International Journal of Advanced Computer Technology (IJACT)

ISSN:2319-7900

REDUCING IPTV CHANNEL CHANGE TIME BY ESTIMATING USER BEHAVIOR WITH RECENT-FIRST-ESTIMATOR VARIATION

Chuan-Ching Sue, Chi-Yu Hsu, Yu-Hsiang Su
Department of Computer Science and Information Engineering
National Cheng Kung University, Taiwan, R.O.C.

Abstract

This article proposes an estimation method called Bayes-RFE to estimate users' behavior including their surfing behavior and channel preference for IPTV service. The Proposed Bayes-RFE is more accurate and precise than previous works. In addition, the proposed prejoin strategy can adapt to satisfy more user's IPTV channel zapping behavior based on the higher hit rate.

Introduction

With the increasing of the network deployment, Internet Protocol TV (IPTV) has become a mature and hot application [1, 2]. IPTV provides higher quality video source and multiple services over the network, e.g. VoD, PVR, VoIP, interactive TV shopping, Internet access and so on. The linear TV service in the traditional TV network also be deployed in the IPTV. In order to promote the Quality of Experience (QoE), when customers are watching TV and changing the channels to reduce channel zapping/change time always is a challenging issue to satisfy various considerations based on new and future networking architecture, e.g. Fiber to the home (FTTH), and future network [4, 5].

Figure 1 shows the procedure of original channel change in IPTV, in which Command processing time: Started from that a user change channel by a remote control (RC), STB receives and resolves the signal from RC, and learns which channel to retrieve. Network delay time: STB sends IGMP Leave message to stop the transmission of old channel, and IGMP Join message to start the transmission of new channel. STB layer delay time: Once STB receives the packets of new channel, it de-capsulate them into the content. STB jitter buffer delay time: Fill the jitter buffer with the contents to the full in case of the adverse effect caused by the packet delay jitter. MPEG decoder time: Take out the frames (content) from jitter buffer, decode and display them; however, only I-frame can be decoded without any references, so an I-frame delay is needed to wait at usual.

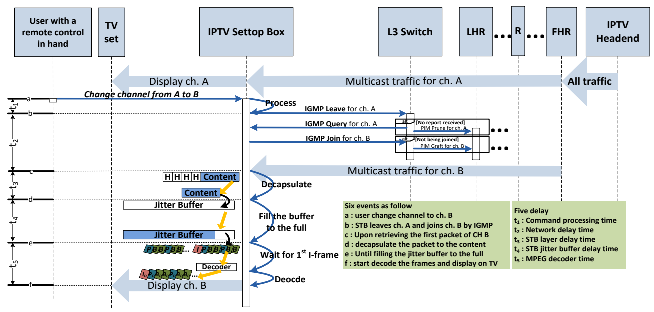

Figure 1. Procedure of original channel change

Table 1. Channel change delay factors

Channel change delay factor	Typical delay time (ms)[11]
Command processing time	25
Network delay time	50 / join, leave
STB layer delay time	~0
STB jitter buffer delay time	200
MPEG decoder time (mainly I-frame delay)	500

The typical delay time of those five delay time is shown in table. 1. According to the experiment result of R. Kooij et al., the acceptable channel change time is 0.43s, but the actual channel change time is far from that and cannot satisfy the users. Therefore, to reduce channel change time is necessary.

Related Works

A. Prejoin method

1. Prejoin adjacent channels

To reduce network delay time, *C.Cho et al.* considered using a Home Gateway (HG) to help STB prejoin adjacent channels [8]. As shown in figure 2, whenever STB requests to join the group of new channel (ch. A), home gateway sends not only IGMP join message for the new channel but also the adjacent channels (ch. A-1, ch. A+1). Therefore, the network delay time is reduced if the user change channel continuously.

ISSN:2319-7900

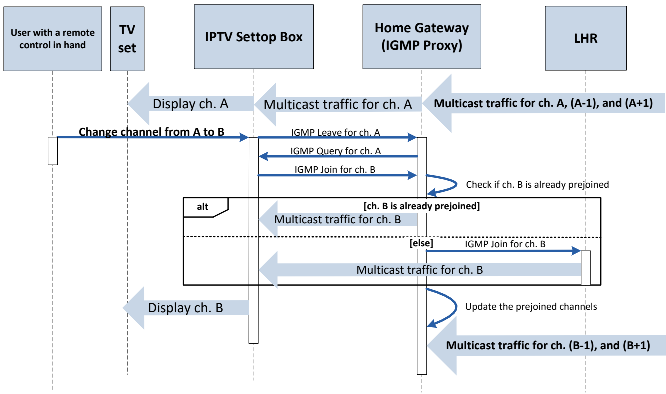

Figure 2. Sequence diagram of prejoining adjacent channels

2. Prejoin adjacent and popular channels

J. Lee et al. considered that the users may not just use up-down buttons to change channel continuously, they may also use other buttons to change channel randomly. As shown in figure 3, the authors used a rating server to help STB prejoin some popular channels in addition to adjacent channels. As a result, this method might reduce the network delay time no matter whether a user changes channel by up-down buttons or not [9].

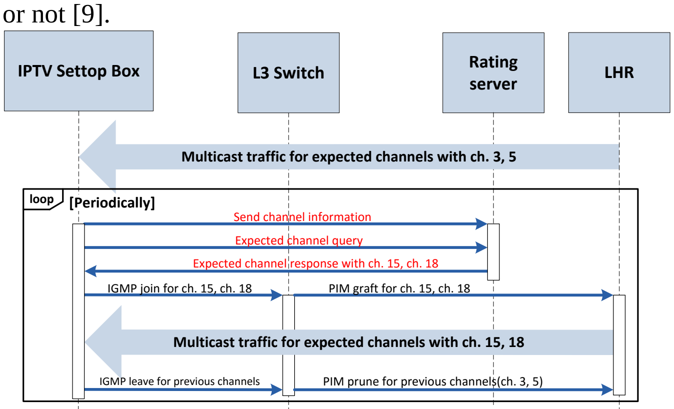

Figure 3. Sequence diagram of prejoining popular channels

3. Prejoin expected and preferred channels

Yuna Kim et al. assumed that users like to keep pushing the same button to change channels [10]. Based on this assumption, they classified the channel change buttons of remote control into 4 types, they are:

- **Up/Down**: Change to adjacent channels.
- **Toggle**: Change to previous number of channel.
- **Favorite**: Preset some favorite channels explicitly, and access these channels in succession.
- **Random**: Change to any channels.

The modeled remote control may look like figure 4. Because the user likes to keep pushing the same button, they use a state-transition diagram to record user's last pushed button shown in figure 5. Therefore, they made STB prejoin expected channel which a user keep pushing the same button to change to. However, if user's last pushed button are random (0~9), the new channel cannot be expected by that assumption. For this reason, they also made STB prejoin a preferred channel which a user likes most.

From their simulation results, the hit rate and bandwidth usage both performs better than previous works [8, 9], but the prerequisite is that their assumption is true.

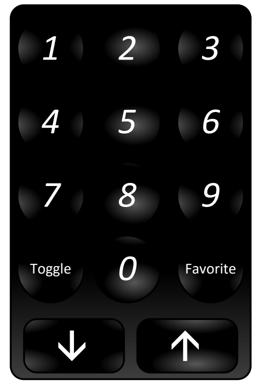

Figure 4. Remote control

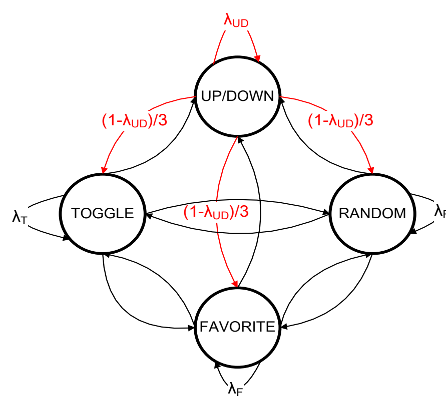

Figure 5. State transition diagram of user's surfing behavior

B. Join popular channels statically

Begić, Z. et al considered that most users like changing to popular channels, so popular channels had better be always replicated to DSLAM no matter whether any STBs request to join or not [11]. The authors called this kind of join method as static join, and called the original IP multicast join method as dynamic join to differentiate. As shown in figure 6, DSLAM would configure popular channels for static join and others for dynamic join. Consequently, the network delay time is reduced in part if the user changes to popular channels.

ISSN:2319-7900

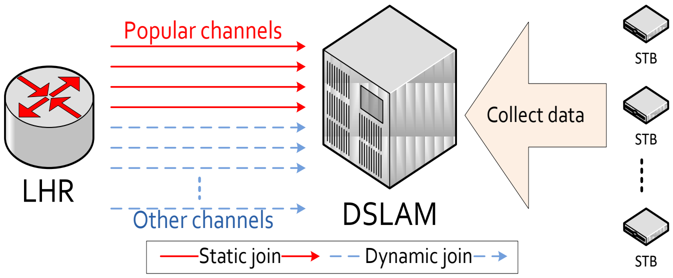

Figure 6. DSLAM configures popular channels as static join

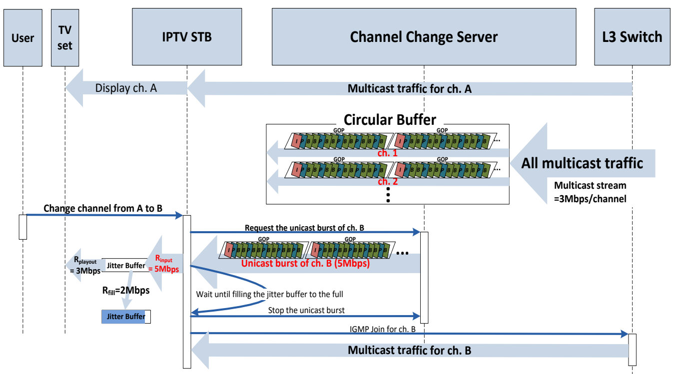

Figure 7. Use circular buffer and unicast burst to reduce channel change time

C. Use circular buffer and unicast burst

This method is different from the above-mentioned methods, it is considered to reduce jitter buffer delay and MPEG decoder time [12]. As shown in figure 7, to reduce MPEG decoder time, they used a circular buffer in a channel change server to prepare the recent content for each channel, so it ensures that the first frame that STB received is I-frame; to reduce jitter buffer delay, channel change server would send a high speed unicast burst to the STB upon channel change request arrived. Because the bitrate of unicast burst is larger than play-out rate, STB can start to decode the first frame without waiting to fill the jitter buffer. This method is good for reducing jitter buffer delay and MPEG decoder time, but it has much bandwidth requirement in either circular buffer or unicast burst.

D. Problem definitions

1. Delay vs. Bandwidth requirement

The first problem is the tradeoff of reduced delay and their bandwidth requirement. In the method of "circular buffer", it could reduce I-frame delay and Jitter buffer delay, but its bandwidth requirement is very high. As shown in figure 8, LHR always has to join the groups of all channels even though the contents are actually unpopular and nobody is watching them because they are used to be cached in the circular for preparation. The method of "static join" is on the contrary, it could only reduce network delay partially, but its bandwidth requirement is similar to original cost owing to the fact that almost all popular channels are watched channels. The prejoin method is in between, it could reduce network delay time fully, but its bandwidth requirement is usually high because all prejoined channels in LHR are usually partially overlapped watched channels.

Table 2. Delay and bandwidth requirement

Method	Delay type	Bandwidth requirement
(a)Prejoin	Network delay	Usually high
(b)Static join	Network delay	Similar to original cost
(c)Circular buffer	I-frame delay & Jitter buffer delay	Very high

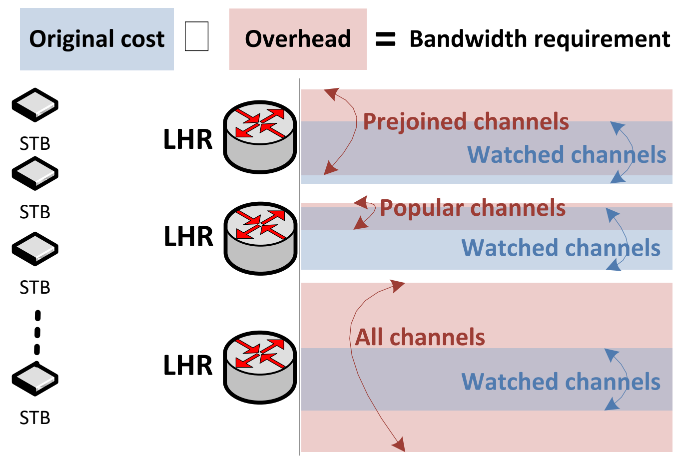

Figure 8. Comparison of bandwidth requirement

2. Previous prejoin methods are inefficient

The second problem is that the previous prejoin methods are inefficient since they made assumptions for user's behavior. For example, as shown in figure 9, the method of prejoining adjacent channels is only useful for the users who like up-down buttons. If the prejoined channels do not fit the user's behavior, then the bandwidth used for prejoin is wasted for nothing.

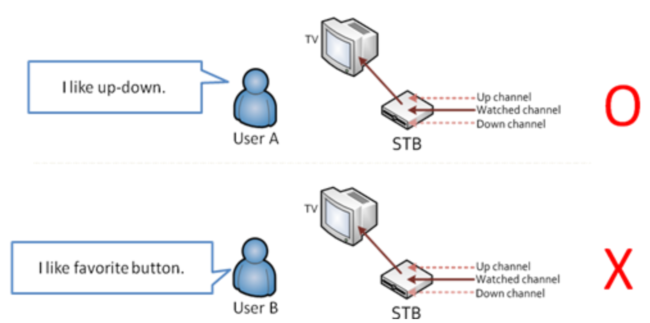

Figure 9. The method of prejoining adjacent method is only useful for the users who like up-down buttons

ISSN:2319-7900

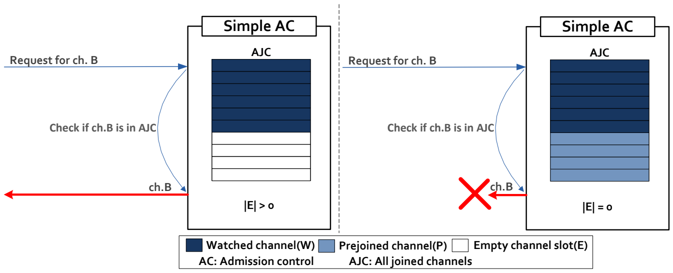

Figure 10. Prejoin method could increase the blocking rate of watched channel in bandwidth-limited environment

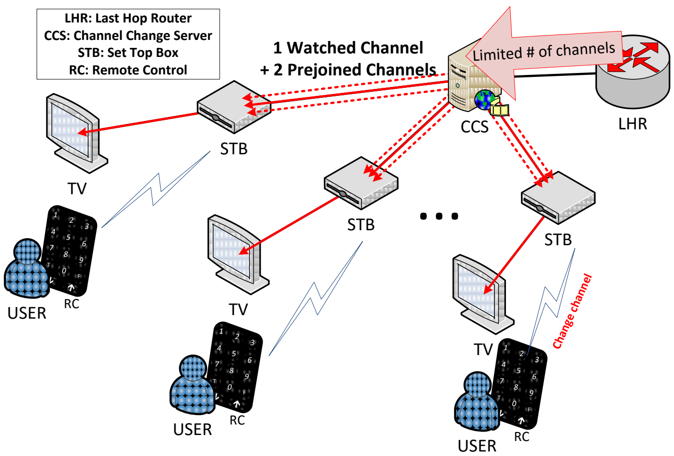

Figure 11. Considered network architecture

Proposed Mechanism

A. Purpose of the proposed mechanism

In previous prejoin methods, their prejoined channels are not used efficiently; besides, they did not consider the bandwidth requirements of those prejoined channels, so the bandwidth requirements are usually high. Therefore, we propose a method to improve the hit rate of prejoined channels in bandwidth-limited environment. Furthermore, we have to avoid increasing the blocking rate of watched channels owing to the prejoined channels as shown in figure 10.

B. Network architecture

Figure 11 shows the network architecture we considered. Two major components are channel change server (CCS) and set-top box (STB). STB is used to collect the user's data and predict the possible next channel, and CCS is used to make the most of the limited number of channel between LHR and CCS.

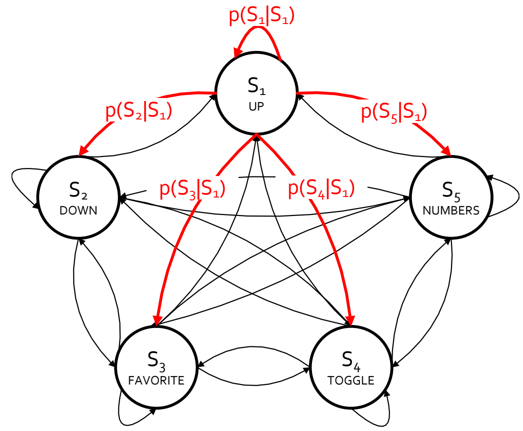

Figure 12. State transition diagram of our user's surfing behavior

C. Model of user behavior

In our proposed method, our model of user behavior is similar to the reference [10]. We also use the remote control as shown in figure 4, but we divide the channel change buttons into five categories which are up, down, favorite, toggle, numbers. Figure 12 shows the state transition diagram of our user's surfing behavior. Each state means the type of user's last pushed button, and the transition probability indicates user's surfing behavior. The surfing behavior contains 25 probability values, which are:

$$p\left(S_j \middle| S_i\right), \forall i, j = 1,2,3,4,5,$$

$$\text{and } \forall S_i \in \{S_1, S_2, S_3, S_4, S_5\}, \sum_{j=1}^{5} p\left(S_j \middle| S_i\right) = 1$$

If this paper can estimate user's surfing behavior, then user's next pushed button is predictable, and the most probable channel which user would like to change to is also expected further. However, the number buttons is an exception because more than one channel is possible if user change channels by number buttons next time. For this reason, we also need to estimate the channel preference of the user which means the degree of how a user likes the channels. The symbol of channel preference is defined as follows:

$$\text{Given } N \text{ channels,}$$

$$pCH_k, \forall k = 1,2,\ldots,N$$

$$\Rightarrow \sum_{k=1}^{N} pCH_k = 1$$

Therefore, to learn the most probable next channel, we have to estimate the surfing behavior and channel preference of the user in advance.

C. Estimation of user behavior

1. Problem definition

To simplify the description, this paper defines the estimation problem at first. As shown in figure 13, user behavior

ISSN:2319-7900

such as surfing behavior or channel preference can be seen as a population with *unknown probability distribution*, and we need to estimate the probability distribution by collecting the data which is generated by the population. Besides, the population $\boldsymbol{p}$ is assumed to be changeable because the user behavior usually changes over time. For example, $\boldsymbol{p}$ may be $[16/31 \quad 8/31 \quad 4/31 \quad 2/31 \quad 1/31]^T$ now, but it becomes $[1/31 \quad 2/31 \quad 4/31 \quad 8/31 \quad 16/31]^T$ soon.

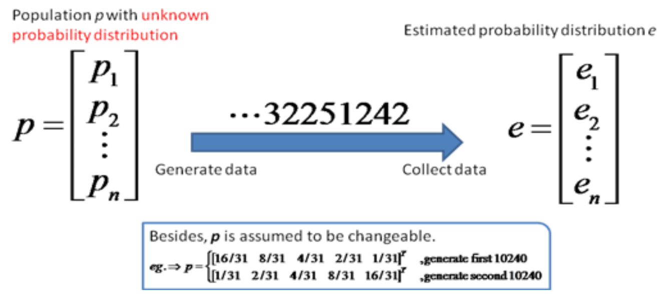

Figure 13. Problem definition of the estimation of user behavior

2. Average (AVG)

Estimating the probability distribution based on the frequency of each behavior may be the most common method, and we call this method Average (AVG). As shown in Figure 14, assumed that the possible generated data is from 1 to 5, this paper collects all the data, calculate the frequencies of each kind of behavior, and estimate them.

This paper executes an experiment to test whether this method is a qualified method to our estimation problem or not. In our experiment, the population $\boldsymbol{p}$ is not always the same, it looks like:

$$p = \begin{cases} [16/31 \quad 8/31 \quad 4/31 \quad 2/31 \quad 1/31]^T & \text{, generate first 10240} \\ [1/31 \quad 2/31 \quad 4/31 \quad 8/31 \quad 16/31]^T & \text{, generate second 10240} \end{cases}$$

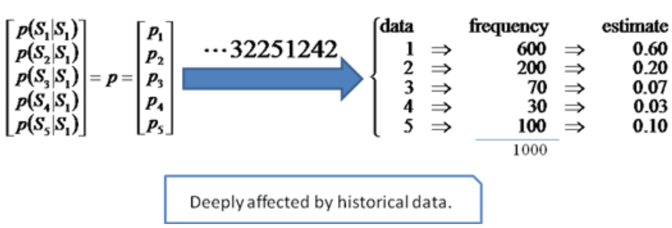

Figure 14. Estimation method of Average

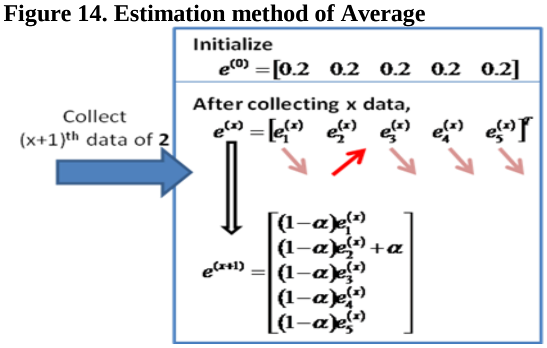

Figure 15. Estimation method of pure-RFE

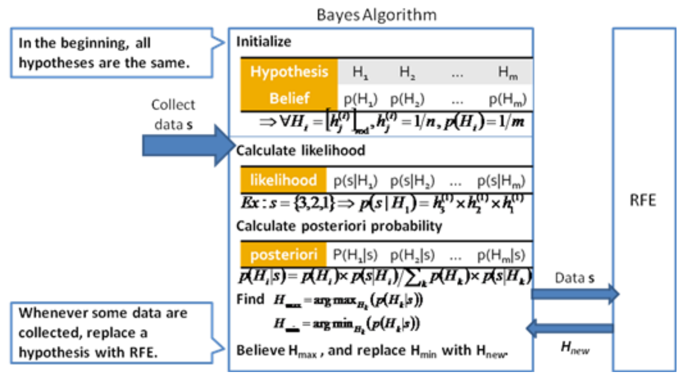

Figure 16. Estimation method of Bayes-RFE

The estimation result is shown in figure 17, and only the first population value (bar) and estimation value (line) are drawn. In addition, the experiment is done repeatedly 100 times; i.e., each estimated value contains the average and its standard deviation (stdev) in 100 tests. As we can see, this method is good until the population has changed; after that, it performs very poor. That is, this article needs another method which is more sensitive to the change of the population.

3. Recent-First-Estimator

Therefore, a method called Recent-First-Estimator (RFE) is proposed, and the operation of RFE is shown in figure 15. In the beginning, the estimated probability distribution is initialized to uniform distribution. After it collects data $\boldsymbol{r}$, RFE updates its probability distribution as follows:

$$e_i' = \begin{cases} e_i \times (1-\alpha) + \alpha & \text{, if } i = r \\ e_i \times (1-\alpha) & \text{, if } i \neq r \end{cases}$$

$$\text{, where } \alpha \in (0,1)$$

The estimation result is shown in figure 18. Obviously, RFE is sensitive to the change of the population; however, its estimated value tends to be unstable which results in its stdev of estimated value is always high.

4. Bayes-RFE

To reduce the unstable phenomenon of RFE, this paper combines RFE with Bayes algorithm and call it Bayes-RFE. In Bayes algorithm, it is used to pick out the hypothesis which is closest to the population from all possible hypotheses according to the collected data. In Bayes-RFE, we don't believe the estimated values of RFE directly, but we regard each one as a new hypothesis of Bayes algorithm, and let Bayes to pick out the hypothesis which is worthy to believe. The operation is shown in figure 16, and the estimation result is shwon in figure 19. As shown in the estimation result, the estimated value of Bayes-RFE is more stable when the

ISSN:2319-7900

collected data increase, and Bayes-RFE is also sensitive to the change of the population.

D. Our prejoin strategy

In this subsection, the proposed prejoin strategy based on the estimated user behavior including surfing behavior and channel preference is introduced. As shown in Fig. 20, we can predict the probability of changing to each channel (PEC) according to the user behavior which can be estimated by Bayes-RFE, so the most appropriate way may be to choose 2 channels with highest PEC and prejoin them.

However, our considered network architecture is a band-width-limited environment, so the prejoin channels decided only by STB may be blocked. As shown in figure 21, we use a CCS to collect the PECs of all STBs, and sum up the PECs as $PEC^{(total)}$ for each channel:

$$PEC^{(x)} = \left[PEC_1^{(x)} \quad PEC_2^{(x)} \quad \cdots \quad PEC_N^{(x)}\right]$$
$$\Rightarrow PEC^{(total)} = \left[PEC_1^{(total)} \quad PEC_2^{(total)} \quad \cdots \quad PEC_N^{(total)}\right],$$
$$\text{and } PEC_y^{(total)} = \sum_x PEC_y^{(x)}, \forall y = 1,2,\ldots,N$$

, where $PEC^{(x)}: PEC$ of the x^{th} STB

Whenever a user change channel, his STB would send the PEC to CCS, and CCS also update $PEC^{(total)}$ accordingly. Because the number of all joined channels (AJC) between LHR and CCS is limited, CCS would choose as many channels with highest $PEC^{(total)}$ as possible and prejoin them excluding the watched channels. After CCS updates AJC, it returns the AJC to the requesting STB. STB would choose 2 channels with its highest PEC from AJC.

Figure 22 shows the procedure of the channel change in our prejoin strategy. At first step, STB would decide what channel change result is. There exist 4 results in total, which includes HitOnSTB, HitOnCCS, Miss, and Block. HitOnSTB means the new channel resides on STB, and HitOnCCS means the new channel does not reside on STB but on CCS. If the new channel resides neither on STB nor CCS, we check whether the watched channels have occupied AJC or not. If no, the new channel preempts one of prejoined channels in AJC, and the result is Miss; however, if yes, the result is Block which means the new channel (watched channel) is blocked. At the second step, STB updates its prejoined channels, which is described in previous paragraph.

Simulation

A. Parameter definition

In this subsection, the parameters is defined in table. 3, in which the parameter definitions and their settings, and this paper introduces them as follows:

Table 3. Parameter definition and setting

Parameter	Definition	Default
$p(S_j \mid S_i)$	Surfing behavior	nearGeo(0.5, 5)
N_{ch}	Channel count	1000
pCH_k	Channel preference	Zipf(1.0, 1000)
$N_{maxChInCCS}$	Max joined channel count in CCS	300
N_{usr}	User count	
p_{hit}	Hit rate	
p_{block}	Blocking rate	

1. Channel count N_{ch} and max joined channel count in CCS $N_{maxChInCSS}$, where N_{ch} is the total channel count, and its value is set as 1000, but the max joined channel count in CCS $N_{maxChInCSS}$ is only 300.
2. Surfing behavior $p(S_j|S_i)$, where $p(S_j|S_i)$ is the user's surfing behavior, and it is generated by a near-Geometric distribution nearGeo(p, M), which is:
$$g_X'(x; p, M) = \frac{(1-p)^{x-1} p}{1-(1-p)^M}$$
$$, \forall p \in (0,1), x = 1,2,\ldots,M$$
, where the surfing behavior is set as nearGeo(0.5, 5) to simulate, and nearGeo(0.5, 5) is equivalent to the probability distribution of $[16/31 \quad 8/31 \quad 4/31 \quad 2/31 \quad 1/31]$.
3. Channel preference pCH_k, where pCH_k is the channel preference, and it is set to follow Zipf's law Zipf(α, N), which is:
$$\because z_X(x;\alpha,N) \propto \frac{1}{x^\alpha}$$
$$\Rightarrow \quad z_X(x;\alpha,N) = \frac{1/x^\alpha}{\sum_{n=1}^{N}\left(1/n^\alpha\right)}$$
$$, \forall \alpha \geq 0, x = 1,2,\ldots,N.$$
4. User count N_{usr}, where N_{usr} is the user count.
5. Hit rate p_{hit} and blocking rate p_{block}, where p_{hit} is the hit rate of the prejoined channels, and p_{block} is the blocking rate of the watched channels.

B. Simulation of one user

In this subsection, this paper compares the hit rate of different methods without consideration of limited bandwidth. The symbols of the methods used to compare are listed in table 4. STB_MostProbK and STB_MostProbE are our proposed prejoin strategy, and K means STB knows user behavior in advance but E means STB estimates user behavior by

ISSN:2319-7900

Bayes-RFE. STB_Adj, STB_AdjPop, and STB_ExpPref are the methods proposed by references [8-10].

Table 4. Symbols used in the simulation of one user

Symbol	Definition
STB_MostProbK	STB **knows** user's surfing behavior and channel preference, and prejoins the most probable 2 channels.
STB_MostProbE	STB **estimates** user's surfing behavior and channel preference in advance, and prejoins the most probable 2 channels.
STB_Adj	STB prejoins 2 adjacent channels.[8]
STB_AdjPop	STB prejoins 2 adjacent channels and 1 popular channel.[9]
STB_ExpPref	STB prejoins 1 expected channels and 1 preferred channel.[10]

The simulation results are shows in figure 23-25, all x-axes are channel change count and all y-axes are the hit rate. In figure 23, the simulated user is randomly selected, which means his surfing behavior and channel preference are the default setting described in table 3. The hit rate of the proposed prejoin strategy (STB_MostProbK and STB_MostProbE) is always higher than others due to the fact that our prejoin strategy is closer to user behavior. Besides, the hit rate of STB_MostProbE is more and more close to that of STB_MostProbK as the channel change count increases because the estimation of user behavior is more and more accurate.

To compare the hit rate on the assumptions of the previous works, we do another two simulations. The first one is to generate a user who prefers changing channels by up-down buttons, which the surfing behavior is set as:

$$\forall S_i, p(S_j|S_i) = \begin{cases} 0.35 & ,\forall S_j = S_1, S_2 \\ 0.1 & ,\text{otherwise} \end{cases}$$

The simulation result is shown in figure 24. As the expected result, to prejoin adjacent channels is a good choice for this kind of users. In the proposed prejoin strategy STB_MostProbK, its hit rate is always the same as STB_Adj, so it explains our prejoin strategy is effective. A similar simulation performed in which the surfing behavior is defnied as:

$$\forall S_i \in \{S_1, S_2, S_3, S_4\} \quad , p(S_j|S_i) = \begin{cases} 0.60 & ,\forall S_j = S_i \\ 0.25 & ,\forall S_j = S_5 \\ 0.05 & ,otherwise \end{cases}$$

$$\forall S_i \in \{S_5\} \quad , p(S_j|S_i) = \begin{cases} 0.60 & ,\forall S_j = S_i \\ 0.10 & ,otherwise \end{cases}$$

This simulation result is shown in figure 25 and STB_ExpPref has higher hit rate than STB_Adj and STB_AdjPop as expected.

Table 5. Symbols used in many users

Symbol	Definition
STB_MostProb_AC	STB prejoins the most probable 2 channels and has an AC nearby.
STB_MostProb_CCS	STB prejoins the most probable 2 channels and has a CCS nearby.
STB_Adj_AC	STB prejoins 2 adjacent channels and has an AC nearby.[8]
STB_AdjPop_AC	STB prejoins 2 adjacent channels and 1 popular channel and has an AC nearby.[9]
STB_ExpPref_AC	STB prejoins 1 expected channels and 1 preferred channel and has an AC nearby. [10]
STB_None_AC	STB prejoins nothing and has an AC nearby.

B. Simulation of many users

Finally, this paper executes a simulation in a bandwidth-limited environment to compare the hit rate and blocking rate of different methods. In this simulation, all the simulated methods are listed in Table. 5. In our proposed prejoin strategy, this paper adopts a CCS to manage all joined channels and prevent increasing blocking rate owing to some channels joined only for prejoin. Therefore, this article also uses a simple admission control (AC) to control the max joined channels for other prejoin methods. As shown in figure 26, CCS allows the requests of watched channels as long as AJC is not occupied by existing watched channels; however, AC treats each request by FCFS (First-Come-First-Server), so it rejects the request of watched channels even though some channels in AJC are only for prejoin.

In the simulation, this paper restricts the max number of all joined channels to **300** in AC or CCS. Figure 27 shows the total number of watched and prejoined channels of all STBs. Three hundred channels of all methods except for STB_None_AC are occupied by the watched and prejoined channels in only 200~300 users. For AC, the next request of watched or prejoined channel would be rejected if AJC is occupied and the requested channel is not in AJC, so it is expected that hit rate decreases and blocking rate increases. Figure 28 and figure 29 show the hit rate and blocking rate of all STBs in all methods. Our prejoin strategy, STB_MostProb_CCS, has highest hit rate and the same blocking rate as STB_None_AC because each STB always can prejoin 2 channels (for chosen from AJC) and CCS gives higher priority to the request of watched channels than prejoined channels. However, the hit rate still decreases as the user count decreases because less and less channels in AJC can be used for prejoin and STB is forced to choose the channels with lower PEC to prejoin. Finally, STB_MostProb_AC is used to compare with STB_MostProb_CCS. STB_MostProb_AC always choose 2 channels with highest PEC to prejoin but has lower hit rate

ISSN:2319-7900

than STB_MostProb_CCS, which is because the request of prejoined channels of STB_MostProb_AC may be rejected due to AJC being occupied.

Conclusions

In this paper, an estimation method called Bayes-RFE is proposed to estimate users' behavior including their surfing behavior and channel preference. As shown in the experiment results (figure 17-19), Bayes-RFE is more accurate and precise with more data collected based on the proposed prejoin strategy to reduce IPTV channel change time. As shown in results, the proposed prejoin strategy can adapt to satisfy more user's behaviors; besides, it has higher hit rate than previous works.

Acknowledgments

This work was supported in part by the Ministry of Science and Technology, Taiwan, R.O.C., under Grant MOST 104-2221-E-006-038.

References

[1] ITU-T FG IPTV-OD-0001, *1st FG IPTV meeting*, Geneva, 10-14 July 2006

[2] Recommendation ITU-T Y.1910 (09/2008), "IPTV functional architecture," *ITU-T, 2008*.

[3] Kwang-Jae Kim, et al., "Analysis of key features in IPTV service quality model," *Industrial Engineering and Engineering Management, 2008. IEEM 2008. IEEE International Conference on*, pp.595-598, 8-11 Dec. 2008

[4] ITU-T FG IPTV-IL-0050. (2007) *Definition of Quality of Experience (QoE)*. ITU-T.

[5] Wikipedia, Quality of Experience (2010), http://en.wikipedia.org/wiki/Quality_of_experience.

[6] Hyun Jong Kim, et al., "The QoE Evaluation Method through the QoS-QoE Correlation Model," *Networked Computing and Advanced Information Management, 2008. NCM '08. Fourth International Conference on*, vol.2, pp.719-725, 2-4 Sept. 2008

[7] R. Kooij, K. Ahmed, and K. Brunnström, "Perceived quality of channel zapping," in *IASTED Int. Conf. Communication Systems and Networks*, 2006.

[8] Chunglae Cho, et al., "Improvement of channel zapping time in IPTV services using the adjacent groups join-leave method," *Advanced Communication Technology, 2004. The 6th International Conference on* , vol.2, pp. 971- 975, 2004

[9] J. Lee, et al., "Advanced Scheme to Reduce IPTV Channel Zapping Time," *LNCS 4773*, 235-243, 2007

[10] Yuna Kim, et al., "Reducing IPTV channel zapping time based on viewer's surfing behavior and preference," *Broadband Multimedia Systems and Broadcasting, 2008 IEEE International Symposium on* , pp.1-6, March 31 2008-April 2 2008

[11] Begić, Z.; Bolić, M.; Bajrić, H., "Effect of multicast on IPTV channel change performance," *ELMAR, 2008. 50th International Symposium* , vol.1, pp.151-155, 10-12 Sept. 2008

[12] Degrande, N., et al., "Increasing the user perceived quality for IPTV services," *Communications Magazine, IEEE* , vol.46, no.2, pp.94-100, February 2008

[13] W. Fenner, "Internet Group Management Protocol, Version 2," IETF, *RFC2236*, Nov. 1997

[14] B. Cain, et al., "Internet Group Management Protocol, Version 3," IETF, *RFC3376*, Oct. 2002

[15] B. Fenner, et al., "Protocol In-dependent Multicast-Sparse Mode (PIM-SM): Protocol Specification (Revised)," IETF, *RFC 4601*, August 2006.

[16] S. H. Hsu, et al., "AIMED – A Personalized TV Recommendation System," LNCS 4471, 166-174, 2007.

[17] Smith, D.E., "IP TV Bandwidth Demand: Multicast and Channel Surfing," *INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE*, pp.2546-2550, 6-12 May 2007

[18] Tony Lancaster, "An Introduction to Modern Bayesian Econometrics," *Wiley-Blackwell*, 2004.

Biographies

CHUAN-CHING SUE became a Member (M) of the IEEE in 1998, and received a B.S. and Ph.D. in electrical engineering from National Taiwan University, Taipei, Taiwan, R.O.C., in 1994 and 2001, respectively. He is currently a Full Professor in the Department of Computer Science and Information Engineering, National Cheng Kung University, Tainan, Taiwan, R.O.C. His Current research interests are fault-tolerant WDM networks and high-speed computer networks. Professor Sue may be reached at suecc@mail.ncku.edu.tw.

CHI-YU HSU received the B.S. and M.S. degree in Department of Management Information System from National Pingtung University of Technology and Science, Pingtung, Taiwan, R.O.C. Currently, he is a Ph.D. candidate in Department of Computer Science and Information Engineering from National Cheng Kung University, Tainan, Taiwan, R.O.C. He also is a founder of the Sunshine Tech. R&D Co., Ltd. and WoT Technology Inc. His research areas include Internet of Things, Web technology, Property Management System, and System Integration. Mr. Chi-Yu Hsu may be reached at cyhsu@stech.tw.

YU-HSIANG SU received B.S. and M.S. in Computer Science and Information Engineering from National Cheng Kung University, Tainan, Taiwan, R.O.C. His research focuses on fast channel change in IPTV networks. Mr. Yu Hsiang Su may be reached at tqsmm628@gmail.com.