A Novel Multichannel Streaming Scheme to Reduce Channel Switching Delay in Application Layer Multicast

Toshiaki Ako, Hiroki Nishiyama, *Member, IEEE,* Nirwan Ansari, *Fellow, IEEE,* and Nei Kato, *Senior Member, IEEE*

***Abstract*—Video streaming services over the Internet are growing rapidly owing to the broadbandization of various networks. However, in such systems where a large number of users simultaneously access the same contents server, the convergence of the network traffic and service requests presents a significant problem. From this point of view, application layer multicast (ALM) has recently attracted much attention due to its scalability and feasibility in the current network systems. In ALM, user nodes construct a multicast tree structure to efficiently deliver an identical video streaming content. Various techniques to construct ALM trees have been designed to achieve improved performance in terms of shorter delay and higher reliability. However, in most of the existing ALM schemes, the performance in channel switching events has not been studied sufficiently; channel switching can be considered as the combination of the procedures for departing from the current and joining the future ALM trees, respectively. In general, node joining and departure processes take a certain time, which can be increased significantly depending on the network conditions whereby users are left waiting for the start of a new program on the switched channel. Therefore, reducing the channel switching delay is a significant issue in multichannel video streaming in ALM. In this paper, we propose a multichannel streaming scheme to reduce channel switching delay in ALM. Our scheme is based on an advanced ALM method dubbed network-aware hierarchical arrangement graph, which constructs node-disjoint multicast trees by utilizing the arrangement graph theory to achieve high robustness on node departures. The performance of the proposed scheme is verified through extensive computer simulations, which demonstrate that our proposed scheme succeeds in reducing the delay in channel switching without system throughput degradation.**

***Index Terms*—Application layer multicast (ALM), channel change, channel switching delay, Internet TV, multichannel streaming.**

I. INTRODUCTION

RAPID DEVELOPMENT and advances of broadband and high-speed networks are fueling widespread audio/video streaming content delivery services over the Internet. Nowadays, a large number of video streaming sites provide many users with various programs over different channels. Most of the currently served video streaming systems are based on unicast communication, i.e., users receive their streaming directly from servers in these existing systems. However, in such unicast-based video streaming systems, the network traffic convergence and the access concentration to the servers are becoming the bottleneck. As a consequence, multicast technologies, which can mitigate some issues in unicast-based schemes, have been developed for next generation video streaming delivery services.

Multicast technologies can be categorized into two types, namely, IP multicast [1] and application layer multicast (ALM) [2]. Although IP multicast has been considered as a powerful solution which is expected to completely solve the problems in video streaming services, it requires drastic changes to the existing networks, e.g., the replacement of all intermediate network equipments to enable a copy of IP packets at each point. In other words, since it is difficult to widely deploy most of the conventional IP multicast technologies in current network infrastructures, these technologies are not readily adopted for the purpose of video streaming. In contrast, the ALM based approach is quite compatible with the existing networks because it allows multicast streaming on the application layer, i.e., all the video contents are replicated at user terminals and redistributed to other users. In addition, no change is required on any of the network-side equipments. ALM has recently attracted a great deal of attention because it is a promising solution to serve single/multichannel video streaming delivery over the Internet.

In most of the existing multichannel video streaming strategies using ALM, an ALM network is used to serve each video channel. In such systems, in general, the channel switching event is similar to the combination of both the termination of the current streaming and the establishment of a new streaming event. Therefore, while users can freely switch their channels according to their preference as in TV broadcasting, they may experience, however, a non-negligible delay upon switching the channel. This channel switching delay occurs because the channel switching procedure in the multichannel streaming system by ALM networks is quite complicated as compared to that of TV broadcasting. Since long channel switching delay can greatly degrade the quality of experience (QoE) of the users, reducing channel switching delay is one

---

Manuscript received November 15, 2010; revised March 31, 2011; accepted April 4, 2011. Date of publication October 6, 2011; date of current version November 23, 2011.

T. Ako, H. Nishiyama, and N. Kato are with the Graduate School of Information Sciences, Tohoku University, Sendai 980-8578, Japan (e-mail: ako@it.ecei.tohoku.ac.jp; bigtree@it.ecei.tohoku.ac.jp; kato@it.ecei.tohoku.ac.jp).

N. Ansari is with the Department of Electrical and Computer Engineering, New Jersey Institute of Technology, Newark, NJ 07102 USA (e-mail: nirwan.ansari@njit.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JSYST.2011.2165611

of the most significant challenges that needs to be addressed in multichannel video streaming systems.

Most of the ALM techniques proposed so far such as NICE [3], Zigzag [4], and Scribe [5] construct a single-tree structure, which is rooted at a media server. Each user node participates in the tree, and relays a media stream from its parent node to its child nodes. The single-tree structure is efficient in terms of the optimization of bandwidth utilization and media delivery delay, but a node departure caused by channel switching must break video streaming. In order to improve the robustness of ALM based on the single-tree structure, multiple tree-based multicast schemes such as CoopNet [6], [7], SplitStream [8], THAG [9] have been proposed. A media stream is divided into several descriptions which are delivered through multiple multicast trees in parallel. Users can playback the content just by receiving one of the descriptions, and the quality is improved by receiving more descriptions. Multiple-tree multicast is highly robust because each node has more than one parent.

In this paper, we focus on an advanced multiple-tree ALM scheme referred to as network-aware hierarchical arrangement graph (NHAG) [10], [11], which can efficiently utilize available network capacities. By utilizing the NHAG technique, our advanced ALM scheme allows effective multichannel video streaming by efficiently utilizing spare bandwidths to reduce channel switching delay.

The rest of this paper is organized as follows. In Section II, several recently proposed techniques to reduce the channel switching delay in IP multicast and ALM are reviewed. The proposed scheme based on NHAG is described in Section IV after an overview of key algorithms in the original NHAG in Section III. The performance of the proposed scheme is verified in Section V. Concluding remarks are given in Section VI.

II. RELATED WORKS

In multichannel streaming over the Internet, the channel switching delay mainly consists of three different kinds of delay, namely, network delay, initial buffering delay, and decoding delay. Network delay is the time required to receive the first delivered data packet belonging to the new stream of a switched channel. At each user terminal, received data packets are queued in the buffer to absorb the data transmission delay jitter until the buffer occupancy reaches the proper threshold, which is referred to as the initial buffering delay. After a sufficient number of packets are received, streaming data are extracted and decoded for playing back. The decoding delay is the time required for this decoding procedure. After these procedures have been completed, users can begin to start watching a new program at the switched channel. The improvement of the delay in channel switching can be achieved by reducing one of the three delays (i.e., network delay, initial buffering delay, and decoding delay). In this section, several different approaches aiming at reducing the channel switching delay by IP multicast and ALM networks are reviewed.

A. Channel Switching Delay Improvement in IP Multicast

The most straightforward method to almost completely remove the delay in channel switching is to simultaneously receive all the channel programs, i.e., not only to receive the video that the user intends to watch but also to obtain other unwatched programs over other channels. By doing so, the network and initial buffering delays are almost equal to zero when a channel switches, given that network resources are sufficient. However, the efficiency in network resource utilization suffers greatly because most of the network traffics are useless as they contain videos, which are just received and will be discarded by users without being viewed. On the other hand, if it is possible to know to which channel each user will switch its currently watched channel, the redundant delivery just for reducing the channel switching delay can be minimized by limiting the redundant delivery only for the expected channels. In fact, several techniques aiming to reduce the channel switching delay by the prior receipt of a few number of expected channels have been developed [12]–[16]. They adopt different algorithms to predict the channel switching events.

In [12], when a user is viewing the nth channel, contiguous $(n - 1)$th and $(n + 1)$th channels are considered as potential channels to be selected at the next channel switching event. By receiving channels adjacent to the currently watched channel at the same time, users can smoothly change their channels. This method is simple and yet effective in channel surfing. In [13], viewers prefetch the popular channels, which have a large number of viewers, in addition to the adjacent channels. Since it is assumed that a popular channel is more likely to be selected as the next channel, popularity is also an important metric for predicting the next channel. In [14], the use of a centralized rating server was proposed to predict the next channel switching in each user. The rating server periodically collects records of past meaningful channel switching events from all the users. Here, meaningful data means that the user keeps watching the selected channel for more than 10 s. The rating server also collects information of the remote controller action, the previous channel, the current new channel, and so on. Thus, the channel switching prediction can be conducted by analyzing the correlation between video program's characteristics and users' channel selection behaviors. The expected channel programs for each user are served in advance of the occurrence of actual channel switching, thus decreasing the channel switching delay.

In contrast to [14], the channel selection in each user is estimated individually based on the user's preference as described in [15]. In each user terminal, the history of channel control is recorded, and is used to analyze the user's preference and habit of pushing buttons of the remote controller, and accordingly appropriate additional channels for future channel switching are determined and delivered. Lee *et al.* [16] studied not only the selection method of prejoining channels, but also the optimal number of prejoining channels. They analyzed the users' channel selection behaviors based on semi-Markov processes, and estimated the average channel switching delay and average bandwidth usage. Based on the above estimation, the optimal number of prejoining channels is obtained depending on the state of the user, i.e., whether the user continues channel switching or keeps watching one channel. This method enables appropriate bandwidth usage to reduce

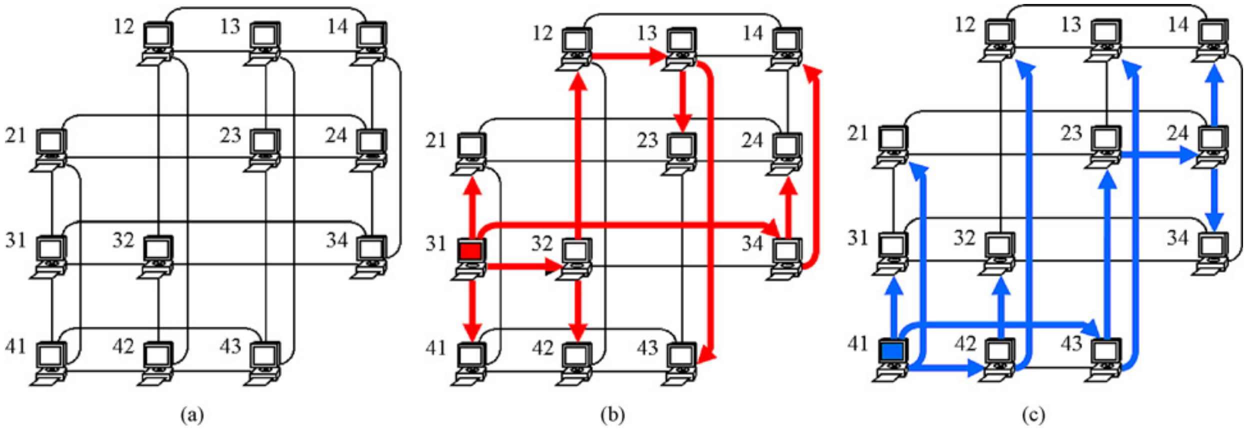

Fig. 1. Tree construction based on arrangement graph. (a) Arrangement graph with size of 4. (b) Tree with 31 as root. (c) Tree with 41 as root.

channel switching delay according to the viewing state of the user.

Decreasing the initial buffering delay is another approach to reduce the channel switching delay. In [17], the accelerator servers which can distribute the copy of the original video stream in a coordinated way with the main server are installed in the system. Upon deciding to switch the channel, each user sends the request for the delivery of the new video program to the main server as well as accelerator servers. Since the main and accelerator servers cooperatively deliver the stream without duplication, the initial buffering delay can be greatly decreased by choosing multiple accelerator servers.

The studies in [18]–[20] focused on reducing the decoding delay in video streaming based on moving picture expert group (MPEG) technology. In the MPEG-2 and H.264/MPEG advanced video coding technologies, the encoded video data consists of three different frames, namely, I-frame, P-frame, and B-frame at a certain ratio. The I-frame is a reference image, and the P-frame and B-frame include information on the difference among adjacent frames. Therefore, even if some P-frames and B-frames have already been received by a user right after switching a channel, it is impossible to play back the video until the I-frame reaches the user. As a consequence, there is a certain decoding delay. To deal with this issue, the works in [18] and [19] proposed to increase the frequency of I-frames. By decreasing the interval between I-frames, the decoding delay can be reduced. On the other hand, multicast assisted zap acceleration [20] presents an alternate approach to decrease the waiting time for receiving the first I-frame. In this system, sub-channels for sub-streams, which are generated by shifting the original stream over different time periods in the time-scale, are served. Thus, each user is able to minimize its decoding delay by selecting an appropriate sub-channel capable of delivering the I-frame as quickly as possible.

B. Channel Switching Delay Improvement in ALM

Although many ALM technologies have been developed in the past decade [3]–[11], almost all of them assume that different contents are independently served by using different ALM trees. Therefore, in general, the channel switching in ALM is similar to changing the participating ALM tree. This is the reason why the degree of channel switching delay in ALM is significantly higher in contrast with that in IP multicast. In ALM, each tree is rooted at the video streaming server, and the participating users are allocated to the interior nodes or leaf nodes of the tree. The interior nodes are required to distribute the stream from its parent node to its child nodes. Leaf nodes only receive the stream from its parent node. In such an application layer network system, when a user changes its participating ALM tree to switch the currently viewed video program, procedures to leave the current ALM tree and to join the new ALM tree have to be completed before starting the process to receive the stream of the new video program. Since it is difficult to skip these necessary procedures to maintain ALM trees where nodes (users) can freely leave and join the trees, achieving small channel switching delay in ALM remains an open and challenging issue.

View upload decoupling (VUD) [21], [22] is one of the few technologies developed to solve the channel switching delay problem. In VUD, ALM trees to deliver each video stream are constructed regardless of the video program being viewed by the node. In other words, nodes of an ALM tree must be required to distribute even those video streams, which are not currently viewed by the nodes. To receive the stream of the desired video program, each node needs to set up a connection to any node in the ALM tree in delivering the video. In this scheme, there is no need to reconstruct ALM trees at any time even if users switch the channels being watched. Therefore, the channel switching delay can be successfully decreased. However, VUD is ineffective especially when there is not enough bandwidth for ALM, because most of the nodes are required to consume their upload and download bandwidths to distribute the channels that they are not viewing. Consuming bandwidth to distribute contents which are not viewed is wasting valuable network resources.

As mentioned above, in the development of multichannel streaming technology for ALM where each user may have different available bandwidth, efficient utilization of their limited bandwidth should be considered. From this point of view, we focus on the advanced multiple-tree-based ALM scheme, namely, network-aware hierarchical arrangement graph or NHAG [10], [11]. While NHAG has been developed for single-channel streaming services in ALM, its performance is superior to the other schemes in terms of the robustness to the changes in ALM trees and the adaptability to heterogeneous physical networks as will be described in the following section. Therefore, we aim to enhance and tailor NHAG for multichannel streaming in ALM to reduce channel switching delay.

III. NHAG

NHAG is a promising ALM technique for video streaming services over the Internet. In this section, we review the motivation of NHAG, and briefly describe the procedures for multicast tree construction and node joining/leaving.

A. *Motivation and Overview of NHAG*

The most notable feature of NHAG is its robustness to node departure. Since video contents are distributed over the overlay networks where nodes freely join and leave the network anytime, the streaming delivery to each node can be suddenly interrupted by the departure of one of its parent nodes that relays the stream. Therefore, robustness to node departure is one of the most important aspects to be taken into account to construct and maintain overlay networks suitable for video streaming services in ALM. To guarantee continuous distribution of content streams, departing nodes must perfectly complete the necessary leaving procedures to enable the alternative delivery of the streams. However, in the Internet, all the nodes may not be able to successfully perform such a procedure because of unexpected termination of connections owing to physical link failures, network congestions, and so forth.

To achieve robustness to node departure, NHAG adopts the multiple-tree-based multicast technique based on multiple description coding (MDC) [23]–[25]. MDC is a sophisticated coding technology, which creates several streams referred to as descriptions from the original stream. In general, all descriptions originated from the same stream have equal bit-rates. While each description is a low quality copy of the original stream, higher quality contents can be obtained by receiving more descriptions together. In NHAG, each description is distributed by using different node-disjoint multicast trees, and each node participates in more than one tree. In node-disjoint multicast trees which can be constructed by using arrangement graph (AG) theory [26] as depicted in Fig. 1, interior nodes in each tree correspond to leaf nodes in all other trees. Therefore, multiple descriptions are delivered to each node through parallel node-disjoint multicast trees. However, the playing back quality of stream contents in each node can be degraded by the departure of its parent nodes. Furthermore, the probability of the occurrence of the stream interruption is quite low, because the descendant node is able to receive descriptions via other alive trees.

In addition to the above-mentioned advantage, NHAG has other strengths in terms of network scalability and heterogeneity. In the construction of node-disjoint trees based on an arrangement graph, the size of the graph needs to be determined prior to acquiring the information regarding the actual number of nodes, which are to join the network [i.e., the AG]. In other words, the capacity of an AG is limited. However, this constraint can be removed by constructing a tree-like hierarchical structure of AGs as shown in Fig. 2. When the number of nodes exceeds the total capacity of the existing AGs, a child AG is newly configured, and the descriptions are delivered from its parent AG to the child AG. Furthermore, the size of each AG can be adjusted in NHAG in order to adapt to the heterogeneity of the network. In NHAG, nodes belonging to a larger AG may be required to distribute many descriptions. This implies that the nodes have to consume much more upload bandwidth to send out descriptions. Therefore, NHAG sets up different sizes of AGs to allow nodes to join appropriate AGs according to their upload bandwidths. Owing to these notable advantages of NHAG, we decide to tailor NHAG to reducing the channel switching delay in multichannel streaming in ALM.

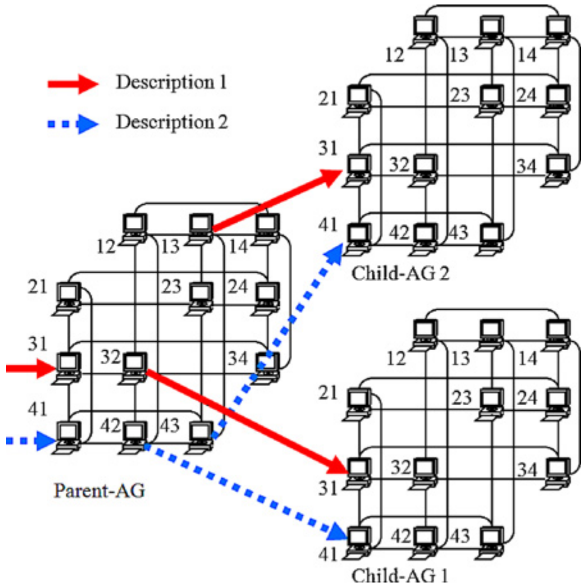

Fig. 2. Hierarchical AG in NHAG.

B. *Algorithm to Construct Node-Disjoint Trees*

Node-disjoint trees can be constructed based on an AG. The AG is identified by two integers, s and t ($1 \leq t < s$), and represented as an undirected graph, $A_{s,t} = (V, E)$, which is defined as follows [26]:

$$\begin{aligned} V &= \{X = x_1 x_2 \ldots x_t | x_i \in \langle s \rangle \text{ and } x_i \neq x_j \text{ for } i \neq j\} \\ E &= \{(X, Y) \mid X \in V,\ Y \in V,\ \exists i,\quad x_i \neq y_i \\ &\qquad\qquad\qquad \text{else for } j \neq i,\ \ x_j = y_j\}. \end{aligned}$$

It should be noted that we denote $\langle s \rangle = \{1, 2, \ldots, s\}$, and x_i is the ith element of X. NHAG uses $A_{s,2}$ to construct $(s - 2)$ node-disjoint trees. Here, s indicates the size of the AG. Fig. 1(a) shows $A_{4,2}$ with the size of 4. The node-disjoint trees are constructed according to the following procedure.

1) Node $k1$ ($3 \leq k \leq s$) becomes the root.
2) Node $k1$ ($3 \leq k \leq s$) becomes the parent of Node kx ($2 \leq x \leq s,\ \ x \neq k$).
3) Node kx ($1 \leq x \leq s,\ \ x \neq k$) becomes the parent of Node yx ($1 \leq y \leq s,\ \ x \neq y$).
4) If k is odd, Node $1(k - 1)$ becomes the parent of Node $1k$. In addition, Node $1k$ becomes the parent of Node zk ($2 \leq z \leq s,\ \ z \neq k$).

5) If k is even, Node $2(k-1)$ becomes the parent of Node $2k$. In addition, Node $2k$ becomes the parent of Node zk $(1 \leq z \leq s,\ z \neq k,\ z \neq 2)$.

It should be noted that k, x, y, and z are integers. After carrying out the above steps, node-disjoint trees are constructed in the AG. Fig. 1(b) and (c) shows the node-disjoint trees, which can be obtained by applying the above steps to $A_{4,2}$. It can be confirmed from Fig. 1(b) and (c) that the parent node in each tree is a leaf node in another tree. In general, there exist some nodes which are leaf nodes in all the trees. One of them is selected as a specific node referred to as "AG entrance" which takes the role of maintaining the status of all AGs and their members. For example, in Fig. 1, Node 21 assumes the role of the AG entrance of $A_{4,2}$.

The maximum number of nodes, which can join the AG with the size equal to s, is $s(s-1)$. When the actual number of nodes participating in an AG reaches its limit, the AG becomes a parent-AG of the newly configured child-AG. To distribute a description to the child-AG, a node in the parent-AG acts as a parent node providing the root in the child-AG with the description. The set of the candidate nodes delivering descriptions to child-AGs are given by the following definition:

$$\{ij \mid i \neq j, i \neq (j \bmod 2) + 1\}(1 \leq i \leq s, 1 < j < s). \quad (1)$$

As shown in Fig. 2, the nodes located on the same column in the parent-AG deliver descriptions to the same child-AG. Here, the set of nodes positioning on the first column are referred to as the AG source. If the size of the child-AG is equal to that of the parent-AG, all of descriptions distributed in the parent-AG are provided to the child-AG. By this hierarchical manner, a large scale streaming delivery networks can be easily achieved.

In an AG, the maximum number of child nodes belonging to a parent node is expressed in terms of the size of the AG, s, i.e., $2(s-2)$. Therefore, the nodes joining the AG with size of s should have a greater upload bandwidth than $2(s-2)r$ where r indicates the bit-rate of each description. In other words, a node having the available upload bandwidth equal to BW is recommended to join the AGs with size smaller than or equal to s_R. This may be formulated as follows:

$$s_R = \left\lfloor \frac{BW}{2 \times r} + 2 \right\rfloor . \quad (2)$$

Here, it should be noted that the minimum size of AGs is three. So, nodes having their s_R values below three are permitted to participate in the AG with size of three. In the node joining process, the value of s_R calculated by each joining node is used to check whether the AG selected by the node is appropriate for itself in terms of the availability of upload bandwidth.

C. Node Joining Procedure

In the node joining procedure, a node looks for a joinable AG, and sends a join request to the AG entrance. Upon receiving the join request, the AG entrance confirms if it is possible to admit the request by comparing its AG size, s, with the requested size, s_R, which is notified by the joining node.

1) In the case where $s > s_R$, it is expected that the joining node is not able to transfer all the descriptions. Therefore, if possible, the AG creates a child-AG with a proper size according to the requested size, and then allows the node to join the child-AG. Otherwise, the AG entrance encourages the joining node to send a new join request to its child-AG with size closest to s_R. If the AG has no child-AG, the joining node is permitted to temporarily participate in the AG.
2) In the case where $s \leq s_R$, the AG entrance allows the joining node to participate in the AG if the AG is not full. Otherwise, the AG entrance compares the requested size of the joining node with the corresponding value of each node, which already exists in the AG. If there is no existing node having a smaller value of s_R than the requested size of the joining node, the AG entrance informs the joining node of the best available child-AG. On the other hand, if there are existing nodes having a smaller value of s_R than the requested size of the joining node, a node having the smallest value of s_R is replaced with the joining node. If multiple existing nodes have the same smallest value of s_R, the network distances between each node and the AG calculated by using network coordinate technologies such as GNP [27] and Vivaldi [28] are used to choose one of them. The node farthest from the AG is replaced with the joining node. The replaced node tries to join the proper child-AG recommended by the AG.

Thus, each node, which has a wider or narrower available upload bandwidth, participates in a higher or lower position in the hierarchical structure of the AGs, respectively. However, these procedures in node joining can become a factor for increasing the channel switching delay in ALM-based multichannel streaming systems where the channel switching event is similar to the combination of node leaving and joining procedures.

D. Node Leaving Procedure

When nodes leave their current AGs, all of their roles need to be delegated to alternative nodes. If the departing node is not the AG entrance, all of its tasks are undertaken by its parent node in the same tree. In contrast, when the AG entrance leaves the AG, both the information on the AG's structure and the function as an AG entrance need to be conveyed to its parent node. On the other hand, if the departing node has child-AGs, one of the non-root nodes in child-AGs is selected as a replacement of the departing node. In the child-AG having the provision of replacement of the parent-AG, similar maintenance can be performed afterwards. Thus, the node departure procedure is not simple because the tree structure needs to be always maintained to serve stable content streaming without any break, which might lead to the increase in the channel switching delay. However, the leaving procedure is performed in parallel with the joining procedure during channel switching. As a consequence, the switching delay is primarily attributed to the joining procedure.

IV. ENHANCEMENT OF NHAG FOR MULTICHANNEL STREAMING DELIVERY

As described in the previous section, NHAG is a promising solution to provision feasible and robust ALM streaming services. However, its advantage is limited to single-channel streaming, i.e., channel switching delay is incurred in applying NHAG for multichannel streaming. In this section, we propose a new multichannel streaming technology based on NHAG to reduce channel switching delay.

A. *Issue in Using NHAG Only for Multichannel Streaming Delivery*

NHAG constructs efficient multicast trees, which address the various bandwidth constraints in heterogeneous networks. However, at the price of having such advantages, the node joining procedure may take a long time to be completed. The joining node has to repeatedly send the participation request, in order, from the higher positioned larger AGs to the lower positioned smaller AGs until its joining request is accepted as shown in Fig. 3. Since no description is delivered to the node before completing the joining procedure, the waiting time can become significantly long, thus degrading the user's QoE. In fact, it is obvious that the nodes having narrow upload bandwidths tend to experience larger channel switching delay because those nodes are preferentially allocated to the lower positioned small AGs for efficient streaming delivery. In a multichannel streaming delivery system where each independent ALM overlay network for each channel is constructed by just using the NHAG technique, the channel switching event involves the node joining procedure. This can incur a significant channel switching delay. Therefore, enhancing NHAG in reducing the channel switching delay is necessary.

B. *Overview of the Proposed Method*

To address the channel switching delay issue in NHAG, we focus on the unused upload bandwidths in AGs. Our key idea is to allow AGs distribute different video contents (channels) to exchange their descriptions amongst one another by utilizing their spare upload capacities. The exchanged description can be distributed in each AG. By doing so, each node can receive additional descriptions of a new channel to which it intends to switch its channel before completing all of the joining procedures. Since the proposed scheme just uses unused bandwidth capacity, the channel switching delay can be reduced without causing network congestion and degrading the performance of delivering the main streams in NHAG.

In this method, each AG has cooperative relationships with certain AGs in other channels. The AG describes each of the other AGs as a "partner-AG" in this relationship. We will explain the selection and renewal of the partner-AGs in detail in Section IV-C. Based on this relationship, each description is exchanged between the AG and the partner-AGs constantly. This description exchange is implemented by using spare upload bandwidth in the AGs. We describe which nodes have spare upload bandwidth in an AG in Section IV-D. In order to exchange the descriptions, two roles are provided, namely, "transmitter" and "intermediate" roles to those nodes having spare upload bandwidth in each AG. Here, a stream is split into several descriptions with low quality, and relayed along multiple trees. When the transmitter node receives descriptions at a certain time, it selects a description from the set of available descriptions. Then, the transmitter node sends the selected description to the intermediate node of the partner-AG. By doing this, the channel's media data is delivered to the intermediate node which is viewing a different channel. The intermediate node keeps the description until the next piece of the description is transferred from that partner-AG.

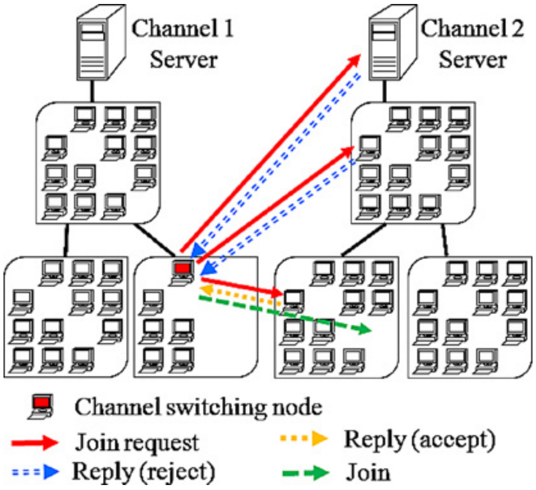

Fig. 3. Node joining procedure.

Basically, when a node requests channel switching, the node sends a join request for the new channel. In our proposed method, additionally, the node sends a request to the intermediate node in its AG. When the intermediate node receives the request, it transfers the most recently received description of the requested channel to the node. Thus, the node can receive the description of the new channel before the joining procedure to the new channel is completed. This method enables smooth channel switching in the multichannel streaming in NHAG. We describe the roles and the selection method of the transmitter and intermediate nodes in more detail in Section IV-E.

C. *Partner-AG*

In this subsection, we show how cooperative relationships between AGs in different channels are established. The search for the partner-AG is performed by the AG entrance, denoted by $E1$, when the AG is created. First of all, $E1$ sends a request for partners to the highest AG's entrance of the other channels' multicast trees. The requested AG's entrance, denoted by $E2$, reacts as follows.

1) If there is an existing cooperative relationship with another AG in $E1$'s channel, the entrance advises $E1$ to send the request to the $E2$'s child-AG.
2) If there is no relationship with $E1$'s channel, $E2$ registers $E1$'s AG as its partner-AG, and sends response of acceptance to $E1$.

The search for partner-AG is repeated according to the above procedure until the request is accepted or rejected by the lowest AG in that channel. If the cooperative relationship is not established, $E1$ starts the search again after the elapse of a certain period of time.

Between partner-AGs, the information of the AG entrance, transmitter, and intermediate node is exchanged whenever this information is updated. The updated information is transferred

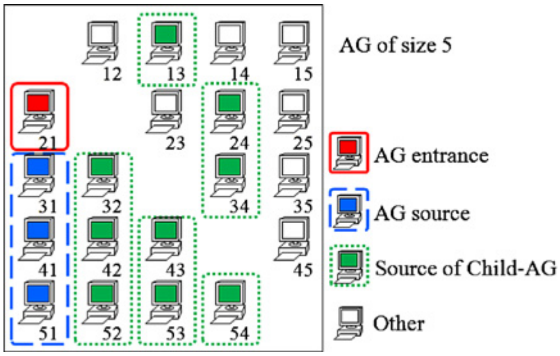

Fig. 4. Role of each node in AG of size 5.

by the AG entrance. When the AG is disrupted owing to the departure of nodes, the AG entrance sends messages regarding these departure events to the partner-AG entrance in order to break the relationship.

D. *Spare Upload Bandwidth in the AG*

The node-disjoint trees are constructed as shown in Section III-B. The construction of these node-disjoint trees implies heterogeneous bandwidth usage, depending on the node's position in the AG. However, each node has the chance to be an AG source, which requires the most bandwidth usage to deliver descriptions. Therefore, each node has spare bandwidth based on its position in the AG. The difference of these spare upload bandwidths results from whether the node is the AG entrance, the AG source, a node which is the source of the child-AG, or a simple node. Fig. 4 depicts the roles of the nodes in an AG of size 5.

In the case of a full AG, we denote the spare upload bandwidth of Node ij as S_{ij}, as described in (3). In the expression, s is the size of the AG. Each description's rate is denoted by r. The upload bandwidth of Node ij is expressed as BW_{ij}. C_j denotes the number of child-AGs belonging to row j. Note that BW_{ij} is basically larger than $2(s-2)r$. The computation of S_{ij} does not require any additional communication, and S_{ij} is estimated by using values which are required for AG management

$$S_{ij} = \begin{cases} BW_{ij}, & \text{if } i = 2 \text{ and } j = 1 \\ BW_{ij} - 2(s-2)r, & \text{if } i > 2 \text{ and } j = 1 \\ BW_{ij} - (s-2)r - C_j r, & \text{if } 1 < j < s \text{ and } i \neq (j \bmod 2) + 1 \\ BW_{ij} - r, & \text{if } 1 < j < s \text{ and } i = (j \bmod 2) + 1 \\ BW_{ij}, & \text{if } j = s \text{ and } i = (j \bmod 2) + 1 \\ BW_{ij} - (s-2)r, & \text{if } j = s \text{ and } i \neq (j \bmod 2) + 1. \end{cases} \tag{3}$$

E. *Transmitter and Intermediate Nodes*

The transmitter and intermediate nodes in an AG are selected by the AG entrance. First, the AG entrance computes the spare upload bandwidth of each node in the AG. Then, the nodes, which have spare upload bandwidth to transmit to the partner-AG, become the candidates for the transmitter or intermediate nodes. The AG entrance prefers to assign the higher node in the multicast tree as the transmitter node. This enables the transmission of the description to the partner-AG as fast as possible. In contrast, the intermediate node is required to be able to transfer descriptions even if more than one node in the AG require channel switching at the same time. Therefore, the lower node in the multicast tree is assigned to be the intermediate node, since it tends to have enough spare upload bandwidth to send more than one description. This assignment is performed for each channel, and in some cases a node acts as the transmitter/intermediate node for more than one channel.

Transmitter and intermediate nodes are assigned for each channel as described above. Since the network conditions of each node are highly fluctuant and heterogeneous, transmitter and intermediate nodes are updated after the elapse of a certain period of time or when the positions of nodes in the AG are updated. When the entrance or transmitter or intermediate node is changed, the AG entrance sends the updated information to the partner-AGs.

We shall next describe the process of exchanging descriptions. Every time when the transmitter node receives the descriptions being watched currently, it selects a description from a set of the available descriptions which are the most recently received, and sends it to the intermediate node of the partner-AG. Note that this communication is performed only by using spare upload bandwidth. Therefore, the transmission of a description never affects the delivery of the currently viewed channel. When the intermediate node receives a description from the transmitter node of the partner-AG, it keeps the description until the next piece of description of that channel is received. When the intermediate node is requested for the channel's description, it responds to the request. This communication is also performed by using only spare upload bandwidth. The node that requests channel switching can receive the description of the new channel in only one step, by requesting the intermediate node. Thus, the node can get the description before joining the new channel's multicast tree, and thus the channel switching delay will be reduced.

V. PERFORMANCE EVALUATION

A. *Simulation Setup*

We used network simulator ns-2 [29] to evaluate our proposed method. In our simulations, the underlying network topology is the Transit-Stub topology generated by GT-ITM [30]. In the simulations, the backbone network consists of a transit domain and 10 stub domains. There are 10 routers per transit domain, and 100 routers per stub domain. Each router in a transit domain has a connection to one of the routers in a certain stub domain. Between a pair of routers in a transit domain, there is a connection 50% of the time. Similarly, between a pair of routers in a stub domain, there is a connection 10% of the time. The total numbers of routers and connections are 1010 and about 5000, respectively. Each connection's link delay is randomly set between 1 and 10 ms. Each participating end-node is randomly connected to a router in the stub domain, and one of them is randomly selected as a media server. Here, all nodes are assumed to know where the media servers are in the network. The other simulation parameters are shown in Table I.

In this simulation, we do not take into account the viewer rate and the preference of channel selection. Therefore, we

TABLE I
PARAMETERS OF NHAG

Parameter	Value
Total number of end-nodes	90–900
Number of channels	3
Number of descriptions divided by MDC	4
Each description rate	500 [kb/s]
Upload bandwidth of each end-node	2–5 [Mb/s]
Download bandwidth of each end-node	10 [Mb/s]

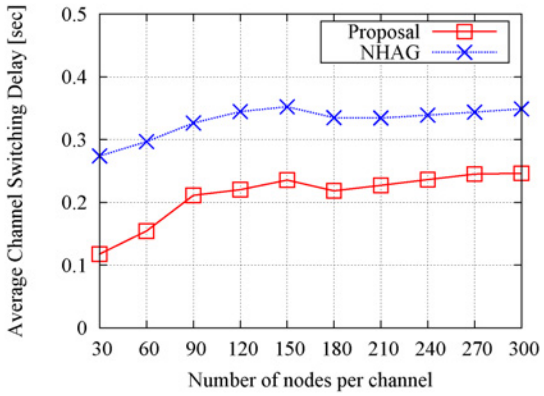

Fig. 5. Average channel switching delay (average CSD).

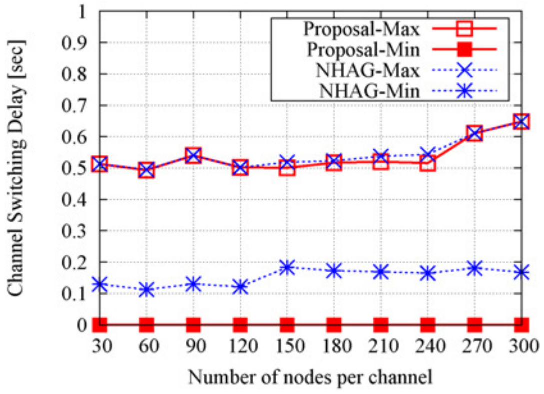

Fig. 6. Maximum and minimum of the channel switching delay (max-min CSD).

assume that the new channel to which the user switches is randomly selected, and we use the scenario described below.

1) After the simulation has commenced, each node joins the channel of which the multicast tree has the least number of nodes. The nodes join the channel one by one every 2 s.
2) After all the nodes have joined, each node switches to a randomly chosen new channel one by one every 2 s.
3) After all nodes have switched to the new channels, each node leaves the multicast tree one by one every 2 s.

We compare the performance of our proposal to that of the original NHAG in terms of the following figure merits: average channel switching delay (average CSD), maximum and minimum of the channel switching delay (Max-Min CSD), total throughput, smooth switching rate (SSR), and cumulative distribution function (CDF) of the channel switching delay.

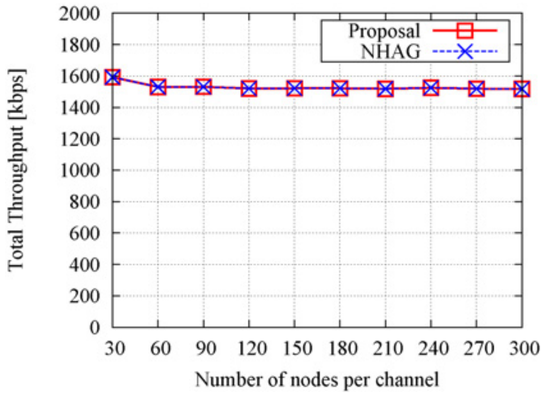

Fig. 7. Total throughput.

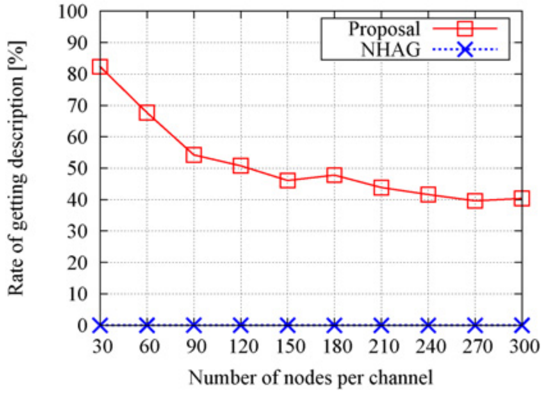

Fig. 8. SSR.

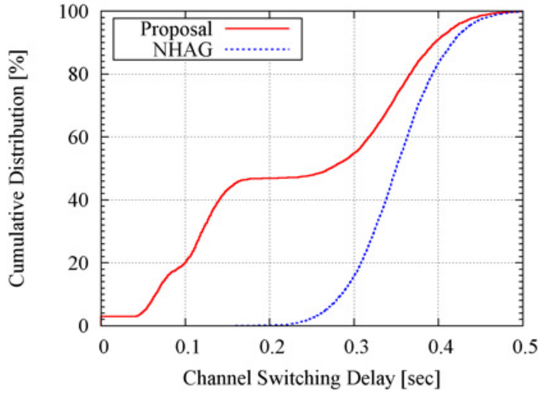

Fig. 9. CDF with 150 nodes per channel.

The CSD is the time interval between the start of the transmission of the channel switching request and the reception of the first description of the requested channel. We use the average, maximum, and minimum of the CSD as the metrics. Total throughput is the total of all the considered descriptions' throughputs that each node is viewing. Here, the descriptions which are transferred to the intermediate node are not included in the throughput. The SSR is the ratio of the number of nodes which received description from the intermediate node before the completion of the joining procedure to the number of nodes which switched channels. The CDF indicates the ratio of the number of nodes which have succeeded in switching with the delays smaller than certain values to the total number of nodes which have switched channels. We adopted the case with 150 nodes per channel for estimating CDF.

B. Simulation Results

1) *Average CSD:* As depicted in Fig. 5, basically in NHAG, the average CSD increases with the increase of the number of nodes per channel. This is because the joining requests tend to be rejected in the higher position of the multicast tree, and take a long time to be accepted. On the other hand, our proposed enhancement to NHAG results in about 0.1s faster switching than that of the original NHAG. In our proposed method, if the channel switching procedure takes a long time, the switching node can receive description of that channel from the intermediate node in the AG.

2) *Max-Min CSD:* Fig. 6 shows the maximum and minimum of the channel switching delay. Our proposed method realizes 0 s at the minimum of the CSD, when the switching node has worked as the intermediate node of that channel. In such a case, the node has already received the description before sending requests for the new channel. The maximum of the CSD, i.e., in the worst case, achieved by our proposed scheme is no greater than that by NHAG. Even if the node fails to receive the description from the intermediate node, it can continue the usual joining procedure in our proposed method.

3) *Total Throughput:* Our proposed method and NHAG achieved almost the same total throughput in terms of receiving the descriptions of the currently viewed channel, as shown in Fig. 7. This means that our proposed method achieves smooth channel switching without adverse effects on the multicast relay of the currently viewed channel. By using spare upload bandwidth, the total throughput is not degraded in our proposed method.

4) *SSR:* As evident from the result shown in Fig. 8, the SSR of NHAG is zero in all cases because there is no enhancement for multichannel content delivery that reduces CSD. On the other hand, in the proposed method, high SSR is achieved. In particular, in the case where there are 30 nodes per channel, over 80% SSR is achieved when the proposed technique is applied. However, this rate is decreased with the increase of the number of nodes per channel. In our proposed method, each AG can have only one partner-AG per channel. Therefore, when the number of nodes is biased toward any channel, several AGs fail to get partner-AGs of less watched channels owing to the imbalance of the numbers of AGs. Since a larger number of nodes lead to greater imbalance, the SSR should degrade as the total number of nodes increases. To achieve high SSR in all the considered scenarios, the required bandwidth should exceed the spare bandwidth in NHAG, and this problem will be addressed in our future work.

5) *CDF:* As shown in Fig. 9, over 40% of channel switching events are achieved within 0.15 s in the proposed method with 150 nodes per channel. In contrast, the original NHAG requires at least 0.2 s to complete channel switching. In our method, only about 3% of the nodes switch channels without any delay, and they are intermediate nodes which receive the other channels' descriptions periodically. Therefore, our method realizes smooth channel switching with a small number of intermediate/transmitter nodes which require additional download/upload bandwidth usages.

In our simulation results, the proposed method achieves lower CSD than the original NHAG without degrading the throughput of the main stream. Our proposal can achieve higher QoE with smooth channel switching in multichannel streaming service.

VI. CONCLUSION

Reducing the channel switching delay is one of the critical issues in multichannel streaming. In this paper, we reviewed our recently proposed NHAG, an ALM method which splits a stream into several descriptions with MDC and delivers them along node-disjoint multicast trees. NHAG provides robust and efficient stream delivery in heterogeneous networks. However, the original NHAG causes long channel switching delay. This delay is attributed to the node re-joining procedure to the new channel's multicast tree. In order to reduce this delay, we proposed a method which enables nodes to get the new channel's streaming data before finishing the node re-joining procedure. In our method, descriptions are exchanged between different channels, and descriptions from other channels are kept for a certain period of time in the AG. When a node requests to change its channel, it can get the new channel's description within its AG. We simulated our proposed method, and the results clearly indicate that our method achieves significant reduction of channel switching delays while maintaining a high throughput of the currently viewed channel.

REFERENCES

[1] A. M. Hamad and A. E. Kamal, "A survey of multicasting protocols for broadcast-and-select single-hop networks," *IEEE Netw.*, vol. 16, no. 4, pp. 36–48, Jul.–Aug. 2002.

[2] M. Hosseini, D. T. Ahmed, S. Shirmohammadi, and N. D. Georganas, "A survey of application-layer multicast protocols," *IEEE Commun. Surveys Tuts.*, vol. 9, no. 3, pp. 58–74, Jul.–Sep. 2007.

[3] S. Banerjee, B. Bhattacharjee, and C. Kommareddy, "Scalable application layer multicast," in *Proc. ACM Special Int. Group Data Commun.*, Aug. 2002, pp. 205–217.

[4] D. A. Tran, K. A. Hua, and T. T. Do, "A peer-to-peer architecture for media streaming," *IEEE J. Sel. Areas Commun.*, vol. 22, no. 1, pp. 121–133, Jan. 2004.

[5] M. Castro, P. Druschel, A.-M. Kermarrec, and A. I. T. Rowstron, "Scribe: A large-scale and decentralized application-level multicast infrastructure," *IEEE J. Sel. Areas Commun.*, vol. 20, no. 8, pp. 1489–1499, Oct. 2002.

[6] V. N. Padmanabhan, H. J. Wang, P. A. Chou, and K. Sripanidkulchai, "Distributing streaming media content using cooperative networking," in *Proc. ACM Int. Workshop Netw. Operating Syst. Support Digital Audio Video*, May 2002, pp. 177–186.

[7] V. N. Padmanabhan, H. J. Wang, and P. A. Chou, "Resilient peer-to-peer streaming," in *Proc. IEEE Int. Conf. Netw. Protocol*, Nov. 2003, pp. 16–27.

[8] M. Castro, P. Druschel, A.-M. Kermarrec, A. Nandi, A. Rowstron, and A. Singh, "SplitStream: High-bandwidth content distribution in cooperative environments," in *Proc. Int. Workshop Peer-to-Peer Syst.*, Oct. 2003, pp. 298–313.

[9] R. Tian, Q. Zhang, Z. Xiang, Y. Xiong, X. Li, and W. Zhu, "Robust and efficient path diversity in application-layer multicast for video streaming," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 8, pp. 961–972, Aug. 2005.

[10] M. Kobayashi, H. Nakayama, N. Ansari, and N. Kato, "Robust and efficient stream delivery for application layer multicasting in heterogeneous networks," *IEEE Trans. Multimedia*, vol. 11, no. 1, pp. 166–176, Jan. 2009.

[11] M. Kobayashi, H. Nakayama, N. Ansari, and N. Kato, "Reliable application layer multicast over combined wired and wireless networks," *IEEE Trans. Multimedia*, vol. 11, no. 8, pp. 1466–1477, Dec. 2009.

[12] C.-C. Sue, C.-Y. Hsu, Y.-S. Su, and Y.-Y. Shieh, "A new IPTV channel zapping scheme for EPON," in *Proc. Int. Conf. Ubiquitous Future Netw.*, Jun. 2009, pp. 131–136.

[13] U. Oh, S. Lim, and H. Bahn, "Channel reordering and prefetching schemes for efficient IPTV channel navigation," *IEEE Trans. Consum. Electron.*, vol. 56, no. 2, pp. 483–487, May 2010.

[14] J. Lee, G. Lee, S. Seok, and B. Chung, "Advanced scheme to reduce IPTV channel zapping time," in *Proc. APNOMS*, 2007, pp. 235–243.

[15] Y. Kim, J. K. Park, H. J. Choi, S. Lee, H. Park, J. Kim, Z. Lee, and K. Ko, "Reducing IPTV channel zapping time based on viewer's surfing behavior and preference," in *Proc. IEEE Int. Symp. Broadband Multimedia Syst. Broadcasting*, May–Apr. 2008, pp. 1–6.

[16] 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. Broadcast.*, vol. 56, no. 3, pp. 321–330, Sep. 2010.

[17] C. Sasaki, A. Tagami, T. Hasegawa, and S. Ano, "Rapid channel zapping for IPTV broadcasting with additional multicast stream," in *Proc. IEEE Int. Conf. Commun.*, May 2008, pp. 1760–1766.

[18] J. Boyce and A. Tourapis, "Fast efficient channel change [set-top box applications]," in *Proc. Int. Conf. Consum. Electron.*, Jan. 2005, pp. 1–2.

[19] U. Jennehag and S. Pettersson, "On synchronization frames for channel switching in a GOP-based IPTV environment," in *Proc. Consum. Commun. Netw. Conf.*, Jan. 2008, pp. 638–642.

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

[21] D. Wu, C. Liang, Y. Liu, and K. Ross, "View-upload decoupling: A redesign of multichannel P2P video systems," in *Proc. IEEE INFOCOM*, Apr. 2009, pp. 2726–2730.

[22] D. Wu, Y. Liu, and K. W. Ross, "Modeling and analysis of multichannel P2P live video systems," *IEEE/ACM Trans. Netw.*, vol. 18, no. 4, pp. 1248–1260, Aug. 2010.

[23] M. Alasti, K. Sayrafian-Pour, A. Phremides, and N. Farvardin, "Multiple description coding in networks with congestion problem," *IEEE Trans. Inf. Theory*, vol. 47, no. 3, pp. 891–902, Mar. 2001.

[24] V. K. Goyal, "Multiple description coding: Compression meets the network," *IEEE Signal Process. Mag.*, vol. 18, no. 5, pp. 74–93, Sep. 2001.

[25] V. Stankovic, R. Hamzaoui, and Z. Xiong, "Robust layered multiple description coding of scalable media data for multicast," *IEEE Signal Process. Lett.*, vol. 12, no. 2, pp. 154–157, Feb. 2005.

[26] K. Day and A. Tripathi, "Arrangement graphs: A class of generalized star graphs," *Inf. Process. Lett.*, vol. 42, no. 5, pp. 235–241, Jul. 1992.

[27] T. S. E. Ng and H. Zhang, "Predicting Internet network distance with coordinates-based approaches," in *Proc. IEEE Conf. Comput. Commun.*, Jun. 2002, pp. 170–179.

[28] F. Dabek, R. Cox, F. Kaashoek, and R. Morris, "Vivaldi: A decentralized network coordinate system," in *Proc. ACM Special Int. Group Data Commun.*, Aug. 2004, pp. 15–26.

[29] *The Network Simulator—ns-2* [Online]. Available: http://www.isi.edu/nsnam/ns

[30] E. W. Zegura, K. L. Calvert, and S. Bhattacharjee, "How to model an Internetwork," in *Proc. IEEE Conf. Comput. Commun.*, Mar. 1996, pp. 594–602.

Toshiaki Ako received the B.E. degree in information engineering from Tohoku University, Sendai, Japan, in 2010. Currently, he is pursuing the M.S. degree with the Graduate School of Information Sciences, Tohoku University.

His current research interests include multimedia systems and overlay networks.

Hiroki Nishiyama (M'08) received the M.S. and Ph.D. degrees in information science from Tohoku University, Sendai, Japan, in 2007 and 2008, respectively.

He was a Research Fellow with the Japan Society for the Promotion of Science (JSPS) during his graduate and doctoral studies, after which he has become an Assistant Professor with the Graduate School of Information Sciences, Tohoku University. His active areas of research include traffic engineering, congestion control, satellite communications, ad hoc and sensor networks, and network security.

Dr. Nishiyama received Best Paper Awards from the IEEE Global Communications Conference in 2010, as well as from the 2009 IEEE International Conference on Network Infrastructure and Digital Content. He was a recipient of the 2009 FUNAI Foundation's Research Incentive Award for Information Technology. He is a member of the Institute of Electronics, Information and Communication Engineers.

Nirwan Ansari (S'78–M'83–SM'94–F'09) received the B.S.E.E. (summa cum laude with a perfect GPA) degree from the New Jersey Institute of Technology (NJIT), Newark, in 1982, the M.S.E.E. degree from the University of Michigan, Ann Arbor, in 1983, and the Ph.D. degree from Purdue University, West Lafayette, IN, in 1988.

He joined the Department of Electrical and Computer Engineering, NJIT, as an Assistant Professor in 1988, tenured and promoted to an Associate Professor in 1993, and has been a Full Professor since 1997. He has also assumed various administrative positions with NJIT. He authored *Computational Intelligence for Optimization* (Berlin, Germany: Springer, 1997, translated into Chinese in 2000) with E. S. H. Hou, and edited *Neural Networks in Telecommunications* (Springer, 1994) with B. Yuhas. He has contributed over 350 technical papers, over one third published in widely cited refereed journals/magazines. His current research interests include various aspects of broadband networks and multimedia communications.

Dr. Ansari guest-edited a number of special issues, covering various emerging topics in communications and networking. He was a Visiting (Chair) Professor with several universities. He has served on the Editorial/Advisory Boards of eight journals, including as a Senior Technical Editor of the IEEE COMMUNICATIONS MAGAZINE from 2006 to 2009. He has served the IEEE in various capacities such as the Chair of the IEEE North Jersey COMSOC Chapter, the Chair of the IEEE North Jersey Section, a member of the IEEE Region 1 Board of Governors, the Chair of the IEEE COMSOC Networking TC Cluster, the Chair of the IEEE COMSOC Technical Committee on Ad Hoc and Sensor Networks, and the Chair/TPC Chair of several conferences/symposia. He has been frequently invited to deliver keynote addresses, distinguished lectures, tutorials, and talks. Some of his recent recognitions include the IEEE Leadership Award from the Central Jersey/Princeton Section in 2007, the NJIT Excellence in Teaching in Outstanding Professional Development in 2008, the IEEE MGA Leadership Award in 2008, the NCE Excellence in Teaching Award in 2009, the Thomas Alva Edison Patent Award in 2010, and designation as an IEEE Communications Society Distinguished Lecturer from 2006 to 2009 (two terms).

Nei Kato (A'03–M'04–SM'05) received the M.S. and Ph.D. degrees in information engineering from Tohoku University, Sendai, Japan, in 1988 and 1991, respectively.

He joined the Computer Center, Tohoku University, in 1991, and has been a Full Professor with the Graduate School of Information Sciences since 2003. He has published more than 200 papers in journals and peer-reviewed conference proceedings. He has been engaged in research on computer networking, wireless mobile communications, network security, image processing, and neural networks.

Dr. Kato currently serves as the Vice Chair of IEICE Satellite Communications TC, the Secretary of IEEE Ad Hoc and Sensor Networks TC, a Technical Editor of IEEE WIRELESS COMMUNICATIONS since 2006, an Editor of the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS since 2008, and an Associate Editor of the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY since 2009. He served as a Co-Guest-Editor for the *Special Issue on Wireless Communications for E-Healthcare*, the IEEE WIRELESS COMMUNICATIONS MAGAZINE, a Workshop Co-Chair of VTC2010-Fall, a Symposium Co-Chair of GLOBECOM'07, ICC'10, ICC'11, ChinaCom'08, ChinaCom'09, and WCNC2010–2011 TPC Vice Chair. His awards include the Minoru Ishida Foundation Research Encouragement Prize in 2003, the Distinguished Contributions to Satellite Communications Award from the IEEE Communications Society, the Satellite and Space Communications Technical Committee in 2005, the FUNAI Information Science Award in 2007, the TELCOM System Technology Award from the Foundation for Electrical Communications Diffusion in 2008, and the IEICE Network System Research Award in 2009. Besides his academic activities, he also serves as a member of the Expert Committee of the Telecommunications Council, Telecommunications Business Dispute Settlement Commission Special Commissioner, Ministry of Internal Affairs and Communications, Japan, and as the Chairperson of ITU-R SG4, Japan. He is a member of the Institute of Electronics and Information and Communication Engineers.