prejoining channel selection method, and the prejoining channel selection methods in the same policy for the number of prejoining channels.

The default values of the parameters for analysis and simulation are: Full Delay (*F*) is 2 seconds [4], total number of channels (*N*) is 50, bandwidth of base layer (BW_{base}) is 1 Mbps, bandwidth of enhanced layers (BW_{enh}) are 8 Mbps, which is required for the HD quality of video [3]. As described above, channel preference is assumed to follow a Zipf-like distribution with $s=1$, and the number of switching prior to watching is assumed to be Poisson-distributed with $\lambda = 3.7$. The amount of the time spent in a viewing period is assumed to be 12 minutes, while the amount of the time spent in a surfing period ($\mu_i, i \neq 0$) is assumed to be 9 seconds.[24]

A remote control has four types of buttons: numeric, up, down, toggle, and the button preference follows the user type, which is described in the Section 6.B. To compare the methods in the same conditions, all methods take advantage of H.264/SVC, which means a prejoining channel uses only the base layer of the channel.

The comparison of the peak bandwidth usage of each method is also performed. The peak bandwidth usage also affects the congestion of the network significantly because the switches caused by commercial break may account for up to 95% of a viewer's total switches [4], meaning that channel switches can be crowded in a certain moment and using too much bandwidth in the surfing period can be critical to the congestion of the network.

A. *The Policy for the Number of Prejoining Channels*

Three policies for the number of prejoining channels are evaluated, comparing channel zapping time and bandwidth usage as shown in Table I. As mentioned before, every policy selects prejoining channels based only on the channel preference for the comparison under the same conditions.

Figure 6 shows the expected channel zapping time of each policy with the different number of prejoining channels. The expected channel zapping time should be below 0.43 second to satisfy QoE [5]. The 'always' method provides the solution with 12 prejoining channels and 'proposed (v=5)' method with 16 prejoining channels in the surfing period and 5 prejoining channels in the viewing period. However, the 'surfing only' method can't provide the solution. It also shows that increasing the number of prejoining channels becomes ineffective gradually.

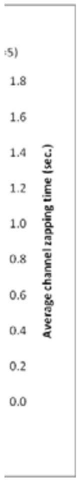
channels

ıannels

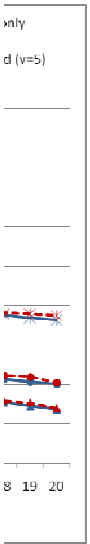
Figure 7 shows the expected bandwidth usage of each policy with the different number of prejoining channels. The ‘always’ method with 12 prejoining channels consumes 20.6 Mbps, while the ‘proposed (v=5)’ method with 16 prejoining channels in the surfing period and 5 in the viewing period consumes 14.2 Mbps, which is 31.1% bandwidth improvement. It also shows that prejoining more channels in the surfing period scarcely affects the average bandwidth usage.

Figure 8 presents the peak bandwidth usage and the expected channel zapping time of each policy. The ‘always’ method with 12 prejoining channels consumes 21.0 Mbps, while the ‘proposed (v=5)’ method requires 17.0 Mbps, which is 19.0% improvement.

We develop a Java-based simulation program to verify our mathematical analysis. We simulate a set-top box which takes 100,000 channel switches in each case of the channel surfing and switching model described in Section 3. By taking every channel zapping time of each switch and bandwidth usage in each state under the various predictive tuning methods average channel zapping time and bandwidth usage of each switch is obtained. Figure 9 shows that our simulation results closely match the analytical results in the aspect of channel zapping time.



er number



To make the ‘surfing only’ method provide a solution, we relaxed the required average channel zapping time to 7 seconds and compared the performance of each policy as shown in Table II. It shows that the ‘surfing only’ method provides a solution using minimum average bandwidth (10.02 Mbps). But its peak bandwidth is too high (28.0 Mbps), because it prejoins too many channels to meet the channel zapping time requirement. On the other hand, the ‘proposed (v=2)’ method provided a solution using suitable average bandwidth (11.05 Mbps) and peak bandwidth usage (12.0 Mbps). In the case of the ‘always’ method, both average bandwidth usage (14.57 Mbps) and peak bandwidth usage (28.0 Mbps) are higher than the proposed method.

TABLE II
COMPARISON OF POLICIES FOR THE NUMBER OF PREJOINING CHANNELS

	Average Channel Zapping Time (seconds)	Average Bandwidth Usage (Mbps)	Peak Bandwidth Usage (Mbps)
Prejoining 6 channels 'always'	0.698	14.57	15.0
Prejoining 27 channels during 'surfing only'	0.7004	**10.02**	28.0
Prejoining 2 channels in viewing period and 11 channels in surfing period	0.6732	11.05	**12.0**

TABLE III
PREJOINING CHANNEL SELECTION METHODS

Method	Description
PREF	Prejoins most preferred channels [5]
ADJ-PREF	Prejoins 2 adjacent channels and most preferred channels [4,7]
EXP-PREF	Prejoins 1 expected channel and most preferred channels [6]
PROPOSED	Prejoins channels based-on combined button-channel preference

TABLE IV
USER TYPES FOR BUTTON PREFERENCE

Method	Description
Numeric only	Uses only numeric buttons with channel preference
Numeric preferred	Uses numeric buttons 60% of times with channel preference
Up/Down preferred	Uses up/down buttons 60% of times
Surfing Model [6]	Has the tendency to keep pushing the same button

B. Prejoining Channel Selection

Four prejoining channel selection methods, including our method, are evaluated comparing channel zapping as shown in Table III. As mentioned before, every method prejoins the same number of channels for the comparison under the same conditions. Every method prejoins 2 channels in the viewing period and 10 channels in the surfing period.

The performance of the prejoining channel selection depends on the user's preference in channels and buttons. As mentioned before, channel preference is usually assumed to follow Zipf-like distribution, but there is no guide for button preference. Therefore, we made four types of users which have unique characteristics to represent various users, as shown in Table IV.

TABLE V
COMPARISON OF USER TYPES FOR BUTTON PREFERENCE

Channel Zapping time (sec.)	Numeric only	Numeric preferred	Up/Down preferred	Surfing Model [6]	Avg.
PREF	**0.668**	0.884	1.147	0.941	0.910
ADJ-PREF	0.891	**0.621**	**0.332**	0.684	0.632
EXP-PREF	**0.668**	0.838	0.708	**0.485**	0.675
PROPOSED	**0.668**	**0.608**	**0.318**	**0.426**	**0.505**

Table V shows the average channel zapping time of each method for the different types of users. In the case of 'Numeric only,' 'PREF,' 'EXP-PREF' and 'PROPOSED' methods provide minimum channel zapping time. If a user only pushes numeric buttons, those three methods perform in the same manner because 'EXP-PREF' and 'PROPOSED' methods consider the button that the user pushed recently or frequently.

In the cases of 'Numeric preferred' and 'Up/Down preferred,' prejoining adjacent channels is more effective. The reason is if the user decides to push numeric buttons, it is hard to predict the next channel. But if he decides to push the up or down button, the STB can predict the next channel. That makes the 'ADJ-PREF' method provide a better solution than other previous methods because the 'ADJ-PREF' method gives more weight to adjacent channels than preferred channels. The proposed method also considers these characteristic. In addition, it uses combined button-channel preference, which provides more accurate weight between adjacent channels and the preferred channel. As a result, the proposed method provides minimum channel zapping time.

In the case of the 'Surfing Model,' introduced in [18], the 'EXP-PREF' method gives minimum channel zapping time among the previous methods because it considers the buttons that the user pushed recently. But the proposed method provides better performance because we can prejoin many channels by using H.264/SVC and we can consider more preferred buttons instead of one recently used button. As a result, the proposed method provides minimum channel zapping time.

VII. ALGORITHM FOR THE PRACTICAL SYSTEM

Numerical results show the outstanding performance of the proposed predictive tuning method. However, it seems hard to apply the proposed method to the practical system because it is analytical rather than practical. We will show the simple algorithm to apply our method in the practical IPTV system, focused on a set-top box.

The proposed algorithm has several characteristics compared with the analytical process. First, it uses collected data of the user's real behavior instead of data from models or statistics, which provide more effective solutions. The STB can aggregate the information on the channel switching easily: how many channels the user switches prior to viewing, how many times a channel is selected and how many times a button is pushed.

From this information, the STB can estimate more accurate channel and button preference, channel zapping time and bandwidth usage. Second, the proposed algorithm uses a finite duration instead of the steady-state, which is assumed in the analysis, which means that the STB re-estimates all values periodically.

The proposed algorithm assumes that the number of prejoining channels is limited because of the lack of the capacity of the access network, and that the channel zapping time of non-prejoined channels is constant.

The process of the proposed algorithm is as follows:

1) In a certain period, the STB aggregates the information.

$n(i)$: The number of visiting states *i* in our model, which can be obtained by counting the number of switches prior to viewing.

$c(j)$: The number of times selecting channel *j* to watch.

$b(k)$: The number of times pushing button *k*.

2) At the end of the period, the STB calculates those values, which are needed to estimate the channel zapping time.

A. Transition probability ($P_{i,i+1}$) and steady-state probability (π_i)

$$P_{i,i+1} = \frac{n(i+1)}{n(i)}, \qquad \pi_i = \frac{n(i)}{\sum_{j=0}^{m} n(j)}$$

B. Button preference (η_k) and channel preference (ρ_j)

$$\eta_k = \frac{b(k)}{\sum_{i=1}^{K} b(i)}, \qquad \rho_j = \frac{c(j)}{\sum_{i=1}^{N} c(i)}$$

3) From those values and the analysis described in Section 5, we can estimate the channel zapping time of the next period. So we can determine the number of prejoining channels in 'viewing period' and 'surfing period' to meet the required channel zapping time as follows:

Step 1: Calculate the average channel zapping time in the previous period.

Step 2: Adjust the objective of the channel zapping time of next period (Obj_{next}). If the average channel zapping time of previous period ($D_{previous}$) is longer than the overall objective of the channel zapping time ($Obj_{overall}$), decrease the objective of the channel zapping time of next period. Otherwise, increase it. It can be describe as follows:

$$Obj_{next} = Obj_{overall} + (Obj_{overall} - D_{previous})$$

TABLE VI
SIMULATION RESULT OF PROPOSED ALGORITHM

Channel Zapping time (sec.)		Numeric only	Up/Down preferred	Surfing Model [6]	Avg.
PREF /always[5]	Zapping (sec.)	**0.445**	-	0.455	0.450
	BW (Mbps)	**20.639**	-	27.638	24.139
ADJ-PREF /always[4,6]	Zapping (sec.)	0.460	**0.422**	0.434	**0.439**
	BW (Mbps)	20.638	**11.637**	17.638	**16.638**
EXP-PREF /always[6]	Zapping (sec.)	0.446	0.440	**0.442**	0.443
	BW (Mbps)	20.639	22.637	**16.638**	19.971
PROPOSED /always	Zapping (sec.)	0.444	0.443	0.447	0.445
	BW (Mbps)	20.638	11.638	13.639	15.305
PROPOSED /adaptive	Zapping (sec.)	**0.429**	**0.447**	**0.428**	**0.435**
	BW (Mbps)	**11.708**	**10.370**	**12.784**	**11.621**

Step 3: Determine the number of prejoining channels in 'surfing period' and 'viewing period.' First, Increase the number of prejoining channels in the 'surfing period' from 1 to the maximum limit, examining whether the expected zapping time, calculated from the data described in 2)-A, meets the objective of channel zapping time in next period (Obj_{next}).

Step 4: If this fails, increase the number of prejoining channels in 'viewing period' and go back to Step 3.

Step 5: Whenever a user switches the channel, calculate combined probability (ω_j) of each channel *j* from the data described in 2)-B, and prejoin the number of channels, determined in the previous steps, in descending order of ω_j .

To evaluate the proposed and previous methods [16, 17, 18], the simulation explained in Section 6 is performed for channel zapping time and average bandwidth usage. The number of prejoining channels is limited to 20 with 20 Mbps capacity. 100,000 channel switches are experimented. The objective of the channel zapping time is 0.43 sec [5].

In each prejoining channel selection method 'always' policy is adopted based on Table I for the number of prejoining channels. Each method chooses the number of prejoining channels to meet the channel zapping time requirement of 0.45 sec. We also added the proposed algorithm, called 'PROPOSED/adaptive,' which chooses the minimum number of prejoining channels for the channel zapping time requirement.

Table VI shows that the proposed algorithm requires the lowest bandwidth in most user types. On the average, the proposed algorithm requires 30% less bandwidth than other methods.

VIII. Conclusion

In this paper, an improved predictive tuning method with two aspects is described. It features a) determining the most efficient number of prejoining channels in a surfing period and a viewing period and b) selecting prejoining channels with combined button-channel preference. Also it takes advantage of H.264/SVC, which alleviates the bandwidth consumption.

The user's channel selection behaviors is analyzed with Semi-Markov Processes, with which we can estimate average channel zapping time and average bandwidth usage and obtain the optimal number of prejoining channels in a surfing period and in a viewing period. To verify its accuracy, a simulation is also conducted. Studies show that the proposed method achieved required channel zapping time with less bandwidth consumption and a more stable network than previous work. This also shows that the proposed prejoining channel selection method with combined button-channel preference provides shorter channel zapping time than any previous methods in the environment of the various types of users.

A simple algorithm to apply the proposed method to the practical IPTV system is also presented. It is demonstrated that the proposed algorithm satisfies the QoE requirement of the channel zapping time. The research on the channel zapping time in this paper is more oriented to the set-top box. For more comprehensive study a comparative analysis on the aggregation and access nodes in the network is necessary.

References

[1] H. Joo, H. Song, D.B. Lee, I. Lee, “An Effective IPTV Channel Control Algorithm Considering Channel Zapping Time and Network Utilization,” Broadcasting, IEEE Transactions on, 2008, art. no. 4446230, pp. 208-216.

[2] H.Uzunalioglu, “Channel Change Delay in IPTV Systems,” 2009 6th IEEE Consumer Communications and Networking Conference, CCNC 2009, art. no. 4784832

[3] DSL Forum, “Triple-play Services Quality of Experience (QoE) Requirements,” DSL Forum, Tech. Rep., Dec. 2006, DSL Forum, Tech. Rep. TR-126.

[4] K. Link, W. Sun, “Switch Delay Analysis of a Multi-Channel Delivery Method for IPTV,” 2008 4th IEEE International Conference on Circuits and Systems for Communications, 2008, ICCSC , art. no. 4536798, pp. 471-476.

[5] R. Kooji, K. Ahmed, and K. Brunnstrom, “Perceived Quality of Channel Zapping,” Proceedings of the 5th IASTED International Conference on Communication Systems and Networks, CSN 2006 , pp. 155-158.

[6] A.C. Begen, N. Glazebrook, W. Ver Steeg, “A Unified Approach for Repairing Packet Loss and Accelerating Channel Changes in Multicast IPTV,” Consumer Communications and Networking Conference, IEEE, 2009, pp. 1-6.

[7] Y. Zhu, W. Liu, L. Dong, W. Zeng, H. Yu, “High Performance Adaptive Video Services based on Bitstream Switching for IPTV Systems,” Consumer Communications and Networking Conference, IEEE, 2009, pp. 1-5.

[8] S.G.Choi, H.J. Park, J.M.Lee, J.K.Choi, “Adaptive Hybrid Transmission Mechanism for On-Demand Mobile IPTV Over WiMAX,” Broadcasting, IEEE Transactions on, June 2009, Vol. 55, Issue 2, pp. 468 – 477.

[9] U. Jennehag, S. Pettersson, “On Synchronization Frames for Channel Switching in a GOP-based IPTV Environment,” IEEE Consumer Communications & Networking Conference, January 2008, pp.638-642.

[10] U. Jennehag, T. Zhang, S. Pettersson, “Improving Transmission Efficiency in H.264 Based IPTV Systems”, IEEE Transactions on Broadcasting, March 2007, pp.69-78.

[11] U. Jennehag and T. Zhang, “Increasing bandwidth utilization in next generation iptv networks,” International conference on Imgage Processing (ICIP), Singapore, Oct 2004, pp. 2075-2078.

[12] Cisco Visual Quality Experience Whitepaper, “Delivering Video Quality in Your IPTV Deployment”, Nov 2006.

[13] J. Caja, “Optimization of IPTV Multicast Traffic Transport over Next Generation”, Metro Networks, 12th International Telecommunications Network Strategy and Planning Symposium, Nov 2006, art. no. 4082440.

[14] R. Shmueli, O. Hadar, R. Huber, M. Maltz, M. Huber, “Effects of an Encoding Scheme on Perceived Video Quality Transmitted Over Lossy Internet Protocol Networks,” Broadcasting, IEEE Transactions on, 2008, Vol. 54, Issue 3, pp. 628 – 640.

[15] M. Rezaei, M. M. Hannuksela, M. Gabbouj, “Tune-in Time Reduction in Video Streaming Over DVB-H,” Broadcasting, IEEE Transactions on, March 2007, Vol. 53, Issue 1, pp. 320 – 328.

[16] C. Cho, I. Han, Y. Jun and H. Lee, “Improvement of Channel Zapping Time in IPTV Services Using the Adjacent Groups Join-Leave Method,” 6th Intern. Conf. on Advanced Communication Technology, 2004, pp. 971-975.

[17] J. Lee, G. Lee, S. Seok, B. Chung, “Advanced Scheme to Reduce IPTV Channel Zapping Time,“ Lecture Notes in Computer Science 4773 LNCS, pp. 235-243.

[18] Y. Kim, J.K. Park, H.J. Choi, S Lee, H Park, J Kim, Z, “Reducing IPTV channel zapping time based on viewer’s surfing behavior and preference,” IEEE International Symposium on Broadband Multimedia Systems and Broadcasting 2008, Broadband Multimedia Symposium 2008, BMSB , art. no. 4536621.

[19] Y. Lee, J. Lee, I. Kim, H. Shin, “Reducing IPTV channel switching time using H.264 scalable video coding,” Consumer Electronics, IEEE Transactions on, 2008, pp. 912-919.

[20] ITU-T FG IPTV-C-0545, “Consideration on Channel Zapping Time in IPTV Performance Monitoring,” 2007.

[21] H. Fuchs, N. Farber, IIS Fraunhofer, G Erlangen, “Optimizing channel change time in IPTV applications,” IEEE International Symposium on Broadband Multimedia Systems and Broadcasting 2008, Broadband Multimedia Symposium 2008, BMSB , art. no. 4536619.

[22] Y. Xiao, X. Du, J. Zhang, F. Hu, S. Guizani, ”Internet Protocol Television (IPTV): the killer application for the

next-generation Internet," IEEE Communications Magazine, 2007, Vol. 45, No. 11, pp. 126-134.
[23] U. Jennehag, T. Zhang, S. Pettersson, "Improving Transmission Efficiency in H.264 Based IPTV Systems," Broadcasting, IEEE Transactions on, March 2007, Vol. 53, Issue 1, pp. 69 – 78.
[24] M Cha, K Gummadi, P Rodriguez, "Channel Selection Problem in Live IPTV Systems", In Proc. of ACM SIGCOMM Poster, Seattle, Washington US, August 2008.
[25] C.Y. Lee, K.J. Oh, "User-Viewing Characteristics Aware Resource Allocation for Mobile IPTV in WMNs," 2009 6th IEEE Consumer Communications and Networking Conference, CCNC 2009, art. no. 4784835, pp.1-2.
[26] W. Sun, K. Lin, Y. Guan, "Performance Analysis of a Finite Duration Multichannel Delivery Method in IPTV," Broadcasting, IEEE Transactions on, 2008 vol. 54, pp. 419-429.
[27] E. Shihab, L. Cai, F. Wan, T. A. Gulliver, and N. Tin, "Wireless mesh networks for in-home IPTV distribution," IEEE Network, 2008, pp. 52-57.
[28] Moriaty Sandra and Everett Shu-Ling, "Commercial breaks: A viewing behaviour study," Journalism Quarterly, 71 (2) , 1994, pp.346-355.