Fast Channel Zapping with Destination-Oriented Multicast for IP Video Delivery

Xiaohua Tian, *Student Member, IEEE,* Yu Cheng, *Senior Member, IEEE,*
and Xuemin (Sherman) Shen, *Fellow, IEEE*

Abstract—Channel zapping time is a critical quality of experience (QoE) metric for IP-based video delivery systems such as IPTV. An interesting zapping acceleration scheme based on time-shifted sub-channels (TSS) was recently proposed, which can ensure a zapping delay bound as well as maintain the picture quality during zapping. However, the behaviors of the TSS-based scheme have not been fully studied yet. Furthermore, the existing TSS-based implementation adopts the traditional IP multicast, which is not scalable for a large-scale distributed system. Corresponding to such issues, this paper makes contributions in two aspects. First, we resort to theoretical analysis to understand the fundamental properties of the TSS-based service model. We show that there exists an optimal sub-channel data rate which minimizes the redundant traffic transmitted over sub-channels. Moreover, we reveal a *start-up effect*, where the existing operation pattern in the TSS-based model could violate the zapping delay bound. With a solution proposed to resolve the start-up effect, we rigorously prove that a zapping delay bound equal to the sub-channel time shift is guaranteed by the updated TSS-based model. Second, we propose a destination-oriented-multicast (DOM) assisted zapping acceleration (DAZA) scheme for a scalable TSS-based implementation, where a subscriber can seamlessly migrate from a sub-channel to the main channel after zapping without any control message exchange over the network. Moreover, the sub-channel selection in DAZA is independent of the zapping request signaling delay, resulting in improved robustness and reduced messaging overhead in a distributed environment. We implement DAZA in ns-2 and multicast an MPEG-4 video stream over a practical network topology. Extensive simulation results are presented to demonstrate the validity of our analysis and DAZA scheme.

Index Terms—IPTV, channel zapping, time-shifted sub-channels, destination oriented multicast.

1 Introduction

The broad adoption of Internet Protocol (IP) as a standard for digital transmission is revolutionizing the traditional way of video delivery. Typical applications such as IPTV and live video streaming have become increasingly popular over current Internet. To efficiently utilize the bandwidth resources of the network in these applications, the video is usually compressed with media coding schemes and then delivered only to subscribers who request it. However, the widely used media coding schemes such as MPEG-2 or MPEG-4/H.264 [1] divide video stream into segments, which could incur unacceptable channel zapping delay. For data delivery, multicast is considered the most efficient paradigm for these applications, but the scalability issue of traditional IP multicast hinders the deployment of the large-scale video delivery system with numerous channels. In this paper, we propose a synergetic approach to accelerate the channel zapping for IP video delivery, which incorporates the fast channel zapping algorithm and the scalable underlying multicast scheme. The generic approach can be applied to any video delivery application that delivers MPEG video with multicast over the IP-based network. For the convenience of demonstration, we will study the IPTV system as an example in this paper.

- *Xiaohua Tian was with the Department of Electrical and Computer Engineering, Illinois Institute of Technology, Chicago, IL, 60616, and is now with Electronic Engineering Department, Shanghai Jiao Tong University, Shanghai, China, 200240. E-mail: xtian@sjtu.edu.cn.*
- *Yu Cheng is with the Department of Electrical and Computer Engineering, Illinois Institute of Technology, Chicago, IL, 60616. E-mail: cheng@iit.edu.*
- *Xuemin (Sherman) Shen is with Department of Electrical and Computer Engineering, University of Waterloo, Waterloo Ontario N2L 3G1 Canada. E-mail: xshen@bbcr.uwaterloo.ca.*

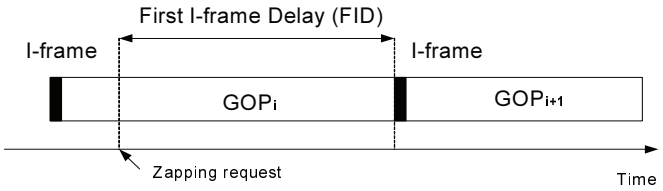


Fig. 1. First I-frame delay in channel zapping.

IPTV is a kind of distributed system that delivers the TV program as a compressed multicast video stream over IP-based network [1], where the stream is divided into groups of pictures (GOPs) and play-out can only start with an I-frame at the beginning of each GOP. The IPTV channel zapping is the act of leaving a stream and joining in another. The time it takes for the new TV program to start playing from the time a request to change to that channel happens is *zapping time*, which is a critical quality of experience (QoE) metric for IPTV systems. As illustrated in Fig. 1, the zapping request possibly occurs at any time in a GOP of the new channel stream, the time between the arrival of the request and the first I-frame (*first I-frame Delay, FID*) could be up to a few seconds, which is a significant contributor to the channel zapping time [2].

A number of techniques have been proposed recently to mitigate the zapping time in the context of IPTV, and most of them are realized with the auxiliary stream that starts

Digital Object Indentifier 10.1109/TPDS.2012.94 1045-9219/12/$31.00 © 2012 IEEE

with an I-frame. The set-top-box (STB) trying to join in a new channel first subscribes to the auxiliary stream, and then migrates to the main multicast stream when enough data are accumulated in the STB play-out buffer. There have been four basic approaches to exploit the auxiliary stream.

- Approach 1: Unicast a full-quality boost stream replicated from the previously stored stream data for each zapping request [1], [3], [4]. The approach can impose significant resource demands on networks and the streaming server.
- Approach 2: Generate the low-quality stream, which is composed of just I-frames [5] or several low-resolution channels [7], [8], to accompany the regular channel stream. Such mechanisms may incur noticeable picture inconsistency at each channel zapping act.
- Approach 3: Join additional multicast groups along with the current channel group, where the additional one could be simply the adjacent multicast group to the current channel [9], or the most likely next channel predicted by users' channel selection behaviors [10], [11]. These solutions require extra streams delivered to the home gateway for each channel being watched, thus incur expensive bandwidth cost in the access network.
- Approach 4: Encode a low-quality auxiliary stream with frequent I-frames into the regular stream for each channel [13], [14]; however, the STB has to be equipped with extra codec and the video has to be recoded.

The most closely related work to our study is multicast assisted zap acceleration (MAZA) scheme [2], which is based on the time-shifted sub-channels (TSS). In MAZA, a video channel is accompanied with several sub-channels that are spaced by T time units, and each sub-channel is a full-quality replica of the main channel media stream. The TSS-based scheme can guarantee a bounded FID, and provide the following advantages over existing solutions. First, the sub-channel stream is delivered through multicast to subscribers upon request, thus consumes less bandwidth resources in contrast to the unicasting boost stream or additional group pre-joining method. Second, there is no picture inconsistency incurred by the low-quality accompanying stream during the transition from the sub-channel to the main channel. Third, there is no new codec introduced to accommodate the auxiliary stream encoded into the regular channel. We summarize major features of zapping acceleration solutions mentioned above in Table 1.

Despite the remarkable advantages, the fundamental properties and performance of the TSS-based service model are not yet well understood from a theoretical perspective. Moreover, the current implementation of TSS-based model utilizes traditional IP multicast, with each channel and its associated sub-channels implemented as separate multicast groups. The per-group based multicasting is well-known as suffering from the scalability issue in a large distributed system, and in particular can incur frequent control messages for the TSS-based channel zapping (to be demonstrated in Section 4.2).

In this paper, we first resort to rigorous theoretical analysis to understand the fundamental properties of TSS-based service model. Specifically, we show that there exists an optimal sub-channel data rate which minimizes the redundant traffic transmitted over sub-channels. We reveal the *start-up effect*, by which the basic sub-channel turn-on policies in [2] could violate the FID bound, and thus introduce augmented sub-channel turn-on policies for guaranteed FID performance. Moreover, we give a rigorous proof that the TSS-based model could ensure a FID bound of T under the augmented turn-on policies. As the second main contribution, we propose a destination-oriented-multicast (DOM) assisted zapping acceleration (DAZA) scheme to implement the TSS-based service model. Our previous work DOM [26], [27] makes each packet carry explicit destinations of receivers, instead of an implicit group address as in IP multicast, to facilitate multicast forwarding. DOM can be readily integrated with the TSS-based model to achieve the DAZA scheme, where the migration from the sub-channel to the main channel can be done seamlessly, incurring no control message exchange over the network. The sub-channel selection is simply accomplished at the data source node, independent of the zapping request signaling time, thus resulting in improved robustness and reduced messaging overhead of DAZA. We implement DAZA in ns-2 and multicast a MPEG-4 video stream over a practical network topology. Extensive simulation results are presented to demonstrate the validity of our analysis and DAZA scheme.

This paper provides a special perspective on designing the channel zapping acceleration scheme for the IP video delivery system, with IPTV as an example. We show that more performance gains can be obtained with the synergetic design, compared with simply relying on the application-layer zapping acceleration mechanism. Specifically, the DOM not only provides a scalable transportation scheme, but also accommodates the features of the TSS-based service model. The coordination between the two schemes outperforms the simple addition of the two, which may shed light on general design guidance for the IP video delivery system.

The remainder of this paper is organized as follows. In Section 2, we briefly describe the process of channel zapping and principles of MAZA and DOM. Section 3 presents the theoretical analysis of the TSS-based service model. Section 4 describes the design and advantages of DAZA. Section 5 presents the simulation results. Section 6 gives the conclusion remarks.

2 PRELIMINARY

In this section, we first describe the channel zapping process in the IP video delivery system, using IPTV as an example. Various factors contributing to the zapping delay are analyzed, among which the FID is a significant one. We then overview the principles of MAZA mechanism, the most related to our work, which reduces the zapping time by constraining the FID. The basic idea of DOM is also discussed, which will be applied in the DAZA design later to improve the performance of MAZA.

2.1 Channel Zapping Delay

The major process of channel zapping in a typical IPTV access network architecture is illustrated in Fig. 2 [16], [17], [19],

TABLE 1
Features comparison of zapping acceleration solutions

Features	Approach 1	Approach 2	Approach 3	Approach 4	TSS-based Model
Server burden	High	Medium	Medium	Medium	Low
Network burden	High	Medium	High	Medium	Low
Picture inconsistency	No	Yes	No	No	No
New codec	No	No	No	Yes	No

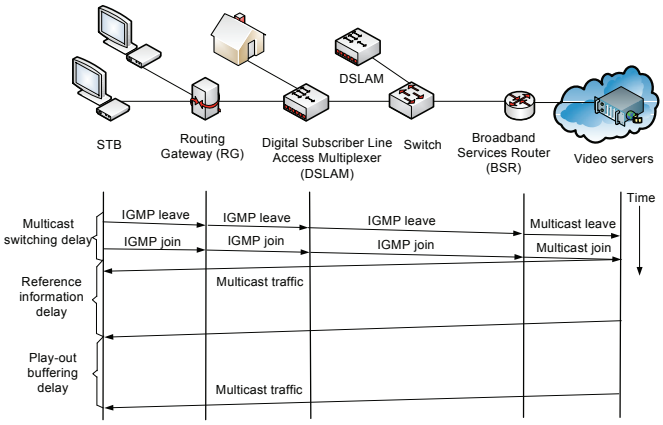


Fig. 2. Channel zapping delay.

[22]–[24]. When a user requests to switch to a new channel, the STB requests to leave the current multicast channel and join in the new one through the Internet Group Management Protocol (IGMP) signaling [30]–[32]. The broadband service router (BSR) is the IGMP router, which will establish/prune the tree branches for multicast groups according to the IGMP information.

In practice, when the multicast stream reaches the STB, the streaming data is encoded in some container format and can not be directly decoded by the decoder [17]. The receiver has to acquire and parse certain reference information before it can process the multicast traffic. The reference information includes control information and the I-frame. The control information must be first obtained, with which the receiver knows how to decode the data from the container format and find the audio/video elements or security related information for the multicast stream [16]–[18]. With the obtained element video stream, the receiver has to wait until the appearance of the first I-frame before the decoder can recover pictures from the data stream. The reference acquisition process must be finished first, and then the generated decoding friendly streaming data can be available for play-out buffering, where the network jitter is smoothed out and some application-specific functionalities are implemented [16], [17].

The time interval from the IGMP leave message sent out to the STB joining in the new channel is typically short and not the primary factor of the channel zapping time [2], [16], [17], [19], [21]. The buffering delay is decoder-implementation specific and unavoidable for decoders. In the reference information delay, the time for acquiring the I-frame is a major contributor [1], [2], [5], [16]. The video object is normally encoded as series of GOPs [1], with each GOP starts with an I-frame. An I-frame contains the representation of an entire picture, and the receiver can recover a GOP only after the corresponding I-frame is acquired. GOP durations can be up to a few seconds and the joining request for a zapping act can arrive right after the transmission of the starting I-frame, thus the time between the arrival of the joining request and the next I-frame (i.e., FID) can be up to a few seconds, which is the dominant channel zapping time contributor solvable through networking techniques [1], [2], [5], [16].

2.2 MAZA

2.2.1 TSS-based Service Model of MAZA

In order to constrain the channel zapping time by reducing the FID, the TSS-based model was recently proposed in MAZA scheme [2]. Specifically, each video channel is accompanied by several time-shifted sub-channels (sCHs) that are generated through replicating the main channel (mCH) media stream as illustrated in Fig. 3. There are $X = \lceil \frac{s}{T} \rceil$[1] sCHs coexisting with the mCH, where T is the time space between two adjacent sCHs and s is the size (in time units) of the largest GOP in the video stream. Every T time units an I-frame will be sent over a sCH. At any time, a joining STB can always find an appropriate sCH, which will send an I-frame within T time units. With TSS-based model, the observed FID can be bounded by T.

However, if sCHs are always active, they will consume network resources continuously. To save network resources, the TSS-based model provides the following two desirable functionalities. First, a user joins a sCH will migrate to the mCH as soon as possible, and any sCH on which no users are listening can be deactivated. Second, a sCH transmits data only when there is a user who will need its service.

For the first functionality, the sCH intuitively only needs to be configured a higher data rate considering mCH's head start. The sCH subscribers can change to join in the mCH after the sCH catches up with the mCH, and the sCH can be deactivated then. The problem is, if the X sCHs are deactivated one after another, there will be only mCH in the system, and the FID of new joining subscribers can not be bounded. The TSS-based model adopts dynamic sCH operations to deal with this issue. Imagine that there are infinite number of sCHs that are spaced by T and configured a higher data rate than the mCH. As we only need X sCHs to guarantee the FID bound, the rest of the sCHs can be imagined as invisible. Suppose the sCH 1 in Fig. 3 catches up with the mCH, we could make sCH 3 visible in order that the new comers could be taken care of by sCH 2 and 3. In this way, there are always X sCHs to ensure the

1. When $\frac{s}{T}$ is an integer, the $\frac{s}{T}$th sCH lags behind the mCH by exactly a GOP and align to the mCH at an I-frame. However, this sCH is necessary in order to guarantee the delay bound of T, to be proved in Section 3.4.

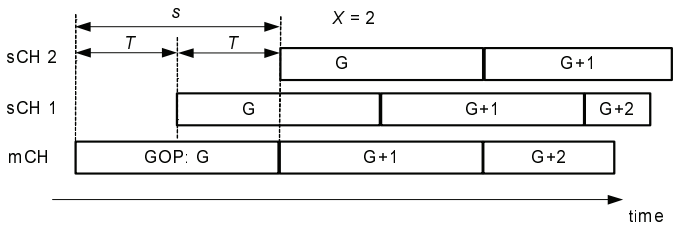


Fig. 3. TSS-based service model.

T-shift property and sCH subscribers will eventually migrate to the mCH.

For the second functionality, TSS-based model does not need sCHs to transmit data all the time. In fact, only some meta information about sCHs needs to be maintained. For example, which sCHs should be visible? Which subscribers are listening on a given sCH? Which part of the media stream should be delivered over a sCH that has users listening on? With these information, TSS-based model can always duplicate correct part of the streaming data from the mCH and dispatch them to corresponding sCH users only when requested.

2.2.2 IP-Multicast based Implementation of MAZA

MAZA implements the TSS-based service model in the streaming server called zapping accelerator (ZA), which delivers the streaming data to STBs with IP multicast. In MAZA, each mCH or sCH is assigned an IP multicast address, and the STB is pre-configured to associate the IP multicast address with the mCH or sCH index. There is a *Meta-Channel* in ZA broadcasting to all STBs in advance, which specifies the sCH offering the earliest I-frame for every GOP. After the STB receives the meta-channel information, it can decide which sCH to join in, with considering the time it takes for the zapping request to reach ZA (signaling delay). When the chosen sCH catches up with mCH, the ZA notifies the STB to migrate to mCH, and continues keeping the sCH on to have enough time for the STB to join in mCH. The IP-multicast based implementation can be affected by the signaling delay and incur frequent control messages. These issues will be eliminated with the facilitation of DOM. The basic principle of DOM is introduced below.

2.3 DOM

Our previous work DOM [26], [27] provides a scalable multicast service. With DOM [26], [27], the subscriber sends joining request to the data source server; the server aggregates the requests for the same multicasting group, and then creates each group a list of subscribers' IP addresses. The multicasting packet will carry this addresses list to facilitate the data forwarding.

When receiving a multicast packet, each router will extract the addresses list from the packet. With the list, the DOM router will check its IP routing table to determine the output interface to each listed address and make necessary aggregation. Specifically, a DOM router performs the following processing: first, check the unicast routing table to determine the output interface for each address listed, and aggregate addresses with the same output interface into a set; second, replicate the packet for each unique interface found in the first step; third, update the addresses list of each packet copy with the aggregated set yielded in the first step, so that the packet copy for a given interface contains only the destination addresses that can be reached via this interface. By removing unnecessary addresses from the list, the downstream router will not generate unnecessary packet copies for those destinations that have been delivered over other sibling subtrees.

With DOM, the messaging and storage overhead for multicast tree management is independent of the number of groups being supported by each node; therefore, the scalability is significantly improved compared with traditional IP multicast. In practice, we resort to the bloom filter technique [28], [29] to encode the addresses list carried by each packet for space efficiency, with elaborated design to accommodate the features of the practical Internet including the longest-prefix matching, route aggregation and asymmetric inter-domain routing [26], [27]. Our proposed DAZA scheme is to incorporates the TSS-based service model with DOM, which brings remarkable performance benefits for IP video delivery systems.

3 Theoretical Analysis of the TSS-based Model

This section presents a fundamental theoretical study of the TSS-based service model. We first summarize in a more understandable manner the basic operational properties of the TSS-based model according to the results in [2]. By examining the lifetime of a sCH, we reveal that an optimal sCH data rate exists, which leads to the least amount of redundant traffic. We then investigate the start-up effect, by which the original channel turn-on policies in [2] could violate the FID bound. We thus propose augmented channel turn-on policies for guaranteed FID performance. Moreover, we give a rigorous proof that the TSS-based model could ensure a FID bound of T under the augmented turn-on policies. The main notations in the paper are summarized in Table 2.

3.1 Operation Pattern of TSS-based Model

The essence of the TSS-based system is to maintain the proper time-shifting relationship among sCHs, considering the dynamics due to channel merging and creation. In accordance with [2], the core operation pattern is the sCH turn-on policies.

sCH Turn-on Policies[2]:

1) *For the first X sCHs, they are created and turned on one after another spaced by T*[3].
2) *For $i > X$, sCH i is turned on when sCH $i - X$ catches up with mCH.*

Specifically, the creation and turn-on times of the first X sCHs are the same, which conserve the T-shift property,

$$C_i = C_m + iT \quad 1 \leq i \leq X, \tag{1}$$

2. See Section IV-C of [2].

3. Upon created/turned on, the sCHs do not necessarily transmit data. Only the timing relationship is maintained. A sCH will physically transmit data only when it has subscribers.

TABLE 2
Main notations

FID	First I-frame delay.
mCH	Main channel.
sCH	Sub-channel.
s	The duration of the largest GOP in the video.
T	The time shift between two successive sub-channels.
X	The number of admissible sCHs, $X = \lceil s/T \rceil$.
C_m	Creation time of the main channel.
C_i	Creation time of sub-channel i.
V_i	Turn-on time of sCH i, when sCH $i - X$ catches mCH
r	Data rate of the main channel.
R	Data rate of the sub-channel.
Δ	*Speed-up* of the sub-channel, $\Delta = \frac{R}{r} - 1$.
θ_i	Initial time gap between sCH i and mCH.
D_i	The lifetime of sCH i, the duration from it is turned on till it catches mCH.
$B(R)$	The amount of traffic delivered over a sCH at rate R during its lifetime.
$Z_m(t)$	The index of the packet being transmitted over mCH at time t.
$Z_i(t)$	The index of the packet being transmitted over sCH i at time t.
$H_i(t)$	The time lag of sCH i regarding the main channel at t.
A	The current time lag of the sCH with the smallest channel index regarding mCH .
t_I	The time the latest I-frame before the zapping request was delivered over mCH.
i^*	The index of the sCH that provides the earliest I-frame.
$P_i(t)$	At any time t, the packet being transmitted over sCH i should have been transmitted over mCH at time point $P_i(t)$.
J	Predicted signaling delay in IP multicast.

where C_i is the creation time of sCH i and C_m is the creation time of the mCH.

A higher sCH rate R compared to mCH rate is purposely configured so that an active sCH can later catch and merge into the main channel to reduce the redundant traffic. It is not difficult to calculate that the sCH i will catch mCH at $C_i + iT/\Delta$, where $\Delta = \frac{R}{r} - 1$, considering that sCH i lags behind the mCH with data volume iTr when it is created. A channel merging event will logically turn on a new sCH, so that the TSS-system always logically maintains X sCHs. The term "logically maintain" means that the time-shifting relationship of each sCH is calculated, but the actual data transmission (i.e., activation) of a sCH depends on the service request. Such a sCH service model could achieve the balance between guaranteed FID performance and controlled traffic redundancy.

For the the convenience of analysis, we sequentially index all the sCHs that will be involved. Since the neighboring sCHs have a clear T-shifting relationship, we can consider that those sCHs turned on later (i.e., sCHs $i > X$) are virtually created at $C_i = C_m + iT$, $i > X$. Note that when a sCH i catches mCH, it will turn on the sCH $i + X$. Let V_{i+X} denote the turn-on time of sCH $i + X$. We have

$$C_{i+X} = C_m + (i + X)T, \quad i \geq 1 \tag{2}$$

$$V_{i+X} = C_i + \frac{iT}{\Delta}, \quad i \geq 1. \tag{3}$$

For the convenience of implementation, it is important to determine the packet index for transmission when a sCH is turned on under the time-shifting constraint. Consider that the packet for transmission over sCH $i+X$ at the turn-on moment V_{i+X} was transmitted on mCH θ_{i+X} time units ago. Since the sCH $i + X$ lags behind the sCH i with time XT and sCH i just catches mCH at V_{i+X}, thus,

$$\theta_{i+X} = \frac{XTR}{r}. \tag{4}$$

The key observation underpinning equation (4) is that the data rates of sCHs and mCH are different[4]. At V_{i+X}, although sCH $i+X$ lags behind the sCH i with time XT and sCH i is transmitting the same packet as mCH, we need to trace back more than XT time units over mCH to retrieve the packet that should be transmitted over sCH $i + X$ at the moment.

We consider the *lifetime* of sCH $i + X$ since it is turned on till it catches mCH. According to the virtual creation time (2), it can be seen that sCH $i+X$ will catch mCH at $C_{i+X} + (i + X)T/\Delta$. Let D_{i+X} denote the life time of sCH $i + X$, and we have

$$\begin{aligned} D_{i+X} &= C_{i+X} + (i + X)T/\Delta - V_{i+X} \\ &= XT(1 + \frac{1}{\Delta}) \quad i \geq 1. \end{aligned} \tag{5}$$

Note that the TSS system has a regular pattern in merging and turning on sCHs. The above analysis generally applies to every sCH dynamically turned on.

3.2 Optimal Sub-Channel Data Rate

In the TSS model [2], all the packets transmitted over a sCH are replicated data of a corresponding portion of mCH, which are the traffic redundancy. Let $B(R)$ denote the amount of traffic delivered over a sCH during its lifetime. Based on (5), we have

$$\begin{aligned} B(R) &= RXT(1 + \frac{1}{\Delta}) \\ &= RXT + \frac{RXT}{\Delta}. \end{aligned} \tag{6}$$

To minimize the redundant traffic, we can determine the optimal sCH data rate R^* in the range (r, ∞) according to

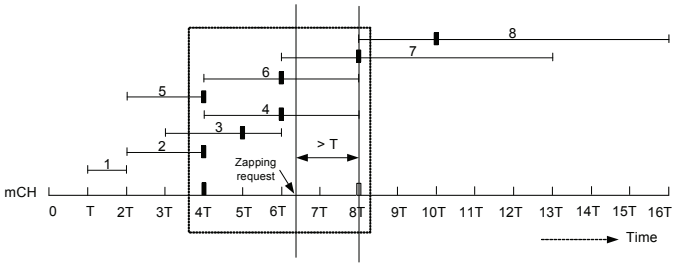
$$\frac{d}{dR}B(R) \mid_{R=R^*} = 0, \tag{7}$$

which gives

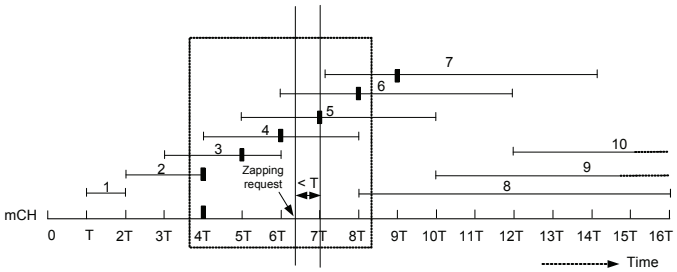
$$R^* = 2r. \tag{8}$$

It is interesting to interpret why there exists an optimal sCH rate and why the optimal rate is $2r$. Based on the analysis above, we can see that the total amount of traffic delivered during the lifetime of a sCH consists of two parts, the *initial lagging traffic* at the moment when the sCH is turned on and the *extra catching traffic* during the catching procedure. The sCH data rate R has contradicting effects on the two parts. The analysis in subsection 3.1 shows that the initial lagging traffic is XTR when a sCH is turned on by a sCH merging event in a general context, which is monotonically increasing with R. Equation (6) shows that the extra catching traffic is $RXT/\Delta = \frac{XTr}{1-\frac{r}{R}}$ (with $\Delta = \frac{R}{r} - 1$), which is monotonically decreasing with R. The optimal value $R^* = 2r$ balances the two contradicting effects, where $R^*XT = R^*XT/\Delta$ and the sum of these two items is minimized.

4. The parameter θ_{i+X} in [2] is actually the initial time gap between sCH $i + X$ and sCH i, instead of mCH



(a) Start-up effect



(b) Augmented turn-on policies

Fig. 4. Start-up effect and solution.

3.3 Start-Up Effect

The *start-up effect* occurs when the new sCHs start to be turned on by the sCH turn-on policy 2. Simply turning on the sCHs according to (3) and starting packet transmission according to the time relationship (4) may lead to the FID greater than T. Consider a scenario shown in Fig. 4(a), where the largest GOP size $s = 4T$ ($X = 4$), $R = 2r$ and the time shift is T. Each section of line represents the duration of a sCH, with its index labeled. The sCHs 1 to 4 are turned on every T time units. According to the sCH turn-on policies defined in [2], when sCH 1 catches mCH, the sCH 5 should be turned on, and the first packet should be on mCH $8T$ time units ago according to (4). However, since mCH just started $2T$ ago, the sCH 5 at best can start from the first packet on the main channel. The same can happen to sCHs 6 to 8. Suppose that an I-frame, represented as a black rectangle, is transmitted at $4T$ over mCH. The time points where it reappears over existing sCHs can be easily identified. For example, sCH 3 starts from packet 0 and the data rate is twice as that of mCH, thus the I-frame will reappear on sCH 3 $2T$ after its creation. Here we assume the first two GOPs are s long. The grey rectangle in the figure is the I-frame of the second GOP. If a zapping request arrives at some time between $6T$ and $7T$, the earliest I-frame will be seen is at $8T$ as illustrated in Fig. 4(a), thus the FID is greater than T.

The start-up effect particularly impacts the sCHs $i + X$, $i \geq 1$. To avoid the situation that a sCH tends to send a packet even before it is generated, a constraint for turning on sCHs by channel merging event should be: the channel turn-on time should not be earlier than the channel creation time, i.e.,

$$V_{i+X} \geq C_{i+X}. \tag{9}$$

After some manipulation, we have

$$i \geq X\Delta.$$

Therefore, the channel merging event can turn on sCHs with index

$$i + X \geq X(\Delta + 1). \tag{10}$$

The constraint (10) indicates that the turn on policies could be augmented.

Augmented sCH Turn-on Policies:

1) *For the first* $\lceil X(\Delta+1)\rceil - 1$ *sCHs, they are created and turned on one after another spaced by* T.
2) *For* $i \geq \lceil X \cdot (\Delta + 1)\rceil$, *sCH* i *is turned on when sCH* $i - X$ *catches up with mCH.*

Applying the augmented policies to the system illustrated in Fig. 4(a), the start-up effect will not happen and the model guarantees a bounded FID, which is shown in Fig. 4(b). It is worth mentioning that the augmented sCH turn-on policies will not impact the results presented in (4) to (8) as long as the sCH turn-on policy according to the channel merging event starts to take effect.

3.4 Delay Bound Guarantee

Although it is intuitively true that using X $(= \lceil \frac{s}{T} \rceil)$ T-shifted sCHs to replicate mCH can ensure a FID bound of T time units, it is meaningful to give a rigorous theoretical proof of such FID bound for the general scenario where the GOPs may have heterogeneous sizes and the augmented sCH turn-on policies are applied.

For the convenience of presentation, we term the regular T-shifted turn-on policy as policy 1 and the policy upon the channel merging event as the policy 2. Notice that for the sCHs turned on under policy 1, their turn-on times are the same as their creation times and thus T-spaced. However, the turn-on times of neighboring sCHs under the policy 2 are not T-spaced. Based on (2) and (3), it can be checked that for a sCH $i(\geq \lceil X(\Delta + 1)\rceil)$ being turned on under policy 2, the gap between successive turn-on moments is $V_{i+1} - V_i = T + \frac{T}{\Delta}$. Regarding the data transmission, it is important to check whether the T-shifting property can still maintain under the policy 2. We have the following lemma regarding the time shifting property.

Lemma 1: For any two neighboring sCHs i and $i+1$, under the augmented sCH turning-on policies, a packet delivered over sCH i will be delivered over sCH $i + 1$ at time $t + T$.

Proof: We prove the lemma by establishing the time lag relationship between a given sCH and mCH. Let $H_i(t)$ denote the time lag of sCH i regarding mCH, that is, the packet being transmitted over mCH at t will be retransmitted over sCH i at time $t + H_i(t)$.

We first examine the time shifting relationship for two neighboring channels turned on under policy 2. At any time t, the index of the packet being transmitted over mCH is $Z_m(t) = r \cdot (t - C_m)$. The sCH turning on operation is that when the sCH $i - X$ merges into the main channel, the sCH i is turned on (at time V_i) and starts transmission at rate R from the packet with an index XTR earlier than that over mCH at time V_i (referring to the discussion in Section 3.1). Under such an operation, at time t the packet index over the

sCH i is $Z_i(t) = r(V_i - C_m) - XTR + R(t - V_i)$. We can then find the time lag

$$H_i(t) = \frac{Z_m(t) - Z_i(t)}{R} = (1 - \frac{r}{R})(V_i - t) + XT, \quad i \geq \lceil X(\Delta + 1) \rceil. \tag{11}$$

Based on (11), the time gap between sCH i and sCH $i+1$ can be obtained as $H_{i+1}(t) - H_i(t) = (1 - \frac{r}{R})(V_{i+1} - V_i) = T$ (recall that $V_{i+1} - V_i = T + \frac{T}{\Delta}$ and $\Delta = \frac{R}{r} - 1$).

For a sCH i ($\leq \lceil X(\Delta + 1) \rceil - 1$), we can easily see that the packet index at t is $Z_i(t) = R(t - C_i)$. The packet index over mCH is still $Z_m(t) = r(t - C_m)$. We can then compute

$$H_i(t) = \frac{Z_m(t) - Z_i(t)}{R} = (1 - \frac{r}{R})(C_m - t) + iT, \quad i \leq \lceil X(\Delta + 1) - 1 \rceil. \tag{12}$$

Based on (12) and $C_i = C_m + iT$ for $i \leq \lceil X(\Delta+1) - 1 \rceil$, the time gap between neighboring sCHs is $H_{i+1}(t) - H_i(t) = T$.

Finally, we consider the gap between channel $\lceil X(\Delta+1) \rceil - 1$ and channel $\lceil X(\Delta + 1) \rceil$, turned on by policy 1 and policy 2, respectively. For convenience, denote $K = \lceil X(\Delta + 1) \rceil$. By applying $V_K = C_{K-X} + \frac{(K-X)T}{\Delta}$ (the merging moment of sCH $K - X$) to (11) and $C_{K-1} = C_m + (K - 1)T$ to (12), we can also have $H_K(t) - H_{K-1}(t) = T$ after some manipulation. ■

With Lemma 1, we can prove the FID bound of T time units. Specifically, we have the following theorem.

Theorem 1: In the TSS-based service model, there is at least one channel (including mCH and sCHs) which guarantees that an I-frame will appear within T time units after the channel zapping request arrives.

Proof: Consider that the channel zapping request arrives at t; the latest I-frame before the zapping request was delivered over mCH at $t_I \leq t$; and the newest I-frame after the zapping request are at $t_{I+1} > t$. Since the maximum GoP is s, we have $t - s < t_I \leq t$ and $t_{I+1} - t_I \leq s$.

Let A denote the time lag of the earliest sCH (the active sCH with the smallest channel index) with respect to mCH at t_I; we have $0 < A \leq T$. Under the turn-on policy 1, it is clear that sCH 1 starts with a time lag of T, but the time lag will reduce because the sCH rate is higher than mCH rate (i.e., $R > r$). Under the turn-on policy 2, upon the channel merging event, the existing earliest sCH has a time lag of T at that moment and again the time lag will reduce along with the time because $R > r$. Given a time t, we can index the current earliest sCH as sCH 1 without lose of the generality, and thus $H_1(t_I) = A$. Since all the neighboring sCHs have a time lag relationship of T according to Lemma 1, we then have

$$H_i(t_I) = A + (i - 1)T, \quad i \in [1, X] \tag{13}$$

for all the X sCHs that are currently logically on. For convenience, we denote mCH as $i = 0$. To prove the theorem, we examine two cases.

Case 1: zapping request starts being served from frame I. If $t - t_I \leq s - T$, we can show that a channel index $i \in [0, X]$ exists so that

$$t \leq t_I + H_i(t_I) \leq t + T. \tag{14}$$

That is,

$$t \leq t_I + A + (i - 1)T \leq t + T \tag{15}$$

and thus

$$\frac{t - t_I - A}{T} + 1 \leq i \leq \left(\frac{t - t_I - A}{T} + 1 \right) + 1 \tag{16}$$

Note that $0 \leq t - t_I \leq s - T$ and $-T \leq -A < 0$, we can see that

$$0 \leq \frac{t - t_I - A}{T} + 1 < \frac{s}{T} \tag{17}$$

Recall that $X = \lceil \frac{s}{T} \rceil$, so the results of (16) and (17) guarantee that a channel index $i \in [0, X]$ exists.

Case 2: zapping request starts being served from frame $I+1$. If $t - t_I > s - T$, the zapping request will be served with the main channel, starting from the frame $I + 1$. Note that $t_{I+1} - t_I \leq s$, thus we have $t_{I+1} - t < T$. ■

In practice, either the GOP size or the data rate of the video object is not necessarily constant; however, the analysis presented in the previous sections is oblivious to these variabilities. Since the case we study is general, where the largest GOP size s is considered and the performance of the model is only related to the factor of R/r regardless of the received rate of the video flow [2].

4 DAZA: DOM Assisted Zapping Acceleration

In this section, we first describe the design of DAZA, which implements the TSS-based model and integrates it into the IPTV infrastructure. We then analyze the properties of DAZA.

4.1 TSS-based Model Facilitated by DOM

The DAZA server can be deployed at the edge or core of the service provider's network to accelerate the channel zapping. TV channels requiring zapping acceleration are first streamed to the DAZA server, where the time-shifted sCHs for each channel are generated. The data source server could also be a DAZA server itself. To implement the TSS-based service model, the DAZA server maintains two fundamental data structures, *Sub-channel States Table* and *DOM Destinations Cache*, as shown in Fig. 5. We consider the single group case (GID_1) for the convenience of presentation.

4.1.1 Sub-Channel Selection

Sub-channel States Table (SST) maintains information of $X + 1$ channels that are logically turned on. For each sCH, the index number, turn-on time and the time gaps between the sCH and mCH at t_I are recorded, where t_I is the transmission time of the latest I-frame over mCH. Initially, the SST table contains X sCHs spaced by T. The SST will be updated according to the augmented sCH turn-on policies as analyzed in Section 3.3. The sCH with the lowest index will first outspace mCH, as the shadowed row of SST in the left side of Fig. 5(a). Then the table entry will be replaced by a new sCH as shown in the left side of Fig. 5(b). For the new sCH added to the SST, its turn-on time is determined by the sCH turn-on policies, while $H_i(t_I)$ can be computed using (11) or (12),

Sub-Ch States Table for GID1

CH #	V_i	$H_i(t_I)$
0 (mCH)	C_m	n/a
1	C_m+T	T
2	C_m+2T	$2T$
...	...	...
X	C_m+XT	XT

DOM Dest. Cache for GID1

CH #	Subscribers
0 (mCH)	R1, R2
1	R3, R4
2	R5
...	...
X	R6, R7, R8

(a) Initial states

Sub-Ch States Table for GID1

CH #	V_i	$H_i(t_I)$
0 (mCH)	C_m	n/a
X+1	C_1+T/Δ	h
2	C_m+2T	$2T$
...	...	...
X	C_m+XT	XT

DOM Dest. Cache for GID1

CH #	Subscribers
0 (mCH)	R1, R2, R3, R4
X+1	...
2	R5
...	...
X	R6, R7, R8

(b) Updated states

Fig. 5. Data structure of DAZA.

based on the value of the sCH index. After the sCH index goes beyond $\lceil X \cdot (\Delta + 1) \rceil$, the SST will be updated every $T + \frac{T}{\Delta}$ time units as analyzed in Section 3.3.

When a zapping request is received, the DAZA server first checks the SST to find the sCH i^* that will provide the earliest I-frame, as illustrated in Algorithm 1.

Algorithm 1: sCH selection

Input: t_I - the time the latest I-frame before the zapping request was delivered over mCH;
t - the time a zapping request is received;
Output: i^* - the sCH to provide the earliest I-frame;
if $\forall i \in [1, X], (t_I + H_i(t_I) - t) < 0$ **then**
  $i^* = 0$;
else
  $i^* = argmin_i\{(t_I + H_i(t_I) - t) \geq 0\}$;
end

Note that Algorithm 1 corresponds to the two cases we proved in Theorem 1, thus the DAZA server can always find at least one of the channels, which will transmit an I-frame within T. The physical meaning of Algorithm 1 is illustrated in Fig. 6 for better understanding. In the example, when a zapping request is received at t, the latest I-frame was transmitted over mCH at t_I, and this I-frame will be sent over each of the 3 sCHs at $H_i(t)$. Fig. 6(a) shows an example of the first case in Algorithm 1, where the zapping request arrives so late that it misses the I-frame transmitted over all 3 sCHs; therefore, the earliest I-frame must be available over mCH. Fig. 6(b) shows an example of the second case of Algorithm 1, where sCH 1's corresponding I-frame has been missed but sCH 2 has the I-frame on the way and the waiting time must be bounded by T.

4.1.2 Data Transmission over Sub-Channels

Among those "logically on" sCHs, only the sCHs with subscribers are actually transmitting data packets. The transmitted

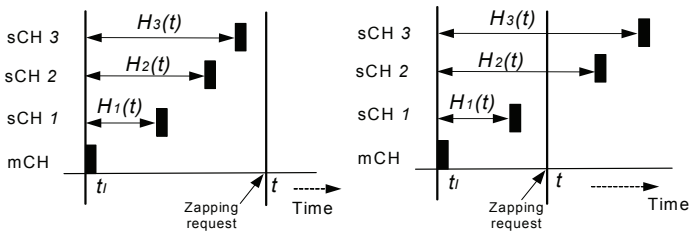


Fig. 6. Sub-channel selection.

packets over the sCHs are replicated from the appropriate portion of mCH. At any time t, the packet being transmitted over sCH i, should have been transmitted over mCH at corresponding time $P_i(t)$. Algorithm 2 shows how to identify the $P_i(t)$.

Algorithm 2: Data to be transmitted over sCH i

Input: t - the time;
V_i - turn-on time of sCH i when sCH $i - X$ catches mCH;
θ_i - initial time gap between sCH i and mCH;
Output: $P_i(t)$ - at any time t, the packet being transmitted over sCH i should have been transmitted over mCH at time point $P_i(t)$;
if $i < \lceil X(\Delta + 1) \rceil$ **then**
  $\theta_i = iT$;
else
  $\theta_i = \frac{XTR}{r}$;
end
if $t \leq V_i$ **then**
  $P_i(t) = V_i - \theta_i$;
else
  $P_i(t) = \Delta(t - V_i) + t - \theta_i$;
end

We here explain the physical meaning of Algorithm 2. If $t \leq V_i$, meaning the sCH i should wait to be turned on, so sCH i should retrieve the data transmitted over mCH θ_i time units before sCH i is turned on. The initial time gap between sCH i and mCH, θ_i, differs according to the index of the sCH as indicated by the augmented turn-on policies in Section 3.3.

If $t > V_i$, the sCH i has been turned on, its starting packet index is $r(V_i - \theta_i)$, and the packet index becomes $r(V_i - \theta_i) + R(t - V_i)$ at t; therefore, the packet with this index has been transmitted over mCH at

$$P_i(t) = \frac{r(V_i - \theta_i) + R(t - V_i)}{r} = \Delta(t - V_i) + t - \theta_i.$$

4.1.3 Seamless Migration to Main Channel

DOM Destinations Cache (DDC) is used to record subscriber addresses for the mCH and active sCHs, and for sCH subscribers to seamlessly migrate to mCH. DDC is initialized by grouping subscribers that select the same sCH as shown in the right side of Fig. 5(a), where mCH is also included.

For a given sCH, if it catches mCH, i.e., the play-outs over the sCH and mCH are the same, the subscribers of the sCH will be merged into the set of main channel subscribers. Refer to the right side of Fig. 5(b), sCH 1 catches mCH, so the original subscribers $R3, R4$ of sCH 1 merge into mCH, as highlighted in the shadowed area. The channel index of DDC is for DAZA server to retrieve correct packet payloads, as different sCHs are transmitting different parts of the video as indicated by the output of Algorithm 2. The addresses of the subscribers (Ri) will be encapsulated into the multicasting packet labeled with group ID GID_1. DOM data forwarding protocol will deliver the packet to corresponding destinations. The subscribers migration needs no message exchange over the network.

4.2 Performance Analysis

DAZA and MAZA both implement the TSS-based service model, while the fundamental difference is that: DAZA selects sCHs at the streaming server while MAZA does it at the STBs.

4.2.1 Signaling Delay Effect

The time taken for a STB to join in a multicasting group (signaling delay) can possibly affect the FID of MAZA, but will not affect DAZA. In MAZA, the meta-channel information is broadcast to STBs in advance. After the STB receives the meta-channel information, it has to take the signaling delay into account to select the sCH to join in. Consider the scenario illustrated in Fig. 7, the appearance of I-frame I is notified to the STB J time units in advance, where J is the signaling delay predicted. If a zapping request arrives at the STB at t, the STB should find out which sCH will provide the earliest I-frame at $t + J$ time units. In Fig. 7, the STB will apparently take sCH 1. However, as it is difficult to predict the traffic condition of access networks, the joining message can be very likely delayed for a longer time than J and miss the time when the selected sCH is transmitting the wanted I-frame. This happens as the late joining message case shown in Fig. 7, and the arrived joining message has to wait up to XT time units for the next I-frame.

The signaling delay could be very short as mentioned in Section 2.1; however, the major issue is that MAZA requires the accurate prediction of the signaling delay for sCH selection, which is quite difficult in practice. For example, even if the signaling delay can be bounded, it is also possible that the joining message arrives at the MAZA server earlier than expected. In the early joining message case of the Fig. 7, the early arrived joining message has to wait more than T to get the first I-frame. If the STB had known the joining message could take such short time to reach the MAZA server, it would have joined in mCH, instead of sCH 1. In contrast, DAZA selects sCHs at the streaming server after the joining message arrives, which is totally independent of the signaling delay.

4.2.2 Control Messages Overhead

DAZA incurs less control messages overhead than MAZA does. The control message exchange process for a zapping act under MAZA is illustrated in Fig. 8. MAZA server broadcasts

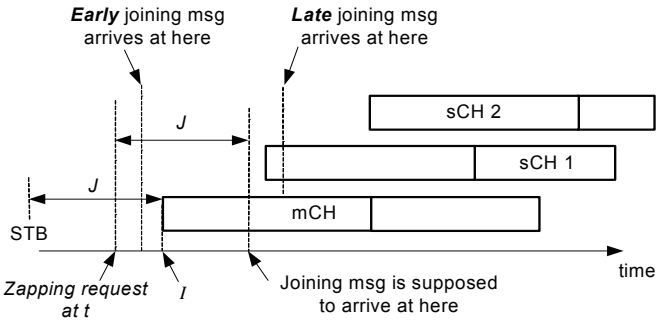


Fig. 7. Signaling delay effect.

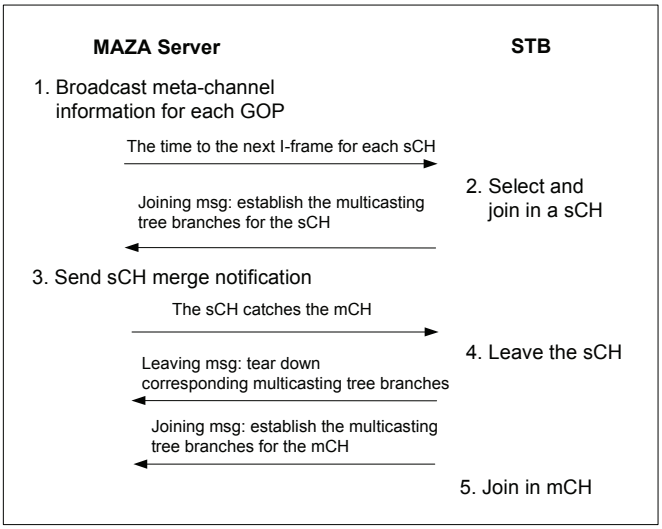


Fig. 8. Control message exchange in MAZA.

the meta-channel information for each GOP of every TV channel provisioned. If the STB chooses a sCH upon a zapping request, it is in the sCH until the sCH merges into mCH. At the moment of merging, the MAZA server sends a notification to indicate the STB to join in mCH. To this end, the STB sends a leaving message to unsubscribe the sCH and then sends another joining message to migrate to mCH [30], [31]. Frequent messages exchange incurs higher probability of system errors. In contrast, the DAZA only needs one joining message to switch to the new channel, and the migration from sCH to mCH is performed without any message exchange over the network. Moreover, the MAZA meta-channel broadcasts to STBs each time an I-frame is sent over mCH, while DAZA needs no meta-channel broadcasting. In DAZA, DOM constructs the source-based multicast tree using destination-specific forwarding states [26], [27]. The zapping operation from a STB does not need to tear down the multicasting tree branch as long as the new channel is still provided by the same data source.

4.2.3 Scalability

A major concern for DAZA is that the multicasting packet may not be able to encode all the subscribers' IP addresses if there are a large number of subscribers, which may hinder DAZA from scaling up. In fact, if a STB requests to switch

to an existing channel that is already being received by other STBs in the receiver domain, the request can be served by the local DAZA server for the domain. Otherwise, the request has to be sent to the DAZA server located in the core network. For both cases, the scalability issue can be handled.

Recall the typical wireline broadband access network architecture as illustrated in Fig. 2, the STB initiates the channel zapping request as an IGMP request for joining/leaving corresponding groups [30], [31], the request passes through the routing gateway (RG), Digital Subscriber Line Access Multiplexer (DSLAM), and will finally reach the broadband services router (BSR) acting as the gateway into the backbone network. The technique of *local content insertion* [20] can be applied to provision a fast local channel zapping process, where a local DAZA server can be connected to the BSR. The DAZA server maintains all channels that are being watched by the local domain. If the requested channel is being viewed by existing subscribers, the zapping request will be served by the local DAZA server. For a given channel, the corresponding sCH subscriber will eventually merge into the mCH, and 90% of receivers tend to be interested in the top 10% of the channels according to the study on IPTV users behavior [12], [25]; therefore, the subscribers' IP addresses list of each popular channel will be considerably long. In order to avoid the expense of this lengthy enumeration in the DOM multicasting header, we can use the *multicast push* model [23], where a channel will be broadcast if the number of its subscribers are beyond certain proportion of the total subscribers in the domain.

If the STBs request new channels that are not maintained in the local DAZA server, the BSR needs to unicast the aggregated membership information towards corresponding DAZA server in the core network to establish the multicasting tree. Consequently, the addresses encoded into the DAZA multicasting packet are the BSRs' addresses. The scalability of DOM in this inter-domain scenario is described in detail in [26], [27]. Particularly, a large number of BSRs still can incur bandwidth overhead in DAZA, as multiple data packets need to be generated to cover all BSRs, with each covers a subset of the BSRs. However, the bandwidth overhead of DAZA is reasonably low, and close to that of MAZA. Section 5.6 gives the simulation results for comparing the bandwidth overhead under DAZA with that under MAZA.

5 Simulation Results

We use ns-2 [34] simulation results to verify our theoretical analysis and demonstrate the performance of DAZA. The network topology for simulation is given in Fig. 9. The topology is widely used in the literature as a hypothetical US backbone network [35], where we deploy major components in the IPTV network architecture [1], [15], [20]. The edges of the network represent the BSRs, and they are connected to the Video Hub Office (VHO), which contains necessary equipments for delivering nation-wide/local television programs. VHOs are then connected to the Super Head-end Office (SHO) through the core network, which encodes programs from nation-wide content providers. The black node denotes the streaming

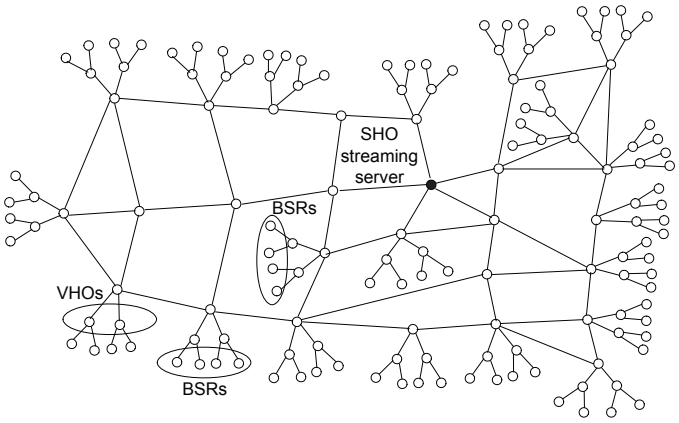


Fig. 9. Simulation topology.

server (e.g., DAZA server) located in the SHO, which realizes different channel zapping acceleration schemes. In order to demonstrate the essence of the proposed scheme, the internal details of SHO, VHOs and access network are purposely omitted. In the simulation, a MPEG-4 video clip is segmented into packets and multicast to BSRs upon request. These packets can be reverted to the video format and displayed with dedicated program [36]. The largest GOP size of the video is set to 1000 ms. We note that the choice of GOP size is based on the expected bandwidth cost by the media stream. Small GOP size in time units means the number of I-frame per second is increased, and the required bandwidth for encoding the media stream is increased; because the I-frame contains the entire picture and its size is significantly larger than other frames in a GOP [2]. However, the choice of the GOP size will not affect the simulation results on DAZA/MAZA as long as the largest GOP size is larger than the FID bound T, which has been proved in Theorem 1. The time for each BSR to issue zapping requests is randomly distributed over the duration of the video clip.

5.1 Visual Effect

We reconstruct the packets received by BSRs under DAZA and MICC [5], [6] into pictures to examine the visual effects. The simulation results confirm that the TSS-based DAZA scheme provides a better QoE than the MICC scheme which uses low-resolution accompanying channels of Approach 2 mentioned in Section 1. Fig. 10(a) displays the first pictures received after the zapping request under the two schemes; Fig. 10(b) shows the pictures received just before the migration from the sCH to mCH. For both of the figures, the picture on the left is yielded with DAZA and the one on the right with MICC. Pictures in Fig. 10(a) are retrieved from a BSR that is 3 hops away from the streaming server, while the BSR observing pictures in Fig. 10(b) is 6 hops away from the streaming server. The visual effect of DAZA is obviously better than MICC, which can be identified by observing the man's eyeballs in the pictures. Moreover, Fig. 10(b) shows that the MICC scheme can incur the picture inconsistency when the receiver is migrating from the auxiliary stream to the main stream. This is because the DAZA sCH for fast zapping is a full-quality stream media

(a) First pictures observed

(b) Pictures observed just before migration

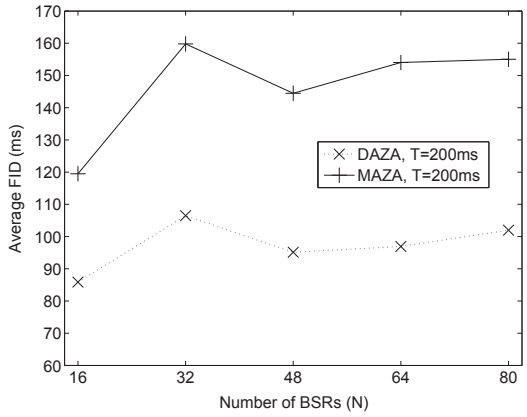
Fig. 10. Visual effects (left: DAZA with TSS-based Model, right: MICC based on Approach 2).

while the auxiliary stream of MICC just contains I-frames. With DAZA, receivers can always get the full-quality pictures of the video even during the channel zapping.

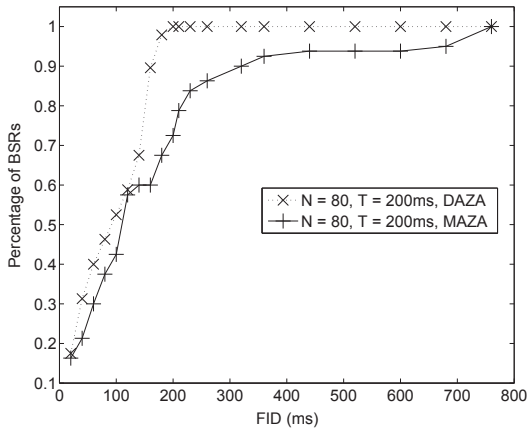
5.2 First I-Frame Delay

In this section, we examine the FID performance of DAZA and MAZA, where the augmented and the original sCH turn-on policies are applied, respectively. The impact of the population of receivers N on the FID by DAZA and MAZA is shown in Fig. 11, with a fixed value of $T = 200ms$. As shown in Fig. 11(a), the average FID under DAZA is around $100ms$ for different values of N. This is because the zapping requests will definitely fall into an interval sized T to bound the FID, which has been proved in Theorem 1. Moreover, the sending time of these requests are randomly distributed over the duration of the video playing out. Thus the average value of the FID should be approaching to $\frac{T}{2}$. The average FID under MAZA is higher than that in DAZA, because the start-up effect of the original sCH turn-on policies can violate the performance bound of TSS-based model as analyzed in Section 3.3.

The cumulative distribution functions (CDFs) of the FID under DAZA and MAZA in the case of $N = 80$ are illustrated in Fig. 11(b), which corroborate the analysis above. We can see that receivers observing FIDs less than $100ms$ account for about 50% under DAZA. Basically, the FIDs are evenly distributed over T. There is no zapping request observing a FID that is more than T. In contrast, a considerable number of zapping requests experience large FIDs that are more than 3 times the performance bound T. Under the sCH turn-on policies of MAZA, zapping requests may fall into an interval that is larger than T, thus the FID distribution is unpredictable; however, the FIDs will be no larger than the largest GOP size



(a) Average FID versus N



(b) CDFs of FID

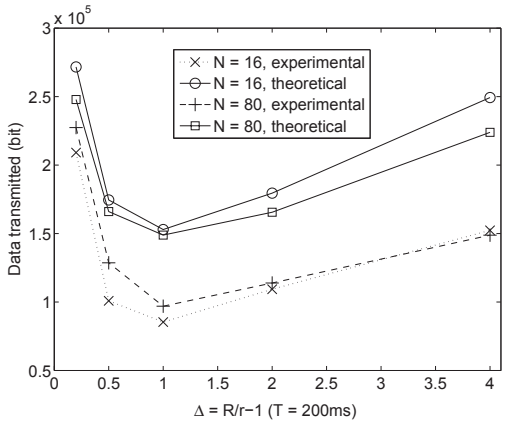
Fig. 11. The impact of N on FID distributions by DAZA.

1000 ms, as the zapping request will definitely fall into a GOP upon its arrival at the streaming server.

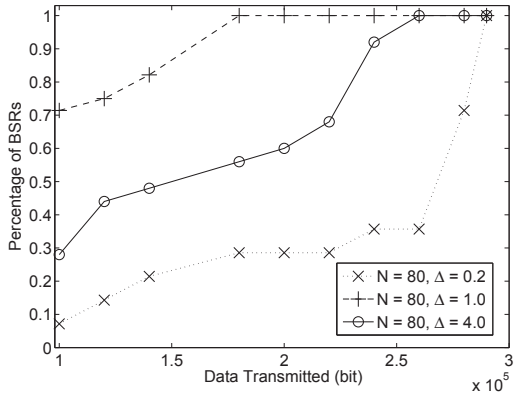
5.3 Optimal Sub-Channel Data Rate

The optimal sCH data rate R^* has been derived in Section 3.2. We here set the ratio of main channel data rate and sCH data rate as different values, and see if $R^* = 2r$ indeed makes the sCH transmit the least redundant traffic.

The amounts of data transmitted over sCHs with different values of $\Delta = \frac{R}{r} - 1$ are illustrated in Fig. 12(a). It is clear that the sCHs transmit the fewest amount of data when $\Delta = 1$, which means $R = R^* = 2r$. In the simulation, the "theoretical" value is the amount of data transmitted over all turned-on sCHs during their lifetimes. Practically, it may happen that the receiver joins in a sCH that has been turned on for a while; the "experimental" value is the amount of sCH data actually received by BSRs. As the zapping requests reach the DAZA server at different times, sCHs to be activated are different for the sparse ($N = 16$) and dense ($N = 80$) cases. Moreover, the lifetimes of sCHs are different according to their sCH indices. This is why the theoretical or experimental values are different in the cases of $N = 16$ and $N = 80$.

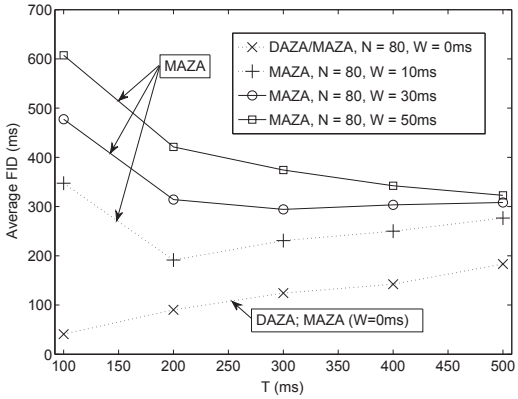


(a) Average data transmission

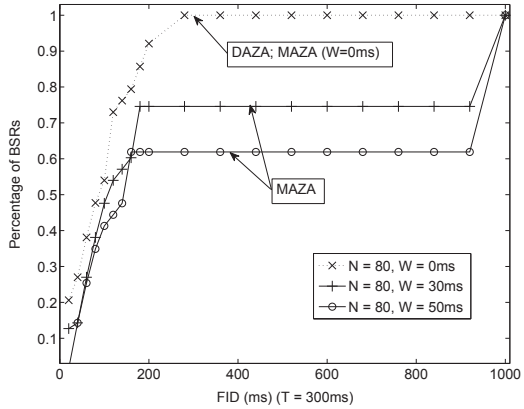


(b) CDFs of data transmission, experimental values

Fig. 12. Optimal sub-channel data rate.



(a) Average FID versus T under MAZA



(b) CDFs of FID ($T = 300ms$) under MAZA

Fig. 13. FID performance with signaling delay effect.

The distributions of the amount of data transmitted over a sCH are illustrated in Fig. 12(b) for $N = 80$. When $R = R^*$, the percentage of sCHs with light data transmitting is higher than that when R takes other values. It shows that all sCHs transmit less than 2×10^5 bits with $R = R^*$, while some sCHs need to transmit about 3×10^5 bits if R takes other values.

5.4 Signaling Delay Effect

In MAZA, the signaling delay can make the zapping request miss the I-frame expected, thus affect the FID performance. To demonstrate the effect of signaling delay, we configure extra traveling time W for zapping requests and apply the augmented sCH turn-on policies to MAZA to bypass the start-up effect.

The average FIDs perceived in MAZA with different values of W are shown in Fig. 13(a), for $N = 80$. As MAZA and DAZA are all implemented from the TSS-based service model, and DAZA is independent of the signaling delay, no matter what values of W are applied to DAZA, the average FID and CDF of FID of DAZA are the same as that under MAZA when $W = 0$. In contrast, the FID performance in MAZA is significantly affected even with a small value of W. The curves representing the average FID in Fig. 13(a) first increase dramatically, then decrease, and gradually approach to certain level. Both T and W can impact the FID performance of MAZA. If T is large, it is more probably that the zapping request will not miss the expected I-frame even with the extra traveling time, and the signaling delay could even luckily further mitigate the FID; however, if T is small, it is easier for the zapping request to fall out of the interval sized T with the extra delay, and the request has to wait the I-frame in the next GOP. Similarly, a larger W will make the I-frame slip away more easily. That is the reason the longest average FID occurs with $T = 100ms$ and $W = 50ms$, and it becomes smaller when T increases. When T becomes much larger than W, the impact from the signaling effect will be wakened as most of the zapping requests can not miss the expected I-frame even with extra traveling time, and the average FID will converge to around $\frac{T}{2}$; however, large T will negatively impact the FID performance of IPTV systems.

Many lucky zapping requests can be found when T becomes larger, where the extra delay for these requests can reduce the FID by T at most; however, if an I-frame is missed, the next I-frame could be up to XT time units away. The negative side of the signaling delay effect is still dominant. Fig. 13(b) shows this fact, where the number of BSRs with FIDs beyond $T = 300ms$ accounts for about $30 - 40\%$ if $W \neq 0$.

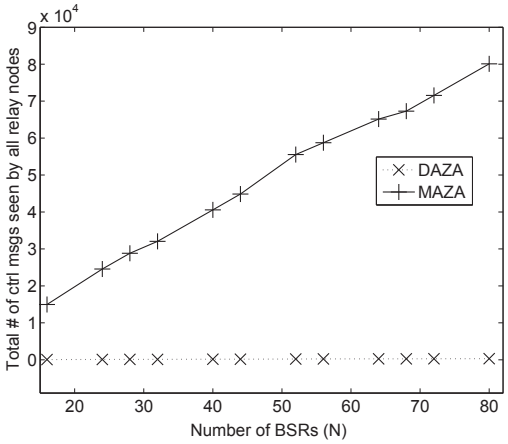
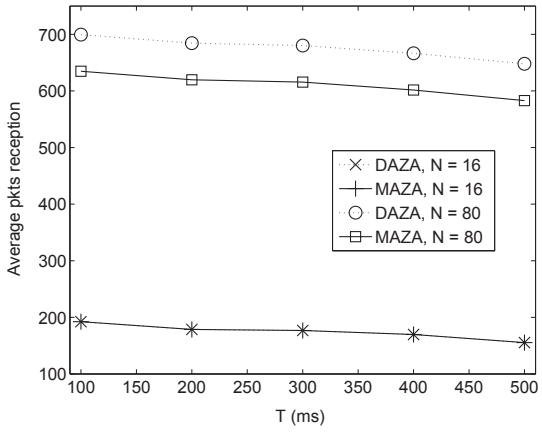
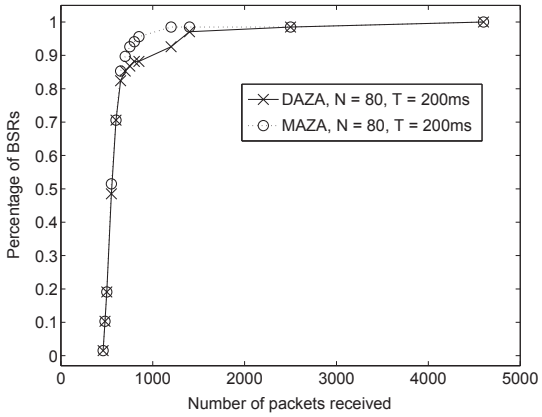


Fig. 14. Control messages overhead.



(a) Average packet reception



(b) CDFs of the packets reception per node

Fig. 15. FID performance with signaling delay effect.

5.5 Control Messages Overhead

We count the total number of control messages that all relay nodes have seen within $10s$ since the video starts to play under MAZA and DAZA. The total number of received messages by all relay nodes versus different receivers populations is depicted in Fig. 14. Most of the messages for MAZA are from meta-channel which broadcasts to receivers each time an I-frame appears over the main channel. There are 264 I-frames identified by the MAZA server within $10s$ in the simulation. Because of its broadcasting nature, the total number of control messages over the network will grow linearly with the number of receivers in the network. In contrast, DAZA only needs the leave-and-join operation for zapping, and the seamless migration from the sCH to mCH is achieved by modifying the destinations list. Thus the number of control messages is negligible compared with that in MAZA. Although the control messages may not consume much bandwidth resources, they incur more chances of system errors. For example, if a receiver missed one or two meta-channel messages when zapping channel, the FID could be more than T.

5.6 Data Forwarding Overhead

This section examines the bandwidth overhead incurred by data forwarding under DAZA and MAZA. We set the limit on the number of destinations that can be encoded into a multicast data header to be 16, and count the number of packets received by each node for $10s$ since the video starts to play. The average numbers of packets reception with FID bound (T) set to different values, and CDFs of packets receptions within $10s$ are shown in Fig. 15. The explicit addressing in DOM causes an extra bandwidth consumption of DAZA compared with MAZA when $N = 80$; however, the difference is small. DOM encodes destination addresses list into the packet to facilitate the data forwarding. If the number of addresses is beyond the capacity of the list, multiple packets have to be generated to cover all the receivers. When $N = 80$, DOM has to generate some redundant traffic to cover all destinations; therefore the bandwidth overhead of DAZA is higher than that of MAZA, as illustrated in Fig. 15(a). However, as DOM leverages some techniques to constrain the extra bandwidth cost [26], [27], the difference of bandwidth consumptions between DAZA and MAZA is insignificant.

For both MAZA and DAZA, the receivers are the same and they just receive almost the same number of packets. Although DAZA uses DOM, there must be a single copy received at each BSR under DAZA for each packet, as there is only one receiver at each part of the network edge. The same thing happens to MAZA, which utilizes the IP multicast. Consequently, the left parts of DAZA and MAZA in Fig. 15(b) are overlapped. The discrepancy occurs because the relay nodes in the core network may receive multiple copies of a single packet under DAZA, each of which cover a subset of the destinations. However, the percentage of relay nodes that receive multiple copies of a packet is very close to that of MAZA using IP multicast as in Fig. 15(b). The bandwidth cost of the DAZA is reasonably low.

6 Conclusion

In this paper, we have presented a systematical analysis of the time-shifted sub-channels (TSS) based service model, which accelerates the channel zapping by constraining the first I-frame delay for IP video delivery over the Internet. The

IPTV system is studied as an example. We have shown that there exists an optimal sub-channel data rate which minimizes the redundant traffic transmitted over sub-channels; moreover, we have revealed the *start-up effect*, where the sub-channel data rate and the time shift between successive sub-channels can impact the behavior of the TSS-based model. With a solution to resolve the start-up effect, we rigorously prove that the TSS-based model can guarantee a zapping delay bound equal to the sub-channel time shift. Furthermore, we have proposed a scalable DAZA scheme which implements the TSS-based channel zapping over the DOM multicasting. DAZA can achieve seamless migration from the sub-channel to the main channel without any control message exchange over the network. The sub-channel selection is independent of the zapping request signaling delay, resulting in improved robustness and reduced messaging overhead of DAZA. The efficiency and feasibility of DAZA have been demonstrated by ns-2 simulations of multicasting an MPEG-4 video stream over a practical network topology, and we plan to develop testbed based experiments in our future work.

REFERENCES

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

[2] Y. Bejerano and P.V. Koppol, "Improving zap response time for IPTV," in *Proc. IEEE INFOCOM*, 2009, pp.1971–1979.

[3] N. Cohen, "USPTO patent application 20060143669: Fast channel switching for digital TV (Assignee: Bitband technologies (filed))," June 2006.

[4] H. A. Goosen and E. J. Rak, "US patent application publication US 2011/0289544 A1: Video streaming system including a fast channel change mechanism," Nov. 2011.

[5] D. Banodkar, K.K. Ramakrishnan, S. Kalyanaraman, A. Gerber, and O. Spatscheck, "Multicast instant channel change in IPTV systems," in *Proc. 3rd International Conference on Communication Systems Software and Middleware*, 2008, pp.370 – 379.

[6] K.K. Ramakrishnan and R.D. Doverspike, "IPTV challenges," in *Proc. Conference on Optical Fiber Communication/National Fiber Optic Engeneers*, 2008, pp.1 – 33.

[7] M. Pelt, "USPTO patent application 20050265374: Broadband telecommunication system and method used therein to reduce the latency of channel switching by a multimedia receiver (Assignee: Alcatel (filed))," Dec. 2005.

[8] B. Dacosta, "USPTO patent application 20060230176: Methods and apparatus for decreasing streaming latencies for IPTV (Assignee: Sony (filed))," Oct. 2006.

[9] C. Cho, I. Han, Y. Jun, and H. Lee, "Improvement of channel zapping time in iptv services using the adjacent groups join-leave method," in *Proc. 6th International Conference on Advanced Communication Technology,* 2004.

[10] J. Farmer, "USPTO patent application 20060075428: Minimizing channel change time for IP video (Assignee: Wave7 Optics (filed))," Apr. 2006.

[11] C. Y. Lee, C. K. Hong, and K. Y. Lee, " Reducing channel zapping time in IPTV based on user's channel selection behaviors," *IEEE Trans. Broadcasting*, vol. PP, no. 99, pp. 1–10, Apr. 2010.

[12] C. Zhang, D. Wu and K. W. Ross, "Unraveling the BitTorrent Ecosystem," *IEEE Trans. Parallel and Distributed Systems*, vol. 22, no. 7, pp.1164–1177, Jul. 2011.

[13] J. Boyce and A. Tourapis, "Patent (WO/2005/112465): Method and apparatus enabling fast channel change for DSL system (Assignee: Thomson)," Nov. 2005.

[14] J.M. Boyce and A.M. Tourapis, "Fast efficient channel change," in *Proc. International Conference on Consumer Electronics,* 2005.

[15] A. Shaikh, J. Wang, J. Yates, Y. Zhang, A. Mahimkar, Z. Ge and Q. Zhao, "Towards automated performance diagnosis in a large IPTV network," in *Proc. ACM SIGCOMM*, 2009, pp.231–242.

[16] B. VerSteeg, A. Begen, T. VanCaenegem, Z. Vax,"Unicast-based rapid acquisition of multicast RTP sessions," Internet-Draft draft-ietf-avt-rapid-acquisition-for-rtp-17, Nov. 2010.

[17] A. Begen and E. Friedrich,"RTP payload format for MPEG2-TS preamble," Internet-Draft draft-begen-avt-rtp-mpeg2ts-preamble-04, Oct. 2009.

[18] H. Guo, K. Lo, Y. Qian and J. Li, "Peer-to-Peer Live Video Distribution under Heterogeneous Bandwidth Constraints," *IEEE Trans. Parallel and Distributed Systems*, vol. 20, no. 2, pp.233–245, Feb. 2009.

[19] ITU-T Telecommunication Standardization Sector,"Quality of experience requirements for IPTV services," Recommendation ITU-T G.1080, Dec. 2008.

[20] Cisco systems, "Optimizing video transport in your IP triple play network," http://www.cisco.com/en/US/prod/collateral/routers/ps368/prod_white_paper0900aecd80478c12.html

[21] Wikipedia, http://en.wikipedia.org/wiki/Zap_time

[22] Juniper Networks, "Introduction to IGMP for IPTV networks: Understanding IGMP processing in the broadband access networks," white paper, http://s-tools1.juniper.net/solutions/literature/white_papers/200188.pdf

[23] Juniper Networks, "DSLAM selection for single-edge IPTV networks," white paper, http://www.juniper.net/us/en/local/pdf/whitepapers/2000189-en.pdf

[24] X. Liao, H. Jin, Y. Liu and L. M. Ni, "Scalable Live Streaming Service Based on Inter-Overlay Optimization," *IEEE Trans. Parallel and Distributed Systems*, vol. 18, no. 12, pp.1663–1674, Dec. 2007.

[25] A. Shaikh, J. Wang, J. Yates, Y. Zhang, A. Mahimkar, Z. Ge and Q. Zhao, "Modeling channel popularity dynamics in a large IPTV system," in *Proc. ACM SIGMETRICS*, 2009, pp.275–286.

[26] X. Tian, Y. Cheng, and B. Liu, " Design of a scalable multicast scheme with an application-network cross-layer approach," *IEEE Trans. Multimedia*, vol. 11, no. 6, pp.1160–1169, Oct. 2009.

[27] X. Tian, Y. Cheng, and X. Shen, "DOM: A scalable multicast protocol for next-generation Internet," *IEEE Network*, vol. 24, no.4, pp.45–51, July 2010.

[28] A. Broder and M. Mitzenmacher, "Network applications of Bloom filters: A survey," *Internet Mathematics*, vol. 1, no. 4, pp.485-509, May. 2004.

[29] B. Xiao and Y. Hua, "Using Parallel Bloom Filters for Multiattribute Representation on Network Services," *IEEE Trans. Parallel and Distributed Systems*, vol. 21, no. 1, pp.20–32, Jan. 2010.

[30] W. Fenner, "Internet group management protocol, version 2," IETF RFC 2236, Nov. 1997.

[31] B. Cain, S. Deering, I. Kouvelas, B. Fenner, and A. Thyagarajan, "Internet group management protocol, version 3," IETF RFC 3376, Oct. 2002.

[32] Y. Yang, J. Wang and M. Yang, "A Service-Centric Multicast Architecture and Routing Protocol," *IEEE Trans. Parallel and Distributed Systems*, vol. 19, no. 1, pp.35–51, Jan. 2008.

[33] K. Egevang and P. Francis, "The IP network address translator (NAT)," IETF RFC 1631, May 1994.

[34] The Network Simulator - ns-2, http://www.isi.edu/nsnam/ns

[35] R. Doverspike, G. Li, K. Oikonomou, K.K. Ramakrishnan and D. Wang, "IP backbone design for multimedia distribution: Architecture and performance," in *Proc. IEEE INFOCOM*, 2007, pp. 1523–1531.

[36] C. Ke, C. Shieh, W. Hwang and A. Ziviani, "An evaluation framework for more realistic simulations of MPEG video Transmission," *Journal of Information Science and Engineering*, vol. 24, no. 2, pp.425-440, Mar. 2008

Xiaohua Tian received his B.E. and M.E. degrees in communication engineering from Northwestern Polytechnical University, Xi'an, China, in 2003 and 2006, respectively. He received the Ph.D. degree in the Department of Electrical and Computer Engineering (ECE), Illinois Institute of Technology (IIT), Chicago, in Dec. 2010. He is currently a research associate in Department of Electronic Engineering in Shanghai Jiaotong University, China. His research interests include application-oriented networking, Internet of Things and wireless networks. He serves as a Technical Program Committee member for Communications QoS, Reliability, and Modeling Symposium (CQRM) of GLOBECOM 2011 and the 6th International Conference on Wireless Algorithms, Systems, and Applications (WASA 2011).

Yu Cheng received the B.E. and M.E. degrees in Electrical Engineering from Tsinghua University, Beijing, China, in 1995 and 1998, respectively, and the Ph.D. degree in Electrical and Computer Engineering from the University of Waterloo, Waterloo, Ontario, Canada, in 2003. From September 2004 to July 2006, he was a postdoctoral research fellow in the Department of Electrical and Computer Engineering, University of Toronto, Ontario, Canada. Since August 2006, he has been with the Department of Electrical and Computer Engineering, Illinois Institute of Technology, Chicago, Illinois, USA, as an Assistant Professor. His research interests include next-generation Internet architectures and management, wireless network performance analysis, network security, and wireless/wireline interworking. He received a Postdoctoral Fellowship Award from the Natural Sciences and Engineering Research Council of Canada (NSERC) in 2004, and a Best Paper Award from the conferences QShine 2007 and ICC 2011. He received the National Science Foundation (NSF) CAREER award in 2011. He served as a Co-Chair for the Wireless Networking Symposium of IEEE ICC 2009, a Co-Chair for the Communications QoS, Reliability, and Modeling Symposium of IEEE GLOBECOM 2011, and a Technical Program Committee (TPC) Co-Chair for WASA 2011. He is an Associated Editor for IEEE Transactions on Vehicular Technology.

Xuemin (Sherman) Shen received the B.Sc.(1982) degree from Dalian Maritime University (China) and the M.Sc. (1987) and Ph.D. degrees (1990) from Rutgers University, New Jersey (USA), all in electrical engineering. He is a Professor and University Research Chair, Department of Electrical and Computer Engineering, University of Waterloo, Canada. He was the Associate Chair for Graduate Studies from 2004 to 2008. Dr. Shen's research focuses on resource management in interconnected wireless/wired networks, wireless network security, wireless body area networks, vehicular ad hoc and sensor networks. He is a co-author/editor of six books, and has published more than 600 papers and book chapters in wireless communications and networks, control and filtering. Dr. Shen served as the Technical Program Committee Chair for IEEE VTC'10 Fall, the Symposia Chair for IEEE ICC'10, the Tutorial Chair for IEEE VTC'11 Spring and IEEE ICC'08, the Technical Program Committee Chair for IEEE Globecom'07, the General Co-Chair for Chinacom'07 and QShine'06, the Chair for IEEE Communications Society Technical Committee on Wireless Communications, and P2P Communications and Networking. He also serves/served as the Editor-in-Chief for IEEE Network, Peer-to-Peer Networking and Application, and IET Communications; a Founding Area Editor for IEEE Transactions on Wireless Communications; an Associate Editor for IEEE Transactions on Vehicular Technology, Computer Networks, and ACM/Wireless Networks, etc.; and the Guest Editor for IEEE JSAC, IEEE Wireless Communications, IEEE Communications Magazine, and ACM Mobile Networks and Applications, etc. Dr. Shen received the Excellent Graduate Supervision Award in 2006, and the Outstanding Performance Award in 2004, 2007 and 2010 from the University of Waterloo, the Premier's Research Excellence Award (PREA) in 2003 from the Province of Ontario, Canada, and the Distinguished Performance Award in 2002 and 2007 from the Faculty of Engineering, University of Waterloo. Dr. Shen is a registered Professional Engineer of Ontario, Canada, an IEEE Fellow, an Engineering Institute of Canada Fellow, and a Distinguished Lecturer of IEEE Vehicular Technology Society and Communications Society.