International Journal of Digital Multimedia Broadcasting

IPTV: Technology, Practice, and Service

Guest Editors: Hsiang-Fu Yu, Jen-Wen Ding, Pin-Han Ho, and János Tapolcai

IPTV: Technology, Practice, and Service

International Journal of
Digital Multimedia Broadcasting

IPTV: Technology, Practice, and Service

Guest Editors: Hsiang-Fu Yu, Jen-Wen Ding, Pin-Han Ho, and János Tapolcai

Editorial Board

Contents

IPTV: Technology, Practice, and Service, Hsiang-Fu Yu, Jen-Wen Ding, Pin-Han Ho, and János Tapolcai
Volume 2012, Article ID 734769, 2 pages

Background Traffic-Based Retransmission Algorithm for Multimedia Streaming Transfer over Concurrent Multipaths, Yuanlong Cao, Changqiao Xu, Jianfeng Guan, and Hongke Zhang
Volume 2012, Article ID 789579, 10 pages

Secure and Reliable IPTV Multimedia Transmission Using Forward Error Correction, Chi-Huang Shih, Yeong-Yuh Xu, and Yao-Tien Wang
Volume 2012, Article ID 720791, 8 pages

QoS Supported IPTV Service Architecture over Hybrid-Tree-Based Explicit Routed Multicast Network, Chih-Chao Wen and Cheng-Shong Wu
Volume 2012, Article ID 263470, 11 pages

A Secure and Stable Multicast Overlay Network with Load Balancing for Scalable IPTV Services, Tsao-Ta Wei, Chia-Hui Wang, Yu-Hsien Chu, and Ray-I Chang
Volume 2012, Article ID 540801, 12 pages

Video Classification and Adaptive QoP/QoS Control for Multiresolution Video Applications on IPTV, Huang Shyh-Fang
Volume 2012, Article ID 801641, 7 pages

Adjustable Two-Tier Cache for IPTV Based on Segmented Streaming, Kai-Chun Liang and Hsiang-Fu Yu
Volume 2012, Article ID 192314, 8 pages

A Seamless Broadcasting Scheme with Live Video Support, Zeng-Yuan Yang, Yi-Ming Chen, and Li-Ming Tseng
Volume 2012, Article ID 373459, 8 pages

An Efficient Periodic Broadcasting with Small Latency and Buffer Demand for Near Video on Demand, Ying-Nan Chen and Li-Ming Tseng
Volume 2012, Article ID 717538, 7 pages

Hindawi Publishing Corporation
International Journal of Digital Multimedia Broadcasting
Volume 2012, Article ID 734769, 2 pages
doi:10.1155/2012/734769

Editorial
IPTV: Technology, Practice, and Service

Hsiang-Fu Yu,[1] Jen-Wen Ding,[2] Pin-Han Ho,[3] and János Tapolcai[4]

[1] *Department of Computer Science, National Taipei University of Education, Taipei 106, Taiwan*
[2] *Department of Information Management, National Kaohsiung University of Applied Sciences, Kaohsiung 807, Taiwan*
[3] *Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada N2L 3G1*
[4] *Department of Telecommunication and Media Informatics, Budapest University of Technology and Economics, Budapest 1521, Hungary*

Correspondence should be addressed to Hsiang-Fu Yu, yu@tea.ntue.edu.tw

Received 10 July 2012; Accepted 10 July 2012

With the advances in video compression and broadband access technologies, the Internet Protocol Television (IPTV) becomes a popular technology for the delivery of multimedia services directly to the end users. A number of operators and vendors are currently working on IPTV standardization efforts (e.g., ATIS/IIF, ITU-T FG IPTV) to support IPTV as a secure, reliable, and managed service. IPTV also provides bundled service offerings that encompass Internet access, telephony (VoIP), multimedia services, and mobile services. Being regarded as a great business opportunity for content providers, service providers, and equipment manufacturers, there are still many issues in the standardization, design, development, and deployment of commercially viable IPTV services.

The objective of this special issue is to provide a forum for sharing knowledge and recent advances in IPTV services. A total of 14 high-quality submissions were received for this special issue. However, we are only able to accept 8 of them due to the page limit.

In the paper entitled "*A secure and stable multicast overlay network with load balancing for scalable IPTV services*," T.-T. Wei et al. propose a secure application-layer multicast overlay network for IPTV, called SIPTVMON. SIPTVMON can secure all the IPTV media delivery paths against eavesdroppers via elliptic-curve Diffie-Hellman (ECDH) key exchange on SIP signaling and AES encryption. The performance analysis shows that SIPTVMON outperforms in quality of privacy protection, stability from user churn, and good perceptual quality for scalable IPTV services over Internet.

In the paper entitled "*Video classification and adaptive QoP/QoS control for multiresolution video applications on IPTV*," S.-F. Huang presents a quality control mechanism in multiresolution video coding structures over WIMAX networks. This work also investigates the relationship between QoP and QoS in end-to-end connections.

In the paper entitled "*Adjustable two-tier cache for IPTV based on segmented streaming*," K.-C. Liang and H.-F. Yu propose a segment-based two-tier caching approach, which divides each video into multiple segments to be cached. This approach also partitions the cache space into two layers, where the first layer mainly caches to-be-played segments and the second layer saves possibly played segments. Comprehensive simulation results show that the approach can yield higher hit ratio than previous work under various environmental parameters.

In the paper entitled "*A seamless broadcasting scheme with live video support*," Z.-Y. Yang et al. present a scalable binomial broadcasting scheme to transfer live videos using constant bandwidth by increasing clients' waiting time. When the scheme finds that the length of a video exceeds the default, it doubles the length of to-be-played segments, and then its required bandwidth is constant.

In the paper entitled "*An efficient periodic broadcasting with small latency and buffer demand for near video on demand*," Y.-N. Chen and L.-M. Tseng improve the fixed-delay pagoda broadcasting (FDPB) scheme to save client buffering space as well as waiting time. In comparison with the staircase broadcasting (SB), the reverse fast broadcasting (RFB), and the hybrid broadcasting (HyB) schemes, the improved FDPB scheme can yield the smallest waiting time under the same buffer requirements.

In the paper entitled "*QoS supported IPTV service architecture over hybrid-tree based explicit routed multicast network*," C.-C. Wen and C.-S. Wu propose a cooperative

scheme of hybrid-tree based on explicit routed multicast, called as HT-ERM to combine the advantages of shared tree and source tree for QoS supported IPTV service. The simulation results show that the HT-ERM scheme outperforms other multicast QoS-based delivery schemes in terms of channel switching delay, resource utilization, and blocking ratio for IPTV services.

In the paper entitled "*Background traffic-based retransmission algorithm for multimedia streaming transfer over concurrent multipaths*," Y. Cao et al. firstly investigate the effect of background traffic on the performance of concurrent multipath transfer SCTP (CMT-SCTP). Motivated by the localness nature of background flow, this work proposes an improved retransmission algorithm named RTX_CSI to yield higher average throughput and better users' experience of quality for multimedia streaming services.

In the paper entitled "*Secure and reliable IPTV multimedia transmission using forward error correction*," C.-H. Shih et al. propose a novel FEC scheme to ensure the secure and reliable transmission for IPTV multimedia content and services. The scheme utilizes the characteristics of FEC including the FEC-encoded redundancies and the limitation of error correction capacity to protect the multimedia packets against the malicious attacks and the data transmission errors/losses.

As we conclude this overview, we would like to express our sincere appreciation to all the reviewers for their timely and insightful comments on the submitted papers, which made this special issue possible.

Hsiang-Fu Yu
Jen-Wen Ding
Pin-Han Ho
János Tapolcai

Hindawi Publishing Corporation
International Journal of Digital Multimedia Broadcasting
Volume 2012, Article ID 789579, 10 pages
doi:10.1155/2012/789579

Research Article

Background Traffic-Based Retransmission Algorithm for Multimedia Streaming Transfer over Concurrent Multipaths

Yuanlong Cao,[1] Changqiao Xu,[1,2] Jianfeng Guan,[1] and Hongke Zhang[1,3]

[1] *State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China*
[2] *Institute of Sensing Technology and Business, Beijing University of Posts and Telecommunications, Jiangsu, Wuxi 214028, China*
[3] *National Engineering Laboratory for Next Generation Internet Interconnection Devices, Beijing Jiaotong University, Beijing 100044, China*

Correspondence should be addressed to Yuanlong Cao, ylcao@bupt.edu.cn

Received 1 December 2011; Accepted 2 May 2012

Academic Editor: János Tapolcai

The content-rich multimedia streaming will be the most attractive services in the next-generation networks. With function of distribute data across multipath end-to-end paths based on SCTP's multihoming feature, concurrent multipath transfer SCTP (CMT-SCTP) has been regarded as the most promising technology for the efficient multimedia streaming transmission. However, the current researches on CMT-SCTP mainly focus on the algorithms related to the data delivery performance while they seldom consider the background traffic factors. Actually, background traffic of realistic network environments has an important impact on the performance of CMT-SCTP. In this paper, we firstly investigate the effect of background traffic on the performance of CMT-SCTP based on a close realistic simulation topology with reasonable background traffic in NS2, and then based on the localness nature of background flow, a further improved retransmission algorithm, named *RTX_CSI*, is proposed to reach more benefits in terms of average throughput and achieve high users' experience of quality for multimedia streaming services.

1. Introduction

The content-rich multimedia streaming, such as video-on-demand (VoD) [1, 2] and Internet Protocol Television (IPTV) will be the most attractive services in the next-generation networks. Most researches have proved the Stream Control Transmission Protocol (SCTP) will be the most promising technology for the large bandwidth consumption of multimedia streaming services [2–4]. Particularly in the future wireless heterogeneous network that the terminals will be equipped with multiple network interfaces and attached multiple heterogeneous access capability at the same time, the SCTP can provide the effective transmission for multimedia streaming services and balance the overhead among multiple access networks.

The SCTP [5] has been proposed and standardized by the Internet Engineering Task Force (IETF) in order to effectively utilize the multihoming environment and support real-time signaling transmission over IP networks, since SS7 has been the only bearer for the signaling traffic in telecommunication networks [6] for many years. SCTP has some important features including: (1) multi-homing. The destination nodes can be reached under the several IP addresses (multi-homed). In SCTP, both sides of the association provide multiple IP addresses combined with a single SCTP port number [7]. (2) Multistreaming which means the parallel transmission of messages over the same association between sender and the receiver. The stream independently carries fragmented messages from one terminal to another, which can achieve a cumulative throughput [8] than other protocols (e.g., TCP). SCTP manages more than one communication path with two major functions: (a) using SACK (selective acknowledgment) to probes primary path connectivity and HEARTBEAT to probe the alternative paths, respectively; (b) fail-over which means once the primary path breaks and selects an alternative path as the primary path.

As an improved version of SCTP, Concurrent Multipath Transfer (CMT) [9] uses the SCTP's multi-homing feature to distribute data across multiple end-to-end paths in a multi-homed SCTP association. CMT is the concurrent transfer of new data from a source to a destination via more than one end-to-end paths, and it is used between multi-homed source and destination hosts to increase the throughputs. Moreover, a CMT sender can maintain more accurate information (such as available bandwidth, loss rate, and RTT) of all the paths, since new data are sent to all destinations concurrently. This feature allows the CMT sender to better decide where to retransmit once data is lost.

There is more and more researches pay attention to multimedia streaming, and CMT-SCTP had been employed as transport protocol as well to study the performance of multimedia streaming services. For example, Stegel et al. [10] proposed solutions on how to provisioning SCTP multi-homing in converged IP-based multimedia environment. Huang and Lin [11] proposed a partially reliable-concurrent multipath transfer (PR-CMT) protocol for multimedia streaming in order to improve the throughput and video quality degrade. In our previous work, we designed a novel Evalvid-CMT platform [3, 4] to investigate and evaluate the performance of CMT for real-time video distribution, and then a meaningful suggestion was pointed out on which strategies for real-time video concurrent multipath transmissions.

Although the advantages of CMT-SCTP has been investigated in variety of attractive services, however, existing evaluation works [1–16] of CMT-SCTP do not consider the impact of background traffic. Actually, Internet measurement studies showed complex behaviors of Internet traffic [17, 18] that are necessary for realistic testing environments. There are several reasons why background traffic is important in performance testing. First, the aggregate behavior of background traffic can induce a rich set of dynamics such as queue fluctuations, patterns of packet losses, and fluctuations of the total link utilization at bottleneck links and can have a significant impact on the performance of CMT-SCTP. Second, network environments without any randomness in packet arrivals and delays are highly susceptible to the phase effect [19], and a good mix of background traffic reduces the likelihood of synchronization [20]. Third, the core of the Internet allows a high degree of statistical multiplexing. Therefore, the performance evaluation of network protocols with little or no background traffic does not fully investigate the CMT-SCTP behaviors that are likely to be observed when it is deployed in the Internet.

On the other hand, there are five retransmission algorithms proposed in [12] to enhance the performance of CMT-SCTP. Previous work [9, 14] take major researching focus on the effects of different retransmission algorithms with different limited receive buffer (*rbuf*) sizes. However, all of the five retransmission algorithms are designed with only one of paths' condition as metric. Liu et al. [16] combines some paths' conditions to select the retransmission path but with an unreasonable metric since loss rate is not be recommended according to RFC4460.

In this paper, taking reasonable background traffic into account, we firstly investigate the effect of background traffic on the performance of CMT-SCTP based on a more realistic simulation topology in NS2 [21]. Considering the nature of background traffic and taking paths' previous states into account, we further propose an improved retransmission algorithm named *RTX_CSI* to achieve more benefits in terms of average throughput and high users' experience of quality for multimedia streaming services.

The rest of paper is organized as follows. Section 2 explains our experimental design for network redundancy in CMT-SCTP. Section 3 presents how effects occurred by designed background traffic. Section 4 addresses the proposed *RTX_CSI* algorithm and its performance evaluation. Section 5 concludes this paper and discusses the future work.

2. Preliminary Work

2.1. Background Traffic Design. In accordance with the Internet survey [22], TCP traffic on the Internet is about 80–83%, and UDP traffic is about 17–20%. Moreover, the content-rich multimedia streaming will be the most attractive services in the future networks, more and more multimedia encoded by VBR will be deployed in Internet. Thus, more reasonable background traffic consists of TCP traffic, CBR traffic, and VBR traffic should be taken into account to evaluate the performance of data delivering.

With the purpose of investigating the effect of background traffic on the performance of CMT-SCTP, our experiments adopt a more realistic simulation scenario for network redundancy in CMT-SCTP, that is, TCP traffic, CBR traffic, and VBR traffic will be employed in our simulation topology designing. Test scenario consists of one path with TCP traffic and UDP/CBR traffic as background traffic (TCP : UDP/CBR is 4 : 1) and another with TCP traffic and UDP/VBR traffic as background traffic (TCP : UDP/VBR is 4 : 1) which is represented by *TCP+UDP/CBR* and *VBR* in below.

2.2. VBR Traffic Generator Loading. Since NS2 still cannot support VBR traffic, in order to enable VBR traffic generator in NS2, we add *PT_VBR* as packet enumeration and then set *VBR* for *PT_VBR*'s value in packet information function [23]. The default values for VBR traffic are set as shown in Table 1.

2.3. Simulation Topology Setup. To investigate the impact of background traffic on the performance of CMT-SCTP completely, a more realistic simulation topology with reasonable background traffic is proposed, which is shown in Figure 1. In the dual dumbbell topology, each router (R1 ~ R4) connects to five edge nodes. The edge nodes are single interfaces and connect to the routes to generate the background traffic. Each edge node attaches with a traffic generator, and four edge nodes generate 80% TCP traffic and one edge node generates 20% UDP traffic (CBR or VBR). According to [24], the propagation delay between the edge nodes and routers is set to 5 ms in order to create the maximum effect occurred by background traffic, and bandwidth is set to 100 Mb. The propagation delay between

Table 1: Parameter settings of VBR traffic.

Variable	Value
*Application/traffic/VBR set rate_*	448 Kb
*Application/traffic/VBR set random_*	0
*Application/traffic/VBR set max pkts_*	268435456
*Application/traffic/VBR set max Size_*	200
*Application/traffic/VBR set min Size_*	100
*Application/traffic/VBR set intervaltime_*	200

two routers is set to 45 ms with 10 Mb of bandwidth in accordance with article [25] (CMT-PF addressed in this article is not employed in our experiments for pure study of the impact occurred by background traffic).

The *S* and *R* stand for CMT-SCTP sender and receiver, respectively, and connected to the network through two interfaces. CMT-SCTP uses concurrent multipath transfer to send data on both paths with the default parameters recommended by RFC4460. After 0.5 seconds of simulation, the CMT-SCTP sender begins to initiate the association with CMT-SCTP receiver. At 1.0 seconds, edge nodes generate the background traffic, and the total simulation time is 30 seconds.

3. Study of the Impact of Background Traffic

To analyze the impact of the background traffic, this section evaluates the average throughput (delay) of CMT-SCTP with and without background traffic, respectively. To measure the presence of background traffic affecting the performance of CMT-SCTP, we define a metric called *Impact Degree* (denoted as Θ) which can be expressed by

$$\Phi(\Theta_\alpha) = \frac{|\Phi(x_\alpha) - \Phi(y_\alpha)|}{\Phi(x_\alpha)} \times 100\%, \quad (1)$$

where α stands for *rbuf*; x stands for CMT-SCTP without background traffic condition and y for CMT-SCTP with *TCP + UDP/CBR* and *VBR* traffic condition; $\Phi(x_\alpha)$ is on behalf of average throughput or average delay achieved by CMT-SCTP without background traffic under different α; $\Phi(y_\alpha)$ for average throughput or average delay created by CMT-SCTP with *TCP + UDP/CBR* and *VBR* traffic under different α; $\Phi(\Theta_\alpha)$ stands for impact degree arisen by *TCP + UDP/CBR* and *VBR* traffic. High $\Phi(\Theta_\alpha)$ means that background traffic has high side effects on CMT-SCTP in terms of throughput and delay, that is, low average throughput (high average delay) CMT-SCTP will be reached.

Since default *rbuf* size commonly used in operating systems today is varied from 16 KB to 64 KB and beyond. Herein, we investigate the impact of background traffic on CMT-SCTP under *rbuf* size with 16 KB, 32 KB, 64 KB, 128 KB, and 256 KB. Figures 2, 3, 4, 5, and 6 show throughput reached by CMT-SCTP with or without background under different size of *rbuf,* respectively (the measuring interval is 0.5 s).

As illustrated in Figures 2, 3, 4, 5, and 6, we can point out that: (1) the background traffic presents an impact on throughput clearly; (2) with the increase of the receive buffer, the impact of the background traffic is increased.

Figure 7 shows the comparison on average throughput with and without background traffic under different *rbuf*.

Figure 8 shows the comparison on average delay with and without background traffic under different *rbuf*, respectively.

Based on (1) and above simulation results, Figure 9 shows the corresponding impact degree which occurred by the designed background traffic.

As it shown in Figure 7, when *TCP + UDP/CBR* and *VBR* is employed as the background traffic, the impact degree on average throughput can be calculated as $\Phi(\Theta_{16}) \approx 0.0025$, $\Phi(\Theta_{32}) \approx 0.0047$, $\Phi(\Theta_{64}) \approx 0.0278$, $\Phi(\Theta_{128}) \approx 0.2652$, and $\Phi(\Theta_{256}) \approx 0.5238$. Figure 9 illustrates that larger *rbuf* will lead to larger impact degree, namely, larger side effect will occur by background traffic in terms of average throughput.

As it shown in Figure 8, when *TCP + UDP/CBR* and *VBR* is employed as the background traffic, the impact degree on average delay can be calculated as $\Phi(\Theta_{16}) \approx 0.05019$, $\Phi(\Theta_{32}) \approx 0.0721$, $\Phi(\Theta_{64}) \approx 0.1235$, $\Phi(\Theta_{128}) \approx 0.4941$, and $\Phi(\Theta_{256}) \approx 0.3037$. From Figure 9, we note that larger *rbuf* will lead to larger impact degree. However, when *rbuf* is set more than 256 KB, the impact will be reduced, the reason maybe that data can be received timely as greater receive buffer is used.

From above experiments and analysis, we can conclude that the background traffic can present an obvious impact on CMT-SCTP's performance in terms of throughput and delay, and it will lead to some known problems like congestion. Thus, we need to take background traffic condition into account during designing the retransmission algorithm.

4. *RTX_CSI* Algorithm

Retransmission algorithms play a more important role in achieving high users' experience of quality for multimedia streaming services. As mentioned in Section 1, there are five retransmission schemes proposed [12] for CMT-SCTP, we call them as *existing retransmission algorithms*. However, all of *exiting retransmission algorithms* do not consider the nature of background traffic. This section will simply introduce the *existing retransmission algorithm* firstly, and then a further improved retransmission algorithm named *RTX_CSI* will be addressed with considering background traffic condition, a necessary performance evaluation will be presented lastly.

4.1. Existing Retransmission Algorithm.

RTX-SAME. Once a new data chunk is scheduled and sent to a destination, all retransmissions of the chunk thereafter are sent to the same destination (until the destination is deemed inactive due to failure).

RTX-ASAP. A retransmission of a data chunk is sent to any destination for which the sender has *cwnd* space available at the time the retransmission needs to be sent. If the sender has available *cwnd* space for multiple destinations, one is chosen randomly.

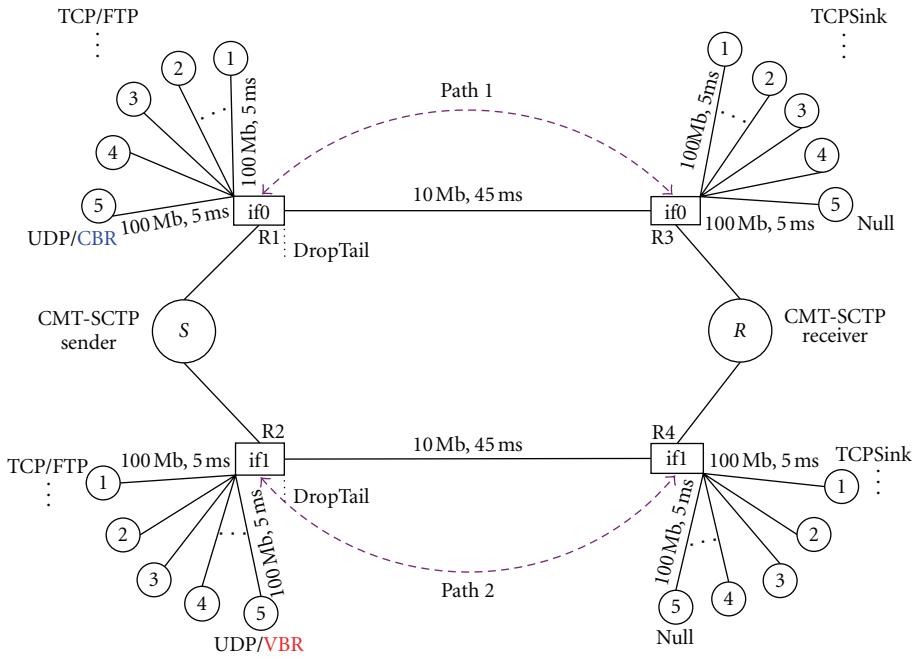


Figure 1: Simulation topology for studying the impact of background traffic.

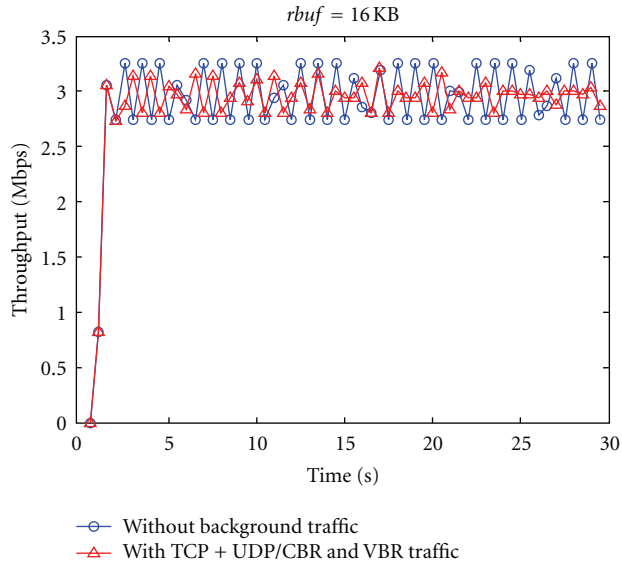


Figure 2: Comparison with *rbuf* = 16 KB.

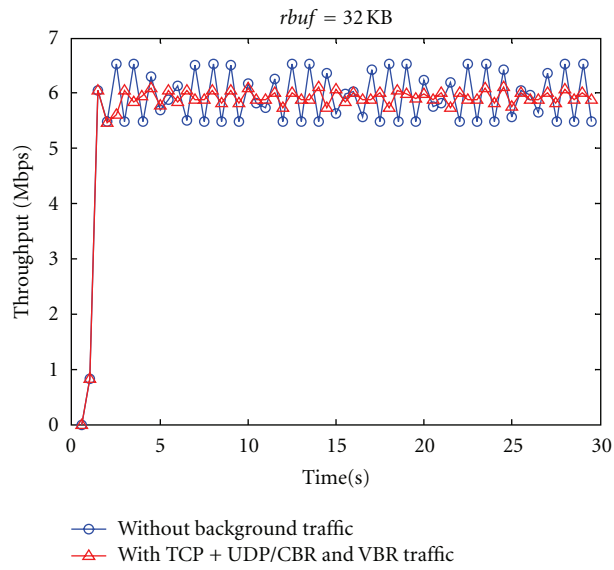


Figure 3: Comparison with *rbuf* = 32 KB.

RTX-LOSSRATE. A retransmission of a data chunk is sent to the destination with the lowest loss rate path. If multiple destinations have the same loss rate, one is selected randomly.

RTX-CWND. A retransmission of a data chunk is sent to the destination for which the sender has the largest *cwnd*. A tie is broken randomly.

RTX-SSTHRESH. A retransmission of a data chunk is sent to the destination for which the sender has the largest *ssthresh*. A tie is broken randomly.

However, according to RFC4460, only the *RTX-CWND* and *RTX-SSTHRESH* are recommended retransmission policies, and the others are just for experimental sake. Moreover, the *RTX-CWND* is recommended as the default retransmission strategy since it can present the best performance [12].

4.2. RTX_CSI Description. As mentioned in Section 1, all of *existing retransmission algorithms* do not take the impact of background traffic into account. To fix this issue, we consider

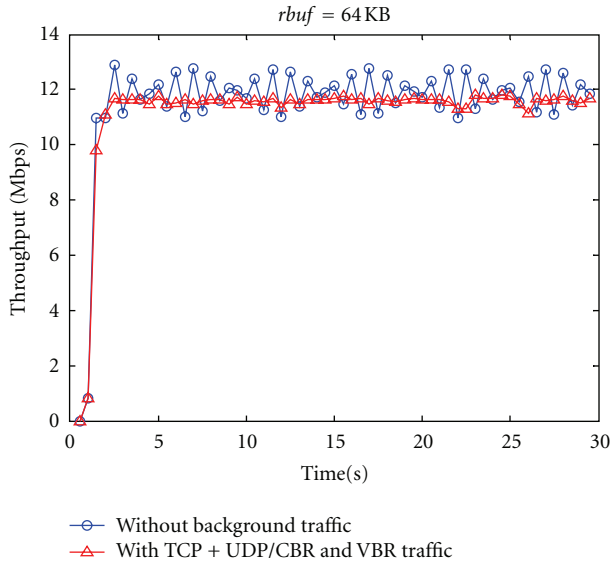


Figure 4: Comparison with *rbuf* = 64 KB.

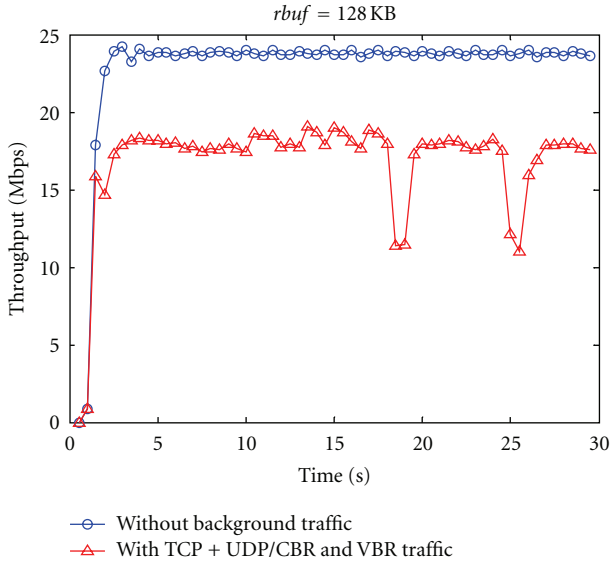


Figure 5: Comparison with *rbuf* = 128 KB.

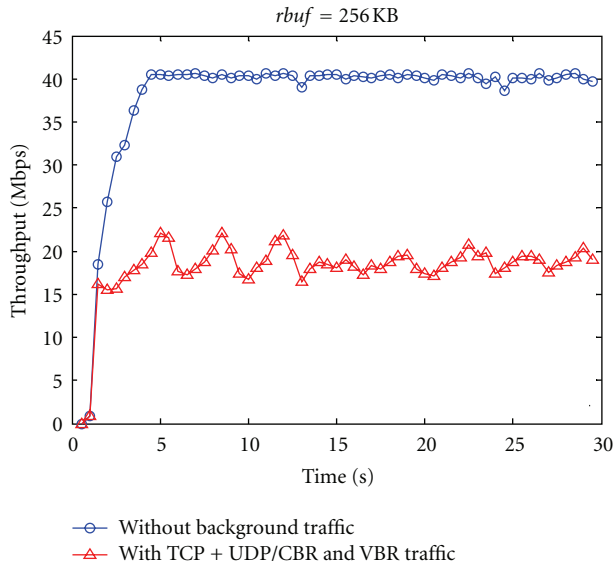


Figure 6: Comparison with *rbuf* = 256 KB.

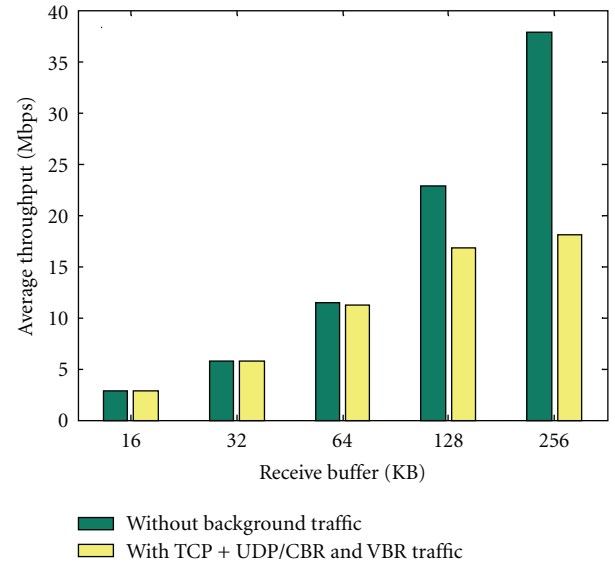


Figure 7: Comparison on average throughput.

the localness nature of background flow [19], that is, coming packets will belong to the flow as previously arrived ones in a short period. Correspondingly, the side effect on CMT-SCTP caused by background flow will be the same in short period. As countermeasure of the localness nature of background traffic, paths' previous states should be considered during designing of retransmission algorithm. Therefore, we take paths' previous states into account to design an improved retransmission algorithm named *RTX_CSI* which consists of four more reasonable paths' conditions during selecting retransmission destination. *RTX_CSI* follows the below steps to select candidate path for data retransmission.

(1) A retransmission is sent to the destination that has the largest *cwnd*;

(2) if more than one destination has the largest *cwnd*, then a retransmission is sent to the one that has the largest *ssthresh* value;

(3) if there are more than one destination has the largest *ssthresh* value, then a retransmission is sent to the one that has the lowest *Time Out Records* (*tor*) in specified time span (denoted as τ);

(4) if more than one destination has lowest timeout records in specified timeslot, then a retransmission is sent to the one that has the largest interval t_{interval} (t_{interval} stands for the interval between the last timeout's time and current time);

Definition:
d_i: the i_{th} path in the destination list of core node
d_{list}: the active destination list of core node
d_i^{cwnd}: the *cwnd* value of the i_{th} path
d_i^{ssth}: the *ssthresh* value of the i_{th} path
d_i^{tor}: total timeouts on d_i in timeslot τ
d^{interval}: the t_{interval} of the i_{th} path
$d_{rtx\text{Dest}}$: the destination selected to retransmit loss data
Once having packet to resend
1: **for** each destination d_i **do** //scan all paths' condition
2: **if** status of d_i == ACTIVE **then**
3: put d_i into d_{list}; //ignore the inactive paths
4: **end if**
5: **end for**
6: set $d_s^{cwnd} = d_{\text{list}(0)}^{cwnd}$, $d_t^{ssth} = d_{\text{list}(0)}^{ssth}$, $d_w^{\text{interval}} = d_{\text{list}(0)}^{\text{interval}}$;
7: **for** ($i = 1$, $i <= \text{count}(d_{\text{list}})$, i++) **do**
//select the destination which has largest *cwnd* value in loop
8: **if** ($d_s^{cwnd} < d_{\text{list}(i)}^{cwnd}$) **then**
9: set $s = i$; set $d_s^{cwnd} = d_{\text{list}(i)}^{cwnd}$;
10: **end if**
//select the destination which has largest *ssthresh* value in loop
11: **if** ($d_t^{ssth} < d_{\text{list}(i)}^{ssth}$) **then**
12: set $t = i$; set $d_t^{ssth} = d_{\text{list}(i)}^{ssth}$;
13: **end if**
//select the destination which has lowest *tor* value in loop
14: **if** ($d_t^{\text{tor}} > d_{\text{list}(i)}^{\text{tor}}$) **then**
15: set $t = i$; set $d_t^{\text{tor}} = d_{\text{list}(i)}^{\text{tor}}$;
16: **end if**
//select the destination which has largest t_{interval} value in loop
17: **if** ($d_w^{\text{interval}} < d_{\text{list}(i)}^{\text{interval}}$) **then**
18: set $w = i$; set $d_w^{\text{interval}} = d_{\text{list}(i)}^{\text{interval}}$;
19: **end if**
20: **end for**
21: **if !**$((d_j \in d_{\text{list}})\&\&(d_j^{cwnd} == d_s^{cwnd})\&\&(j <> s))$ **then**
//path with largest *cwnd* is set as retransmission destination
22: set $d_{rtx\text{Dest}} = d_s$;
23: **else if** $!((d_j \in d_{\text{list}})\&\&(d_j^{ssth} == d_t^{ssth})\&\&(j <> t))$ **then**
//path with largest *ssthresh* is set as retransmission destination
24: set $d_{rtx\text{Dest}} = d_t$; //
25: **else if** $!((d_j \in d_{\text{list}})\&\&(d_j^{\text{tor}} == d_t^{\text{tor}})\&\&(j <> t))$ **then**
//path with lowest tor is set as retransmission destination
26: set $d_{rtx\text{Dest}} = d_t$;
27: **else if** $!((d_j \in d_{\text{list}})\&\&(d_j^{\text{interval}} == d_w^{\text{interval}})\&\&(j <> w))$
//path with largest t_{interval} is set as retransmission destination
28: set $d_{rtx\text{Dest}} = d_w$;
//a tie will be broken by random selection
29: **else** set $d_{rtx\text{Dest}}$ with random d_i ($d_i \in d_{\text{list}}$);
30: **end if**
let loss data be retransmitted on $d_{rtx\text{Dest}}$

Algorithm 1: *RTX_CSI* Algorithm.

(5) if multiple destinations have the largest t_{interval}, then a tie will be broken by random selection.

The details of RTX_CSI algorithm are shown in Algorithm 1.

4.3. Simulation Topology Setup. In this section, we adopt the average throughput as the metric in our experiments. Figure 10 shows the simulation topology. Related simulation parameters are set the same as that mentioned in Section 3.

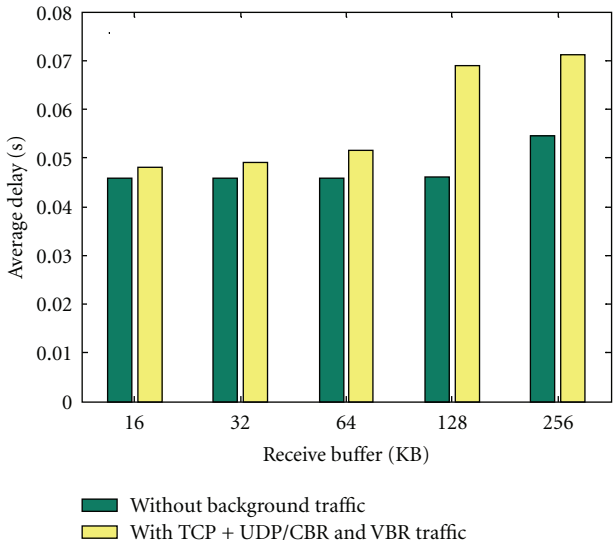


Figure 8: Comparison on average delay.

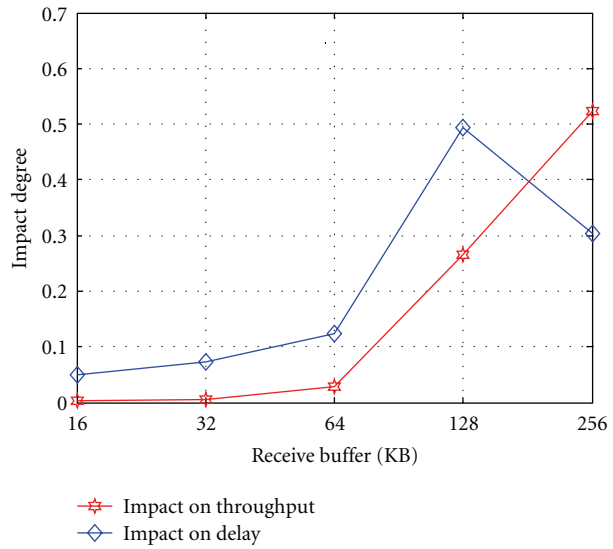


Figure 9: Impact degree on average throughput and delay.

We perform two experimental scenarios named Case 1 and Case 2 are examined as follows to study the performance of *RTX_CSI*. In our experiments, the *rbuf* is set to 16 KB, 32 KB, 64 KB, 128 KB, and 256 KB, respectively, and τ is set to 30 s.

Case 1. The loss rate on *TCP + UDP/VBR* traffic path is always kept at 1%, and on *TCP + UDP/CBR* traffic path, it is varied from 1% to 10%;

Case 2. The loss rate on *TCP + UDP/VBR* traffic path is varied from 1% to 10%, and on *TCP + UDP/CBR* traffic path, it is always kept at 1%.

4.4. Performance Evaluation. As mentioned above, according to RFC4460, only the *RTX-CWND* and *RTX-SSTHRESH* are recommended retransmission policy. So we compare the performance of *RTX_CSI* with the *RTX-CWND* and *RTX-SSTHRESH*.

To compare conveniently, we use (2) illustrated below to express the advantage in terms of average throughput achieved by algorithm A (denoted as A_{adv}) over algorithm B.

$$A_{\text{adv}} = \frac{A_{\text{aveTh}} - B_{\text{aveTh}}}{B_{\text{aveTh}}} \times 100\%, \tag{2}$$

where A_{aveTh} and B_{aveTh} are on behalf of average throughput achieved by retransmission algorithm A and B, respectively.

Firstly, we evaluate the performance of *RTX_CSI*, *RTX_CWND,* and *RTX_SSTHESH* with experimental condition illustrated in Case 1; Figure 11 shows the performance of the three algorithms under different *rbuf*. We can get that the average throughput achieved by the *RTX_CWND*, *RTX_SSTHRESH*, and proposed *RTX_CSI* will rise with the increase of the *rbuf*. But when *rbuf* is set to more than 64 KB, the increments of average throughput are reduced whichever the three algorithms is employed. This phenomenon verifies again that the background traffic presents more serious side effects on performance of CMT-SCTP as larger *rbuf* is used. However, the *RTX_CSI* performs the best performance in the three algorithms, the *RTX_CWND* comes next, and the *RTX_SSTHRESH* presents the worst behavior.

For Case 1, per calculated by (2), detailed comparison on the average throughput can be pointed out as follows.

(1) Comparing to the RTX_SSTHRESH, the RTX_CWND achieves a more advantage about −4.42%, 4.94%, 18.2%, 56.39%, and 55.15% when *rbuf* is 16 KB, 32 KB, 64 KB, 128 KB, and 256 KB, respectively. So, it can be conculded that the RTX_CWND can also present better performance over the RTX_SSTHRESH [12] even under background traffic condition (Case 1).

(2) Comparing to the RTX_CWND, the proposed RTX_CSI achieves more benefits about 1.14%, 0.56%, and 1.62% when rbuf is set to 16 KB, 32 KB, and 64 KB, respectively. Since larger rbuf leads to less packet loss; therefore, when the rbuf is set to 128 KB and 256 KB, the proposed RTX_CSI presents same performance in terms of throughput as the RTX-CWND algorithm.

Secondly, we compare the performance of the *RTX_CWND* and *RTX_SSTHRESH* with the proposed *RTX_CSI* under different *rbuf* with designed experimental scenario addressed in Case 2. As it shown in Figure 12, the average throughput achieved by the three algorithms will rise with the increase of *rbuf*. But with same reason mentioned in Case 1, when *rbuf* is set to more than 64 KB, the increments of average throughput are reduced whichever the three algorithms are employed. In this case, the *RTX_CSI* still performs the best performance. But different from conclusion addressed in [12] and Case 1, only if *rbuf* is larger than 64 KB, the *RTX_CWND* presents a better performance than the *RTX_SSTHRESH*.

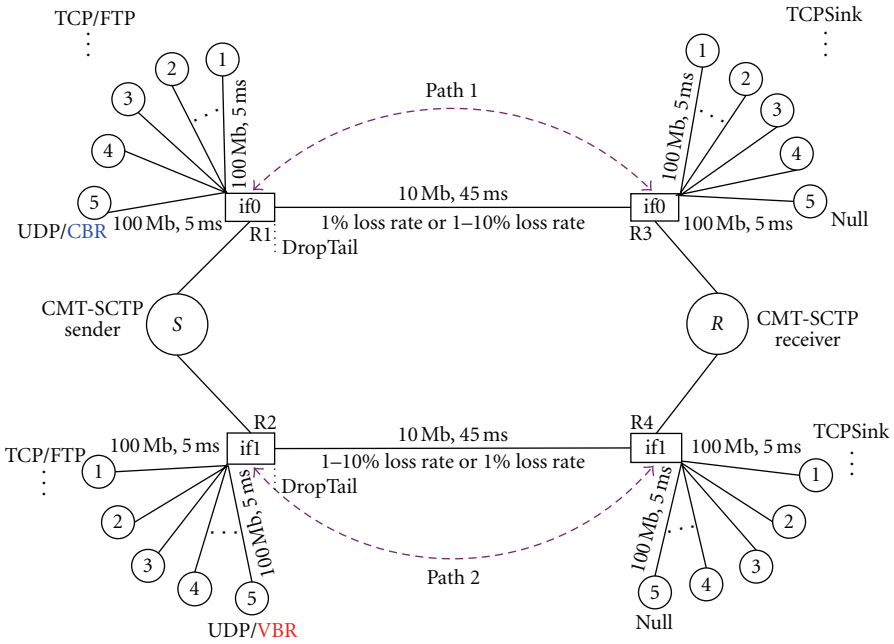


Figure 10: Simulation topology for evaluating *RTX_CSI*.

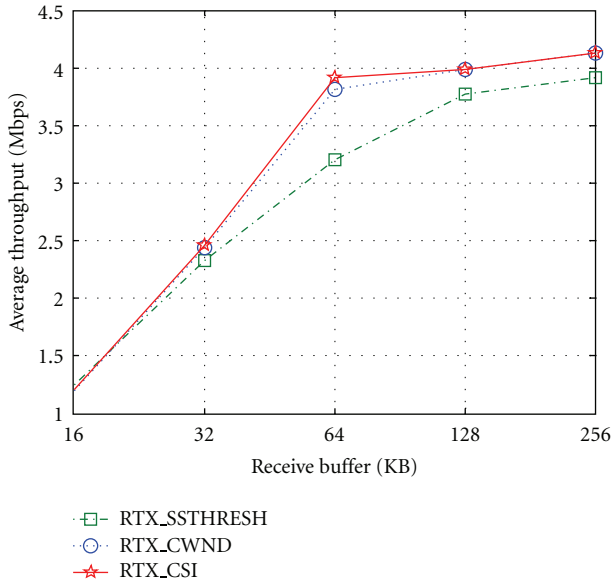


Figure 11: Path 1 loss rate is varied from 1–10%, Path 2 is always kept at 1%.

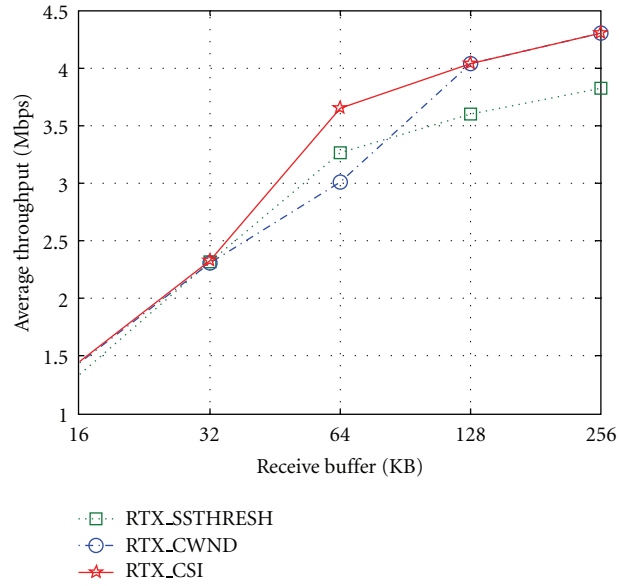


Figure 12: Path 2 loss rate is varied from 1–10%, Path 1 is always kept at 1%.

Likewise, for Case 2, detailed comparisons are pointed out per calculated by (2).

(1) Comparing to the *RTX_SSTHRESH*, the *RTX_CWND* achieves a more advantage about 7.26%, −0.15%, −8.02%, 12.05%, and 12.57% when *rbuf* is 16 KB, 32 KB, 64 KB, 128 KB, and 256 KB, respectively. The comparison results are different from the conclusion mentioned in [12], in our experiment (Case 2), comparing to the *RTX_SSTHRESH*, the *RTX_CWDN* only can achieve obvious more advantages in terms of average throughput when *rbuf* is set to the value which is larger than the default *rbuf* (64 KB). The reason may be that when enormous of variable bite rate packets are lost, their retransmissions not only deteriorate the path's quality but also enlarge the unpredictability of path's condition. Those unexpected conditions make the CMT-SCTP sender cannot tune its congestion window accurately. But

when the *rbuf* is set to larger one (more than 64 KB), the sender can correct its congestion window since lots of packets can be received and acknowledged timely thus, the *RTX_CWND* can outperform the *RTX_SSTHRESH*. This phenomenon further verifies the proposed *RTX_CSI* more reasonable with considering the nature of background traffic. Our future work will investigate the reason in detail.

(2) Comparing to the *RTX_CWND*, the proposed *RTX_CSI* achieves more benefits about 0.47%, 0.57%, and 17.56% when rbuf is set to 16 KB, 32 KB, and 64 KB, respectively. As Case 1, when the rbuf is set to 128 KB and 256 KB, the proposed *RTX_CSI* presents same performance in terms of throughput as the *RTX_CWND* with the reason that larger rbuf leads to less packet loss.

From experiments and analysis for Case 1 and Case 2 respectively, we can conclude that the proposed *RTX_CSI* algorithm can achieve a better performance than the *existing retransmission algorithm*, especially for *rbuf* such as 16 KB, 32 KB, and 64 KB which are used commonly. The reason *RTX_CSI* can achieve a more advantage over the two *existing retransmission algorithm* is that the *RTX_CSI* can select a more efficient path as the retransmission destination with considering more reasonable rules such as paths' *cwnd*, *ssthresh* value, and historical states to meet some known problems like congestion and packet loss caused by background traffic.

5. Conclusions

In this paper, we designed realistic simulation topologies and examined the performance of CMT-SCTP in terms of throughput, end-to-to packet delay by considering reasonable background traffic. We discussed *how* the presence of background traffic affects the performance of CMT-SCTP in detail, which are generally ignored by most current researchers.

Based on above work, we proposed an improved retransmission algorithm called *RTX_CSI* for CMT-SCTP. *RTX_CSI* takes background traffic into account and considers paths' comprehensive characteristics during selecting retransmission destination to meet the localness nature of background traffic. The simulation results show *RTX_CSI* achieves better efficiency than CMT-SCTP's original retransmission algorithm. So the proposed *RTX_CSI* can be employed to improve the users' experience of quality for multimedia streaming service when CMT-SCTP is used for the multimedia transport protocol.

Acknowledgments

This work is partially supported by the National High-Tech Research and Development Program of China (863) under Grant no. 2011AA010701, in part by the National Natural Science Foundation of China (NSFC) under Grants nos. 61001122 and 61003283 and the Beijing Natural Science Foundation under Grant no. 4102064, in part by the Fundamental Research Funds for the Central Universities under Grants nos. 2012RC0603 and 2011RC0507, and in part by the Natural Science Foundation of Jiangsu Province under Grant no. BK2011171.

References

[1] C. Xu, G. M. Muntean, E. Fallon, and A. Hanley, "Distributed storage-assisted data-driven overlay network for P2P VoD services," *IEEE Transactions on Broadcasting*, vol. 55, no. 1, pp. 1–10, 2009.

[2] Z. Liu, C. Wu, S. Zhao, and B. Li, "UUSee: large-scale operational on-demand streaming with random network coding," in *Proceedings of the IEEE Conference on Computer Communications (INFOCOM '10)*, San Diego, Calif, USA, March 2010.

[3] C. Xu, E. Fallon, Q. Yuansong, Z. Lujie, and M. Gabriel-Miro, "Performance evaluation of multimedia content distribution over multi-homed wireless networks," *IEEE Transactions on Broadcasting*, vol. 57, no. 2, pp. 204–215, 2011.

[4] C. Xu, E. Fallon, M. Gabriel-Miro, X. Li, and A. Hanley, "Performance evaluation of distributing real-time video over concurrent multipath," in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC '09)*, Budapest, Hungary, April 2009.

[5] K. Zheng, M. Liu, Z. C. Li, and G. Xu, "SHOP: an integrated scheme for SCTP handover optimization in multihomed environments," in *Proceedings of the IEEE Global Telecommunications Conference*, pp. 1–5, New Orleans, La, USA, December 2008.

[6] R. Stewart, Q. Xie, K. Morneault et al., "Stream control transmission protocol," IETF RFC 2960, October 2000.

[7] P. Natarajan, F. Baker, P. D. Amer, and J. T. Leighton, "SCTP: what, why, and how," *IEEE Internet Computing*, vol. 13, no. 5, pp. 81–85, 2009.

[8] Y. Wang, R. Injong, and H. Sangtae, "Augment SCTP multistreaming with pluggable scheduling," in *Proceedings of the 30th IEEE International Conference on Computer Communications Workshops*, pp. 810–815, Shanghai, China, April 2011.

[9] J. R. Iyengar, P. D. Amer, and R. Stewart, "Concurrent multipath transfer using SCTP multihoming over independent end-to-end paths," *IEEE/ACM Transactions on Networking*, vol. 14, no. 5, pp. 951–964, 2006.

[10] T. Stegel, J. Sterle, U. Sedlar, J. Bešter, and A. Kos, "SCTP multihoming provisioning in converged IP-based multimedia environment," *Computer Communications*, vol. 33, no. 14, pp. 1725–1735, 2010.

[11] C. M. Huang and M. S. Lin, "Multimedia streaming using partially reliable concurrent multipath transfer for multihomed networks," *IET Communications*, vol. 5, no. 5, pp. 587–597, 2011.

[12] J. R. Iyengar, P. D. Amer, and R. Stewart, "Retransmission policies for concurrent multipath transfer using SCTP multihoming," in *Proceedings of the 12th IEEE International Conference on Networks (ICON '04)*, pp. 713–719, Singapore, November 2004.

[13] J. R. Iyengar, P. D. Amer, and R. Stewart, "Performance implications of a bounded receive buffer in concurrent multipath transfer," Tech. Rep., CIS Department, University of Delaware.

[14] J. Liao, J. Wang, and X. Zhu, "cmpSCTP: an extension of SCTP to support concurrent multi-path transfer," in *Proceedings of*

the IEEE International Conference on Communications, pp. 5762–5766, Beijing, China, 2008.

[15] Ł. Budzisz, R. Ferrús, F. Casadevall, and P. Amer, "On concurrent multipath transfer in SCTP-based handover scenarios," in *Proceedings of the IEEE International Conference on Communications*, pp. 1–6, Dresden, Germany, June 2009.

[16] J. M. Liu, H. X. Zou, J. X. Dou, and Y. Gao, "Reducing receive buffer blocking in concurrent multipath transfer," in *Proceedings of the IEEE International Conference on Circuits and Systems for Communications (ICCSC '08)*, Shanghai, China, May 2008.

[17] P. Barford and M. Crovella, "Generating representative web workloads for network and server performance evaluation," in *Proceedings of ACM SIGMETRICS*, pp. 151–160, Madison,Wis, USA, June 1998.

[18] S. Floyd and V. Paxson, "Difficulties in simulating the Internet," *IEEE/ACM Transactions on Networking*, vol. 9, no. 4, pp. 392–403, 2001.

[19] S. Floyd and E. Kohler, "Internet research needs better models," *ACM Computer Communications Review*, vol. 33, no. 1, pp. 29–34, 2003.

[20] S. Ha, L. Le, I. Rhee, and L. Xu, "Impact of background traffic on performance of high-speed TCP variant protocols," *Computer Networks*, vol. 51, no. 7, pp. 1748–1762, 2007.

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

[22] M. Fomenkov, K. Keys, D. Moore, and K. claffy, "Longitudinal study of Internet traffic in 1998–2003," in *Proceedings of the Winter International Symposium on Information and Communication Technologies (WISICT '04)*, pp. 1–6, Cancun, Mexico, January 2004.

[23] http://www.isi.edu/nsnam/archive/ns-users/webarch/2001/msg05051.html.

[24] A. Caro, P. Amer, and J. Iyengar, "Retransmission policies with transport layer multihoming," in *Proceedings of the 11th IEEE International Conference on Networks*, pp. 255–260, Sydney, Australia, November 2003.

[25] P. Natarajan, J. R. Iyengar, P. D. Amer, and R. Stewart, "Concurrent multipath transfer using transport layer multihoming: performance under network failures," in *Proceedings of the Military Communications Conference (MILCOM '06)*, pp. 1–7, Washington, DC, USA, 2006.

Hindawi Publishing Corporation
International Journal of Digital Multimedia Broadcasting
Volume 2012, Article ID 720791, 8 pages
doi:10.1155/2012/720791

Research Article

Secure and Reliable IPTV Multimedia Transmission Using Forward Error Correction

Chi-Huang Shih, Yeong-Yuh Xu, and Yao-Tien Wang

Department of Computer Science and Information Engineering, HungKuang University, Taichung 433, Taiwan

Correspondence should be addressed to Yeong-Yuh Xu, yyxu@sunrise.hk.edu.tw

Received 14 December 2011; Accepted 22 May 2012

Academic Editor: Hsiang-Fu Yu

With the wide deployment of Internet Protocol (IP) infrastructure and rapid development of digital technologies, Internet Protocol Television (IPTV) has emerged as one of the major multimedia access techniques. A general IPTV transmission system employs both encryption and forward error correction (FEC) to provide the authorized subscriber with a high-quality perceptual experience. This two-layer processing, however, complicates the system design in terms of computational cost and management cost. In this paper, we propose a novel FEC scheme to ensure the secure and reliable transmission for IPTV multimedia content and services. The proposed secure FEC utilizes the characteristics of FEC including the FEC-encoded redundancies and the limitation of error correction capacity to protect the multimedia packets against the malicious attacks and data transmission errors/losses. Experimental results demonstrate that the proposed scheme obtains similar performance compared with the joint encryption and FEC scheme.

1. Introduction

As digital technologies process, Internet Protocol Television (IPTV) has emerged in the past years to deliver high-quality multimedia services to end users over IP broadband networks. Generally, IPTV has multifaceted content such as video/audio/text/graphic/data and needs to provide the user-required quality of experience (QoE), interactivity, security, and reliability in the IP-based networks [1]. The typical IPTV applications include the cable TV-like service and video on demand (VoD). In the cable TV-like service, the service provider can provide the entertainment, news, and sports programs in a regular standard definition (SD) or further high definition (HD) format, while the VoD service supports more personal options to select their favorite multimedia content. Because the IP technology is basically the same for the IPTV and Internet applications, IPTV is likely to integrate the existing and independent services over the home network connection.

Since the content delivered through IPTV is mostly of high economic value and of copyright with user's subscription, secure and reliable transmission becomes an important issue in provisioning IPTV content and services. The basic principle of the service and content protection is to ensure that users are only able to obtain the services they are entitled to access and use the content in accordance with the right they have been granted [2, 3]. For content and service protection, the conditional access system (CAS) and digital right management (DRM) are two primary protection technologies on IPTV [4]. CAS is employed in the conventional TV industry to restrict certain television programs to certain users according to a billing mechanism. On the other hand, DRM is often utilized in the information technology (IT) industry to protect the digital data against illegal copy and redistribution. In both CAS and DRM, data encryption is one common security tool to provide the robust security control on the valued data. An essential requirement for data encryption in IPTV is the need to transmit a single encrypted stream to many users. Since different users can be authorized to receive different packages of services, this requirement is generally met by using multiple layers of encryption. In addition, frequent update of encryption keys is desired to avoid unauthorized data sharing due to illegal key extracting. Although the data encryption is a well-designed technique

in protecting sensitive data, its efficacy in IPTV environment can be affected based on the IP delivery characteristics such as packet loss [5, 6]. In order to protect the multimedia stream against transmission errors/losses, forward error correction (FEC) deliberately produces redundant data to enable the reconstruction of any multimedia packets which are lost during transmission. The IPTV standard developed by Digital Video Broadcasting (DVB) project specifies an application-layer FEC to perform packet loss repair for IPTV streaming media [7, 8].

One of the possible solutions to support a secure and reliable transmission is the integration of data encryption and FEC recovery [9–12]. The typical operation of this integration is to first encrypt the source data using a secret key which is available only at the end nodes, and then to generate the redundant FEC data for loss recovery purpose by encoding the encrypted data. Both the encrypted and redundant data are transmitted along the network path and the receiver processes the received data in a reverse order (i.e., FEC decoding and decryption). In [13], an image-coding scheme has been proposed to provide encryption and FEC based on Error Correction Codes (ECCs) over noisy channels. Related works in [14, 15] use turbo-codes-based error control scheme to combine with encryption for secure data transmission. Moreover, the cryptographic encryption scheme based on Advanced Encryption Standard (AES) [16] and the FEC protection scheme using turbo codes have been integrated to ensure a reliable and secure transmission [17]. Although the joint encryption and FEC scheme is effective enough, several performance problems arise in terms of computational cost and management cost. All costs contribute to the delay time, which is critical to the multimedia services, and complicate the IPTV system design. In general, the computational cost largely originates from the processing overhead including FEC encoding/decoding and encryption/decryption, while the management cost derives from the generation of multiple keys, frequent key updates, channel feedback messages carrying network conditions, and so on. In [18, 19], an iterative decoding approach for digital signatures has been developed to perform the error correction, in addition to the authentication capacity provided by the digital signature itself. However, this approach becomes more effective as an FEC scheme is present and most importantly the transmission data remains unsafe to the malicious attacks. It is therefore necessary to design a IPTV transmission system with light performance cost in delivering multimedia content securely and reliably.

In this paper, we propose a security-enhanced FEC scheme which achieves a secure and reliable transmission for valued IPTV content, by means of packet-level Reed-Solomon (RS) codes [20] with a set of security constraints on the FEC coding parameters. The proposed secure FEC focuses on providing the content or service protection to prevent malicious users from acquiring the unauthorized data, while aiming at improving data goodput by recovering the potential transmission errors/losses. Two key features of the secure FEC are (1) to transmit FEC-encoded data only and hence the original content data are prevented from exposing to the malicious users directly and (2) to deliberately control the amount of FEC-encoded data so that the error correction capacity provided by the FEC-encoded data fails to reconstruct source content data. As to the authorized user, the successful data reconstruction relies on an additional data storage between the content server and user. The experimental results show that the proposed secure FEC obtains the same performance as the joint AES encryption scheme with 128-bit key and the packet-level FEC scheme based on Reed-Solomon codes, in the data transmission.

The remainder of this paper is organized as follows. Section 2 reviews the standard packet-level FEC protection scheme using Reed-Solomon codes. Section 3 introduces the proposed secure FEC scheme. Section 4 describes the exposure rate of source data for measuring the security level in this paper and establishes an analytical model associated with the exposure rate. The performance evaluation results are presented and discussed in Section 5. Finally, Section 6 provides some brief concluding remarks and future works.

2. Standard FEC

Without loss of generality, we use systematic Reed-Solomon erasure codes RS (n, k_1) to protect multimedia data from channel losses. The RS encoder chooses k_1 multimedia data items as an FEC block and generates $(n - k_1)$ redundant data items for the block. Every data item has its own sequence number used to indicate the corresponding position within the block. With this position information, the RS decoder can locate the position of the lost items and then correct up to $(n - k_1)$ lost items. Furthermore, a packet-level RS code is applied as FEC since it has a high efficiency over error-prone channels [21]. Figure 1 illustrates the operations of packet-level FEC scheme. Packet-level FEC schemes group the source data packets into blocks of a predetermined size k_1 and then encode $n = k_1 + h_{\text{std}}$ packets for network transmission, where $h_{\text{std}} \geq 0$ is the number of redundant packets. The coding rate is thus defined as k_1/n. Provided that k_1 or more packets are successively received, the block can be completely reconstructed. In the standard packet-level FEC, given the target recovery probability R_{std}, the estimated packet loss rate P_B and fixed k_1, the lower bound on n can be computed in the sender using

$$R_{\text{std}}(n, k_1, P_B) = \sum_{i=k_1}^{n} \left[\binom{n}{i} (P_B)^{n-i} (1 - P_B)^{i} \right]. \quad (1)$$

On the other hand, the feedback packets are sent periodically from the receiver to the sender in order to obtain the timely channel information about P_B. Note that packet-level FEC extends the media stream simply by inserting redundant packets into the stream, and, therefore, the method requires only minor modification to the source packets.

3. Secure FEC

The secure FEC scheme aims at supporting reliable and secure transmission for multimedia IPTV flows. To achieve the secure transmission, the proposed FEC scheme is based on the packet-level RS codes and has two features: (1) only the

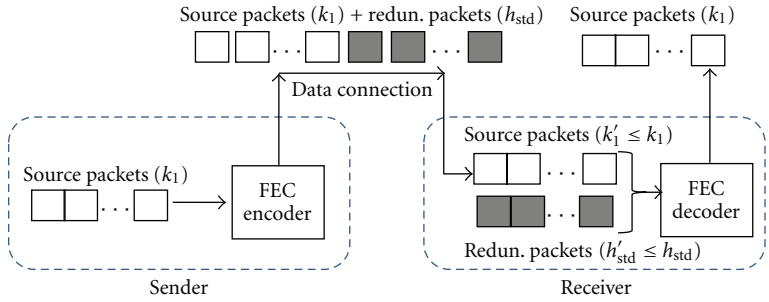


Figure 1: Overview of standard packet-level FEC scheme. This figure shows the FEC coding operations at both the sender and receiver. The maximum amount of loss packets that can be recovered is h_{std}.

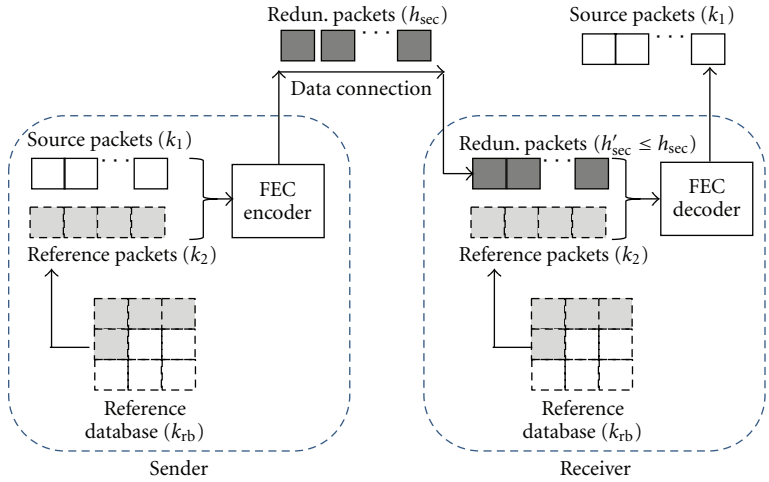


Figure 2: Overview of secure FEC scheme. Note that the reference packets k_2 are included in the FEC encoding/decoding, and the sender transmits redundant packets h_{sec}.

redundant data are delivered along the transmission path; in other words, the original source data are used in the encoding stage, and (2) both the data sending side and data receiving side need to maintain a consistent reference database where the reference data are selected to perform the FEC encoding with the source data at the data sending side, and the FEC decoding with the received redundant data at the data receiving side. Transmitting redundant data can avoid that the malicious host directly inspect the content of source data. It is noted that in the standard FEC, the transmission data include source and redundant data. Furthermore, the use of reference data in the FEC encoding/decoding stage causes the FEC decoding failure for the malicious host even if the malicious host attempts to decode the intercepted redundant data. Figure 2 illustrates the operations of secure FEC. The detained procedures can be summarized into five steps.

(1) Both the data sending side and receiving side have the similar k_2 reference packets.

(2) The data sending side generates the FEC redundant packets h_{sec} based on the source packets k_1 and the reference packets k_2.

(3) The data sending side transmits h_{sec} redundant packets through the network to the data receiving side.

(4) The data receiving side receives h_{sec}' packets and $h_{sec}' \leq h_{sec}$.

(5) The data receiving side uses the reference packets k_2 and the received packets h_{sec}' to reconstruct the source packets k_1.

According to the procedures described above, the condition that a block can be successfully recovered is given by

$$h_{sec} + k_2 \geq k_1 + k_2 \longrightarrow h_{sec} \geq k_1. \tag{2}$$

To prevent that the malicious host intercepts the transmitted packets h_{sec} between the data sending side and data receiving side, the value of h_{sec} must not exceed the amount of FEC-encoded source packets. That is

$$h_{sec} < k_1 + k_2. \tag{3}$$

Then the recovery probability in the secure FEC is shown as follows:

$$R_{sec}(h_{sec}, k_1, P_B) = \sum_{i=k_1}^{h_{sec}} \left[\binom{h_{sec}}{i} (P_B)^{h_{sec}-i} (1-P_B)^{i} \right], \quad \text{subject to } k_1 \leq h_{sec} < k_1 + k_2. \tag{4}$$

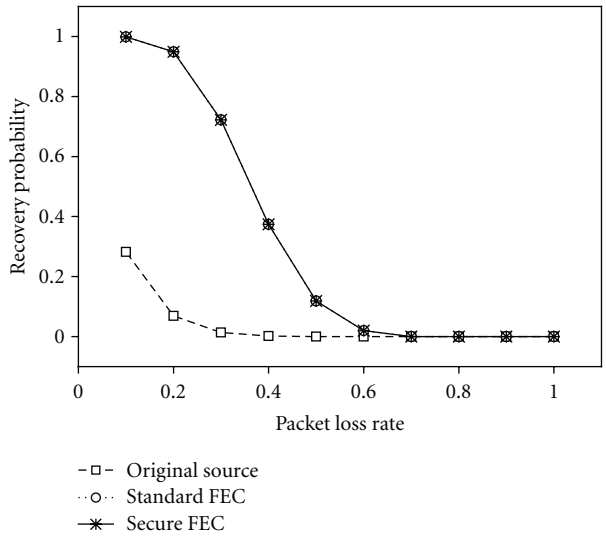


Figure 3: Coding rate for both the standard FEC and secure FEC is 2/3 with $k_1 = 12$.

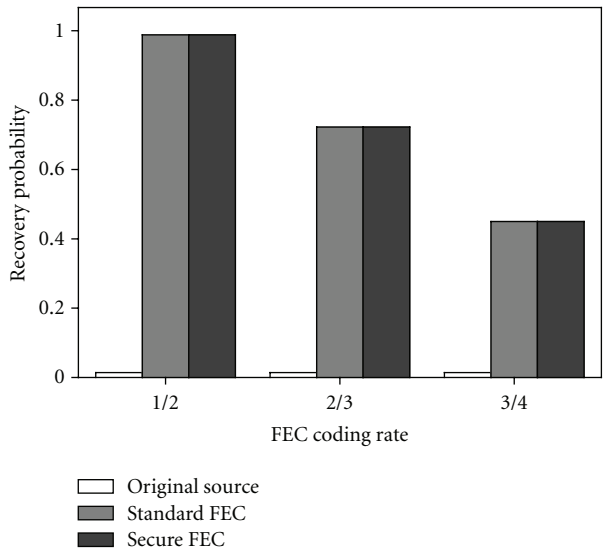


Figure 4: Comparison of recovery probability with varied FEC coding rates, when keeping $k_1 = 12$. The packet loss rate is fixed to 0.3.

According to (4), the amount of reference packets k_2 determines the efficiency of the secure FEC scheme since k_1 is typically a predefined value. Larger the value of k_2, higher the FEC recovery rate for a given packet loss rate P_B.

In keeping the consistent reference database between connection ends, the reference data can be initially set up as the secure FEC is installed to start its service and could be updated or expanded by selecting reference data from the reconstructed source data. It is noted that the source data are available only at the connection ends under the decoding constraint on the amount of redundant data (i.e., $k_1 \leq h_{\text{sec}} < (k_1 + k_2)$).

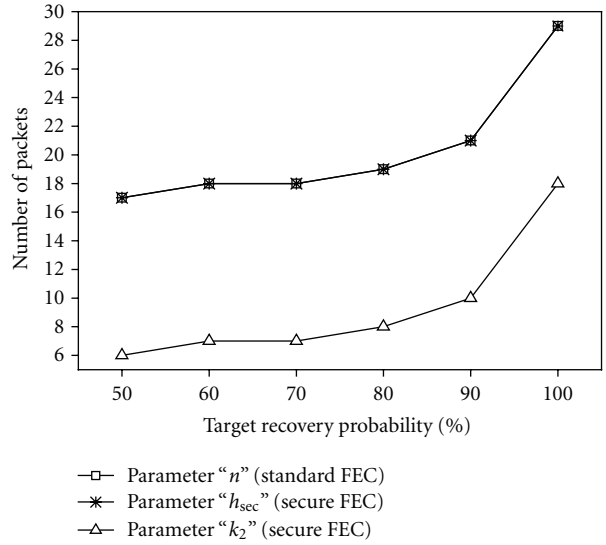


Figure 5: Observation on parameter changes with varied target recovery probabilities. The values of k_1 and P_B are 12 and 0.3, respectively.

4. Exposure Rate

In this section, we calculate the exposure rate to observe the degree of data inspection for the malicious host. For a data flow transmitting the source packets k_1 along the data path, the exposure rate (ER) can be easily obtained as the malicious host intercepts k_1' packets:

$$\text{ER}_{\text{src}}(k_1', k_1) = \frac{k_1'}{k_1}. \tag{5}$$

Let us assume that the malicious user can intercept the transmission packets in the presence of packet loss rate P_B. Then the value of k_1' can be computed as $k_1' = k_1 \times (1 - P_B)$ and (5) becomes $\text{ER}_{\text{src}}(k_1', k_1) = 1 - P_B$. The exposure rate represents the degree that the source data are exposed to any malicious host with the data interception capacity. As the value of ER approaches to 1, the malicious host can inspect more source content.

In the standard FEC, the delivery blocks inspected by the malicious host fall into one of two different categories: (1) the block is successfully reconstructed and (2) the block is not successfully decoded since the number of received packets is less than the number of source packets. In the first category, all k_1 source packets are completely reconstructed and the expected number of source packets, $E_{\text{FEC}(n,k_1),1}$, can be calculated as

$$E_{\text{FEC}(n,k_1),1} = k_1 \times R_{\text{std}}(n, k_1, P_B). \tag{6}$$

As to the second category, the number of lost packets is greater than $n - k_1$ within n transmitted packets and the

expected number of source packets, $E_{\mathrm{FEC}(n,k_1),2}$, is derived as follows:

$$E_{\mathrm{FEC}(n,k_1),2} = \frac{k_1}{n} \times \sum_{j=n-k_1+1}^{n} \left[\binom{n}{j} \times P_B^j \times (1-P_B)^{n-j} \times (n-j) \right]. \tag{7}$$

Therefore, the total expected number of source packets after decoding an FEC block is given by

$$E_{\mathrm{FEC}(n,k_1)} = k_1 \times R_{\mathrm{std}}(n,k_1,P_B) + \frac{k_1}{n} \times \sum_{j=n-k_1+1}^{n} \left[\binom{n}{j} \times P_B^j \times (1-P_B)^{n-j} \times (n-j) \right]. \tag{8}$$

Then, the exposure rate for an FEC block with k_1 source packets and n total transmission packets is given by

$$\mathrm{ER}_{\mathrm{std}}(n,k_1) = \frac{E_{\mathrm{FEC}(n,k_1)}}{k_1}. \tag{9}$$

For our proposed secure FEC scheme, only FEC-encoded redundant packets are injected into the transmission channel, and the amount of injected packets has to be less than the sum of total source packets $(k_1 + k_2)$ for FEC encoding. It is noted that an FEC block can be completely reconstructed at the data receiver only when the amount of received packets is not less than the amount of total source packets. Letting the amount of intercepted packets be h'_{sec} in the secure FEC, we can obtain the following relation

$$h'_{\mathrm{sec}} \le h_{\mathrm{sec}} < k_1 + k_2. \tag{10}$$

Based on the relation above, in the secure FEC scheme, the malicious host receives ER = 0 since the malicious host cannot reconstruct the source packets k_1 with the intercepted packets h'_{sec}, and all intercepted packets are FEC-encoded redundancies. Considering that the malicious host might have k'_2 reference packets and $0 \le k'_2 \le k_2$, the recovery probability for the malicious host with k'_2 is computed as

$$R_{\mathrm{sec}}(h_{\mathrm{sec}},k_1,k'_2,P_B) = \begin{cases} 0, & \text{if } h_{\mathrm{sec}}+k'_2 < k_1+k_2, \\ \sum_{i=k_1+k_2-k'_2}^{h_{\mathrm{sec}}} \binom{h_{\mathrm{sec}}}{i} \times P_B^{h_{\mathrm{sec}}-i} \times (1-P_B)^i, & \text{if } h_{\mathrm{sec}}+k'_2 \ge k_1+k_2. \end{cases} \tag{11}$$

Then the exposure rate for the malicious host is given by

$$\mathrm{ER}_{\mathrm{sec}}(h_{\mathrm{sec}},k_1,k'_2,P_B) = \begin{cases} 0, & \text{if } h_{\mathrm{sec}}+k'_2 < k_1+k_2, \\ \left(k_1 \times R_{\mathrm{sec}}(h_{\mathrm{sec}},k'_2,P_B) + \frac{k_1}{h_{\mathrm{sec}}} \times \sum_{j=h_{\mathrm{sec}}+k'_2-(k_1+k_2)+1}^{h_{\mathrm{sec}}} \left[\binom{h_{\mathrm{sec}}}{i} \times P_B^i \times (1-P_B)^{h_{\mathrm{sec}}-i} \times (h_{\mathrm{sec}}-i) \right] \right) / k_1, & \text{if } h_{\mathrm{sec}}+k'_2 \ge k_1+k_2. \end{cases} \tag{12}$$

5. Performance Analysis and Discussions

In this section, the performance of the proposed secure FEC scheme has been evaluated in terms of FEC recovery capacity and data exposure degree. The standard FEC and secure FEC employed packet-level RS codes. In the standard FEC, the values of parameters (k_1, n) were set to $(12, 18)$, and in the secure FEC, the values of k_1 and h_{sec} were 12 and 18, respectively.

5.1. FEC Recovery Capacity. To observe the FEC capacity of the proposed scheme, we compare the secure FEC with the standard FEC and the original source flow. For the original source flow, the source packets are directly transmitted into the network, while the standard FEC transmits both the source packets and redundant packets. Figure 3 shows the results of the recovery probability as the packet loss rate varies. In Figure 3, all source packets are nearly lost as the packet loss rate is larger than 0.3. For the standard FEC and the secure FEC, both schemes have the decay curve as the packet loss rate increases and their curves are exactly the same for all values of packet loss rates. It is noted that the standard FEC has the loss recovery capacity of $(n-k_1)$ packets while the loss recovery capacity in the secure FEC is given by $(h_{\mathrm{sec}} + k_2) - (k_1 + k_2)$ and therefore $(n - k_1)$. As shown in Figure 3, based on the assumption that both schemes require

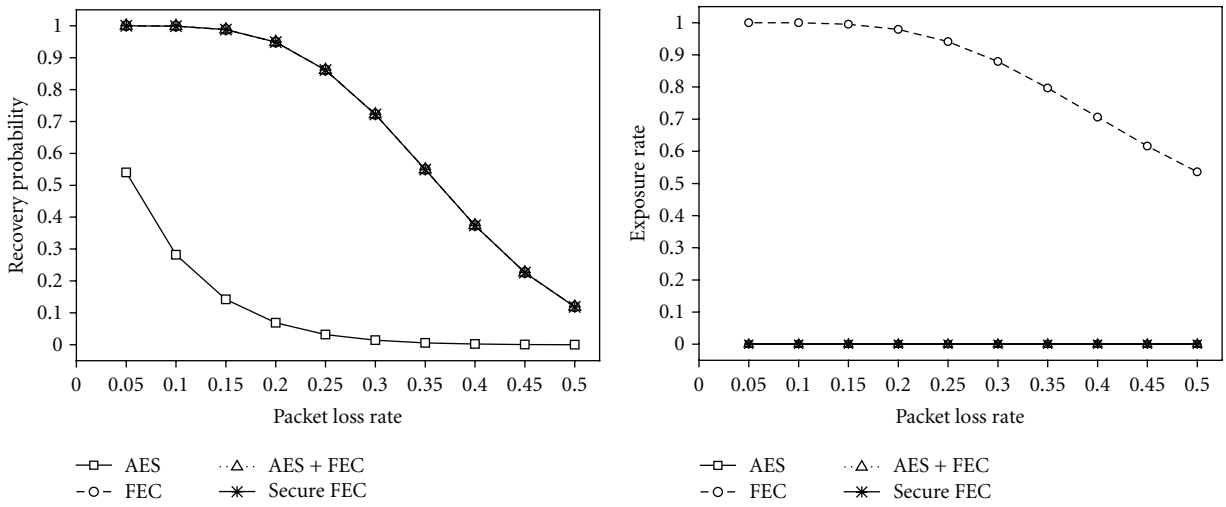


Figure 6: System performance comparison with varied packet loss rates. (a) Recovery probability; (b) exposure rate. Noted that label "FEC" represents the standard FEC.

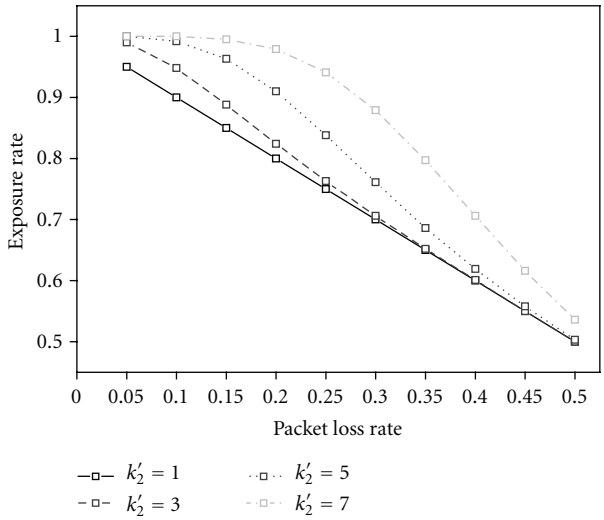


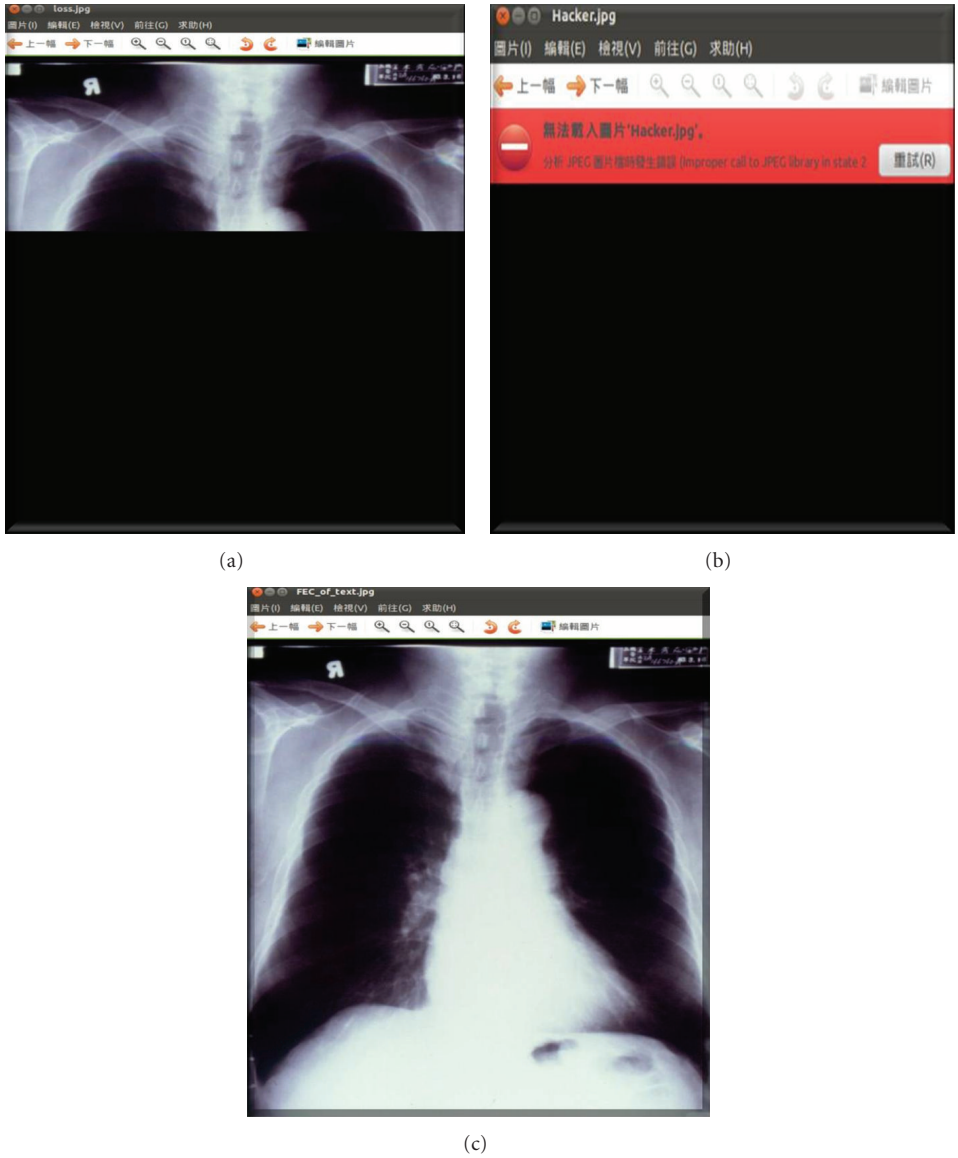
Figure 7: Exposure rate performances when different k_2' are applied to the secure FEC. This figure shows the performance impact as the reference packets are leaked to the malicious user.

the same bandwidth consumption (i.e., $h_{\text{sec}} = n$), the secure FEC can obtain the similar performance as the standard FEC. The similar observations can also be found in Figure 4. Figure 4 shows the results of the recovery probability for three FEC coding rates of 1/2, 2/3, and 3/4.

To study the operating behavior of the secure FEC, Figure 5 shows the values of parameters (n, h_{sec}, k_2) in packets as the target recovery probability is given. For the secure FEC, the required redundant packets h_{sec} are increased in order to achieve a higher recovery probability. Meanwhile, to ensure the secure transmission, the amount of reference packets k_2 needs to be increased accordingly. It is worth to note that the necessary condition of h_{sec} used in the secure FEC to avoid the successful FEC decoding by the malicious host is $h_{\text{sec}} < (k_1 + k_2)$, and the lower bound of k_2 is hence $k_2 > (h_{\text{sec}} - k_1)$. The curve of k_2 presented in Figure 5 is plotted by using its minimum value for a given target recovery probability (i.e., $k_2 = (h_{\text{sec}} - k_1) + 1$).

5.2. Data Exposure Degree. We then compare the proposed secure FEC with the encryption scheme, and the joint encryption and FEC scheme. In this study, AES with 128-bit key and packet-level FEC using RS codes are considered. Throughout the evaluation, we assume that a malicious host is located at the receiver side and is capable of performing FEC decoding. Four cases are studied: AES, standard FEC, joint AES and standard FEC, and secure FEC. Figure 6 shows the performance results in terms of recovery probability and exposure rate, from the perspective of a malicious host. From Figure 6, it can be seen that (1) AES has a exposure rate of 0 to guarantee the secure transmission in Figure 6(b) and in Figure 6(a), it has the much lower recovery probability than other three cases; (2) in Figure 6(b), standard FEC obtains the higher values of exposure rates than other cases with secure protection capacities, and as the packet loss rate increases, the exposure rate of FEC is decreased since the recovery probability of FEC is decreased accordingly to receive less source data for the malicious host; and (3) the secure FEC achieves the same performance as the joint AES and FEC scheme in Figures 6(a) and 6(b) to ensure the secure and reliable transmission.

Figure 7 shows the performance impacts when the malicious host is assumed to be capable of acquiring the reference packets. As shown in Figure 7, leaking more reference packets has a higher probability to expose source data to the malicious host. Furthermore, a higher exposure rate is also observed in the presence of lower packet loss rate because the FEC process at the malicious host is easier to reconstruct the source data.



(a) (b)

(c)

Figure 8: Snapshots of experiment results at three different receiving sides. (a) AES receiver without FEC capacity; (b) malicious receiver with FEC capacity; (c) secure FEC receiver.

5.3. Implementation and Experimental Results. To examine the effectiveness of our proposed secure FEC, the secure FEC scheme was implemented on the Linux platform and employed to transmit a sequence of images. In the experimental setup, five machines were connected with a Fast Ethernet LAN. They included an AES sender, an AES receiver, a secure FEC sender, a secure FEC receiver, and a malicious receiver attempting to peak the transmission data. The packet size is 1246 Bytes and all receivers apply the same packet loss traces of $P_B = 0.1$ to the received packet stream.

Figures 8(a)–8(c) presents the snapshots for the AES receiver, malicious receiver, and secure FEC receiver, respectively. From Figure 8, we can observe that (1) the AES receiver can only obtain a part of source packets after decrypting the received data, in the presence of packet loss; (2) the malicious receiver is unable to inspect the content of the transmitted data for either the AES connection or the secure FEC connection; and (3) the secure FEC receiver can receive the complete source data after the successful FEC reconstruction.

6. Conclusions

In this paper, a novel FEC scheme, which is equipped with both the error correction and security-enhanced capacity, is proposed so as to provide the secure and reliable transmission for valued IPTV content. We have derived the mathematical model to calculate data recovery rate and exposure rate for performance analysis purpose and conducted experiments to demonstrate the validity of the proposed secure FEC. To conclude, the secure FEC can protect the content data against the transmission losses and the unauthorized

access. Our future works are (1) to further study the secure FEC applications on the security issues such as authentication and data alteration and (2) to incorporate with encryption and watermarking to achieve robust security promise to end users while the overall performance cost can be minimized.

Acknowledgment

This work was supported by the National Science Council, Taiwan, under Grant no. NSC100-2221-E-241-014.

References

[1] J. Maisonneuve, M. Deschanel, J. Heiles et al., "An Overview of IPTV Standards Development," *IEEE Transactions on Broadcasting*, vol. 55, no. 2, pp. 315–328, 2009.

[2] M. Jeffrey, S. Park, K. Lee, G. Adams, and S. Savage, "Content security for IPTV," *IEEE Communications Magazine*, vol. 46, no. 11, pp. 138–146, 2008.

[3] S. O. Hwang, "Content and service protection for IPTV," *IEEE Transactions on Broadcasting*, vol. 55, no. 2, pp. 425–436, 2009.

[4] H. Zhang, C. Chen, L. Zhao, S. Yang, and L. Zhou, "Content Protection for IPTV-current state of the art and challenges," in *Proceedings of the IMACS Multiconference on Computational Engineering in Systems Applications (CESA '06)*, pp. 1680–1685, Beijing, China, October 2006.

[5] M. Ellis and C. Perkins, "Packet loss characteristics of IPTV-like traffic on residential links," in *Proceedings of the 7th IEEE Consumer Communications and Networking Conference (CCNC '10)*, Las Vegas, Nev, USA, January 2010.

[6] M. Cha, G. Choudhury, J. Yates, A. Shaikh, and S. Moon, "Case study: resilient backbone design for IPTV services," in *Proceedings of the International World Wide Web Conference, IPTV Workshop*, 2006.

[7] Society of Motion Picture and Television Engineers, "Forward error correction for real-time video/audio transport over IP networks," SMPTE specification 2022-1, 2007.

[8] Digital Video Broadcasting (DVB), "IP datacast over DVB-H: content delivery protocols," ETSI TS 102 472.

[9] M. A. Haleem, C. N. Mathur, R. Chandramouli, and K. P. Subbalakshmi, "Opportunistic encryption: a trade-off between security and throughput in wireless networks," *IEEE Transactions on Dependable and Secure Computing*, vol. 4, no. 4, pp. 313–324, 2007.

[10] M. Ruiping, L. Xing, and H. E. Michel, "A new mechanism for achieving secure and reliable data transmission in wireless sensor networks," in *Proceedings of the IEEE Conference on Technologies for Homeland Security: Enhancing Critical Infrastructure Dependability*, pp. 274–279, Woburn, Mass, USA, May 2007.

[11] A. Neri, D. Blasi, L. Gizzi, and P. Campisi, "Joint Security and Channel Coding for Ofdm Communications," in *Proceedings of the European Signal Processing Conference (EUSIPCO'08)*, Lausanne, Switzerland, 2008.

[12] X. Zhu, Q. Sun, Z. Zhang, and C. W. Chen, "A joint ECC based media error and authentication protection scheme," in *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME '08)*, pp. 13–16, Hannover, Germany, June 2008.

[13] C. Nanjunda, M. A. Haleem, and R. Chandramouli, "Robust encryption for secure image transmission over wireless channels," in *Proceedings of the IEEE International Conference on Communications (ICC '05)*, pp. 1287–1291, Seoul, Korea, May 2005.

[14] A. Neri, D. Blasi, P. Campisi, and E. Maiorana, "Joint authentication and forward error correction of still images," in *Proceedings of the European Signal Processing Conference (EUSIPCO'10)*, pp. 2111–2115, Aalborg, Denmark, August 2010.

[15] L. Yao and L. Cao, "Turbo codes-based image transmission for channels with multiple types of distortion," *IEEE Transactions on Image Processing*, vol. 17, no. 11, pp. 2112–2121, 2008.

[16] J. Daemen and V. Rijmen, *The Design of Rijndael: AES—The Advanced Encryption Standard*, Springer, 2002.

[17] H. Cam, V. Ozduran, and O. N. Ucan, "A combined encryption and error correction scheme: AES-turbo," *Journal of Electrical and Electronics Engineering*, vol. 9, no. 1, pp. 891–896, 2009.

[18] N. Zivic, "On using the message digest for error correction in wireless communication networks," in *Proceedings of the International Workshop on Wireless Distributed Networks*, Istanbul, Turkey, September 2010.

[19] N. Zivic and C. Ruland, "Parallel joint channel coding and cryptography," *International Journal of Computer Science and Engineering*, vol. 4, pp. 140–144, 2008.

[20] L. Rizzo, "Effective erasure codes for reliable computer communication protocols," *ACM SIGCOMM Computer Communication Review*, vol. 27, no. 2, pp. 24–36, 1997.

[21] F. Borgonovo and A. Capone, "Efficiency of error-control schemes for real-time wireless applications on the Gilbert channel," *IEEE Transactions on Vehicular Technology*, vol. 54, no. 1, pp. 246–258, 2005.

Hindawi Publishing Corporation
International Journal of Digital Multimedia Broadcasting
Volume 2012, Article ID 263470, 11 pages
doi:10.1155/2012/263470

Research Article

QoS Supported IPTV Service Architecture over Hybrid-Tree-Based Explicit Routed Multicast Network

Chih-Chao Wen and Cheng-Shong Wu

Department of Electrical Engineering, National Chung Cheng University, Chia-Yi 62102, Taiwan

Correspondence should be addressed to Chih-Chao Wen, ccwen@ccu.edu.tw

Received 16 November 2011; Revised 14 March 2012; Accepted 30 March 2012

Academic Editor: János Tapolcai

With the rapid advance in multimedia streaming and multicast transport technology, current IP multicast protocols, especially PIM-SM, become the major channel delivery mechanism for IPTV system over Internet. The goals for IPTV service are to provide two-way interactive services for viewers to select popular program channel with high quality for watching during fast channel surfing period. However, existing IP multicast protocol cannot meet above QoS requirements for IPTV applications between media server and subscribers. Therefore, we propose a cooperative scheme of hybrid-tree based on explicit routed multicast, called as HT-ERM to combine the advantages of shared tree and source tree for QoS-supported IPTV service. To increase network utilization, the constrained shortest path first (CSPF) routing algorithm is designed for construction of hybrid tree to deliver the high-quality video stream over watching channel and standard quality over surfing channel. Furthermore, the Resource Reservation Protocol-Traffic Engineering (RSVP-TE) is used as signaling mechanism to set up QoS path for multicast channel admission control. Our simulation results demonstrated that the proposed HT-ERM scheme outperforms other multicast QoS-based delivery scheme in terms of channel switching delay, resource utilization, and blocking ratio for IPTV service.

1. Introduction

As the rapid growth of broadband network applications with streaming transport over Internet, the Internet Protocol Television (IPTV) system has been widely deployed to provide multimedia service anywhere at any time. This is because IPTV enables digital service convergence of communications, computing, and media content over IP network with desired QoS guarantee [1]. From the perspective of the quality of experience (QoE), IPTV system operates as the same with broadcasting TV service, which would deliver the watching and surfing programs over different channels. However, the most difference is that IPTV works in a two-way interactive communications between service providers and subscribers. We need to consider the effective channel and delivery control problem to achieve video streaming with desired quality over Internet.

To efficiently satisfy multiple viewers' own quality requirements, IP multicast is considered a promising solution for IPTV application. Nevertheless, quality of service (QoS) support to IPTV system still poses challenging issues for multicast channel delivery and resource utilization through IP networks. The QoS-supported IPTV multicast service architecture is to deploy an efficient multicast transmission system via IP multicast delivery tree with the integration of resource provisioning and channel admission control. The IP multicast delivery has the merit of efficient bandwidth saving; however, it is difficult to assign effective multicast channel to meet QoS requirements in consideration of multicast channel state labeling and channel switching delay [2]. Therefore, the original IP multicast is not designed for multimedia application to transport time-sensitive packet streaming with bandwidth reservation and QoS gurantee along point-to-multipoint (P2MP) multicast path for large amount of IPTV channel subscribers.

From the perspective of QoS requirement, IPTV channel change will impair the content quality of video streaming to speed up the surfing streams transmission. It usually depends on the group of picture (GOP) size between Intracoded frames (I-frames) in video stream sequence to determine

the quality of watching channel and surfing channel. According to high quality video coding, a typical high definition (HD) video stream requires at least 10 Mbps of I-frames transfer rate for IPTV watching channels, while a standard definition (SD) video stream requires 2–5 Mbps for lower quality video stream in IPTV surfing channels upon channel change [3].

From the perspective of network transmission performance, current IP multicast protocols can be enhanced by different multicast QoS routing mechanisms. The QoS-aware multicast routing protocol (QMRP) [4] was first to propose feasible multicast paths computation based on QoS metric for single path or multipath. Afterwards, the protocol independent multicast (PIM) protocol is based on receiver initiated routing decision to find the shortest path regardless of underlying unicast routing. For example, the typical PIM protocols are represented by source specific multicast (SSM) [5], and PIM-sparse mode (SM) [6]. Those two multicast protocols are integrated by QoS routing algorithm with traffic control to construct source tree, shared tree, and even hybrid-tree structures. Especially, this hybrid-tree multicast can be an alternative to improve IPTV QoS and achieve load balance of multicast traffic by combining advantages of above two multicast tree types: shared tree and source tree.

As the previous paper mentioned in [7], the hybrid-tree multicast is considered as a suitable solution for IPTV channel control and delivery to satisfy multicast QoS requirements. However, existing core functionality of PIM-SM protocol in rendezvous point (RP) node still lacks the efficient control mechanism for hybrid-tree switchover operations to realize IPTV QoS multicast during channel change period. The reasons are explained as follows.

(i) *High-level traffic control mechanism*: RP router will aggregate all channel source streaming into the single shared tree until the link efficient bandwidth is over-threshold. The status report is a high-level control message detected by receivers. Therefore, the reaction time may be too slow to deal with unexpected QoS degradation and traffic congestion.

(ii) *Two-pass switchover control operation*: for channel change, multicast tree switchover is executed by RP node after receiver member-leave and rejoin request. By using two-pass switchover operation, RP node suspends the traffic aggregation from the specified source node; then channel traffic can be changed to new source node inefficiently.

Therefore, IPTV service provider must provide a cost-effective multicast network control mechanism as an efficient channel delivery solution. In this paper, we propose the enhanced hybrid-tree-based multicast delivery scheme with explicit routed multicast, called as HT-ERM. To improve performance of QoS-supported IPTV multicast channel, our HT-ERM routing algorithm is designed based on constrained shortest path first (CSPF) [8], and HT-ERM channel admission control is employed by RSVP-TE mechanism [9]. In performance evaluation, the proposed HT-ERM scheme can improve IPTV delivery and channel control as compared with the other QoS multicast schemes.

The rest of this paper is organized as follows. We summarize past works of multicast QoS for IPTV in Section 2. In Section 3, we made assumptions of hybrid-tree multicast in related IPTV models. The Section 4 describes the HT-ERM control algorithm and multicast channel operations. Section 5 presents simulation results for IPTV service, and in Section 6 we give conclusions.

2. Related Works

QoS-supported IPTV services need to consider QoS guarantee, which involves with multicast delivery through core network, and channel selective control in user access network. Most of researches focus IPTV multicast QoS on two crucial subjects: multicast network resource control for IPTV watching channels and multiple surfing channel change delay control.

2.1. IPTV Multicast Channel Delivery and Change. The analysis of IPTV channel control for content delivery and channel change depends on those factors such as command processing time, network transmission delay time, streaming switchover delay time, and video-decoding time [10–12]. The most important key factors for channel control are affected by content transmission and streaming switchover through networks. In [11], the authors proposed multicast proxy IGMP scheme for channel prejoining to the expected IPTV channels by bulk delivery the popular watching channels and other subscribers can filter watching channel and switchover surfing channels in the same local network.

2.2. IPTV Multicast Network Resource Control. Many QoS multicast routing algorithms are proposed to compute the feasible multicast tree, so that can reduce traffic transmission delay and achieve efficient resource utilization. In traditional IP multicast network, the multicast routing algorithm lacks QoS control for network P2MP connections and traffic load balancing. Recently, QoS-aware multicast approaches, such as ECMP [13] and QMRP [4], are developed to solve scalability and resource allocation when a large amount of different multicast streams transmit to heterogeneous receivers through Internet.

The modified multicast equal-cost multipath (ECMP) scheme [13] has been applied in shared tree to enhance PIM-SM or SSM-related IP multicast protocols. To achieve traffic balancing, multiple paths with equal cost are constructed to split the traffic from RP-shared tree. The centralized multicast traffic control approach is usually utilized by RP node to aggregate multicast traffic into the shared tree for IPTV watching channels and adjacent surfing channels together.

The QMRP is a well-known QoS-aware multicast routing control approach [4], which can compute optimal tree-path in single-path or multi-path mode to join multiast tree for IPTV channel delivery. When the traffic load on multicast tree link is over threshold, the multipath traffic distribution will be activated to diverse the traffic load, and thus switchover to the specific multicast path.

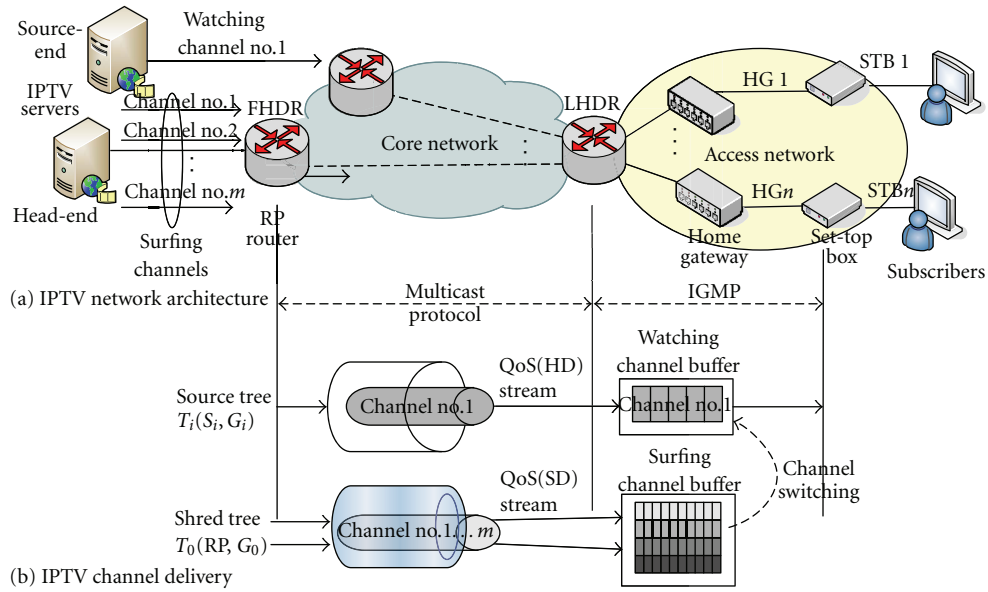


FIGURE 1: IPTV multicast network architecture consists of (a) core network and access network; IPTV channel delivered by (b) hybrid-tree based channel distribution scheme.

Furthermore, the proposed HT-ERM algorithm is to improve the overall IPTV channel delivery with efficient network resource utilization and fast channel change. The hybrid-tree-based multicast HT-ERM protocol integrates with hybrid multicast advantages to enhance functionality of current IP multicast protocol for IPTV QoS channel delivery control.

3. IPTV System Model

We use the IPTV system model to simplify the complicated service interaction between application level and network level. The IPTV channel quality states are assumed to abstract the channel dynamic behavior so as to introduce the proposed HT-ERM scheme in following sections.

3.1. Network Architecture for IPTV Service. The IPTV network architecture consists of multicast core network for channel distribution and local network for channel access as shown in Figure 1(a). In core network, IPTV channels are distributed and delivered by multicast protocol from first-hop designated router (FHDR) to last-hop designated router (LHDR). In local access network, the subscribers with terminal devices access IPTV channels via set-top box (STB) through home gateway (HG). For IPTV media service provider, the head-end (HE) server is to aggregate different basic quality channels for viewers' channel surfing and change behavior. The source-end server is to supply high quality program stream as a unique watching channel.

We model the IPTV system over IP multicast network. Assume that IPTV media servers can provide IPTV channels with source streams denoted by $S_i = \{S_1, \ldots, S_m\}$, and the subscriber members $M_i = \{M_1, \ldots, M_n\}$ can join any watching or surfing channel from its attached LHDR node R_i to form the channel group $G_i = \{R_1, \ldots, R_n\}$. The collection of IPTV channels are delivered through multipoint-to-multipoint (MP2MP) connections (S_i, G_i) between FHDR and LHDR over core network.

3.2. State Parameters for IPTV QoS Channel. In Figure 1(b), IPTV multicast channels are distributed between FHDR and LHDR. The source tree T_i links carry the multiple high quality streams from their specific sources to the corresponding group. Each watching stream is transferred from watching channel buffer to STB. The surfing channel is aggregated by surfing streams over RP-based shared tree T_0 links with low quality. Each surfing channel is extracted from surfing channel buffer for fast channel switching during channel change.

The QoS-supported IPTV channel state is defined by two types of video quality: high definition (HD) and standard definition (SD). The HD video stream is paid per channel for high quality watching program, and SD video stream is normal quality used for fast channel surfing and free watching. Accordingly, the channel stream with higher QoS level is assigned to source-based multicast channel, and lower QoS level stream is delivered by shared multicast channel.

For QoS-supported IPTV channel state, the QoS level with HD is in steady-state function, and SD is in dynamic

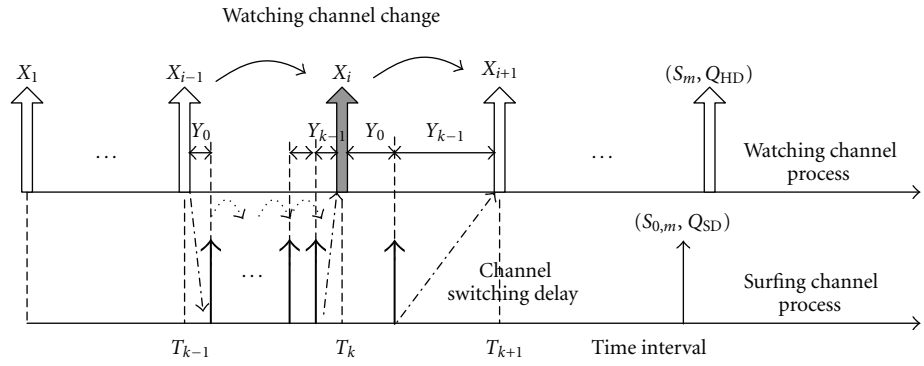


Figure 2: IPTV channel change process between watching and surfing channel state.

state. The IPTV QoS channel state can be represented by multicast tree symbol with QoS level $T(S, G, Q)$, which is characterized by

$$T(S,G,Q) = \begin{cases} T_{\text{watch}}\left(S_i, G_i, Q_q\right), & i = 1,\ldots,m;\ q = \text{hd}, \\ T_{\text{surf}}\left(S_{i,j}, G_0, Q_q\right), & i = 0;\ j = 1,\ldots,m, \\ & q = \text{sd}, \end{cases} \tag{1}$$

where the parameter S denotes the source nodes of IPTV channel, G denotes the group identification for receiver joining, and Q denotes the required stream QoS level with HD/SD. The G_i is the multicast group joining to the specified source S_i for watching channel no. i, and G_0 is the shared multicast group joining to the common source $S_{i,j}$ with RP router for aggregated surfing channels. In other words, the watching channel set is delivered by source tree $T_i(S_i, G_i, Q_{\text{hd}})$ for HD quality stream. The surfing channel is delivered by RP shared tree $T_0(S_{0,j}, G_0, Q_{\text{sd}})$ for SD quality stream.

3.3. IPTV Channel Change Behavior. According to the definition of channel state, IPTV QoS channel can be obtained by watching state T_{watch} and channel surfing state T_{surf} in channel change process. Assume that viewers usually stay in watching channel X_i with the state parameters (S_i, Q_{HD}), and the last surfing channel state with the state parameters $(S_{0,m}, Q_{\text{SD}})$ will stop before next watching channel.

As the viewer making multiple channel changes, the random behavior of channel changing in surfing state can be modeled by a terminating renewal process [3]. Once channel change is occurred, the surfing state is zapping between surfing channels in transition state with random time interval $Y_i, i = 0, 1, 2, \ldots$. channel switching process as shown in Figure 2. When channel change is stopped, we observed that the events X_i of watching channel state always stay in steady state within the time interval $[T_k, T_{k+1}]$, where $T_k = Y_0 + Y_1 + \cdots + Y_{k-1}$.

As IPTV channel change is a random process, the channel state may occur either in watching states or surfing states at any given time. We can figure out the joint channel state probability density function by

$$P_T = p(k, q), \tag{2}$$

where index k is the channel number depending on program popularity, and index q is the QoS stream quality distributed ratio over core network.

The channel popularity is the preference to the desired watching channel for most of viewers. According to Zipf's law [2], the probability Z_k that a viewer will choose the kth most popular channel is given by

$$Z_k = \frac{c}{k^s}, \tag{3}$$

where c is a constant to make the probabilities sum to 1, and exponent s is set to 1. The selection of watching channel is first determined by Zipf's law. Then, the watching channel changes are occurred by poison process and terminated at next watching channel from surfing channel selection.

4. The HT-ERM Scheme for IPTV Channel Delivery

The proposed HT-ERM scheme has efficient hybrid-tree multicast operation for IPTV channel control and delivery as compared with the function of PIM-SM protocol.

4.1. HT-ERM Protocol Design. As shown in flow chart of Figure 3, we design the HT-ERM algorithm for multicast IPTV channels, including the hybrid-tree initialization, shared tree aggregation, source tree switchover and multicast admission control in following subsections.

4.1.1. Channel Initialization and Join. The RP node is initialized by IPTV channel aggregation for surfing channel joining requests in random process before watching channel selection. In additions, the initial parameters setting are shared tree T_0, shared group G_0, channel quality Q, and resource allocation threshold BW_{th}.

While receiving the channel request message, RP node checks the control message types (Join or Switchover) by source node (rp or s) and group member ($g0$ or gi). The RP node is channel concentrator to update the surfing channel states through the shared tree T_0. The explicit routes of shared tree T_0 can be derived by computing CSPF (constrain

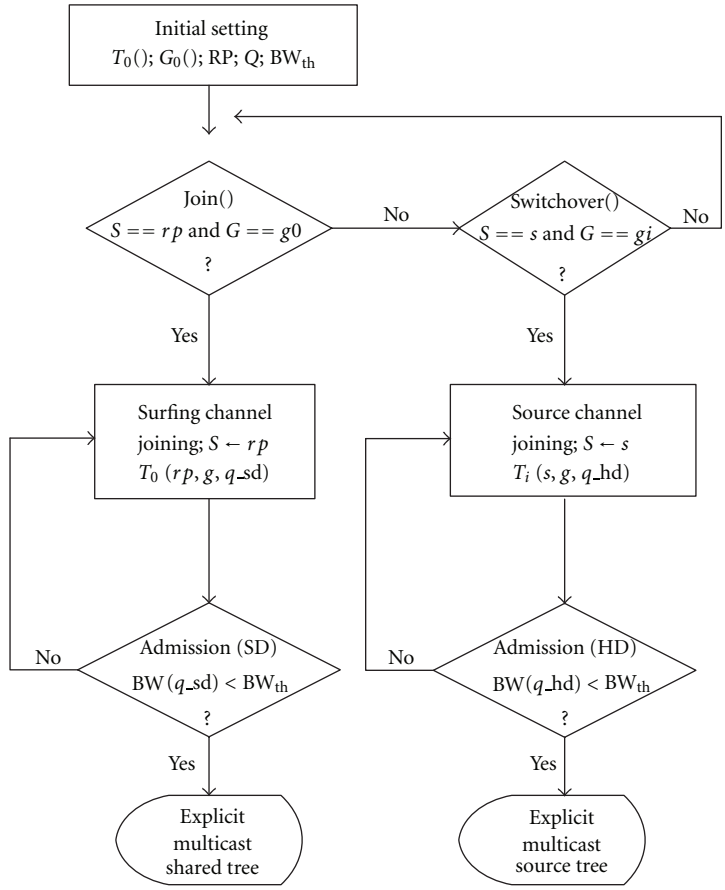


Figure 3: IPTV channel control algorithm for RP node.

shortest path first). The group source mapping table is generated. The surfing channels are expressed by explicit routed states in core routers as the multicast branching toward members. The more different IPTV channels are joined, the more network resource is utilized by multicast packet delivery.

4.1.2. Channel Change by Hybrid-Tree Switchover. As mentioned in Section 2, the channel change is a random process for viewer's behavior to select preferable watching channel. To reduce IPTV channel change time, the switchover control message is fast operated between RP shared tree and source tree simultaneously. The fast switchover mechanism, which is considered by link bandwidth utility and channel change time, can insert temporary SD quality stream from the same RP-aggregated shared tree as soon as possible to HD quality stream buffer for watching channel. After the new source tree switchover is finished, the HD quality media stream is delivered by specific source tree.

When the channel change request is occurred, the RP node executes switchover process for group members. After being informed by switchover control message, the source node computes the explicit routes for source tree T_i by CSPF and then updates the channel states for watching stream delivery.

4.1.3. Channel Setup with Multicast Admission Control. According to the group member joining request with QoS requirements, the HT-ERM admission control will check the link bandwidth status for multicast hybrid tree. To guarantee QoS for multicast stream delivery, the upper bound of link utilization is defined by *bandwidth threshold* (BW_{th}) for efficient resource allocation that IPTV channel can carry the media streams through specific source tree T_i and shared tree T_0. The bandwidth threshold (BW_{th}) is to maximize the bandwidth usage for the total watching channel demands and additional channel change bandwidth estimation for network links.

Because the RSVP-TE is explicit routed signaling protocol, the admission control is used to reserve resource for hybrid-tree-based multicast. The surfing channel joining and watching channel switchover are admitted by comparing between available resource and bandwidth threshold (BW_{th}). When the total bandwidth of T_i and T_0 exceed the threshold BW_{th}, the blocking ratio for joining requests will be increasing due to detection of bandwidth overthreshold.

4.2. Channel Delivery Operational Differences between HT-ERM and PIM-SM. The differences between proposed HT-ERM and PIM-SM multicast protocol for channel delivery are listed as follows.

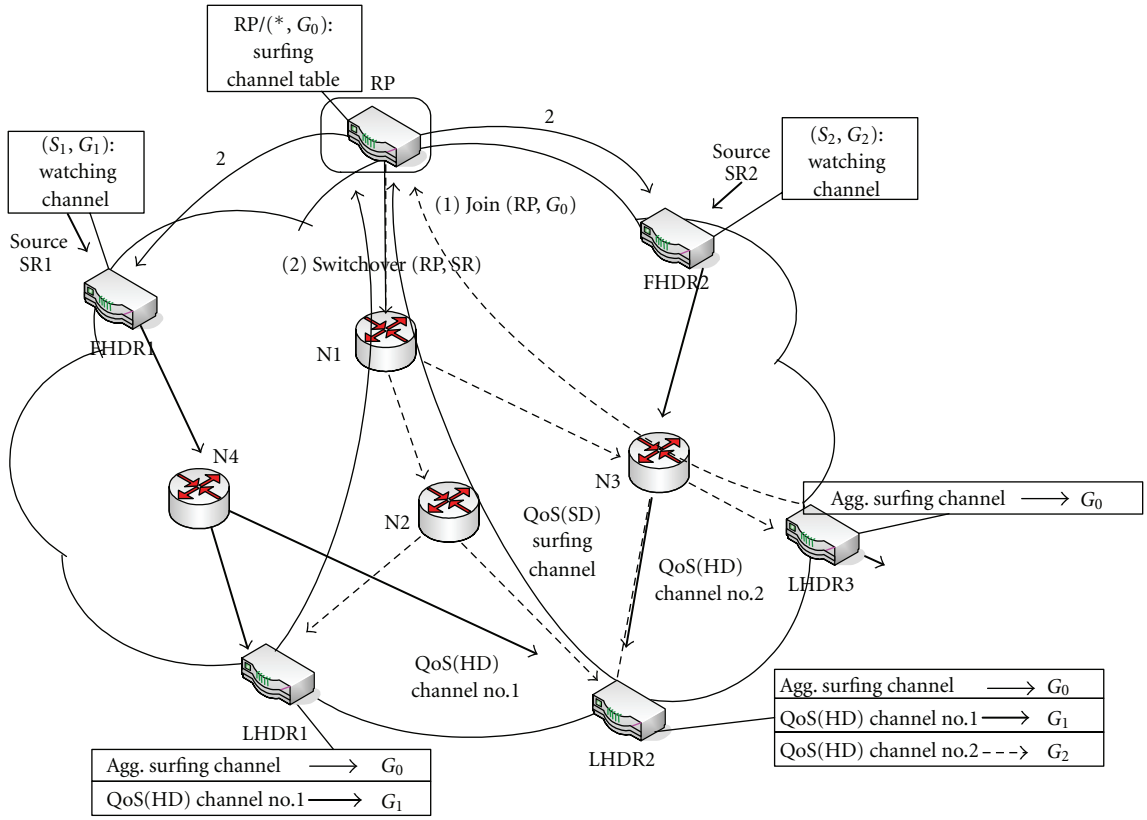


Figure 4: IPTV channel delivery through HT-ERM operations.

(1) *Source group mapping table*: because PIM-SM multicast routing is based on reverse path check, it is difficult to make accurate QoS estimation for bandwidth allocation from FHDR router of source node (or RP node) to multiple LHDR routers. The proposed HT-ERM scheme is based on mapping table to compute the source multicast explicit routed path forwarding toward all joining multicast group members.

(2) *The modified control messages*: Join/Leave (RP, G_0) and Switchover (RP, SR_i) are adopted by the extended multicast protocol message of PIM-SM for IPTV channel setup and change. The control messages of HT-ERM are operated to enhance delivery performance between FHDR and LHDR.

To explain the HT-ERM protocol operations, we take an illustrative example in Figure 4. Firstly, the source node SR1 and SR2 will register to RP for surfing streams aggregation in RP shared tree. All group members distributed in LHDR nodes request RP to join to group G_0 for surfing channels. Secondly, upon receiving the join requests from LHDR 1–3, the RP function performs multicast channel initialization, and the group-source mapping table is updated by shared tree for surfing stream aggregation from SR1 and SR2. By information collecting from all sources and group members, the shared tree in hybrid multicast tree can be computed by CSPF for the explicit multicast routes of surfing channels.

In the scenario of LHDR1, the switchover message is completed to request new watching channel no. 1 from node SR1 via RP. However, in the scenario of LHDR2, not only one viewer requests watching the channel no. 1, one of views desires to change channel from the channel no. 1 to new channel no. 2. Instead of rejoining to SR2 for new channel no. 2 (S2, G2), RP can relay the control message *Switchover* (RP, SR) to inform old source node SR1 to retain the routes of subpath to LHDR2 for channel no. 1. The new source node SR2 is admitted to submit the media stream from channel no. 2 to LHDR2 by using the explicit routes with available bandwidth over (S2 and G2) source tree through core router N3. As for the scenario of LHDR3, the viewer is still in surfing state without decision making for specific watching channel.

5. Performance Evaluation

The performance measures of the proposed HT-ERM scheme for IPTV service are in terms of blocking ratio of admission control, resource utilization, and channel change delay over multicast core network and access network employed as performance. As compared with two underlying protocols, PIM-SM with ECMP algorithm and QMRP with QSPF algorithm, the HT-ERM with CSPF algorithm will be verified as a valid QoS supported IPTV multicast approach.

5.1. Simulation Parameters Setup. The simulation is conducted over two different network topologies. The first one

is the random graph RandNet with 100-node and 294-link, generated by GT-ITM [14] network topology generator. The other is the fixed backbone graph NSFNet [15], which is abstracted from a real network model with 14 nodes and 42 bidirectional links. The multicast core network environment is setup by sources and receivers randomly attached to any network edge node as the designated router. In experiments, the number of join requests is measured from 500 to 5000 per 500 increasing step, to join different multicast channels. The group size is proportional to the total number of group member, and the joining requests from each node are uniformly distribution. From the aspect of viewer's random behavior, channel change joint state probability equals to the relationship with channel switching ratio (α) and channel popularity. The popular channels are usually assumed to stay within HD quality stream distribution ratio (β). According to the empirical estimation by Zipf's law in (3), the cumulative probability of channel popularity is over 50% when β is set to 0.2 (i.e., top 10 popular channels over total 50 channels). In our IPTV channel test scenario, those simulation parameters are summarized in Table 1.

Table 1: IPTV multicast simulation parameters.

IPTV channel delivery test conditions	Parameter
Total number of IPTV channels	50
Channel switching ratio α	0.25, 0.5, 0.8
HD QoS stream distribution ratio β	0.2
HD/SD QoS stream bitrate	10/2 Mbps
Bandwidth threshold	80%
Link bandwidth capacity	500 Mbps

Table 2: Hybrid multicast scheme with routing algorithms.

Hybrid-tree protocol	Multicast routing algorithm
PIM-SM	Equal cost multipath (ECMP)
QMRP	QoS shortest path first (QSPF)
HT-ERM	Constrained shortest path first (CSPF)

5.2. Performance Metrics. In HT-ERM channel control simulation, the performance evaluations for hybrid-tree-based multicast scheme with different multicast routing algorithms are employed by PIM-like protocols. Table 2 lists hybrid multicast with three different multicast routing algorithms. The proposed HT-ERM scheme is used by CSPF algorithm. The conventional PIM-SM multicast scheme can be adopted by the equal cost multi-path (ECMP) algorithm. The QoS-aware multicast routing protocol (QMRP) is used as PIM-like protocol based on QoS shortest path first (QSPF or QoS-SPF). The proposed HT-ERM scheme is source initial explicit routed multicast based on available bandwidth. The other schemes, multicast ECMP and multicast QSPF algorithms, are receiver-based multicast tree join by traversing the single path and/or multiple shortest path computations according to the link cost.

The performance metrics for multicast network delivery and QoS channel control are defined as following.

(i) Multicast Tree Setup Ratio. The multicast tree is computed by the specific on-tree node in RP node or source node. The tree setup ratio can be represented as the average number of multicast source and shared trees for multicast channel (S, G) established by the joining requests from group members.

(ii) Multicast Forwarding Entries. The total number of multicast forwarding entries can be represented by the number of forwarding entries in multicast routing table per multicast router and the number of multicast router on multicast distribution tree T with group member g. The number of multicast forwarding entries ε is calculated by

$$\varepsilon(T, g) = N_e \times N_T = (N_s + N_b + N_t + N_l) \times N_T \quad (4)$$

where N_e is the total number of forwarding entries in multicast routing table, that is, the total number of (S, G) entries in all multicast nodes for distribution tree T, including the root node number N_s, branching node number N_b, transit node number N_t, and leaf node number N_l. The number of multicast tree is denoted by N_T.

(iii) Maximum Multicast Resource Usage. The ratio of the utilized bandwidth is calculated in most traffic-congested link over multicast tree. The metric of resource usage U is the total bandwidth consumption BW for QoS channels with HD and SD streams through multicast tree T from joining requests of group member g by

$$U(Q, T) = \sum_{i \in T, q \in Q} \frac{\mathrm{BW}(i, q)}{C_T}, \quad (5)$$

where C_T is the total tree link capacity over multicast network, and $\mathrm{BW}(i, q)$ is the reserved bandwidth for HD stream and SD stream over each link i through distributed tree T.

(iv) Blocking Ratio. The ratio of rejection service request is divided by total requests under the admission control with bandwidth threshold $\mathrm{BW_{th}}$. The metric of blocking ratio B is defined

$$B(Q, G) = \sum_{g \in G} \frac{N_R}{N_g}, \quad (6)$$

where N_g is the number of join requests from group member G for available IPTV channels, and N_R denotes the number of rejected member requests for specified QoS channel by admission control.

(v) Multicast Channel Change Delay. it is considered by channel processing delay, watching channel stream transmission, and surfing channel switching delay over multicast trees

T with group member size g. The metric of total switching delay time D can be summed up by

$$D(T,g) = D_{1,\text{channel processing delay}} + D_{2,\text{surfing channel switching delay}} + D_{3,\text{watching channel transmission delay}}, \quad (7)$$

where D_1 denotes channel processing delay, D_2 denotes surfing channel switching delay, and D_3 denotes watching channel transmission delay.

5.3. Comparisons of Multicast Network Delivery. We carried out following simulations over 100-node and 294-link random graph RanNet to evaluate the performance for multicast routing algorithms such as multicast ECMP, receiver-based QMRP- and RP-based HT-ERM scheme.

(1) Multicast Tree Setup Ratio Comparisons. The *multicast tree setup ratio* is measured by the average number of multicast source and shared trees computed by specific on-tree nodes per joining requests from group members. The main effect of multicast tree computation is determined by the number of active joining requests during channel change. The popular channel ratio β with HD QoS stream distribution is set to 0.2. With incremental channel switching ratio α by 0.25, 0.5, and 0.8, we observed the setup trend of multicast tree for multicast channel (S, G) computed by proposed HT-ERM routing schemes in RP and source nodes. Figure 5 shows the results for multicast tree setup ratio versus the average number of group member requests that compared HT-ERM scheme with CSPF, PIM-SM protocol with ECMP, and PIM-SSM protocol with QSPF. We found that the lowest setup ratio of multicast tree for ECMP scheme can afford for large amount of new channel change requests to diverse traffic flows into multiple links of shared tree. It means more multicast tree setup leads to more computation and delivery resource consumption. We also derive that multicast trees setup is almost identical to channel switching ratio in HT-ERM scheme with parameter $\alpha = 0.25, 0.5, 0.8$. When the switching ratio is increasing to 0.8, the multicast tree setup ratio of HT-ERM scheme still outperforms that of QMRP. That is because RP node uses HT-ERM with CSPF algorithm to gain the minimum hybrid-tree setup ratio with shared and source tree, instead QMRP is based on receiver multipath for source trees setup.

(2) Multicast Forwarding Entry Comparisons. Multicast *forwarding entry* is represented as the control overhead of multicast channel state maintenance in multicast forwarding table for network delivery through the multicast delivery tree. The total number of forwarding entries (S, G) consists of root node number, branching node number, transit node number, and leaf node number over the distribution tree with group member. The increasing size of multicast forwarding entries is proportional to the number of group member joining requests because of grafted subpath from multicast branching node. Therefore, the performance for forwarding entry scalability may be affected by routing control scheme.

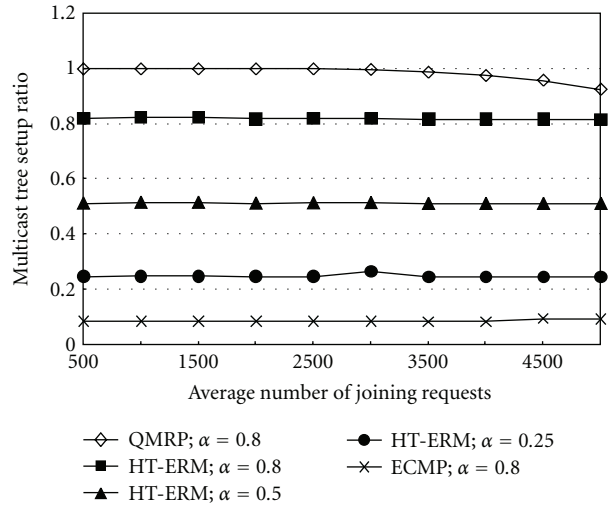


Figure 5: Multicast tree setup ratio versus group member joining requests.

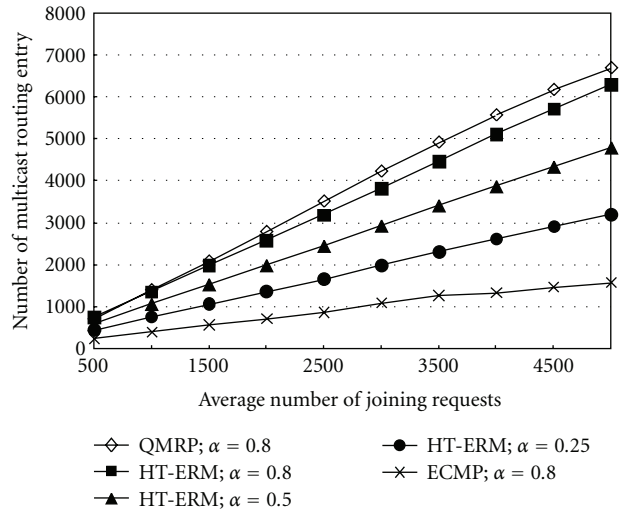


Figure 6: Comparisons of multicast forwarding entry size.

Figure 6 shows the comparison of ECMP, QMRP, and HT-ERM scheme with number of the forwarding entry number versus the average number of group joining requests. We found that the number of multicast forwarding entries for HT-ERM is increasing largely from 3,199 to 6,301 (i.e., the raising ratio is over 50%) in condition of HD QoS stream distribution ratio ($\beta = 0.2$) and multicast channel switching ratio ($\alpha = 0.8$, 0.5 and 0.25), when group joining member requests are more than 5000 times. When switching ratio is set to 0.8, the growth of forwarding entry size computed by QMRP is a little greater than that of HT-ERM. In contrast, HT-ERM uses the explicit routes in RP-shared tree that can reduce more forwarding entries in multicast trees. However, ECMP is used to compute the unique shared tree, so that forwarding entry size is almost the same.

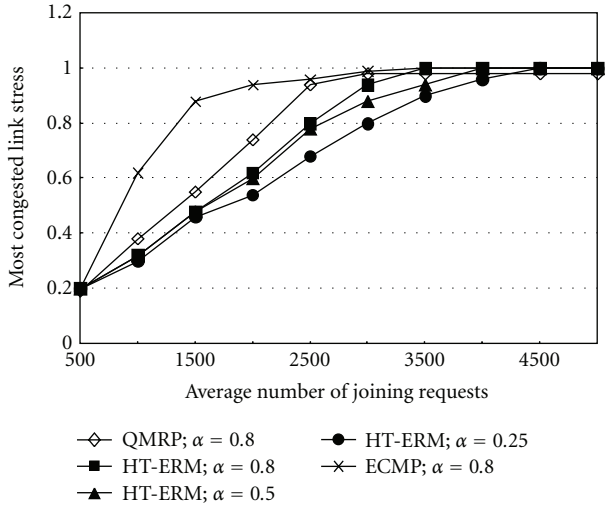


Figure 7: Multicast tree link resource usage versus group joining requests.

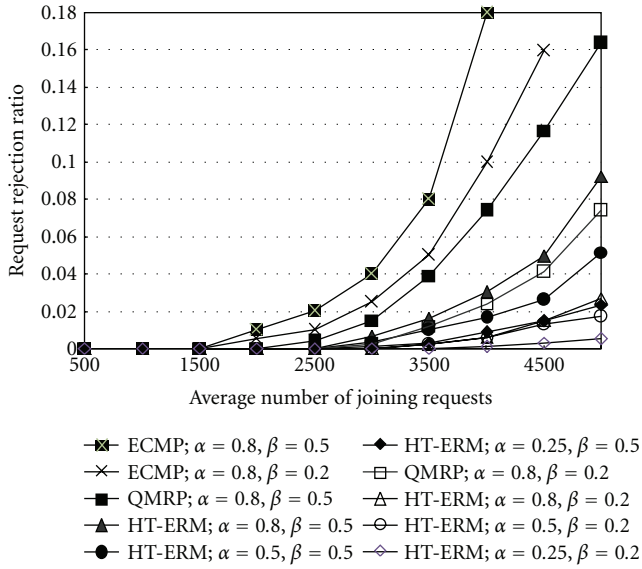


Figure 8: Request rejection ratio versus group joining requests.

(3) Multicast Tree Link Resource Usage Comparisons. In multicast resource usage comparison, the multicast tree link stress is estimated by summation with the various channel switching ratio ($\alpha = 0.8$, 0.5, and 0.25), under the HD QoS stream distribution ratio ($\beta = 0.2$). In Figure 7, we observed that HT-ERM achieves the best resource efficiency utilized on the most congested multicast tree link as compared with ECMP and QMRP approach before link resource is overutilized by at the number of joining requests up to 4000. In other words, the resource usage control of the proposed HT-ERM scheme is efficient by using CSPF algorithm because the explicated multicast routing can limit bandwidth threshold to redistribute heavy traffic loads between the shared tree and source trees over entire network topology.

(4) Joining Request Rejection Ratio Comparisons. The blocking for joining request is caused by insufficient resource allocation and high-level QoS request. The admission control can detect the available bandwidth in advance before accepting the joining requests with channel access. On the other hand, high-level QoS request may be rejected by either source node or RP node when multicast tree is switching over the specific links of source tree. By adjusting the various channel switching ratio ($\alpha = 0.8$, 0.5, and 0.25) and HD QoS stream distribution ratio ($\beta = 0.2, 0.5$), we evaluate the performance for large amount of joining requests for multicast tree setup by comparing multicast routing algorithm with ECMP, QMRP, and HT-ERM. We found that the rejection ratio of each routing algorithm is increasing by large amount of the joining requests from group members at higher channel switching ratio ($\alpha = 0.8$) and higher QoS stream distributed ratio ($\beta = 0.5$), as shown in Figure 8. Especially, even in the worst conditions of channel switching ratio ($\alpha = 0.8$) and HD QoS distribution ratio ($\beta = 0.5$), the performance of rejection ratio for HT-ERM routing algorithm is better than that of ECMP and QMRP routing algorithm. As the result, we can prove that the CSPF algorithm of proposed HT-ERM can achieve efficient resource allocation and effective admission control for QoS requirements.

5.4. Comparisons of IPTV Channel Control. We carried out following simulations over real backbone network NSFNet with 14-node and 42-link to evaluate the performance for different IPTV multicast protocols PIM-SM, QMRP, and HT-ERM applied by associated channel control scheme.

(1) Channel Change Delay. In metric definition (7), the major channel delay effects are caused by surfing channel switching delay D_2, and watching channel transmission delay D_3. We simulated a large amount of groups to receive different IPTV watching channels over NSFNet topology and obtained the results of channel change delay. The Figure 9 demonstrates comparisons among different multicast tree construction for successful channel change from 10 to 50 at each access node.

The traditional RP-shared tree setup with ECMP algorithm leads to the largest transmission delay during channel change. The traffic of multipath is separated on the shared tree links so as to result in the large switching control delay; however, the multi-path can reduce the traffic load. As the QRMP approach constructs the source tree by QoS routing algorithm (i.e., QSPF), the IPTV channel traffic distribution can be diversified by different source tree links. The proposed HT-ERM can construct hybrid multicast tree based on CSPF algorithm to reduce the switching latency by source and shared tree switchover. As compared by QSPF routing algorithm, the proposed CSPF of HT-ERM scheme improves the performance for channel change delay. The importance of this simulation result indicates that the source-initiated QoS routing algorithm can achieve better channel switching

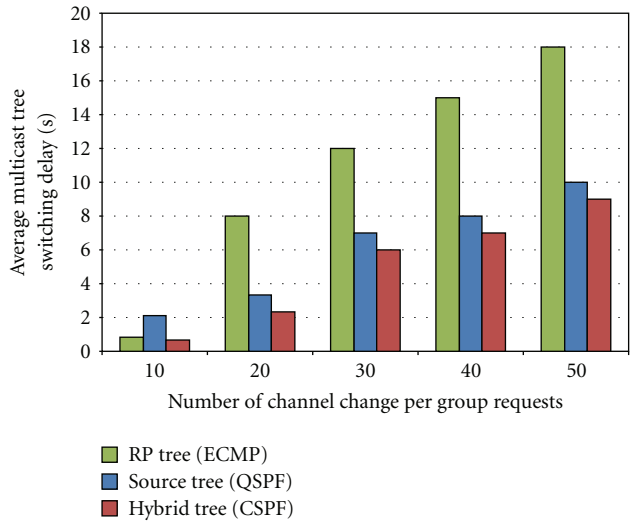


Figure 9: Channel change delay comparisons by multicast tree switching and transmission.

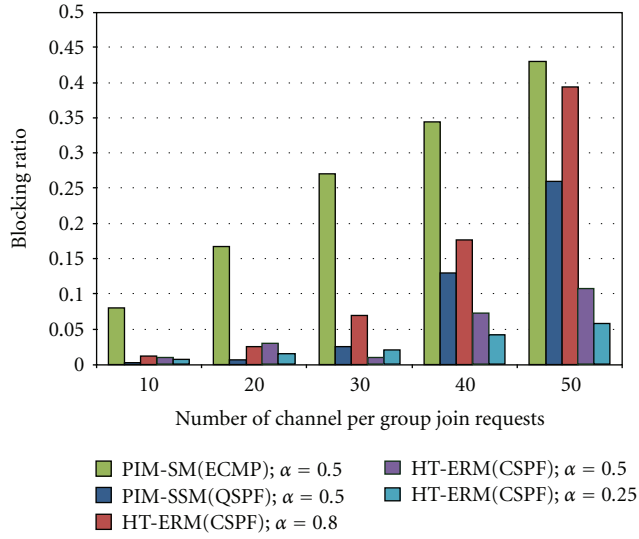


Figure 10: Comparisons of blocking ratio for channel control.

control performance than that of the receiver-initiated routing algorithm. Therefore, the proposed HT-ERM scheme can reduce the channel change delay by enhancing the hybrid-tree operation with efficient switchover mechanism for RP-centralized control.

(2) Channel Blocking Ratio. To simulate the blocking ratio for practical IPTV channel service, multicast routing protocols are employed by using different routing algorithm for number of IPTV channel request per group joining. By adjusting the channel switching ratios (α = 0.25, 0.5, and 0.8) for different test scenarios, the HD QoS IPTV channels are requested to join per group member using PIM-SM with ECMP, QMRP with QSPF, and the proposed HT-ERM with CSPF. The results of Figure 10 indicate that the HT-ERM scheme with CSPF outperforms the other two QSPF and ECMP algorithm beneath the average switch ratio (α = 0.5), while the number of channel joining requests is increasing from 10 to 50. We observe that the traditional PIM-SM multicast using ECMP resulted in the highest blocking ratio due to the traffic aggregation over shared tree with the same routing path, regardless of the multiple parallel links. In additions, note that the channel blocking ratio of HT-ERM is rising sharply by admission control at higher switching ratio α = 0.8.

(3) Channel Resource Usage. Figure 11 shows the results of resource usage for delivering watching streams over the requested IPTV channels using different multicast approaches. The resource consumptions are nearly even with those multicast approaches: PIM-SM (ECMP), QMRP (QSPF), and proposed HT-ERM (CSPF) at any channel switching ratio (α = 0.8, 0.5, 0.25). For the effect of traffic load balancing, QMRP and HT-ERM scheme can achieve the better link usage performance as compared by PIM-SM while the number of channel joining is increasing to 50. Because

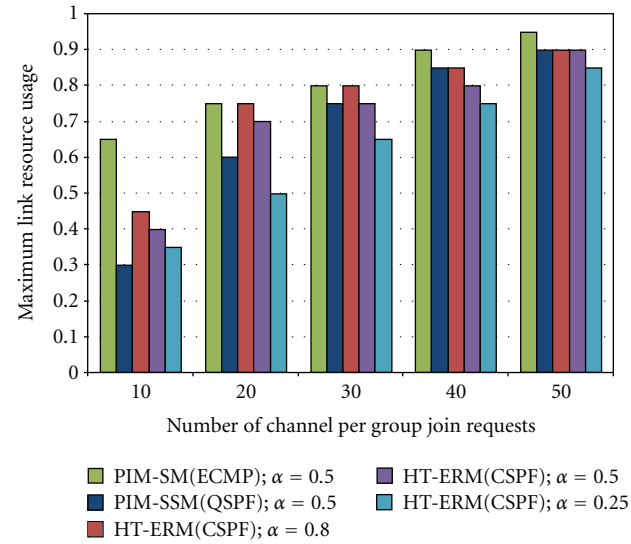


Figure 11: Comparisons of resource usage for IPTV channels.

the QSPF routing algorithm is based on receiver's multiple tree-nodes joining decision, the channel traffic distribution can perform better than that of the proposed HT-ERM at lower channel joining requests of 10 and 20. However, the maximum link utilization of HT-ERM performs nearly the same resource usage at higher channel joining requests more than 30. This is the cross-effect caused by network congestion and traffic control. Therefore, the proposed HT-ERM scheme can operate hybrid-tree efficiently via enhanced RP functionality which makes better resource management for HD/SD channel stream utilization and traffic load distribution by hybrid-tree multicast channel service.

6. Conclusions

As the killer application in future Internet, IPTV service needs to provide the effective and efficient operations for channel delivery and control. The hybrid-tree-based multicast IPTV has been validated to supply watching channels and surfing channel services. However, we need to overcome the drawbacks in RP functionality and switchover mechanism to achieve effectiveness and efficiency in hybrid-tree multicast channel control and delivery. To solve the QoS degradation problems of multicast network utilization and IPTV channel change delay, the hybrid-tree-based explicit routed multicast (HT-ERM) scheme is proposed to enhance current multicast protocols for QoS-supported IPTV service. The main contributions focus on performance improvement for multicast network delivery and IPTV channel control, including the reduction of maximum link resource utilization, fast switchover mechanism for channel change delay, and admission for multicast QoS channel setup. In network-layer level, HT-ERM provides flexible approach to design hybrid-tree-based IPTV multicast for IP multicast protocols. We also validate multicast solution to QoS-supported IPTV channel control and delivery by simulation. The result shows that the proposed HT-ERM outperforms existing QoS multicast approaches applied by PIM-related protocols while retaining QoS guarantee.

References

[1] Y. Xiao, X. Du, J. Zhang, F. Hu, and S. Guizani, "Internet protocol television (IPTV): the killer application for the next-generation internet," *IEEE Communications Magazine*, vol. 45, no. 11, pp. 126–134, 2007.

[2] W. Sun, X. Luo, K. Lin, and Y. Guan, "Performance analysis of a finite duration multichannel delivery method in IPTV," *IEEE Transactions on Broadcasting*, vol. 54, no. 3, pp. 419–429, 2008.

[3] D. E. Smith, "IP TV bandwidth demand: multicast and channel surfing," in *Proceedings of the 26th IEEE International Conference on Computer Communications (INFOCOM '07)*, pp. 2546–2550, Anchorage, Alaska, USA, May 2007.

[4] S. Chen, K. Nahrstedt, and Y. Shavitt, "QoS-aware multicast routing protocol," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 12, pp. 2580–2592, 2000.

[5] H. Holbrook and B. Cain, "RFC 4607: Source-Specific Multicast for IP," Internet Engineering Task Force, August 2006.

[6] B. Fenner, M. Handley, H. Holbrook, and I. Kouvelas, "Protocol Independent Multicast-Sparse Mode (PIM-SM): Protocol Specification (Revised)," RFC 4601, August 2006.

[7] C. C. Wen, C. S. Wu, and M. T. Yang, "Hybrid tree based explicit routed multicast for QoS supported IPTV service," in *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM '09)*, December 2009.

[8] M. Bag-Mohammadi, N. Yazdani, and S. Samadian-Barzoki, "On the efficiency of explicit multicast routing protocols," in *Proceedings of the 10th IEEE Symposium on Computers and Communications (ISCC '05)*, pp. 679–685, June 2005.

[9] S. Yasukawa, M. Uga, H. Kojima, and K. Sugisono, "Extended RSVP-TE for Multicast LSP Tunnels," IETF draft, June 2003.

[10] C. Sasaki, A. Tagami, T. Hasegawa, and S. Ano, "Rapid channel zapping for IPTV broadcasting with additional multicast stream," in *Proceedings of the IEEE International Conference on Communications (ICC '08)*, pp. 1760–1766, May 2008.

[11] Y. Kim, J. K. Park, H. J. Choi et al., "Reducing IPTV channel zapping time based on viewer's surfing behavior and preference," in *Proceedings of the IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB '08)*, April 2008.

[12] H. Joo, H. Song, D. B. Lee, and I. Lee, "An effective IPTV channel control algorithm considering channel zapping time and network utilization," *IEEE Transactions on Broadcasting*, vol. 54, no. 2, pp. 208–216, 2008.

[13] Cisco Technical Support Module, *Load Splitting IP Multicast Traffic over ECMP*, Cisco Systems, 2007.

[14] B. Chinoy and H.-W. Braun, "The national science foundation network," Technical Report GA-A210029, SDSC, 1992.

[15] K. L. Calvert, E. W. Zegura, and S. Bhattacharjee, "How to model an internetwork," in *Proceedings of the 1996 15th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '96)*, pp. 594–602, March 1996.

Hindawi Publishing Corporation
International Journal of Digital Multimedia Broadcasting
Volume 2012, Article ID 540801, 12 pages
doi:10.1155/2012/540801

Research Article

A Secure and Stable Multicast Overlay Network with Load Balancing for Scalable IPTV Services

Tsao-Ta Wei,[1,2] Chia-Hui Wang,[1] Yu-Hsien Chu,[2,3] and Ray-I Chang[3]

[1] *Department of Computer Science and Information Engineering, Ming Chuan University, No. 5, Deming Road., Guishan Township, Taoyuan County 333, Taiwan*
[2] *Technical Division, Develop Department, YES Information Incorporated, Xinyi District, Taipei 110, Taiwan*
[3] *Department of Engineering Science and Ocean Engineering, National Taiwan University, Taipei 106, Taiwan*

Correspondence should be addressed to Chia-Hui Wang, wangch@mail.mcu.edu.tw

Received 31 January 2012; Accepted 31 March 2012

Academic Editor: Pin-Han Ho

The emerging multimedia Internet application IPTV over P2P network preserves significant advantages in scalability. IPTV media content delivered in P2P networks over public Internet still preserves the issues of privacy and intellectual property rights. In this paper, we use SIP protocol to construct a secure application-layer multicast overlay network for IPTV, called SIPTVMON. SIPTVMON can secure all the IPTV media delivery paths against eavesdroppers via elliptic-curve Diffie-Hellman (ECDH) key exchange on SIP signaling and AES encryption. Its load-balancing overlay tree is also optimized from peer heterogeneity and churn of peer joining and leaving to minimize both service degradation and latency. The performance results from large-scale simulations and experiments on different optimization criteria demonstrate SIPTVMON's cost effectiveness in quality of privacy protection, stability from user churn, and good perceptual quality of objective PSNR values for scalable IPTV services over Internet.

1. Introduction

Due to the prevalent broadband Internet access and advanced video compression techniques, the Internet Protocol Television (IPTV) has been emerging as one of the most popular Internet applications. IPTV can further benefit Internet users by entertainment, social, and business values, but IPTV faces more challenges of scalability, privacy, and service quality over the public Internet due to the conventional client-server architecture applied.

The success of well-known P2P video-streaming systems such as PPStream [1], PPLive [2], Sopcast [3], and TVants [4] has proven that P2P paradigm is a feasible solution to deliver bandwidth-hunger IPTV media content in large scale over the pervasive Internet. However, the above-mentioned proprietary P2P video-streaming systems still suffer the issues of long startup delays, significant video-switching delays, large peer playback lags, and security due to the peer heterogeneity and churn [5–7].

Therefore, the P2P overlay networks for future multimedia Internet should overcome the previous shortcomings to further promise quality of service, security, and experience to the IPTV end users. Moreover, the P2P overlay architecture for promising IPTV services should be not only easily convergent in heterogeneous networks, but also feasibly integrated with other Internet applications.

In this paper, we apply SIP protocol [8] to construct an application-layer multicast (ALM) overlay network, which is called SIPTVMON, with privacy protection, load balancing, and stability for IPTV to overcome the disadvantages previously mentioned in P2P video-streaming systems. Our contributions in proposed SIPTVMON for IPTV service are summarized as follows.

(i) Stability for Peer Heterogeneity and Churn. Since the peers (i.e., users) with ever-changing Internet access bandwidth may join and leave the IPTV service anytime as they wish, we continue to optimize SIPTVMON overlay tree with minimum SIP signaling overhead to achieve stable IPTV service by the product of average link bandwidth and service life time in peers.

(ii) Security Provision. Most of the IPTV content preserves intellectual property rights, so the content delivery paths in SIPTVMON tree are secured against Internet eavesdroppers via elliptic-curve Diffie-Hellman (ECDH) [9] key exchange algorithm and AES [10] encryption.

(iii) Shorter Service Delays. Users may enjoy rapidly switching different IPTV channels over P2P networks and suffer the significant video-switching delay and larger playback lag, so our proposed SIPTVMON provides not only the peer's graceful leaving procedures realized by SIP protocols, but also the above-mentioned stability optimization for peers' joining and leaving to minimize average service delay from the user churn.

(iv) Interoperability with Other Internet Applications. The applied SIP protocol in SIPTVMON has been widely and successfully deployed in Voice over IP (VoIP) applications. Moreover, the core of IP Multimedia Subsystem (IMS) [11] in 3G telecommunication is also constructed by SIP protocol. We believe that the proposed SIPTVMON framework not only can cooperate with SIP-enabled Internet applications like prevalent VoIP applications, but also is feasible to help IMS in 3G mobile networks to achieve scalable IPTV service provision in more cost-effective way.

The remainder of this paper is organized as follows. In Section 2, we describe the related works of ALM overlay network, privacy protection for IPTV media streaming, and SIP-signaling protocol. The details of proposed SIPTVMON architecture with security, load balancing, and stability are presented in Section 3. Section 4 describes simulation experiments for SIPTVMON and their performance results for P2P IPTV. Finally, we conclude the paper and future work.

2. Related Work

As shown in Figure 1, the logical topology applied in current P2P video-streaming technologies is roughly classified into tree, mesh (i.e., multiple trees), and hybrid of tree and mesh [12]. Though tree-based P2P structure preserves simplicity, it is vulnerable to peers' dynamics of heterogeneity and churn. Mesh-based P2P structure improves the resilience to the dynamics from peers, but it preserves more complex peer partnership relations. In this paper, we adopt tree-based, rather than mesh-based, P2P architecture in proposed SIPTVMON to cost-effectively achieve low service latency, security provision, and stability for IPTV users churn over Internet.

We briefly review the related works in proposed solutions to construct an application-layer multicast overlay network with privacy protection, load sharing, and stability for scalable IPTV service.

2.1. Application Layer Multicast (ALM). ALM is an application-level traversal method for IP multicast packets without the help from routers through unicast tunneling. ALM is also known as a cost-effective tool to construct overlay networks for large-scale Internet multimedia applications. ALM has the advantages of less overhead of maintenance than routers, provision of much larger multicast groups than IP multicast, less compatibility issues than IP multicast and, easier extension to new features like security, error control, stability, and so forth.

Because routers usually disable forwarding IP multicast packets to prevent the flooding of multicast data, self-organization algorithms for effective transmissions in logical topology of multicast overlay network become the essential of ALM mechanism.

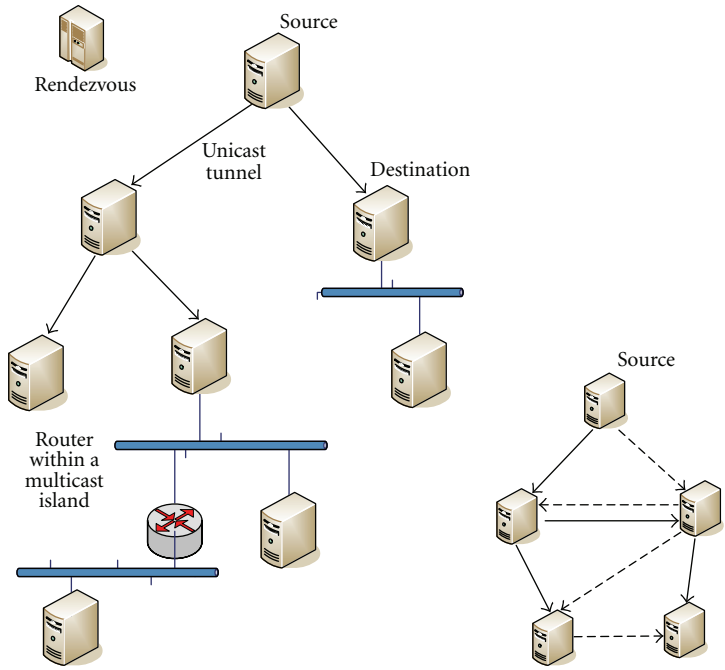
As shown in Figure 1(a), a single-tree ALM aims to provide best-effort single-source data streaming with the optimization of reduced latency and loss rates. For example in [13, 14], to join an ALM tree for streaming data, a new member (i.e., multicast agent, MA, USA) must first connect to a directory server, a rendezvous point to which every member MA must connect at the first time. Then, it will be able to obtain a member list to find the lowest round trip time (RTT), loss rate, or bandwidth among existing members for the better quality of multimedia-streaming service from source ALM.

The mesh topology illustrated in Figure 1(b) is applied to provide bandwidth-consuming peer-to-peer video streaming with the advantages of avoiding replicating group management across multiple (per-source) trees, and resilience of member failure [15]. ALM mesh can be regarded as the superposed overlay of multiple spanning trees. Compared with the single-tree approach, the multiple-tree approach is more complicated.

In [16], a low-delay high-bandwidth mesh called Fast-Mesh for peer-to-peer live streaming is proposed. In this work, the authors propose a centralized heuristic with complete mesh knowledge to minimize the maximum delay of all peers in the mesh. They demonstrate via simulations that their solution can reach a small average delay of 50 ms to 100 ms for hundreds peers in the mesh with a source data rate of 10 kbps. In their experimental test bed over the Internet across several countries, the implemented Fast-Mesh still shows low delay, ranging from tens to 500 ms with a small data source rate of 30 kbps.

In our proposed SIPTVMON, a single-source ALM tree scheme is applied for its simplicity in not only the applied privacy protection, but also the tree adjustment in optimization of both scalability and stability for IPTV services. Moreover, the tree-based SIPTVMON can also lessen average service delays for large-scale peers and reduce the flow of control messages.

2.2. Privacy Protection for IPTV. The privacy protection for Internet video streaming is usually done by symmetric encryption for less time consuming. There are two kinds of symmetric encryption mechanisms. One is the full encryption approach; the other is selective encryption. Generally speaking, the main disadvantages of selective encryption are insufficient security and video content dependent. On the contrary, full encryption is often criticized for the longer calculation delay which is not suitable for real-time video



(a) Single-source tree

(b) Single-source mesh (multiple-tree)

Figure 1: P2P IPTV structure of tree and mesh.

streaming. However, based on the results of a processing speed experiment [17] on full encryption in 2007, the criticism of longer delays is not an issue. In this test, which involved a computer with Intel dual core 1.83 GHz, the encryption bit rate for triple DES, DES, and AES can attain to 104 Mbps, 272 Mbps, and 792 Mbps, respectively. The bit rate of video-streaming service on Internet is very rarely higher than 10 Mbps, thus showing that full encryption has the potential to support real-time video transmission like IPTV.

For the cost-effective privacy protection of real-time video streaming in IPTV, advanced encryption standard (AES), which is a symmetric-key encryption standard adopted by the US government, is most applicable to protect voice data from eavesdropper over Internet. The standard comprises three block ciphers, AES-128, AES-192, and AES-256, adopted from a larger collection originally published as Rijndael. Each of these ciphers has a 128-bit block size, with encryption key sizes of 128, 192, and 256 bits, respectively.

Since AES is a symmetric crypto system, it needs a key management infrastructure to issue a common secrete key (i.e., session key) for later video encryption/decryption between sender and receiver in IPTV.

As summarized in [18], three different methods of the session key distribution are preshared key, public-key encryption, and the Diffie-Hellman (DH) key exchange [19]. There is only a very small amount of data that has to be exchanged in the preshared key method. But, it will incur the scalability issue in large group of communication peers. The public-key encryption can be used to create a scalable privacy-protection IPTV system and usually requires public key infrastructure (PKI) to distribute public key. Its disadvantage is consuming much more resource than the preshared key.

Generally, the third method of DH also has the scalability to protect large-scale IPTV services without the need of PKI. To prevent DH from man-in-the-middle (MITM) attack, authentication [20] between sender and receiver is needed further. Thus, applying DH session key negotiation to protect IPTV services will consume more resource of bandwidth and computation than the previous ones but without the need of centralized PKI.

In SIPTVMON, we use the popular modified DH called elliptic curve DH (i.e., ECDH [9]) key exchange via SIP-signaling protocol, since ECDH preserves much less computation overhead to construct the secure multicast overlay tree for IPTV's privacy protection.

2.3. SIP Signaling for P2P IPTV. As illustrated in Figure 2, SIP [8] is the currently widely used signaling standards for VoIP call setup and management (e.g., registration, resource administration, status, and capability exchange). Session description protocol (SDP) [21] is SIP's companion protocol to explicitly present parameters of functions applied in call setup and session management, such as the key exchange information for negotiating secret key for DH signaling. real-time transport protocol (RTP) [22] is the well-known application-layer protocol for delivering real-time media data like IPTV video packets.

As shown in Figure 3, the option k in SDP within SIP can carry the public keys for DH signaling to negotiate a common secrete key for encrypting the IPTV video in RTP

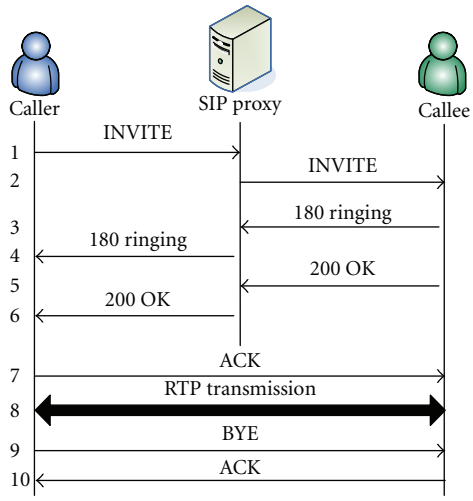


Figure 2: Basic operations of VoIP session via SIP signaling.

payload from source MA (sMA) to destination MA (dMA). Then, dMA can use this common secrete key to decrypt the encrypted RTP payload.

Since SIP with companion SDP not only can handle the setup, modification, and teardown of multimedia session, but also supports many extensions, enhancements, resource management, and interworking with other heterogeneous systems, such as privacy protection mentioned above, transferring information during ongoing session, instant messaging, and so forth, SIP is the best signaling protocol over Internet for the control messages applied to furnish the proposed solutions of security, load balancing, and stability in SIPTVMON for scalable IPTV services.

3. SIPTVMON: Secure ALM Overlay with Load-Balance and Stability for IPTV Using SIP

SIPTVMON is an overlay network composed of a super agent (SA, i.e., rendezvous) and different MAs on different multicast islands over Internet to effectively provide IPTV service for a dedicated media source from content server. SIPTVMON's MAs are dedicated computer systems or software applications to receive content data from sMA, then multicast it to their local subscribers, or unicast it again to one or more other dMAs over different multicast islands as illustrated in Figure 4. Besides, SA [23] will not only take the responsibilities to keep locations of MA and the detailed topology information of SIPTVMON but also help to forward the video from one source MA to destination MA, which is located in a private multicast island to furnish the ubiquitous IPTV service over the pervasive Internet.

However, the topology of SIPTVMON will change from time to time because the Internet users can subscribe or unsubscribe the IPTV service at any time, and the corresponding MA may join or leave the SIPTVMON while either its local users subscribe or no user subscribes the IPTV service. Meanwhile, every dMA in its multicast island may preserve different capabilities of system resources and outbound network bandwidth to forward the media content, so we propose a load-sharing scheme for SIPTVMON to prevent overloading sMA from jeopardizing perceptual quality of IPTV service for end users.

3.1. MA Joins/Leaves SIPTVMON with Security Provision. According to the long tail theory of customer demographic [24], usually most of the customers, like newly joined MAs of SIPTVMON, will not stay with SIPTVMON for a long period. To further reduce the processing overhead of reconnecting disjoint trees in SIPTVMON while a nonleaf node of MA occasionally leaves SIPTVMON, the new dMA should be joined to the leaf node of MA in SIPTVMON.

The procedures of a new MA-joining SIPTVMON are illustrated in Figure 5 and described as follows.

(*J1*) New MA denoted as MA_{new} sends a SIP "REGISTER" request with specified content identifier and registration identifier to SA to ask SA for the connection address of a leaf MA, to which the MA_{new} can be connected and join the SIPTVMON tree.

(*J2*) SA plays a role of SIP proxy and oracle of SIPTVMON topology information to send back the SIP "OK" response with SDP body of corresponding connection address of a leaf node MA_{leaf}, in which the new MA_{new} can be connected to SIPTVMON. Therefore, the SA must maintain all up-to-date SIPTVMON topology information to effectively and correctly reply the access request with a capable leaf node of handling the forwarding requested video to the new subscriber MA_{new}. That is the reason why SA is called the super agent.

(*J3*) After successful registration, MA_{new} needs to prepare the public parameters and key for the peer by ECDH key exchange algorithm. Elliptic curve [9] function's parameter $Eq_1(a_1, b_1)$, base point $G_1 = (x_1, y_1)$, and a random private key k_{new} are generated by MA_{new}. Then, a public key P_{new} can be calculated by k_{new}, Eq_1 and G_1.

(*J4*) MA_{new} sends a SIP "INVITE" request message to remote MA_{leaf} via SA proxy. The SDP body in the SIP message includes the ECDH public data of Eq_1, G_1, and P_{new}.

(*J5*) While MA_{leaf} receives the MA_{new}'s "INVITE" message and MA_{new} is authorized to join the SIPTVMON, it will randomly generate a private key k_{new} and then calculate a public key P_{leaf} according to k_{new} and receive Eq_1 and G_1. Besides, the common private key K_c for encrypting video content can be computed by k_{leaf}, received P_{new}, Eq_1 and G_1.

(*J6*) Then MA_{leaf} responses a SIP "OK" message with SDP body including the public key P_{leaf} back to MA_{new}.

(*J7*) While MA_{new} receives the public key P_{leaf} from MA_{leaf}'s SIP "OK" message, MA_{new} can use P_{leaf}, private key k_{new}, Eq_1 and G_1 to compute the common private key K_c for later decrypting the encrypted video.

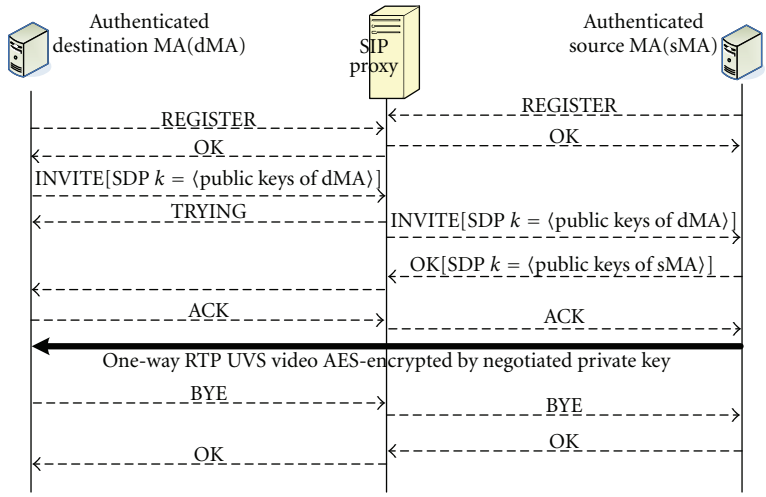


Figure 3: DH key negotiation via SIP/SDP signaling.

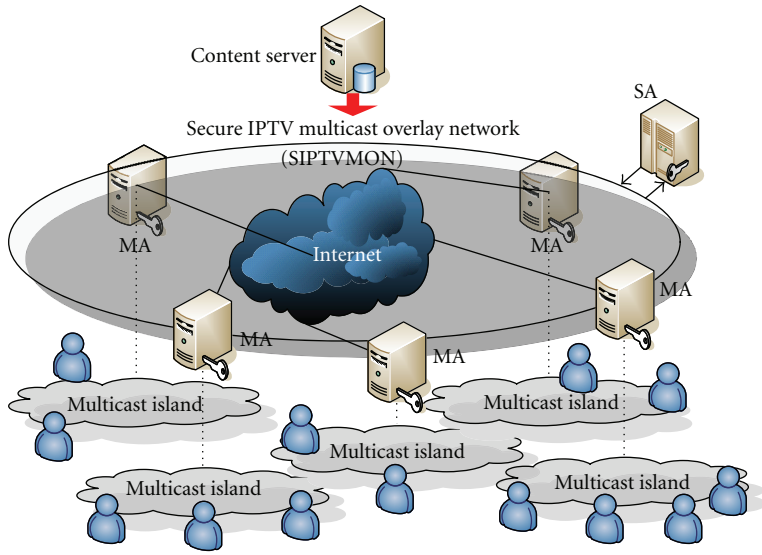


Figure 4: Primary SIPTVMON architecture.

(*J8*) Then, MA_{new} will send to MA_{leaf} a SIP "ACK" message via SA to confirm the completion of ECDH key exchange and the successful member join in SIPTVMON. Meanwhile, SA can also update its SIPTVMON tree topology information of new member join accordingly.

While a nonleaf MA needs to leave SIPTVMON after its local users in multicast island sequentially unsubscribe the IPTV service and it has no obligation to forward video for other MAs (i.e., child nodes of the leaving MA), the child nodes must reconnect to other MAs in SIPTVMON to continue the IPTV service.

As shown in the example of Figure 6, procedures of a nonleaf node MA_l leaving SIPTVMON without breaking IPTV service of its child nodes are illustrated, and the details are described as follows.

(*L1*) The MA_l will first send leaving requests of SIP "BYE" with SDP body of the video content identifier and its registration identifier to acknowledge not only its parent node to cease forwarding video later, but also its child nodes (i.e., MA_{c1} and MA_{c2}) to seek other new parent nodes of MA to replace their old parent MA_l for continuing IPTV video streaming.

(*L2*) The acknowledged child nodes (i.e., MA_{c1} and MA_{c2}) will send reregistration requests of SIP "REGISTER" with SDP body of the same video content identifier, their registration identifiers, and the reason of reregistration to the oracle SA to ask for the connection information of new parent nodes, respectively.

(*L3*) SA will send back response messages of SIP "OK" with SDP body of the connection information of new parents (i.e., MA_{s1} and MA_{s2}) to the leaving-acknowledged child nodes (i.e., MA_{c1} and MA_{c2}).

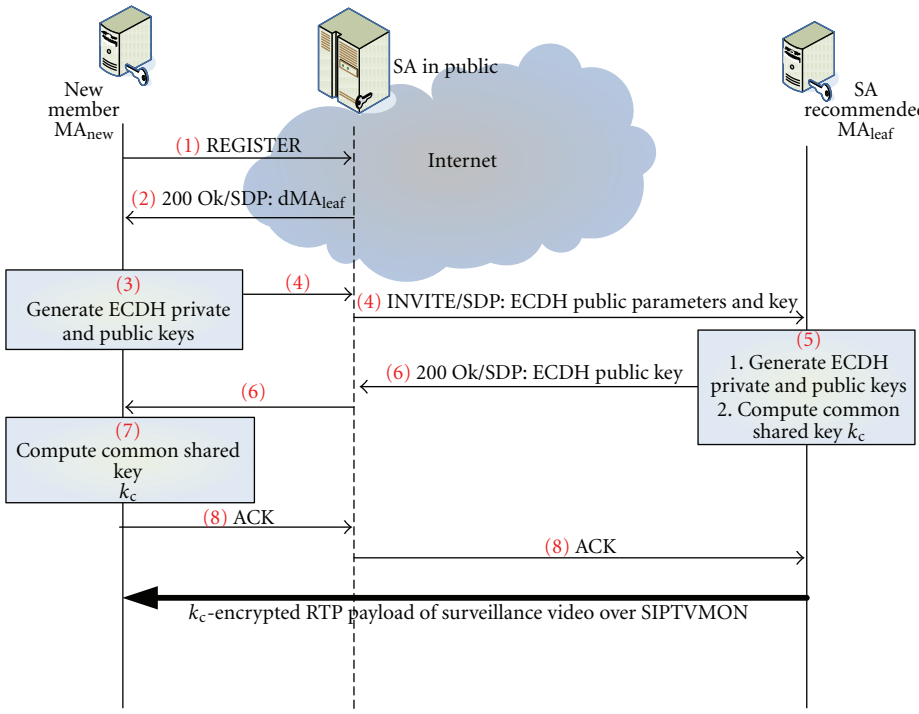


Figure 5: SIPTVMON's join procedures for a new MA (i.e., MA_{new}).

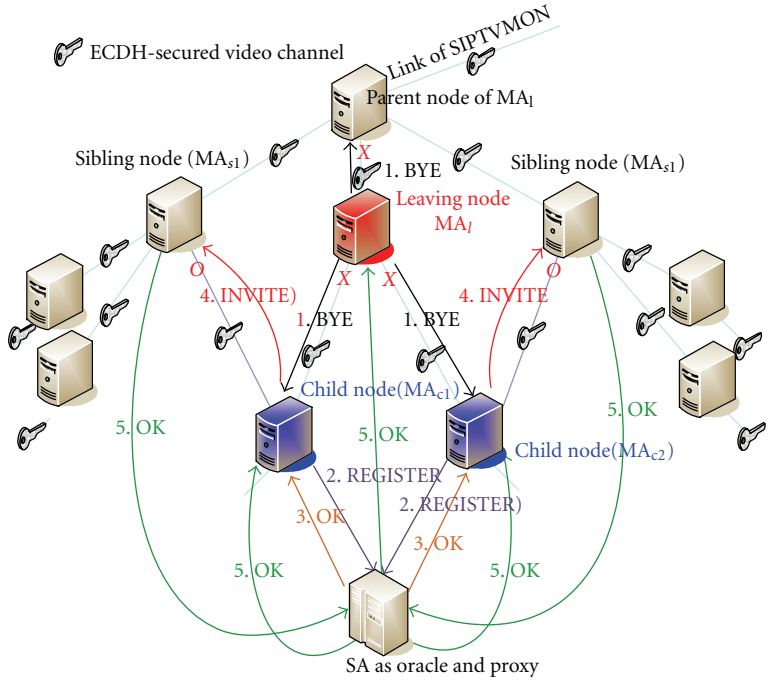


Figure 6: Message flow in leaving procedures of a nonleaf MA_l with two children.

(*L4*) The acknowledged child nodes for parent node's leaving will be able to directly send reconnect requests of SIP "INVITE" message with SDP body of the video content identifiers, their registration identifiers and the ECDH public information like previous step (*J4*) in new MA's join procedures to their new parent nodes (i.e., MA_{s1} and MA_{s2}) respectively.

(*L5*) The new IPTV connections will be immediately established from new parents to the leaving-acknowledged child nodes, right after new parents (i.e., MA_{s1} and MA_{s2}) respond the positive SIP "OK" message via the oracle SA to acknowledged child nodes and leaving MA with SDP body of the corresponding video content identifier, corresponding

registration identifier, and the ECDH public information like previous step (*J6*) in new MA's join procedures. Meanwhile, the sibling nodes MA_{s1} and MA_{s2} will start, respectively, forwarding video to children of MA_{c1} and MA_{c2}, and simultaneously the leaving MA_l stops forwarding video to children of MA_{c1} and MA_{c2}.

Such gracefully leaving procedures for a nonleaf MA_l leaving SIPTVMON can minimize the IPTV service disruption from subsequent video packets loss for the descendant nodes below the leaving node MA_l to maintain the overall quality of IPTV services for users.

Since all the SIP messages including request and response in the previous procedures in new MA's join and old MA's leaving will be forwarded via the so-called SIP proxy (i.e., oracle SA), these messages can easily help SA updating its SIPTVMON topology information to cost-effectively provide correct information upon later requests from SIPTVMON members.

3.2. Optimizations for SIPTVMON of Load Sharing and Stability. IPTV is a near real-time application service, and the delay constraint is not as strict as video conferencing and voice over IP. Therefore, IPTV service stability is more important than service latency. As the SIPTVMON's member MAs, which preserve different capabilities of system resources and network bandwidth, may join, leave, or fail in the overlay network of SIPTVMON during IPTV service session, the avoidance of IPTV service disruption must be considered in proposed SIPTVMON architecture.

The larger outbound link bandwidth of MAs in SIPTVMON not only supports higher bit rate of IPTV video, but also more connections to remote dMA with good quality of IPTV service. In previous researches [25, 26], the ALM tree's node with more outdegree (i.e., higher bandwidth) should be moved to the top of the tree to perform the optimization of load sharing in overlay network to pursue better quality of service.

Furthermore, another important factor of optimization, which will affect stability of SIPTVMON, is user lifetime, because Internet users may join and leave IPTV services in different timing. In the paper [27], authors presented that Internet user's lifetime in video-streaming systems will follow the long-tailed distribution [24]. It means that just a few users will stay in the system for a long time and most users will stay in the system for a short period of time.

To apply both factors mentioned above, which may affect stability of overlay network, [26] uses the product of bandwidth and life time (i.e., Bandwidth and life-Time Product, BTP) for load-sharing optimization of overlay network. The BTP value function, as shown in (1), is used as a criterion to optimize the overlay network service with low service latency and disruption. However, this BTP function is too sensitive to the variation of bandwidth on Internet to frequently reconstruct the overlay network. We have

$$\text{BTP} = \text{bandwidth} \times \text{lifetime}. \tag{1}$$

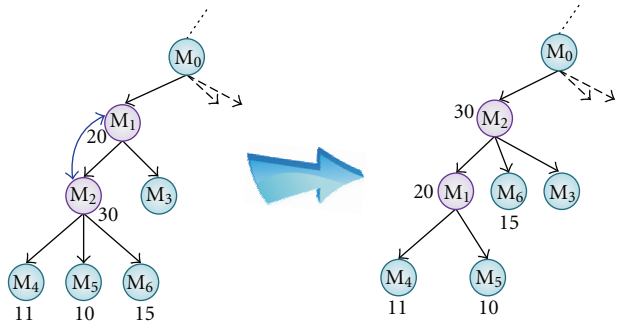


Figure 7: Example of ABTP optimization for a SIPTVMON tree.

Therefore, we further propose an improved criterion called averaging bandwidth life-time product (ABTP) to effectively minimize service disruption during optimization for SIPTVMON. The improved value function of ABTP is defined as follows:

$$\text{ABTP} = \frac{\sum_{i=1}^{n} \text{Bandwidth}_i}{n} \times \text{lifetime}. \tag{2}$$

As shown in (2), ABTP is evaluated as a criterion of load-sharing optimization for SIPTVMON by averaging the latest n measured bandwidth values of a node (i.e., MA) and then multiplying by the value of this node's life-time in SIPTVMON. According to the ABTP value of each node in SIPTVMON, we can reconstruct the SIPTVMON as the examples illustrated in Figure 7. At the left hand side of Figure 7, node M_2 preserves higher ABTP value than node M_1, and node M_2 should be moved to higher level than node M_1. But, node M_1 preserves one less degree than node M_2, and then two child nodes of M_2 with less ABTP values will link to the child nodes of M_1 for leaving them in the same level. Meanwhile, the original child node of M_2 with largest ABTP value will be moved with node M_2 to upper layer. Then, this optimization can keep the degrees of nodes M_1 and M_2 unchanged. The example of optimized SIPTVMON tree is illustrated at the right hand side of Figure 7.

Because the oracle SA also plays the role of SIP proxy, not only the running topology of SIPTVMON tree, but also both of MAs' bandwidth and life-time can be recorded by all the forwarded SIP messages in SA. Consequently, optimization score like ABTP can be calculated by SA. Then, SA can recommend the optimization procedures via SIP "UPDATE" request messages to corresponding SIPTVMON's MAs participated in the ongoing IPTV session to initiate the optimization. In SIP protocol, the request message "UPDATE" is designed to enable the modification of session information.

The SIP message flow shown in Figure 8 illustrates to the optimization of the example from Figure 7, and the procedures are described in details as follows.

(*O1*) While oracle SA detects that the optimization score of M_2 at lower level is larger than the M_1 at higher level in SIPTVMON tree, SA sends M_2 the SIP "UPDATE" with the SDP body of M_2's video content identifier, registration identifier (for authentication),

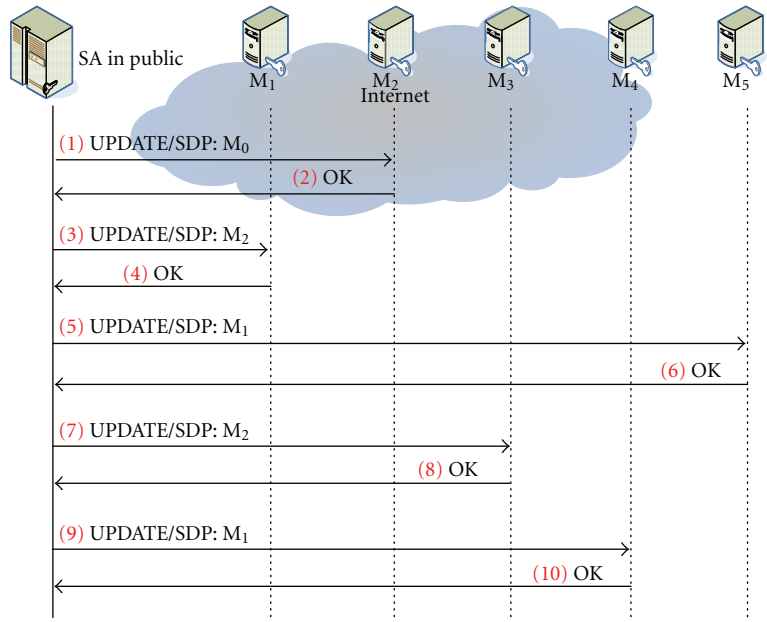


Figure 8: SIP message flow for SIPTVMON optimization in Figure 7.

and the recommended new parent M_0's connection information.

(*O2*) After M_2 receives the recommendation of new parent from SA, it will first start the similar leaving procedures mentioned above but without acknowledging its children to rejoin other parents, and then M_2 and its children start the join procedure with ECDH scheme mentioned above to connect to new parent M_0. When the join procedure is completed, M_2 sends the SIP "OK" response message to SA.

(*O3*) SA then sends M_1, which preserves low optimization score, by the SIP "UPDATE" message with the SDP body of authentication identifiers and the recommended new parent M_2's connection information.

(*O4*) After M_1 receives the recommendation of new parent from SA, it will first start the similar leaving procedures mentioned above but without acknowledging its current children to rejoin other parents, and then M_1 with its current children starts the join procedure with ECDH scheme mentioned above to connect to new parent M_2. When the join procedure is completed, M_2 sends the SIP "OK" response message to SA.

(*O5*) Because M_1's child M_3 has been moved to lower level of SIPTVMON than M_2's children M_4, M_5, and M_6, M_3 with higher optimization score must first exchange position with M_2's child M_5 with the lowest score. Therefore, SA sends M_5 the SIP "UPDATE" message with similar SDP body to ask M_5 to reconnect to new parent M_1.

(*O6*) After M_5 rejoins M_1 like previous steps, M_5 sends the SIP "OK" response message to SA.

(*O7*) Then M_1's child M_3 has to move to the same level as M_2's children. SA sends M_3 the SIP "UPDATE" message with similar SDP body to ask M_3 to reconnect to new parent M_2.

(*O8*) After M_3 rejoins M_2 like previous steps, M_3 sends the SIP "OK" response message to SA.

(*O9*) Since current out degree of M_2 is higher than before, the child M_4 with lowest score should move to the lower level of M_1 with one available degree. Therefore, SA sends M_4 the SIP "UPDATE" message with similar SDP body to ask M_4 to reconnect to new parent M_1.

(*O10*) After M_4 rejoins M_1 like previous steps, M_4 sends the SIP "OK" response message to SA to finish the procedures of optimization.

The adjustment of SIPTVMON tree for optimization of load-sharing and stability may incur the service disruption, but ABTP can smooth the variation of bandwidth on Internet to avoid unnecessary adjustment of SIPTVMON and then further effectively reduce the service disruption. ABTP optimization has two major advantages to the load-sharing and stability for proposed SIPTVMON.

(i) ABTP optimization follows the long-tailed distribution: The one who stays longer in the system can have larger ABTP value easily. Even if a node has large bandwidth, it may not have larger ABTP value because his lifetime is short. As time goes by, a node in SIPTVMON with large bandwidth can have an extremely large ABTP value, and ABTP optimization will move this node to a very high level of the SIPTVMON tree to achieve stable SIPTVMON.

(ii) ABTP value can avoid the unnecessary optimizations: Since the Internet bandwidth varies from

Table 1: Simulation parameters for SIPTVMON.

Parameter	Values
MA quantity	2000, 4000, 6000, 8000, 10000
Degree of MA	2 to 5 outdegree (uniform distribution)
Link bandwidth	Mean: 400 kbits, Std: 10, normal distribution
Link delay	Mean: 0.08 s, Std: 0.05, normal distribution
Optimization (Opt.)	Bandwidth only (B), life time only (T), BTP, Averaging bandwidth only (ABO), ABTP
Simulation time	20000 seconds
Opt. cycle	40 seconds

time to time, the averaging bandwidth value can cost-effectively reduce the unnecessary optimization overhead and possible service disruptions.

In following section, we will present the simulations and performance results of the optimization for load-sharing and stability via ABTP criterion for the proposed SIPTVMON.

4. Experiments and Performance of SIPTVMON

To demonstrate our proposed optimization scheme via ABTP criteria for SIPTVMON, we use the well-known simulation tool OMNeT++4.0 [28] to construct a vital SIPTVMON with new MAs joining, old MAs leaving and timely SIPTVMON adjustment for optimization at load-sharing and stability (i.e., criteria of bandwidth only, life time only, BTP [26], averaging bandwidth only, and ABTP). The optimization criterion of averaging bandwidth only is abbreviated as ABO, and it considers only the averaging bandwidth part of ABTP. The video source's average bit rate is 280 Kbps from the popular MPEG-4-coding data. The detailed simulation parameters and corresponding test values are listed in Table 1.

During the simulation for SIPTVMON, we recorded the count of control messages and service disruptions, tree depths, service latency, average packet loss rate, count of MA with packet loss, and average perceptual PSNR values for different optimization criteria. We will perform each test case five times to find out the mean and standard deviation of these simulation results.

As shown in Figure 9, ABTP criterion outperforms other four optimization criteria in less overhead of control messages. Those criteria (i.e., B and ABO) without considering life time preserve more control messages. It is because the possibility of high-level nodes leaving from SIPTVMON tree will be higher. It also indicated that more control messages are needed to repair the SIPTVMON for continuing IPTV service if MA's life time is not considered in optimization for load sharing.

While an old MA leaving SIPTVMON or SIPTVMON's adjustment for load sharing, service disruption may occur to degrade the quality of IPTV video-streaming service for those dMAs under detached parent in SIPTVMON. According to the results from Figure 10, ABTP also preserves less average count of service disruption to help SIPTVMON to achieve better quality of service.

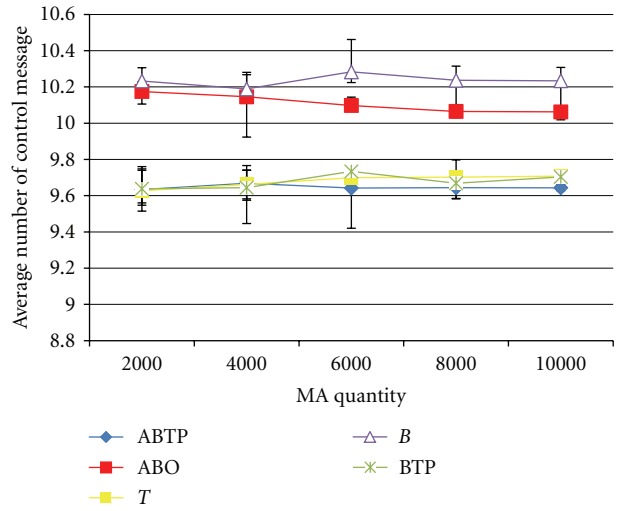


Figure 9: Average number of control messages on different MA quantity and optimization criteria.

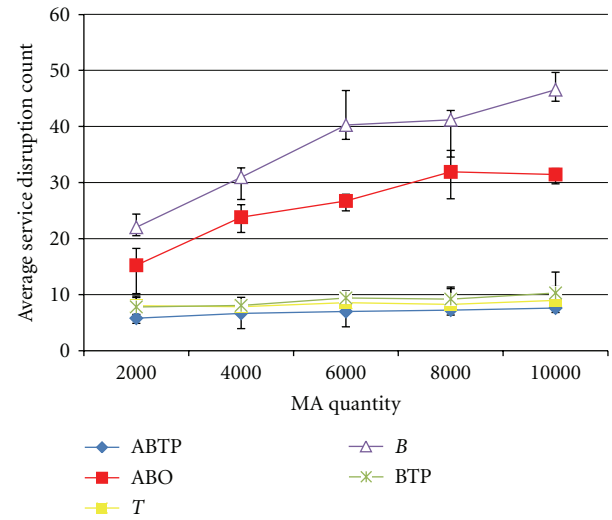


Figure 10: Average count of service disruption on different MA quantity and optimization criteria.

As shown in Figure 11, ABTP optimization criterion keeps lower depths of SIPTVMON tree than other criteria, except the test case of small MA quantity of 2000 nodes. Those optimization criteria considering both bandwidth and life time usually keep less depth than others in most test cases.

While the depth of SIPTVMON tree getting larger, the IPTV service latency of bottom dMAs is also getting longer. However, as shown in Figure 12 for the test case of MA quantity 6000, the value of service latency will keep growing after simulation starts. It is because the tree is growing and the increasing depth indicates increasing service latency. While the member of MA in SIPTVMON tree reaches 6000 and some of these MA may start to leave SIPTVMON tree, the service latency will stop increasing due to the adjustment of SIPTVMON tree through different optimization criteria. The optimization criterion T preserves much more service latency during simulation time than other criteria since the

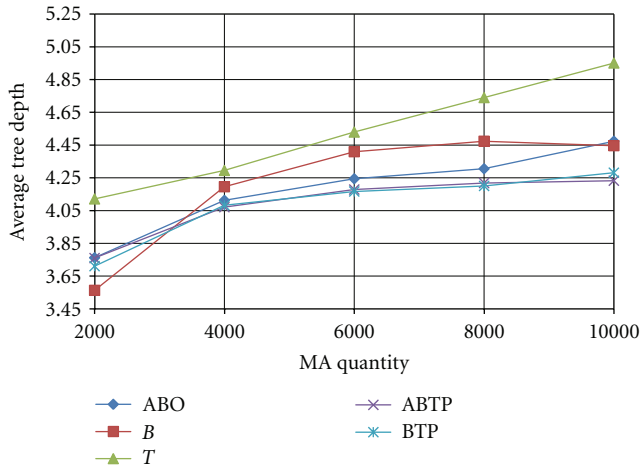


Figure 11: Average tree depth on different MA quantity and optimization criteria.

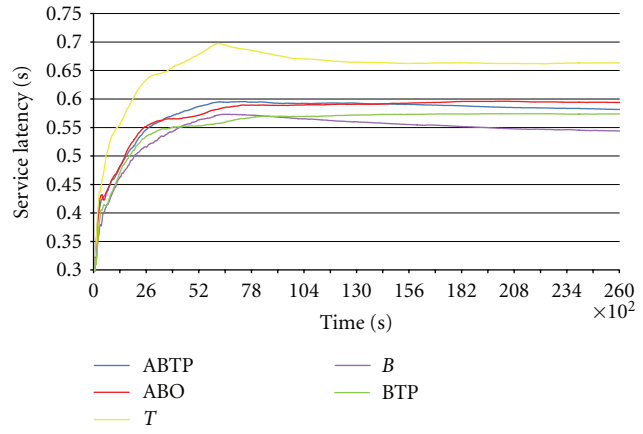


Figure 12: Service latency from different optimization criteria on MA quantity of 6000.

bandwidth criterion was not considered to effectively reduce both the depth of SIPTVMON tree and corresponding service latency.

ABO considers same criteria of bandwidth only with criterion B, but ABO preserves larger service latency than criterion B. This is because the averaging bandwidth from ABO will not decrease the tree depth of SIPTVMON in the same large scale as B.

As shown in Figure 12, the reason why optimization criterion ABTP cannot achieve best results in service latency is because it considers both of the averaging bandwidth and life time. It is not very possible for criterion ABTP to outperform less service latency than criterion B.

Due to the possible service disruption from the SIPTVMON's optimization procedures, we count the packet loss for every MA during the service disruptions within simulation to calculate the average packet loss rate for the whole SIPTVMON tree. The calculation of average packet loss rate l_{avg} is shown in (3). In (3), l_i indicates the packet loss rate of MA_i on the SIPTVMON, and it is derived from the ratio of

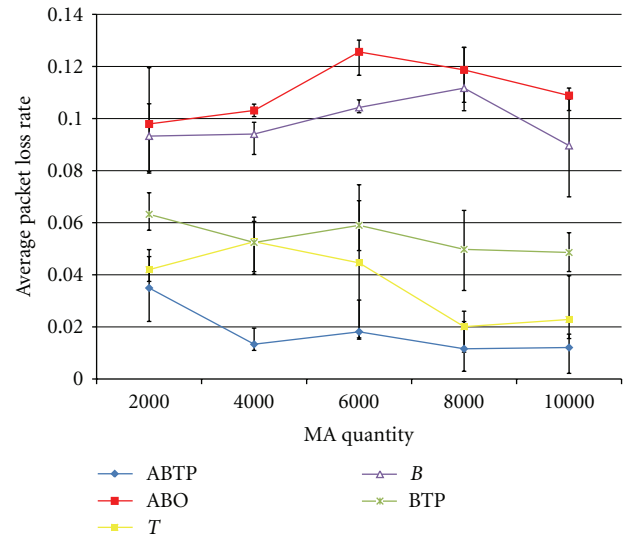


Figure 13: Average packet loss rates on different MA quantity and optimization criteria.

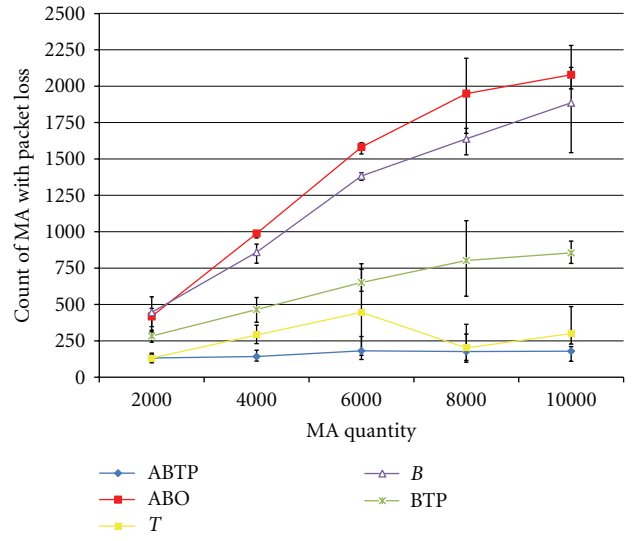


Figure 14: Count of MA with packet loss on different MA quantity and optimization criteria.

the count packet loss in MA_i to the total packets of original video source. Then, n is the MA quantity in Table 1. One has

$$l_{\text{avg}} = \frac{\sum_{i=1}^{n} l_i}{n}. \tag{3}$$

As shown in Figure 13, proposed ABTP optimization criterion for SIPTVMON outperforms other criteria in less average packet loss rates. While we count the number of SIPTVMON's MA preserving packet loss, as illustrated in Figure 14, the ABTP optimization criterion further demonstrates much more scalability for SIPTVMON than other criteria applied.

Moreover, the packet loss may jeopardize the playback quality of the IPTV video, so we derive the average perceptual PSNR values from the packet loss rates previously recorded

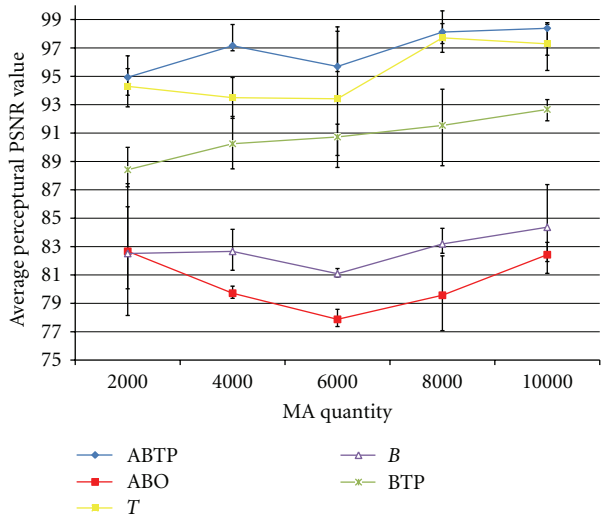


Figure 15: Average perceptual PSNR values on different MA quantity and optimization criteria.

for all MAs on SIPTVMON. The calculation of average perceptual PSNR value p_{avg} is shown in (4). As the same definition in (3), l_i indicates the packet loss rate of MA_i on the SIPTVMON and n is the MA quantity. Then, the PSNR function in (4) provides an estimated PSNR value for the input loss rate of l_i. One has

$$p_{\text{avg}} = \frac{\sum_{i=1}^{n} \text{PSNR}(l_i)}{n}. \tag{4}$$

As shown in Figure 15, similar with most of previous experimental results, the proposed ABTP optimization criterion applied in SIPTVMON outperforms much better perceptual quality for end users than other criteria applied. In our simulation results, the average perceptual PSNR value still reaches the best even at the largest scale of MA quantity to 10000 in SIPTVMON for IPTV service.

5. Conclusion and Future Work

In this paper, we propose a secure overlay network-called SIPTVMON using application-layer multicast with load-sharing and stability schemes to cost-effectively provide scalable IPTV services for users churn such as frequent joining and leaving. The proposed SIPTVMON constructed by SIP signaling can provide Internet users with scalable and stable IPTV video streaming with privacy protection, and the simulations and results demonstrate that our SIPTVMON's proposed optimization criterion (i.e., ABTP) considering the product of averaging bandwidth and life time in peers not only has the better performance in overhead of control message, service disruption, and service latency from tree depth than other optimization criteria but also preserves very acceptable perceptual quality in objective PSNR values with privacy provision.

In near future, we plan to deploy SIPTVMON over the global Internet test-bed (e.g., PlanetLab [29]) for further demonstration. We also like to investigate SIPTVMON's reliability features of open-loop and close-loop error controls such as peer's packet cache for retransmission and adaptive forward error correction (FEC) to further improve P2P IPTV's quality of service over Internet.

Acknowledgment

The work was partially supported by National Science Council, Project Numbers NSC 100-2221-E-130-009, NSC 100-2628-H-002-003-MY2, and NSC 100-2218-E-002-007, Taiwan.

References

[1] PPStream, http://www.ppstream.com.
[2] PPLive, http://www.pptv.com.
[3] SopCast, http://www.sopcast.org.
[4] TvAnts, http://tvants.en.softonic.com.
[5] X. Hei, C. Liang, J. Liang, Y. Liu, and K. W. Ross, "A measurement study of a large-scale P2P IPTV system," *IEEE Transactions on Multimedia*, vol. 9, no. 8, pp. 1672–1687, 2007.
[6] X. Hei, Y. Liu, and K. W. Ross, "IPTV over P2P streaming networks: the mesh pull approache," *IEEE Communications Magazine*, vol. 46, no. 2, pp. 86–92, 2008.
[7] D. Ciullo, M. A. Garcia, A. Horvath et al., "Network awareness of P2P live streaming applications: a measurement study," *IEEE Transactions on Multimedia*, vol. 12, no. 1, pp. 54–63, 2010.
[8] H. Schulzrinne and J. Rosenberg, "The IETF internet telephony architecture and protocols," *IEEE Network*, vol. 13, no. 3, pp. 18–23, 1999.
[9] K. Malhotra, S. Gardner, and R. Patz, "Implementation of elliptic-curve cryptography on mobile healthcare devices," in *Proceedings of the IEEE International Conference on Networking, Sensing and Control (ICNSC '07)*, pp. 239–244, London, UK, April 2007.
[10] C. H. Wang, Y. H. Chu, and T. T. Wei, "SIPTVMON: a secure multicast overlay network for load-balancing and stable IPTV service using SIP," in *Proceedings of the 30th IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS '11)*, pp. 97–102, Shanghai, China, April 2011.
[11] G. Camarillo and M. A. Garcia-Martin, *The 3G IP Multimedia Subsystem-Merging the Internet and the Cellular Worlds*, John Wiley & Sons, New York, NY, USA, 2004.
[12] F. Wang, Y. Xiong, and J. Liu, "MTreebone: a collaborative tree-mesh overlay network for multicast video streaming," *IEEE Transactions on Parallel and Distributed Systems*, vol. 21, no. 3, pp. 379–392, 2010.
[13] Y. Chu, S. G. Rao, S. Seshan, and H. Zhang, "Enabling conferencing applications on the internet using an overlay multicast architecture," in *Proceedings of the ACM Applications, Technologies, Architectures, and Protocols for Computers Communications (SIGCOMM '01)*, pp. 55–67, August 2001.
[14] C. K. Yeo, B. S. Lee, and M. H. Er, "Application layer multicast architecture for media streaming," in *Proceedings of the 7th IASTED International Conference on Internet and Multimedia Systems and Applications*, pp. 455–460, Honolulu, Hawaii, USA, August 2003.
[15] C. K. Yeo, B. S. Lee, and M. H. Er, "A survey of application level multicast techniques," *Computer Communications*, vol. 27, no. 15, pp. 1547–1568, 2004.

[16] D. Ren, Y.-T. H. Li, and S.-H. G. Chan, "Fast-mesh: a low-delay high-bandwidth mesh for peer-to-peer live streaming," *IEEE Transactions on Multimedia*, vol. 11, no. 8, pp. 1446–1456, 2009.
[17] http://www.cryptopp.com/benchmarks.html.
[18] J. Arkko, E. Carrara, F. Lindholm, M. Naslund, and K. Norrman, "Multimedia Internet KEYing (MIKEY)," IETF, RFC 3830, August 2004.
[19] M. E. Hellman, "An overview of public key cryptography," *IEEE Communications Magazine*, pp. 42–49, 2002.
[20] C. C. Yang, R. C. Wang, and W. T. Liu, "Secure authentication scheme for session initiation protocol," *Computers & Security*, vol. 24, no. 5, pp. 381–386, 2005.
[21] M. Handley and V. Jacobson, "Session Description Protocol (SDP)," IETF, RFC4566, July 2006.
[22] H. Schulzrinne et al., "RTP: A Transport Protocol for Real-Time Applications," IETF RFC 3550, July 2003.
[23] Y. Chu, S. G. Rao, S. Seshan, and H. Zhang, "A case for end system multicast," *IEEE Journal on Selected Areas in Communications*, vol. 20, no. 8, pp. 1456–1471, 2002.
[24] C. Anderson, "The long tail," in *Wired*, 2004.
[25] D. Andersen, "Resilient overlay networks," *SIGOPS*, vol. 35, pp. 131–145, 2001.
[26] G. Tan and S. A. Jarvis, "Improving the fault resilience of overlay multicast for media streaming," *IEEE Transactions on Parallel and Distributed Systems*, vol. 18, no. 6, pp. 721–734, 2007.
[27] K. Sripanidkulchai, A. Ganjam, B. Maggs, and H. Zhang, "The feasibility of supporting large-scale live streaming applications with dynamic application end-points," in *Proceedings of the ACM Conference on Computer Communications (SIGCOMM '04)*, pp. 107–120, Portland, Ore, USA, September 2004.
[28] OMNet++, http://www.omnetpp.org/.
[29] PlanetLab, an open platform for developing, deploying and accessing planetary-scale serviceshttp://www.planet-lab.org/.

Hindawi Publishing Corporation
International Journal of Digital Multimedia Broadcasting
Volume 2012, Article ID 801641, 7 pages
doi:10.1155/2012/801641

Research Article

Video Classification and Adaptive QoP/QoS Control for Multiresolution Video Applications on IPTV

Huang Shyh-Fang

Department of Information Communication, MingDao University, Changhua 52345, Taiwan

Correspondence should be addressed to Huang Shyh-Fang, hsfncu@gmail.com

Received 1 February 2012; Revised 22 March 2012; Accepted 5 April 2012

Academic Editor: Pin-Han Ho

With the development of heterogeneous networks and video coding standards, multiresolution video applications over networks become important. It is critical to ensure the service quality of the network for time-sensitive video services. Worldwide Interoperability for Microwave Access (WIMAX) is a good candidate for delivering video signals because through WIMAX the delivery quality based on the quality-of-service (QoS) setting can be guaranteed. The selection of suitable QoS parameters is, however, not trivial for service users. Instead, what a video service user really concerns with is the video quality of presentation (QoP) which includes the video resolution, the fidelity, and the frame rate. In this paper, we present a quality control mechanism in multiresolution video coding structures over WIMAX networks and also investigate the relationship between QoP and QoS in end-to-end connections. Consequently, the video presentation quality can be simply mapped to the network requirements by a mapping table, and then the end-to-end QoS is achieved. We performed experiments with multiresolution MPEG coding over WIMAX networks. In addition to the QoP parameters, the video characteristics, such as, the picture activity and the video mobility, also affect the QoS significantly.

1. Introduction

With the development of heterogeneous networks, multiresolution video coding becomes desirable in various applications. It is important to provide a flexible scalable framework for multiresolution video services, where video resolution, quality, and network quality-of-service (QoS) parameters are determined according to the requirements of user equipment and network resources [1–4]. Worldwide Interoperability for Microwave Access (WIMAX) communication is suitable for supporting video delivery because it guarantees the service quality. The network control reserves adequate resources in the network to support video delivery based on QoS parameters, which, in general, includes the peak rate, the mean rate, the mean burst length, the delay, the jitter, the cell loss rate, and so forth [5–7]. A negotiation process may be involved in QoS parameter determination for efficient network resource utilization. As long as the video application requests a suitable set of QoS parameters, the network should be able to deliver the video signals with guaranteed quality [8].

A user could specify a set of QoS parameters satisfying the requirements of video quality before executing an application. The selection of suitable QoS parameters is, however, not trivia for video service users. The QoS must be set based on specific application programming interface (API) and transport mechanism provided by vendors. An ordinary user may not have the knowledge on such network details. Instead, a user may only concern the size of the pictures, that is, the resolution, the video quality, that is, the PSNR, and the frame rate, defined as the quality-of-presentation (QoP) [9]. It is desirable to have a mechanism which shields video applications from the complexity of QoS management and control. It is also much easier to define the QoP parameters than the QoS parameters because QoP directly defines the quality of the user interface to viewers. In multiresolution video services, this approach becomes more important because of the existence of different QoS requirements [10, 11].

2. Multiresolution Video System Architecture

In 1993, the International Standard Organization (ISO) developed MPEG-2, a scalable coding method for moving pictures. The MPEG-2 test model 5 (TM-5) is used in

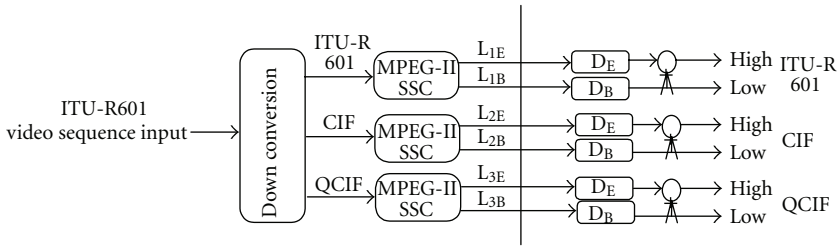


Figure 1: Layered coder.

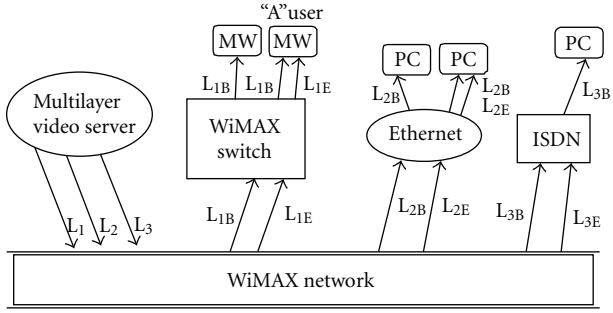


Figure 2: Scalable multiresolution video services with multilayer transmission.

the course of the research for comparison purposes [1]. The MPEG-2 scalability methods include SNR scalability, spatial scalability, and temporal scalability. Moreover, combinations of the basic scalability are also supported as hybrid scalability. In the case of basic scalability of MPEG-2 TM-5, two layers of video, referred as the lower layer and the enhancement layer, are allowed, whereas in hybrid scalability up to three layers are supported. However, owing to huge variations of video service quality with different network bandwidth and terminal equipment, the two or three layer schemes are still not adequate. A more flexible multiresolution scalable video coding structure may be needed.

The structure of a layered video coder is shown in Figure 1. The input signal is compressed into a number of discrete layers, arranged in a hierarchy that provides different quality for delivering across multiple network connections. In each input format, SNR scalability provides two quality services: basic quality service (lower quality) and enhanced quality service (higher quality). The input video is compressed to produce a set of different resolutions ranging from HDTV to QCIF and different output rates, for example, L1B and L1E. The encoding procedure of the base layer is identical to that of a nonscalable video coding. The input bit stream to the encoder of the enhancement layer is, however, the residual signal which is the quantization error in the base layer. The decoder modules, DB and DE, are capable of decoding base layer and enhancement layer bit strings, respectively. If only the base layer is received, the decoder DB produces the base quality video signals. If the decoder receives both layers, it combines the decoded signals of both layers to produce improved quality. In general, each additional enhancement layer produces an extra improvement in reconstruction quality.

By combining this layered multiresolution video coding with a QoP-/QoS-controlled WiMAX transmission system, we can easily support multicast over heterogeneous networks. For multiresolution video systems, we focus on SNR scalable schemes with various video formats, such as HDTV, ITU-R 601, CIF, and QCIF. The input video signal is compressed into a number of discrete layers which are arranged in a hierarchy that provides different quality for delivery across multiple network connections. In this QoP/QoS control mechanism, the multicast source produces video streams, each level of which is transmitted on a different network connection with a different set of QoP requirements, shown in Figure 2. For example, the user "A", who is equipped with a multimedia workstation terminal and an QoS connection, receives both base (L1B) and enhanced (L1E) layers of highest resolution, while a PC user with ISDN connection may only receive the base layer of the lowest resolution stream (L3B). With this mechanism, a user is able to receive the best quality signal that the network can deliver.

3. QoP/QoS Control Scheme

We discuss the QoP/QoS control scheme and the negotiation process in a video server-client model. The mulitresolution video server consists of a scalable encoder and a QoP/QoS mapping table. The video client consists of a scalable MPEG decoder, a QoP regenerator, and a call control unit. A video user specifies a set of QoP parameters which satisfies the requirements based on the terminal capability and network connection capacity. The QoP is sent to the server and is translated to a set of QoS parameters by the QoP/QoS mapping table. The QoS is sent back to the client. The call control on the client side performs schedulability test to check if the resources running along the server-network-client path are capable of supporting the tasks. If the schedulability test is passed, the connection is granted. Otherwise, the connection is rejected, and the QoP regenerator produces a degraded QoP set. Then the former negotiation procedure is supposed to be repeated.

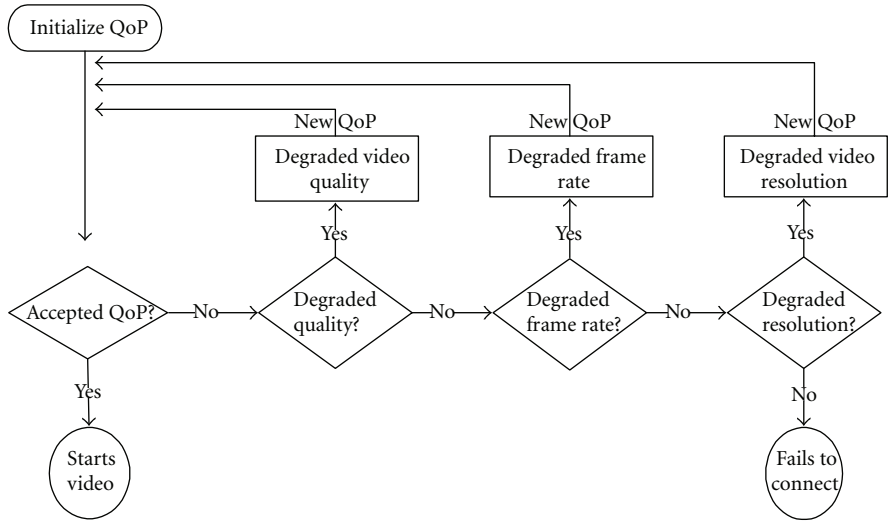


Figure 3: QoP regeneration procedure.

3.1. QoP Negotiation. If the original QoP/QoS pair is not affordable, a new QoP is generated with lower quality. The QoP regeneration procedure is shown in Figure 3. A new set of QoP should have lower requirements. However, it is not expected to have large degradation in one change. The degrading of QoP is in the order of the video quality (PSNR), the frame rate, and the resolution. The reason is that we want to make a small change of QoS at the beginning when the original QoP cannot be satisfied. The resolution parameter has the most impact to QoS because in each step of degrading it reduces the image size to 1/4 and changes the rate to roughly 1/4 of the original rate. On the other hand, the PSNR can be changed in a much finer granularity and the impact to the subjective image quality is also the least. Hence, we downgrade the QoP with the order of SNR scalability, temporal scalability, and spatial scalability. Namely, if the image quality can be degraded, we reduce the SNR requirement, because the slight degradation of quality can be accepted by most customers, and it makes the smallest QoS degradation in network. Otherwise, we degrade the frame rate. It can be archived by dropping some of the frames, such as skipping *B*-frames. Dropping some frames only causes slight degradation of the viewing quality and makes less QoS modifications in network rather than reducing the spatial resolution. If the frame rate can be reduced, we downgrade it. Otherwise, we reduce the spatial resolution. If all the QoP parameters are already set to the lowest levels and still cannot match the requirements, the service is denied. It is noteworthy that a QoS parameter may be restored to a higher level in the negotiation procedure. For example, when the spatial resolution is reduced to a lower level, the SNR requirement is restored to the highest level to avoid large change in the bitrates.

4. QoP and QoS Computations

In this section, the definitions and the computations of QoP and QoS used in this work are given and described. Many QoS parameters are generally discussed in technical articles but cannot be simply calculated. Here the QoS parameters in existing WIMAX network products are considered. Based on WIMAX API, the QoS parameters defined by Fore company are used in our experiments. Also the video characteristics affecting the QoP/QoS mapping significantly are discussed.

4.1. QoP Parameters. The QoP parameters represent the requirements of the video quality specified by video users. The QoP is relied on the subjective assessment of viewers and is generally constrained by the terminal equipment and the network capacity. We choose three parameters to represent the QoP: the spatial resolution, the temporal frame rate, and the image fidelity. The spatial resolution ranges from HDTV, ITU-R 601, CIF, to QCIF. The temporal frame rate ranges from 30, 15, to 10 frames/second or even lower. In our experiments, the image fidelity, represented by the PSNR of the reconstructed video, is divided into three grades (high, medium, and low) with 3 dB difference in each adjacent grades.

4.2. Video Characteristics. The purpose of defining the QoP parameters is to estimate the QoS parameters accurately. In addition to the QoP parameters we have defined, however, the video characteristics existing in each video sequence that affect the QoS setting significantly. We define the spatial activity and temporal mobility as two important video characteristics in the QoP/QoS mapping. The QoP is selected by video users while the video characteristics exist along with the video sequences. Both are considered in the QoS calculations.

4.2.1. Spatial Activity (A). The spatial activity represents the degree of variations in image pixel values. Since the removal of the redundancy in the temporal domain is not considered by *I*-frame encoding, we define the spatial activity measure of a video sequence as the average pixel variance of the *I*-frame.

$$A = \frac{1}{K}\sum_{i=1}^{K}\left[\frac{1}{256}\sum_{j}^{256}\left(P_{i,j} - \overline{P}_i\right)^2\right] \quad (1)$$

$P_{i,j}$: the jth pixel value in ith MicroBlock (MB),

$$\overline{P}_i = \frac{1}{256}\sum_{j=1}^{256} P_j : \text{the mean of pixel values in } i\text{th MB}, \quad (2)$$

K: the number of MBs in a frame.

4.2.2. Temporal Mobility (M). The temporal mobility reflects the degree of motion in a video sequence. It is more difficult to perform accurate motion estimation for a sequence of higher temporal mobility. Thus, the temporal mobility is defined as the percentage of the intracoded MBs in all P-frames in a sequence

$$M = \frac{1}{N_P}\sum_{i=1}^{N_P} M_P(i),$$
$$M_P(i) = \frac{K_a(i)}{K}, \quad (3)$$

where M_P is the percentage of intra-MBs in ith P-frame, K_a is the number of intra-MBs, and N_P is the total number of P-frames in a sequence.

4.3. QoS Parameters. QoS parameters that we discuss are related to the video transmission over WIMAX networks. In general, QoS parameters include a broad range of measures, such as, the peak bandwidth, the mean bandwidth, the mean burst length, the end-to-end delay and jitter, and the cell loss rate. Three parameters, the mean bandwidth, the peak bandwidth, and the mean burst length, are computed. A minimum value and a target value for each parameter are requested. The minimum value is chosen as the average value of all tested video sequences, while the target value is chosen as the maximum value of all tested video sequences.

4.3.1. Mean Bandwidth (B). This is the average bandwidth, expected over the lifetime of the connection and measured in kilobits per second. The mean bandwidth B_k of video sequence k is computed as

$$B_k = \frac{\sum_{i=1}^{n} f_{k,i}}{T_k}, \quad (4)$$

where n is the total number of frames in sequence k, $f_{k,i}$ is the total number of bits of the ith frame in sequence k, and T_k is the total playback time of sequence k.

The total playback time of sequence k is computed as

$$T_k = \sum_{i=1}^{n} t_{k,i}, \quad (5)$$

where $t_{k,i}$ is the playback time of ith frame in sequence k and is supposed to be equal to 1/29.97. The mean bandwidth is calculated as the total number of bits in a sequence divided by the total playback time. The minimum mean bandwidth $B_{\min}$ is the average value of all tested sequences,

$$B_{\min} = \frac{\sum_{i=1}^{r} B_i}{r}, \quad (6)$$

where r is the total number of video sequences. The target mean bandwidth is the maximum value

$$B_{\text{target}} = \max_{1\le k\le r}(B_k) \quad (7)$$

among all tested sequences.

4.3.2. Peak Bandwidth (P). This is the maximum or burst rate at which the transmitter produces data and which is measured in kilobits per second. In MPEG coding, the I-frames usually have the highest rate. Thus the peak bandwidth in sequence k is calculated as the maximum I-frame rate

$$P_k = \max_{1\le i\le n}\left(\frac{f_{k,i}}{t_{k,i}}\right), \quad (8)$$

in sequence k. In all tested video sequences, the minimum peak bandwidth is set to be the average

$$P_{\min} = \frac{\sum_{i=1}^{r} P_i}{r}, \quad (9)$$

and the target peak bandwidth is set to be the maximum

$$P_{\text{target}} = \max_{1\le k\le r}(P_k). \quad (10)$$

4.4. The Mapping between QoP and QoS Parameters. The QoP parameters that directly specify the video quality are friendly to video users. Each QoP set needs to be supported by a particular set of network QoS parameters. In general, higher QoP requires higher QoS. We first determine the mapping for general video services. For a given set of QoP, a corresponding set of QoS is obtained by computing the statistics of the encoded video data. A general QoP/QoS mapping table that consists of many QoP-QoS pairs is then established.

In addition to the QoP parameters, many video characteristics, such as activity and mobility, can also affect the corresponding QoS parameters significantly. In order to make the mapping more accurate, we classify the video sources based on the activity and the mobility. For each class of the video source, a classified QoP/QoS mapping table is then established by the above method. The video characteristics can easily be obtained in a pre-coding application. For real-time video applications, the initial mapping can be obtained from either the general mapping or a realtime analysis based on the first few video frames.

5. Simulation Results

We choose the spatial resolution and the image quality as the set of parameters of QoP. The frame rate is considered fixed in simulations because the current experimental hardware cannot support the full rate (30 fps) video coding. The video sequences include "Garden," "Table Tennis," "Football," "Mobil," "Hockey," "Bus," and "MIT" with ITU-R 601 format (704 × 480 pels, 4 : 2 : 0 chrominance format). CIF and QCIF formats (352 × 240 pels, and 176 × 120 pels) are converted from the ITU-R 601 format. The frame quality is represented by the PSNR with 3 dB difference between two adjacent levels.

Table 1: Activity and mobility of video sequences.

Class	Video source	Spatial activity	Temporal mobility
1	Salesman	92.4	2.9%
	Suzie	37.3	3.7%
	Miss American	14.8	0.1%
2	Football	74.8	51.5%
	Hockey	35.8	43.4%
3	Mobil	689.3	2.8%
	MIT	234.1	0.1%
	Tennis	134.0	7.3%
4	Garden	573.2	15.0%
	Bus	509.2	30.9%

Table 2: General QoP/QoS mapping table.

QoP parameters		QoS parameters					
Frame Resolu.	Quality	MMB (Kbps)	TMB (Kbps)	MPB (Kbps)	TPB (Kbps)	MMBL (Kbits)	TMBL (Kbits)
QCIF	Low	230	278	250	298	15	18
QCIF	Normal	294	348	281	339	19	22
QCIF	High	373	434	319	350	23	27
CIF	Low	461	863	987	1598	43	87
CIF	Normal	1267	1824	1702	2554	116	186
CIF	High	3786	4983	5021	6385	351	413
ITU-R 601	Low	5132	7013	7552	8977	493	613
ITU-R 601	Normal	7961	9592	10384	12096	740	836
ITU-R 601	High	11324	13866	14231	15731	986	1137

MMB: minimum mean bandwidth. TMB: target mean bandwidth. MPB: minimum peak bandwidth. TPB: target peak bandwidth. MMBL: minimum mean burst length. TMBL: target mean burst Length.

5.1. Analysis of Video Characteristics. For limited number of video sequences available for experiments, we divide the video sequences into four unique classes.

Class 1: low-spatial activity, low-temporal mobility: Salesman, Suzie, Miss American,

Class 2: low-spatial activity, high-temporal mobility: Football, Hockey,

Class 3: high-spatial activity, low-temporal mobility: MIT, Mobil, Tennis,

Class 4: high-spatial activity, high-temporal mobility: Bus, Garden.

Table 1 gives the activity and mobility of video sequence. Accordingly, the video sequences are classified into the four classes. After the classification, a set of mapping relations between video presentation quality (QoP parameters) and throughput/traffic specifications (QoS parameters) can be found. The threshold of classification for the spatial activity is set to 120, and the threshold for the mobility is 20%. These values are acquired by experiments.

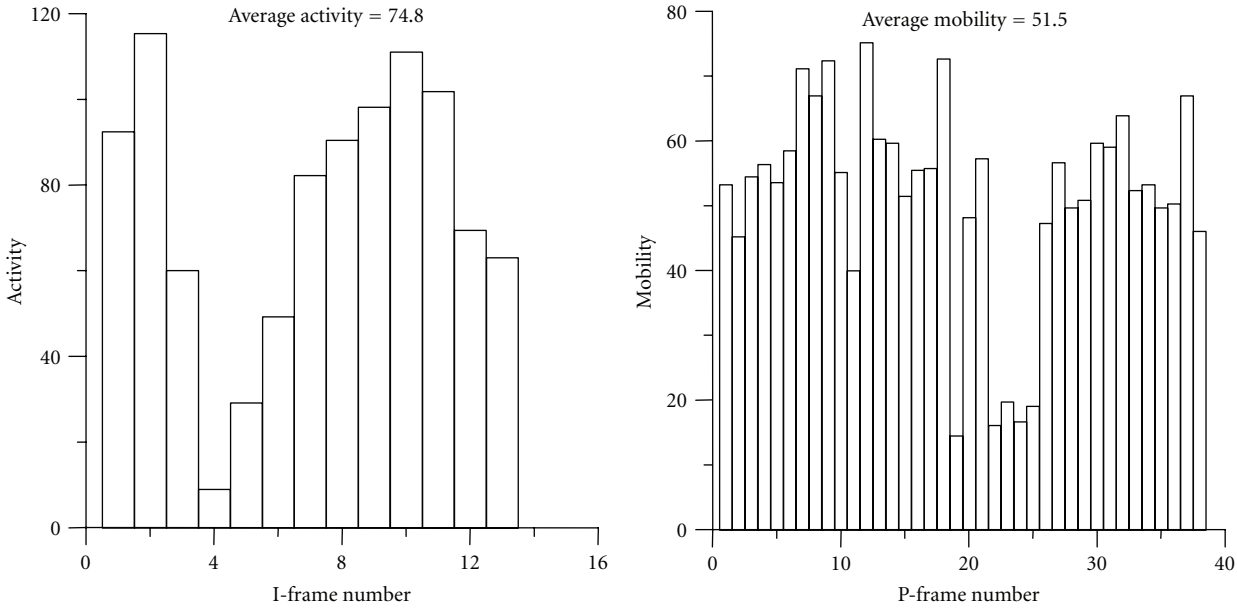
The spatial activity represents the pixel variations and also reflects the coding bit rate. Figure 4(a) shows the activity of *I*-frames in the sequence "Football". Since the peak rate of a video sequence is mainly determined by the *I*-frame bitrate, the peak bandwidth of QoS is highly correlated to the spatial activity.

Figure 4(b) shows the mobility of *P*-frames in "Football". The temporal mobility represents the percentage of the intra-coded MBs in *P*-frames and it directly reflects the coding bitrate of *P*-frames and *B*-frames, since both are motion-compensated coding. Because most frames in MPEG are *B*- or *P*-frames in general, the temporal mobility is highly related to the mean bandwidth of QoS.

5.2. QoP/QoS Mapping. We establish the QoP/QoS mapping for two cases. One is the general case in which the video characteristics are unknown. The other is the classified case in which the video characteristics are known and the QoS setting can be more precise. Table 2 shows the general QoP/QoS mapping. The low frame quality, represented by PSNR, is set to 30 dB, 30 dB, and 24 dB for QCIF, CIF, and ITU-R 601 format, respectively. Higher frame quality requires 3 dB more for each level. Pictures of smaller size are given higher PSNR because the receiver often upsamples the signals to get larger size pictures. The receiver can adjust their best trade-off between the larger picture size and less mosaics in the picture. The frame resolution is the most important

Table 3: Classified QoP/QoS mapping table on CIF.

Class	QoP parameters			QoS parameters					
	Activity	mobility	Frame Quality	MMB (Kbps)	TMB (Kbps)	MPB (Kbps)	TPB (Kbps)	MMBL (Kbits)	TMBL (Kbits)
1	Low	Low	Low	488	587	869	921	53	61
	Low	Low	Normal	547	632	985	1142	58	69
	Low	Low	High	625	829	1145	1378	65	78
2	Low	High	Low	734	902	1302	1639	72	83
	Low	High	Normal	862	1125	1834	2421	87	103
	Low	High	High	1104	1230	2268	2700	109	123
3	High	Low	Low	1207	1430	2588	2958	126	139
	High	Low	Normal	1230	1536	3205	3589	135	158
	High	Low	high	1540	1798	3786	4023	158	172
4	High	High	Low	2198	2388	3906	4366	162	182
	High	High	Normal	3528	3816	5616	6240	234	260
	High	High	High	4503	4792	6901	7658	287	319



(a) Activity of *I*-frames

(b) Mobility of *P*-frames

Figure 4: Activity and mobility of video sequence "footfall".

factor affecting the QoS requirements. At the same frame quality level, ITU-R 601 may need 20 times more bandwidth than QCIF. The frame quality also affects the QoS requirements significantly. A 3 dB improvement in PSNR may increase 50% bandwidth requirement. The target values are significantly larger than the minimum values because of the large variations of the video characteristics in all sequences. Thus, before the video characteristics are acquired, the QoS setting for guaranteed service quality may be wasteful in many cases.

Basing on different video classes, we then make QoP/QoS mapping. Table 3 shows the mapping for CIF format. High activities result in high peak bandwidth requirement. Both high activity and high mobility contribute to high mean bandwidth requirements. It is noteworthy that the differences between the target values and the minimum values are much smaller than that without classifications. Thus the video classification gives more accurate QoS setting than the case with no classifications.

6. Conclusion

We have presented a mechanism of QoP/QoS control in multiresolution MPEG scalable coding structure. The user specifies the video quality represented by a set of QoP parameters. The system maps the QoP setting to the network requirements represented by the QoS parameters by means of mapping tables based on video statistics. The classification

of video source improves the accuracy of the QoP/QoS mapping significantly.

References

[1] N. Kamaci, Y. Altunbasak, and R. M. Mersereau, "Frame bit allocation for the H.264/AVC video coder via cauchy-density-based rate and distortion models," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 15, no. 8, pp. 994–1006, 2005.

[2] "ISO/IEC/JTC1/SC29/WG11 MPEG 93/457," Test Model 5, Draft Vision 1, April 1993.

[3] S. M. Canne, M. Vetterli, and V. Jacobson, "Low-complexity video coding for receiver-driven layered multicast," *IEEE Journal on Selected Areas in Communications*, vol. 15, no. 6, pp. 983–1001, 1997.

[4] H. Doi, Y. Serizawa, H. Tode, and H. Ikeda, "Simulation study of QoS guaranteed ATM transmission for future power system communication," *IEEE Transactions on Power Delivery*, vol. 14, no. 2, pp. 342–348, 1999.

[5] A. Shehu, A. Maraj, and R. M. Mitrushi, "Analysis of QoS requirements for delivering IPTV over WiMAX technology," in *Proceedings of the 18th International Conference on Software, Telecommunications and Computer Networks (SoftCOM '10)*, pp. 380–385, September 2010.

[6] H. Y. Tung, K. F. Tsang, L. T. Lee, and K. T. Ko, "QoS for mobile WiMAX networks: call admission control and bandwidth allocation," in *Proceedings of the 5th IEEE Consumer Communications and Networking Conference (CCNC '08)*, pp. 576–580, Las Vegas, Nev, USA, January 2008.

[7] A. Sayenko, O. Alanen, and J. Karhula, "Ensuring the QoS requirements in 802.16 scheduling," in *Proceedings of the 9th ACM Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems (ACM MSWiM '06)*, pp. 108–117, New York, NY, USA, October 2006.

[8] B. Jung, J. Choi, Y. T. Han, M. G. Kim, and M. Kang, "Centralized scheduling mechanism for enhanced end-to-end delay and QoS support in integrated architecture of EPON and WiMAX," *Journal of Lightwave Technology*, vol. 28, no. 16, Article ID 5452987, pp. 2277–2288, 2010.

[9] X. Mei, Z. Fang, Y. Zhang, J. Zhang, and H. Xie, "A WiMax QoS oriented bandwidth allocation scheduling algorithm," in *Proceedings of the 2nd International Conference on Networks Security, Wireless Communications and Trusted Computing (NSWCTC '10)*, pp. 298–301, April 2010.

[10] N. Liao, Y. Shi, J. Chen, and J. Li, "Optimized multicast service management in a mobile WiMAX TV system," in *Proceedings of the 6th IEEE Consumer Communications and Networking Conference (CCNC '09)*, January 2009.

[11] J. F. Huard, I. Inoue, A. A. Lazar, and H. Yamanaka, "Meeting QOS guarantees by end-to-end QOS monitoring and adaptation," in *Proceedings of the 5th IEEE International Symposium on High Performance Distributed Computing*, pp. 348–355, Los Alamitos, Calif, USA, August 1996.

Hindawi Publishing Corporation
International Journal of Digital Multimedia Broadcasting
Volume 2012, Article ID 192314, 8 pages
doi:10.1155/2012/192314

Research Article

Adjustable Two-Tier Cache for IPTV Based on Segmented Streaming

Kai-Chun Liang and Hsiang-Fu Yu

Department of Computer Science, National Taipei University of Education, Taipei 106, Taiwan

Correspondence should be addressed to Hsiang-Fu Yu, yu@tea.ntue.edu.tw

Received 31 January 2012; Accepted 10 March 2012

Academic Editor: Pin-Han Ho

Internet protocol TV (IPTV) is a promising Internet killer application, which integrates video, voice, and data onto a single IP network, and offers viewers an innovative set of choices and control over their TV content. To provide high-quality IPTV services, an effective strategy is based on caching. This work proposes a segment-based two-tier caching approach, which divides each video into multiple segments to be cached. This approach also partitions the cache space into two layers, where the first layer mainly caches to-be-played segments and the second layer saves possibly played segments. As the segment access becomes frequent, the proposed approach enlarges the first layer and reduces the second layer, and vice versa. Because requested segments may not be accessed frequently, this work further designs an admission control mechanism to determine whether an incoming segment should be cached or not. The cache architecture takes forward/stop playback into account and may replace the unused segments under the interrupted playback. Finally, we conduct comprehensive simulation experiments to evaluate the performance of the proposed approach. The results show that our approach can yield higher hit ratio than previous work under various environmental parameters.

1. Introduction

Internet protocol TV (IPTV) is a promising Internet killer application, which integrates video, voice, and data onto a single IP network, and offers viewers an innovative set of choices and control over their TV content. Many major telecommunication companies, such as AT&T, Verizon, and Bell, have announced their IPTV solutions by replacing the copper lines in their networks with fiber optic cables to create sufficient bandwidths for delivering many TV contents. The Bell Entertainment Service in Bell Canada, for example, uses a single VDSL line with a consistent download speed of 20 Mbps and an upload rate of 8 Mbps to provide a converged Internet and television service. The trend is similar in other areas, such as Europe and Asia. Major cities in Japan, for example, already provide high-speed networks which allow customers to obtain video over IP. In Taiwan, the largest telecommunication company, Chunghwa Telecom, offers the multimedia on-demand (MOD) services, which allow clients to watch traditional TV contents over IPTV infrastructures.

A conventional solution to provide IPTV services is through a content distribution network (CDN), in which the service provider installs multiple video servers at different locations to transmit video contents to local customers. In general, multimedia streaming objects are far bigger than web objects. Additionally, real-time transmission is necessary for continuous video playback. A video server thus yields much larger disk load and bandwidth consumption than a web server. Once viewer arrival rate increases significantly, the video server is easily overloaded and reduces service quality. To alleviate the limits, most CDNs are based on cache servers. Figure 1 depicts a popular architecture, which is composed of a video server, cache servers, and clients. Usually, the video server is in WAN, and the cache servers are deployed near clients. An incoming video request is first forwarded to the cache server, instead of the video server. Once receiving the request, the cache server checks whether

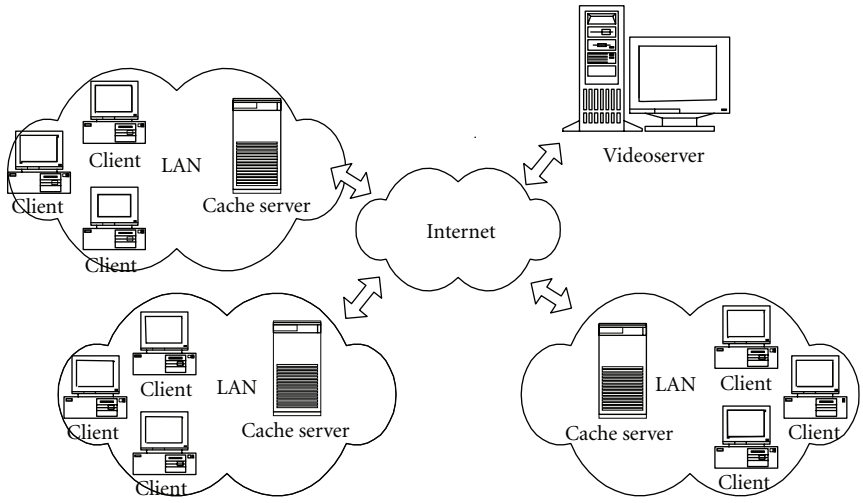


Figure 1: Basic streaming cache architecture.

the video data are available in the cache storage. If so, the cache server transmits the data to the client directly. Otherwise, the cache server connects to the video server for the video data, which are then forwarded to the client.

The studies in [1–3] investigated how to allow time-shifted IPTV services, which could enable the end user to watch a broadcasted TV program with a time shift. Wauters et al. [1, 2] proposed a network-based time-shifted television (tsTV) solution using cooperative proxy caches. The study in [3] proposed a hybrid strategy combining genetic algorithms to determine the optimal cache locations for supporting time-shifted IPTV services. Simsarian and Duelk [4] analyzed the bandwidth requirements in metropolitan area networks (MANs) for providing IPTV services and developed a model of the IPTV network to determine the optimum location of the cached video content. The study in [5] introduced the concept of content cacheability and proposed a cache-partition algorithm using the cacheability to serve the maximum amount of video requests subject to constraints on cache memory and throughput. Sofman and Krogfoss [6] indicated that a portion of the video content could be stored in caches closer to clients to reduce the IPTV traffic and further presented an analytical model of hierarchical cache optimization depending on traffic, topology, and cost parameters. A heuristic model [7] was proposed for hierarchical cache optimization in an IPTV network. Chen et al. [8] presented an IPTV system based on a peer-to-peer hierarchical cache architecture. The work [9] devised a caching algorithm that tracked the popularity of objects to make intelligent caching decisions in IPTV services.

A web cache server generally considers a web page an atomic object, and thus caches a complete page. However, caching a complete streaming object is not suitable to a streaming cache server. If a streaming cache server always caches a complete video object, the number of cached videos will be very small because a streaming object is much larger than a web page. When incoming requests increase, cached videos are easily swapped out because the cached objects are not enough, leading to poor cache performance. In addition, caching an entire video also results in long playback latency because the data transmission time is too large to be ignored. Suppose that a client can download a 100-minute MPEG2 video encoded by 6 Mbps at a bandwidth of 10 Mbps. The video size is $100 * 60 * 6$ bits, and the data transmission time equals 3600 seconds. Clearly, playback after downloading is unrealistic. To alleviate these problems, many studies [10–19] partition a streaming object into multiple smaller segments, which are cached partially. A prefix caching [10] stores the initial frames of popular videos. Upon receiving a video request, the cache server transmits the initial frames to the client and simultaneously requests the remaining frames from the video server. Wu et al. [11, 12] investigate how to partition videos to achieve higher hit ratio. Three video-segmentation approaches—fixed, pyramid, and skyscraper—are proposed. Their simulation results indicate that the pyramid segmentation is the best segmentation approach. Compared with whole video caching, segmentation-based caching is more effective in increased byte-hit ratio. Lazy segmentation approach [13] delays the video partition until a video is accessed. The study in [14] introduces the proxy jitter, which results in playback jitter at the client side due to proxy delay in fetching the uncached segments. The proposed hyperproxy [14] can generate minimum proxy jitter with a low delayed startup ratio and a small decrease of byte-hit ratio. SProxy [15] implements a segment-based streaming cache system on Squid [16]. The study in [17] devises a segment-based cache mechanism to support VCR functions on the client side. extending popularity-aware partial caching algorithm (PAPA) [18], dynamic segment-based caching algorithm (DECA) [19] determines the segment size according to segment popularity.

This paper proposes a two-layer segment-based cache for streaming objects, as shown in Figure 2. The cache server divides the cache storage into two layers—L1 and L2—in

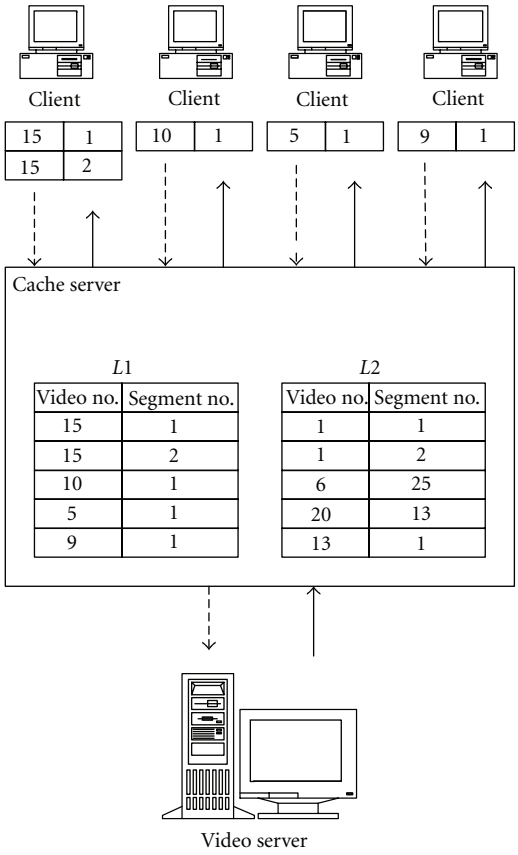


Figure 2: The proposed streaming cache.

which cache L1 mainly caches to-be-played segments and cache L2 saves possibly played segments. As the segment access becomes frequent, the proposed approach enlarges cache L1 and reduces cache L2, and vice versa. Once the space of cache L1 is not enough, cache L1 uses LRU to choose a victim, which is then moved to cache L2. If cache L2 also has not enough space, cache L2 first swaps out a selected segment according to LRU-K [20] and saves the segment coming from cache L1. Because requested segments may not be accessed frequently, this work further designs an admission control mechanism to determine whether an incoming segment should be cached or not. The cache architecture takes forward/stop playback into account and may replace the unused segments under the interrupted playback. Table 1 briefly compares the proposed caching architecture with previous approaches. This work conducts a comprehensive simulation to evaluate the proposed cache under various cache size, video popularity, request arrival rate, and playback interrupt rate. In comparison with the segment-based LRU, the video-based LRU, and Wu's approach [12], our approach mostly yields higher hit ratio.

The rest of this paper is organized as follows. Section 2 presents the proposed cache. The simulation results and performance comparisons are shown in Section 3. Brief conclusions are drawn in Section 4.

$\text{Ratio}_{\text{viewing}} = C_{\text{viewing}}/C_{\text{L1+L2}}$
If($\text{Ratio}_{\text{viewing}} \geq \text{L1\_Cache\_Ratio}_{\max}$){
  $C_{\text{L1\_new\_size}} = C_{\text{L1+L2}} \cdot \text{L1\_Cache\_Ratio}_{\max}$
  $C_{\text{L2\_new\_size}} = C_{\text{L1+L2}} - C_{\text{L1\_current\_size}}$}
Else{
  $C_{\text{L1\_new\_size}} = \text{Ratio}_{\text{viewing}} \cdot C_{\text{L1+L2}}$
  $C_{\text{L2\_new\_size}} = C_{\text{L1+L2}} - C_{\text{L1\_new\_size}}$}

Algorithm 1: The algorithm to determine the sizes of caches L1 and L2.

2. The Proposed Cache

The work devises a segment-based two-layer caching approach to increase byte-hit ratio. Suppose that the bandwidth between a video server and a cache server is unlimited, and the bandwidth between the cache server and a client is larger than playback rate. Each video is partitioned into multiple fixed-length segments, as indicated in Figure 2. The proposed approach periodically adjusts the sizes of caches L1 and L2 according to the size of segments accessed, as shown in Figure 3. With the decreasing of the number of the segments currently played, the proposed approach reduces the size of cache L1 and enlarges the size of cache L2, and vice versa. In order to avoid frequent size adjustment, the adjustment is executed periodically, rather than when a segment request arrives. Table 2 lists the parameters to determine the sizes of caches L1 and L2. Algorithm 1 shows how to determine the sizes of caches L1 and L2. The size of cache L1 has an upper bound to avoid the size of cache L2 being zero when many video requests arrive. Upon determining the cache sizes, the cache server adjusts the cache space by moving recently or seldom used segments, as indicated in Algorithm 2.

Besides LRU, the replacement of cache L1 is also based on the observation that a video is played continuously. If a video is played currently, its segments not played yet are very possibly accessed later. Accordingly, cache L1 avoids swapping out these segments. It is well known that the popularity of video data varies with time. The condition that video segments in cache L1 are no longer played may reflect that the segments become less popular. We thus move the segments to cache L2 when cache L1 is full. Figure 4 depicts the complete operation of the proposed approach once cache hits. When a requested segment hits the cache, the segment can be either in cache L1 or in cache L2. If the segment hits cache L1, cache L1 reorders the segment according to LRU and transmits the segment to the requested client. Otherwise, the segment hits cache L2 and is moved to cache L1. If cache L1 has enough space, cache L2 directly moves the segment to cache L1; else, cache L1 swaps cache L2 a played segment for the hit segment.

When a segment is neither in cache L1 nor in cache L2, the segment is missed. Figure 5 shows how to process a missed segment. If cache L1 has enough space, the segment is saved in cache L1 according to LRU. Otherwise, if cache L2 has free space, cache L1 swaps cache L2 a played segment

```
If(Ratio_viewing > C_L1_current_size/C_L1+L2){ // if cache L1 is not enough
   While(C_L1_current_size < C_L1_new_size){
      Select recently-used segments in cache L2 to move to the bottom of cache L1.
      Update current cache sizes of caches L1 and L2. }}
Elsif(Ratio_viewing < C_L1_current_size/C_L1+L2){
   While (C_L1_current_size > C_L1_new_size){
      Select seldom-used segments in cache L1 to move to the top of cache L2.
      Update current cache sizes of caches L1 and L2.}}
```

Algorithm 2: The algorithm to adjust the sizes of caches L1 and L2.

Table 1: Comparison among related streaming caches.

Approach	Wu's approach	Hyperproxy	SProxy	PAPA	DECA	Segment-based two-layer caching
Segment partition	Pyramid	Lazy	Fixed	Segment prefix	Dynamic segment prefix	Fixed
Cache replacement	Cost function + LRU	Cost function	NA	Cost function	Cost function	LRU + LRU-K
Number of cache layers	2	1	1	1	2	2
Admission control	YES	YES	NA	NA	NA	YES
Precache	NA	YES	YES	NA	NA	NA

Table 2: Terms used by the algorithm to determine the cache size.

Term	Definition
C_{L1+L2}	Entire cache size
$C_{L1\_current\_size}$	Current size of cache L1
$C_{L1\_new\_size}$	New size of cache L1
$C_{L2\_current\_size}$	Current size of cache L2
$C_{L2\_new\_size}$	New size of cache L2
$C_{viewing}$	Size of playing segments
$L1\_Cache\_Ratio_{max}$	Maximum ratio of size of cache L1 to entire cache size
$Ratio_{viewing}$	Ratio of playing segments to entire cache size

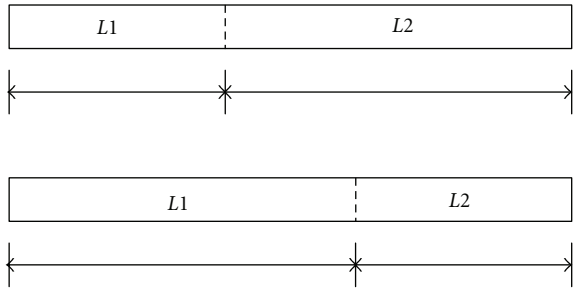


Figure 3: Size adjustment on caches L1 and L2.

for the missed segment. If cache L2 is also full, cache L2 performs an admission control to determine whether to cache the segment. If not, the missed segment is not cached and simply transmitted to the client. If so, cache L2 drops a victim segment according to LRU-K. Cache L1 then moves a segment chosen by LRU to cache L2 and saves the missed segment.

The admission control is based on LRU-K. When both of the caches L1 and L2 are full, the admission control compares the previous Kth access time of a missed segment with that of the victim segment chosen by cache L2. If the access time of the missed segment is later than that of the victim segment cache L2 drops the victim segment and saves the missed segment.

3. Performance Analysis and Comparison

We implemented an event-driven simulator by Perl to evaluate the performance of the proposed cache approach. The simulation settings are listed in Table 3. Suppose that the number of videos equals 2000. Assume that the video size is uniformly distributed between 10 segments and 110 segments, where the length of each segment equals 1 minute. The cache size is expressed in terms of ratio of total videos, and the default value is 0.2. The interarrival time is assumed to follow a Poisson distribution. For each request, it is generated by a Poisson process, which is exponentially distributed with a mean of $1/\lambda$, where λ is the request arrival rate. The default value is 6 requests per minute. The requested videos are drawing from a total of M distinct videos. The popularity of each video follows a Zipf-like distribution Zipf(x, M) [21]. A Zipf-like distribution contains two parameters, x and M, the former corresponding to the degree of skew. The distribution of each video i equals $p_i = c/i^{1-x}$, where $i \in \{1, \ldots, M\}$ and $c = 1/\sum_{j=1}^{M}(1/j^{1-x})$. Setting $x = 0$ corresponds to a pure Zipf distribution, which is highly skewed. On the other hand, setting $x = 1$ corresponds to a uniform distribution without skew. The default value for x is 0.2 and that for M is 2,000. The popularity of each video changes

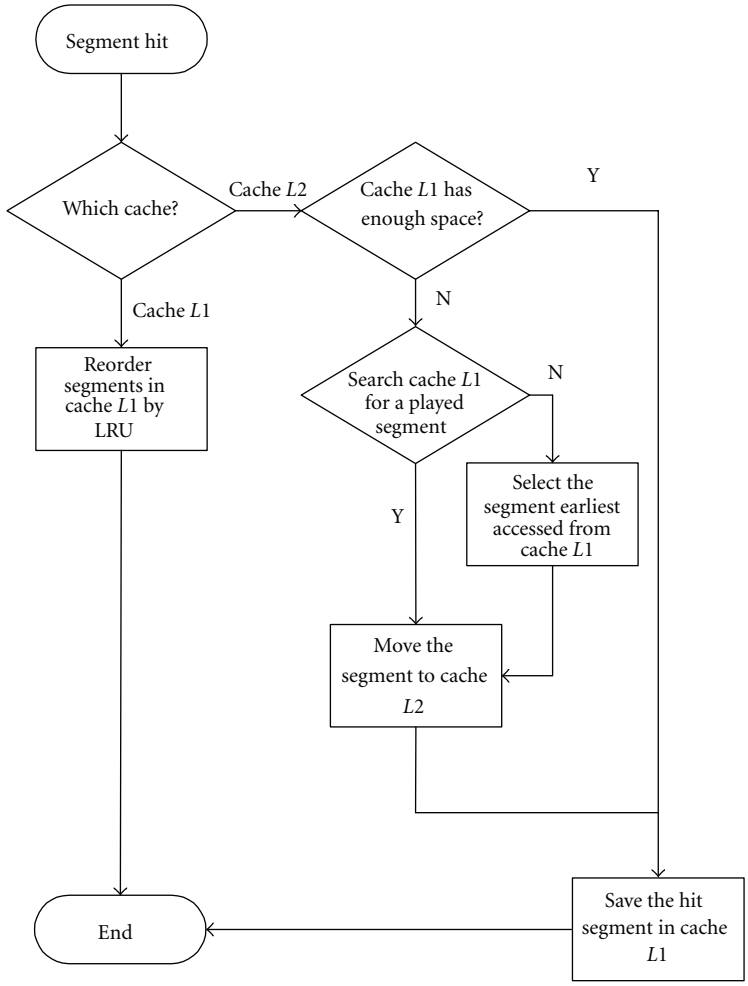


Figure 4: The algorithm to process the segment hit.

Table 3: Simulation parameters and default values.

Number of videos	2000
Video length	10–110 minutes
Request arrival rate	6 requests per minute
Simulation duration	43200 minutes
Cache size	0.2 (in the percentage of the entire video size)
Video popularity	Zipf-like distribution for video titles, Zipf(0.2, 2000)
Probability of forward/stop playback	0.2
Popularity shift distance	10 videos every 21600 minutes

over time to simulate the scenario that there may be different user groups accessing the videos at different time and their interest may be different. Similar to Wu's study [11], the popularity distribution changes every 21600 minutes, and the shift distance equals 10 videos. The default probability of forward/stop playback is 0.2.

This work compares the proposed approach with the video-based LRU, the segment-based LRU, and Wu's approach [12]. The video-based LRU caches a complete video and selects a replaced video according to LRU. For the segment-based LRU, a video is partitioned into multiple equal-size segments. Wu's approach divides a video into unequal segments under pyramid segmentation. The simulator is installed on FreeBSD 8.0 running on HP ProLiant DL380G6 and HP ProLiant DL320G6.

Figure 6 shows the impact of the cache size on the byte-hit ratio. For a wide range of the cache size, the proposed approach has higher byte-hit ratio than other approaches. The advantage in byte-hit ratio of our approach is more significant for a smaller cache size. For instance, the hit ratio of our approach is 11% higher than that of Wu's approach at the cache size of 0.1, while 26% better than those of the video-based LRU and the segment-based LRU. With the growth of the cache size, all the schemes can cache most videos, and thus their performance is similar.

We next examine the impact of the skew in video popularity on the byte-hit ratio, as indicated in Figure 7. The proposed approach has the higher byte-hit ratio under skewed video popularity. When the video popularity becomes normal distribution, our scheme performs less effectively. For example, the hit ratio of the proposed scheme is 7% better

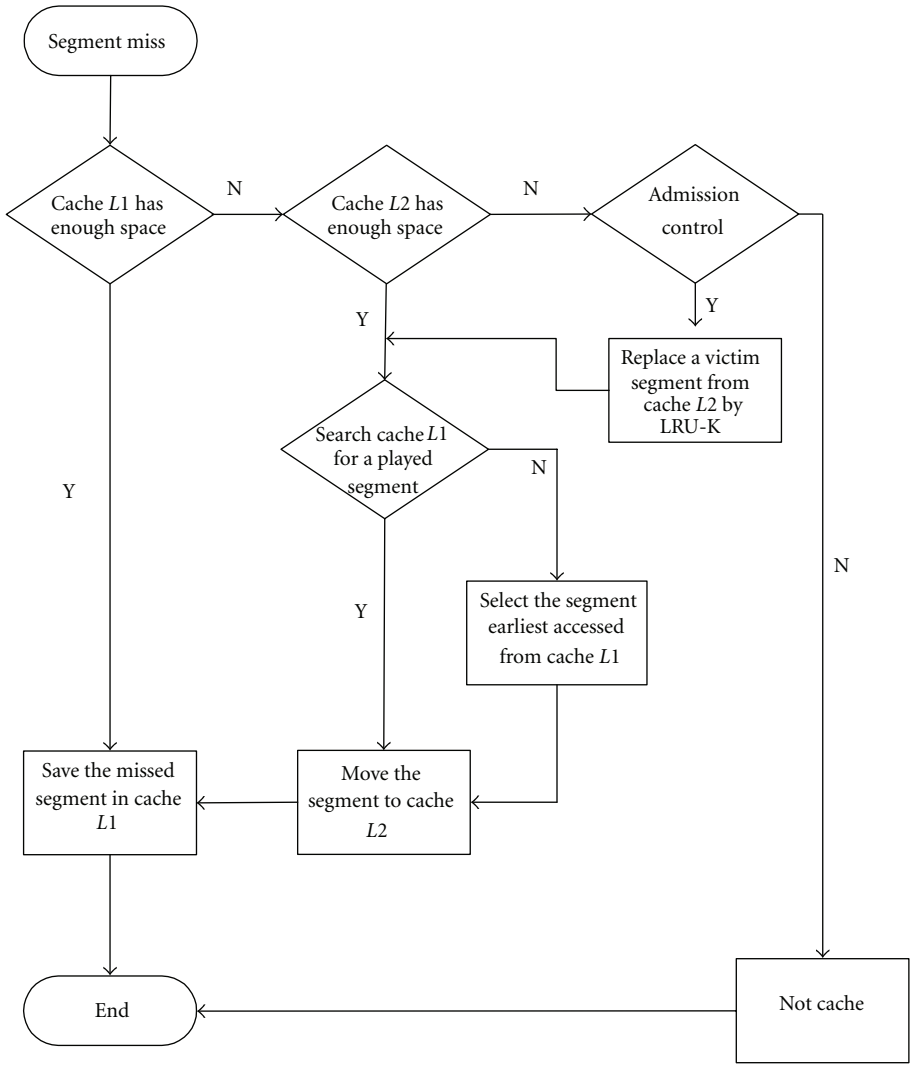


Figure 5: The algorithm to process the segment missed.

than that of Wu's approach at the skew factor of 0.2. The scheme also outperforms the video-based LRU and the segment-based LRU. However, when the skew factor is larger than 0.6, Wu's approach performs better than the proposed.

Figure 8 shows the impact of the request arrival rate on the byte-hit ratio. For a wide range of the arrival rate, the proposed approach outperforms other schemes. In comparison with Wu's approach, our approach yields 3–11% higher-hit ratio. The hit ratio of the approach is also 11–13% larger than those of the video-based LRU and the segment-based LRU. The results reflect that the proposed cache performs steadily under various request arrival rates.

Figure 9 depicts the impact of the rate of the forward/stop playback on the byte-hit ratio. The rate indicates the probability that a user performs forward/stop playback during watching a video. The rate of 0.1 represents that one of ten videos happens forward/stop playback. The figure shows that the proposed cache yields 8-9% larger hit ratio than Wu's

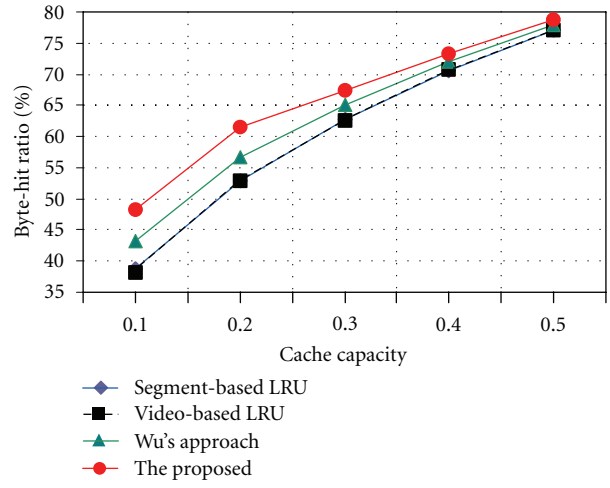


Figure 6: Influence of cache size on byte-hit ratio.

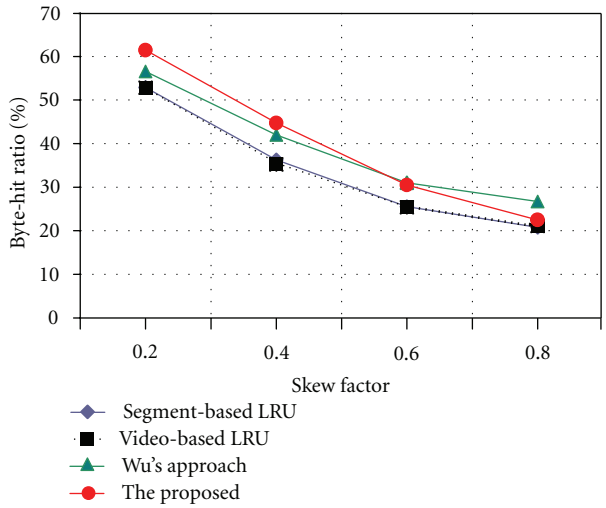


Figure 7: Impact of skew in video popularity on byte-hit ratio.

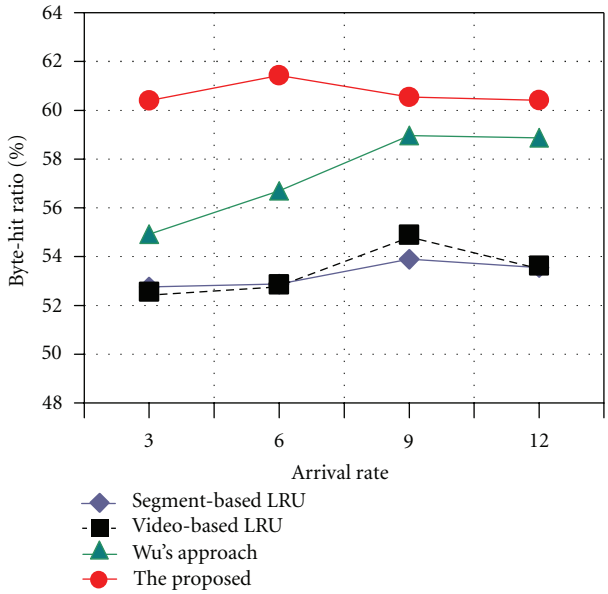


Figure 8: Influence of request arrival rate on byte-hit ratio.

approach. In comparison with the video-based LRU and the segment-based LRU, the proposed approach also achieves 13–17% better.

4. Conclusions

Internet protocol TV (IPTV) is a promising Internet killer application, which integrates video, voice, and data onto a single IP network, and offers viewers an innovative set of choices and control over their TV content. To provide high-quality IPTV services, this work proposes a two-layer segment-based cache, which divides the cache storage into caches L1 and L2, and dynamically adjusts their sizes according to video popularity. As the segment access becomes frequent, the proposed approach enlarges cache L1 and reduces cache L2, and vice versa. Once the space of cache

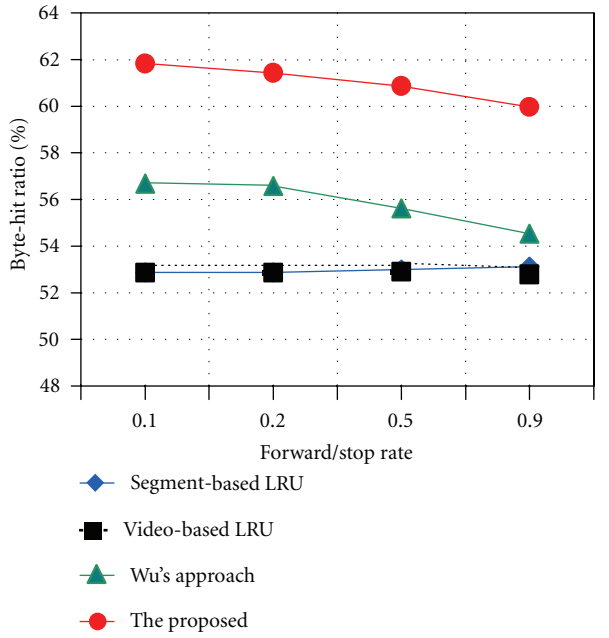


Figure 9: Impact of forward/stop playback on byte-hit ratio.

L1 is not enough, cache L1 uses LRU to choose a victim, which is then moved to cache L2. If cache L2 also has not enough space, cache L2 first swaps out a selected segment according to LRU-K and saves the segment coming from cache L1. To enlarge the hit ratio, this study also presents an admission control to determine which accessed segments should be cached. The proposed cache further considers the situations that clients may suddenly perform forward/stop playback. This work conducts a comprehensive simulation to evaluate the proposed cache under different cache size, video popularity, request arrival rate, and playback interrupt rate. The simulation results indicate that our approach mostly outperforms the segment-based LRU, the video-based LRU, and Wu's approach under various settings.

Acknowledgment

The work was financially supported by National Science Council, Taiwan under a research Grant no. NSC 100-2221-E-152-005.

References

[1] T. Wauters, W. Van De Meerssche, F. De Turck et al., "Co-operative proxy caching algorithms for time-shifted IPTV services," in *Proceedings of the 32nd Euromicro Conference on Software Engineering and Advanced Applications (SEAA '06)*, pp. 379–386, September 2006.

[2] T. Wauters, W. Van De Meerssche, F. De Turck et al., "Management of time-shifted IPTV services through transparent proxy deployment," in *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM '06)*, pp. 1–5, December 2006.

[3] Z. Juchao, L. Jun, and W. Gang, "Study of cache placement for time-shifted TV cluster using genetic algorithm," in

Proceedings of the Genetic and Evolutionary Computation Conference (GEC '09), pp. 781–785, June 2009.

[4] J. E. Simsarian and M. Duelk, "IPTV bandwidth demands in Metropolitan Area Networks," in *Proceedings of the 15th IEEE Workshop on Local and Metropolitan Area Networks (LANMAN '07)*, pp. 31–36, June 2007.

[5] L. B. Sofman, B. Krogfoss, and A. Agrawal, "Optimal cache partitioning in IPTV network," in *Proceedings of the 11th Communications and Networking Simulation Symposium (CNS '08)*, pp. 79–84, April 2008.

[6] L. B. Sofman and B. Krogfoss, "Analytical model for hierarchical cache optimization in IPTV network," *IEEE Transactions on Broadcasting*, vol. 55, no. 1, pp. 62–70, 2009.

[7] B. Krogfoss, L. B. Sofman, and A. Agrawal, "Hierarchical cache optimization in IPTV networks," in *Proceedings of the IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB '09)*, pp. 1–10, May 2009.

[8] L. Chen, M. Meo, and A. Scicchitano, "Caching video contents in IPTV systems with hierarchical architecture," in *Proceedings of the IEEE International Conference on Communications (ICC '09)*, pp. 1–6, June 2009.

[9] D. De Vleeschauwer and K. Laevens, "Performance of caching algorithms for IPTV on-demand services," *IEEE Transactions on Broadcasting*, vol. 55, no. 2, pp. 491–501, 2009.

[10] S. Sen, J. Rexford, and D. Towsley, "Proxy prefix caching for multimedia streams," in *Proceedings of the 18th Annual Joint Conference of the IEEE Computer and Communications Societie (INFOCOM '99)*, pp. 1310–1319, New York, NY, USA, March 1999.

[11] K. L. Wu, P. S. Yu, and J. L. Wolf, "Segment-based proxy caching of multimedia streams," in *Proceedings of the International Conference on World Wide Web (WWW '01)*, Hongkong, May 2001.

[12] K. L. Wu, P. S. Yu, and J. L. Wolf, "Segmentation of multimedia streams for proxy caching," *IEEE Transactions on Multimedia*, vol. 6, no. 5, pp. 770–780, 2004.

[13] S. Chen, H. Wang, X. Zhang, B. Shen, and S. Wee, "Segment-based proxy caching for Internet streaming media delivery," *IEEE Multimedia*, vol. 12, no. 3, pp. 59–67, 2005.

[14] S. Chen, B. Shen, S. Wee, and X. Zhang, "Segment-based streaming media proxy: modeling and optimization," *IEEE Transactions on Multimedia*, vol. 8, no. 2, pp. 243–256, 2006.

[15] S. Chen, B. Shen, S. Wee, and X. Zhang, "SProxy: a caching infrastructure to support internet streaming," *IEEE Transactions on Multimedia*, vol. 9, no. 5, pp. 1062–1072, 2007.

[16] http://www.squid-cache.org/.

[17] J. Z. Wang and P. S. Yu, "Fragmental proxy caching for streaming multimedia objects," *IEEE Transactions on Multimedia*, vol. 9, no. 1, pp. 147–156, 2007.

[18] L. Shen, W. Tu, and E. Steinbach, "A flexible starting point based partial caching algorithm for video on demand," in *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME '07)*, pp. 76–79, Beijing, China, July 2007.

[19] W. Tu, E. Steinbach, M. Muhammad, and X. Li, "Proxy caching for video-on-demand using flexible starting point selection," *IEEE Transactions on Multimedia*, vol. 11, no. 4, pp. 716–729, 2009.

[20] E. J. O'Neil, P. E. O'Neil, and G. Weikum, "LRU-K page replacement algorithm for database disk buffering," in *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pp. 297–306, May 1993.

[21] G. K. Zipf, *Human Behavior and the Principles of Least Effort*, Addison-Wesley, Cambridge, Mass, USA, 1949.

Hindawi Publishing Corporation
International Journal of Digital Multimedia Broadcasting
Volume 2012, Article ID 373459, 8 pages
doi:10.1155/2012/373459

Research Article

A Seamless Broadcasting Scheme with Live Video Support

Zeng-Yuan Yang,[1] Yi-Ming Chen,[2] and Li-Ming Tseng[1]

[1] *Department of Computer Science and Information Engineering, National Central University, Jhongli 32001, Taiwan*
[2] *Department of Information Management, National Central University, Jhongli 32001, Taiwan*

Correspondence should be addressed to Zeng-Yuan Yang, yzy@dslab.csie.ncu.edu.tw

Received 30 January 2012; Accepted 9 February 2012

Academic Editor: Hsiang-Fu Yu

Broadcasting schemes, such as the fast broadcasting and harmonic broadcasting schemes, significantly reduce the bandwidth requirement of video-on-demand services. In the real world, some history events are very hot. For example, every year in March, thousands of people connect to Internet to watch the live show of Oscar Night. Such actions easily cause the networks contested. However, the schemes mentioned previously cannot alleviate the problem because they do not support live broadcasting. In this paper, we analyze the requirements for transferring live videos. Based on the requirements, a time skewing approach is proposed to enable the broadcasting schemes to support live broadcasting. However, the improved schemes require extra bandwidth for live broadcasting once the length of live shows exceeds the default. Accordingly, we proposed a scalable binomial broadcasting scheme to transfer live videos using constant bandwidth by increasing clients' waiting time. When the scheme finds that the length of a video exceeds the default, it doubles the length of to-be-played segments and then its required bandwidth is constant.

1. Introduction

With the growth of broadband networks, the video-on-demand (VOD) [1] becomes realistic. Many studies start investigating VOD. One of important areas is to explore how to distribute the top ten or twenty so-called hot videos more efficiently. Broadcasting is one of the promising solutions. It transfers each video according to a fixed schedule and consumes constant bandwidth regardless of the presence or absence of requests for the video. That is, the system's bandwidth requirement is independent of the number of users watching a given video. A basic broadcasting scheme is the batch scheme [2], which postpones the users' requests for a certain amount of time and serves these requests in batch so that its bandwidth consumption is reduced. However, the batch scheme still requires quite large bandwidth for a hot video. For example, given a video of 120 minutes, if the maximum clients' waiting time equals 10 minutes, the required bandwidth is $12b$, where b is the video playout rate.

Many broadcasting schemes were proposed to further reduce the bandwidth requirement by using a set-top box (STB) at the client end. The schemes include the fast broadcasting (FB) [3, 4], pagoda broadcasting (PB) [5], new pagoda broadcasting (NPB) [5], recursive-frequency splitting (RFS) [6], staircase broadcasting (SB) [7], and harmonic broadcasting (HB) [8, 9] schemes, which divide a video into multiple segments and distribute them through several independent data channels. As well, these schemes require the STB to receive the segments from the channels when users start watching the video. The schemes substantially reduce the bandwidth requirements for hot videos. For example, if a video server allocates 4 video channels to transmit a 120-minute video by FB, then its maximum waiting time is merely 8 minutes. The FB scheme, in comparison with the batch scheme, reduces the bandwidth requirements and waiting time remarkably.

In the real world, some history events are very hot. For example, every year in March, many people connect to Internet to watch the live show of Oscar Night. Such actions easily cause the networks congested. However, the schemes, such as PB, NPB, SB, RFS, and HB, cannot broadcast live programs to alleviate the congestion. In order to overcome this obstacle, we analyzed the requirements for live broadcasting and proposed an approach, called time skewing, to enabling these schemes to distribute live shows. However, the improved schemes require extra bandwidth for

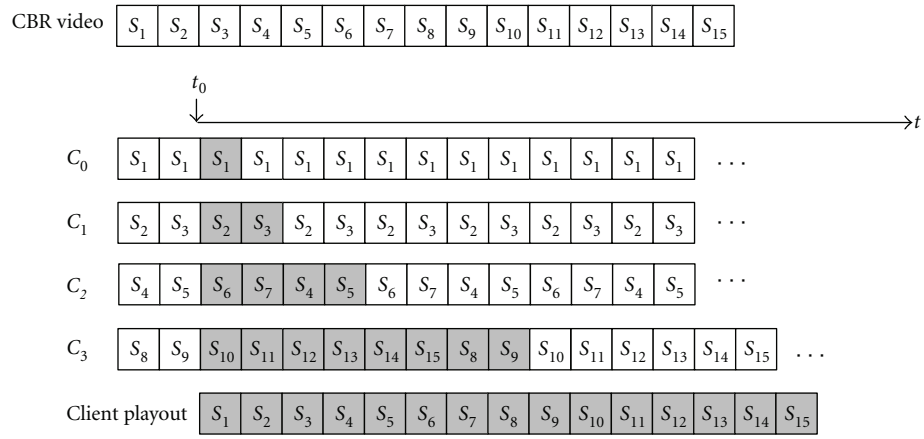


Figure 1: The FB scheme ($N = 15$ and $k = 3$).

live broadcasting once the length of live shows exceeds the default. Accordingly, a scalable binomial broadcasting scheme was proposed to demonstrate how to transfer live videos at the constant bandwidth regardless of video length.

The remainder of the paper is as follows. Section 2 analyzes the requirements of live broadcasting. This section also introduces the time skewing and the scalable binomial broadcasting scheme. An analysis and comparison is presented in Section 3. Finally, we make a brief conclusion in Section 4.

This work briefly introduces the FB and HB first. Suppose that there is a video with length L (e.g., 120 minutes). The consumption rate of the video is b (e.g., 10 Mbps). The size of the video is $S = L * b$ (e.g., 9 Gbytes). Suppose that the desired viewer's waiting time is less than $l = L/N$, where N is a positive integer. Both schemes involve the following steps.

(1) The video is equally divided into N segments. Suppose that S_i is the ith segment of the video. The concatenation ($\bullet$) of all the segments, in the order of increasing segment numbers, constitutes the whole video, $S = S_1 \bullet S_2 \bullet \cdots \bullet S_N$.

(2) On the server side, the FB and HB schemes involve the following steps, respectively.

2. The Live Broadcasting Schemes

2.1. Background.

(a) For FB, there exists an integer k such that $\sum_{i=0}^{k-1} 2^i < N \leq \sum_{i=0}^{k} 2^i$. The server then periodically transfers segments S_{2^i} to $S_{2^{i+1}-1}$ on channel C_i, where $0 \leq i \leq k$, as shown in Figure 1. Hence, the total bandwidth allocated for the video is $B = (k+1) * b$.

(b) For HB, the server further equally divides segment S_i into i subsegments $\{S_{1,i}, S_{2,i}, \ldots, S_{i,i}\}$. The subsegments of segment S_i are then broadcast on channel C_i, where $1 \leq i \leq N$, as indicated in Figure 2. The bandwidth of C_i thus equals b/i. The total bandwidth for the video equals $B = \sum_{i=1}^{N} b/i = H_N * b$, where $H_N = \sum_{n=1}^{N} 1/n$ is the harmonic number of N.

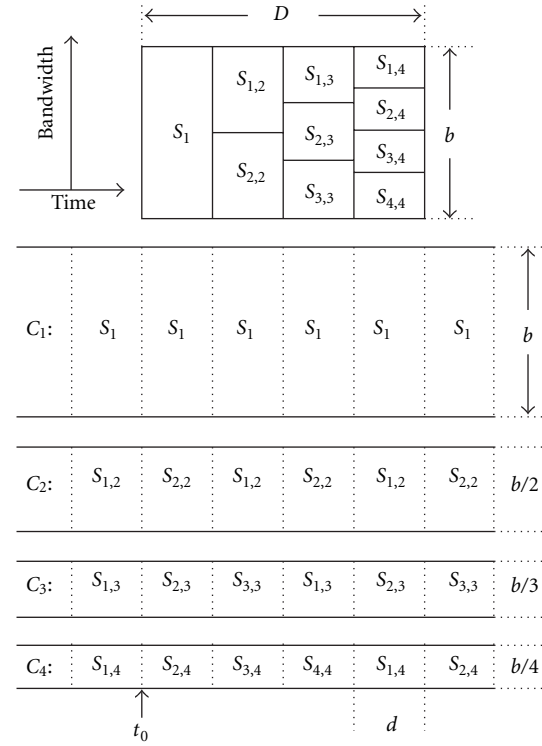


Figure 2: The HB scheme ($N = 4$).

(3) At the client end, suppose that there are enough buffer spaces to store data segments of a video. The steps to watch a video include the following.

(a) The client downloads the first data segment at the first occurrence on the first channel and then downloads other related data segments from other channels concurrently.

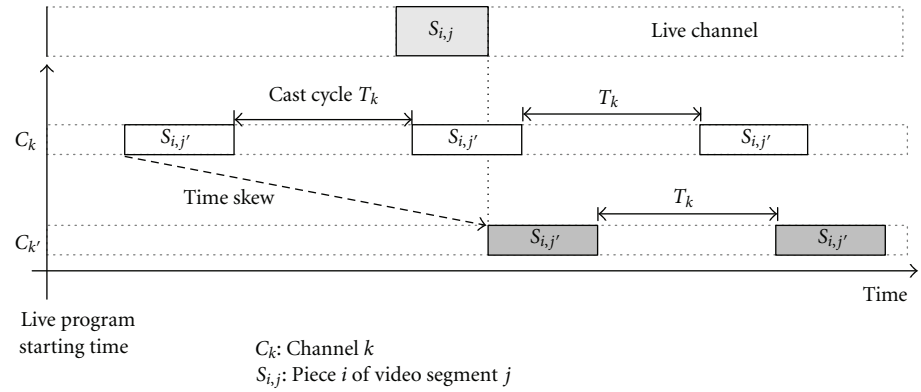


Figure 3: An illustration of time skewing.

(b) Once finishing the downloading of the first segment, the client starts to play the video with its normal speed in the order of $S_1 \bullet S_2 \bullet \cdots \bullet S_N$.

(c) The client stops loading from channels when all data segments are received.

2.2. The Requirements of Live Broadcasting. The section describes three important requirements for live broadcasting.

(R1) *The data transfer rate of a channel must be less or equal to media production rate*. In the case of live broadcasting, the new media are produced at constant speed such that the broadcasting schemes transferring data on a channel at a higher rate than media production rate are unable to support live broadcasting.

(R2) *The broadcasting schemes cannot transmit the rearward and unavailable video segments in advance.*

(R3) *A live broadcasting scheme has to tolerate the varying length of live videos.* In the real world, live shows often end either early or late, rarely on time. Most current broadcasting schemes [3–9] assume that the length of a video is known and fixed. In the case of early ending, such schemes simply free the allocated channels or repeat the last or blank video segments. Hence, the viewer watching the video is not affected. However, in the case of late ending, the schemes require additional bandwidth to handle the situation of program elongation. If the bandwidth is not available, the live broadcasting will be interrupted.

Being subject to the aforementioned requirements, we examine whether several proposed broadcasting schemes support live-program broadcasting. First, the pyramid broadcasting [10] scheme violates the requirement R1 because it uses multifold bandwidth channels to distribute video segments. Second, HB and SB conflict with the requirement R2 because they attempt to broadcast the nonexistent rearward video segments. Third, all of the mentioned broadcasting schemes cannot satisfy the requirement R3. They need to allocate additional channels to transfer the over length of a live video. Hence, they cannot distribute live videos using constant bandwidth. Accordingly, we proposed the time skewing to enable the schemes to satisfy the requirement R2. Furthermore, a scalable binomial broadcasting, based on FB, was proposed to demonstrate how to fulfill the requirement R3.

2.3. Time Skewing. Suppose that a broadcasting scheme schedules a subsegment $S_{i,j}$ of segment S_j onto channel C_k in a constant cycle time T_k. That is, subsegment $S_{i,j}$ will appear once on channel C_k every time T_k. This work also assumes that in the beginning of broadcasting the video, segment S_j is not available yet. Hence, the video server must put off the transmission of segment S_j until the segment is available. Once receiving segment S_j from a video source, such as a video camera, the video server transfers the deferred subsegment $S_{i,j}$ via channel C_k as it does for prerecorded videos. We call this postponement as time skewing. Figure 3 shows the differences between a regular broadcasting scheme on C_k and a broadcasting scheme with the time skewing.

The time skewing is a general approach. If a broadcasting scheme cannot be rescheduled to fit time skewing, then it cannot support live broadcasting. In the paper, we merely demonstrate how HB takes the advantage of the approach to support live broadcasting. This is because HB was proved to require the minimum bandwidth under the same average waiting time in [9].

Live Harmonic Broadcasting. Suppose that the default length of a live video is L and the maximum user's waiting time is $L/5$. According to HB, the video is equally divided into five segments, and the jth segment is further divided into j subsegments. With the time skewing, we put off the transmission of the posterior video segments until they are available. Figure 4 displays the scenario.

The channel C_L broadcasts the live video sequentially. Meanwhile, the other channels, that is, C_1 to C_5, distribute the recorded segments of live program with time skewing. Whenever the entire live video is recorded, it is distributed over the HB scheme.

2.4. The Scalable Binomial Broadcasting. The time skewing successfully allows the previous broadcasting schemes [3–9] to distribute live videos; however, it does not address

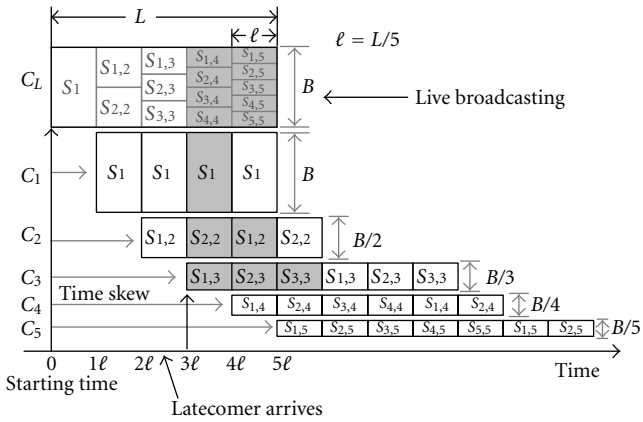


Figure 4: Live harmonic broadcasting scheme.

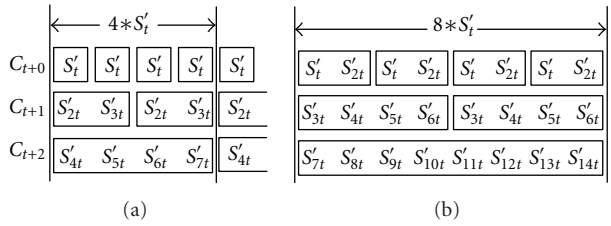


Figure 5: The binomial relations on the fast broadcasting scheme.

the over-length problem. For example, in Figure 4, if the live program's length is longer than L, we must allocate an additional channel C_6 to transmit the excess part of the video. If the bandwidth is unavailable, the system has to stop video distribution. To overcome this obstacle, we develop a scalable binomial broadcasting that broadcasts a live video using constant bandwidth, regardless of its length.

The idea comes from the binomial relationship of FB [3]. The FB scheme reveals the binomial relationship between two conjunctive channels (Figure 5(a)). For channel C_{t+1}, the length of basic cycle unit ($S'_{2t} + S'_{3t}$) is twice larger than that (S'_t) of channel C_{t+0}. The binomial relationship is independent of the length of each video segment. Therefore, the server can broadcast a video of double length on the same number of channels by doubling the length of basic cycle unit (Figure 5(b)). Namely, by increasing the length of the cyclic unit on demand, the over-length part of a live video can be broadcast via preallocated channels.

The work then presents how to seamlessly lively broadcast an over-length video at a constant bandwidth.

(1) Schedule the allocated channels by regular FB.

(2) When the last timeslot of a basic cycle unit on the highest numbered channel (S'_{7t} on C_{t+2} in Figure 6(a)) is allocated and no addition channel is available for the live video, the elongation process of the basic cycle unit starts. The process doubles the length of basic cycle unit of each channel ($2 * S'_t$ on channel C_{t+0}, $4 * S'_t$ on channel C_{t+1}, and $8 * S'_t$ on channel C_{t+2}). As well, the starting segment $S'_{t+i} = S'_w$ of basic cycle unit on each channel C_{t+i} is derived based on the following formula:

$$\begin{aligned} &\text{if } i = 0, \text{ then } w = 1; \\ &\text{else } w = 1 + \sum_{n=1}^{i} 2^k \times 2^{n-1}, \end{aligned} \tag{1}$$

where k is the times which the elongation process is applied to. (As shown in Figure 6(b), the starting segments of channel C_t, C_{t+1}, and C_{t+2} are S'_{1t}, S'_{3t}, and S'_{7t}, resp.).

(3) If the starting segment S'_w has been broadcasted via this channel, the VOD server scans backward one by one till to the first occurrence of S'_w and mark it as a new run of broadcasting cycle.

(4) If a new data cycle has been broadcasted completely and the live video is not over yet, then the video server jumps back to Step 2 and begins a new elongation process (Figure 6(c)). Otherwise, the video server cyclically broadcasts the double length of basic cycle unit as usual.

By the scalable binomial broadcasting, a video server can distribute live videos using a constant number of channels. The cost of the scheme is that the latecomers' waiting time and buffer requirements grow in the power of two.

3. Analysis and Comparison

This section presents the requirements of network bandwidth, buffer size, and I/O load on client sides with respect to each live broadcasting scheme. The assumptions include the following. Let L be the video length. (1) The media playback rate is b. (2) The length of a segment is l. (3) The number of video segments is N. The media size of the live video is blN bytes. The over-length portion is M. (4) A playing time slot denotes n.

3.1. Bandwidth Requirements. When the length of the live video is longer than L, the broadcasting scheme must allocate extra channels to distribute the over-length portion, as mentioned in Section 2.3. The bandwidth requirements of each broadcasting scheme are listed as follows.

(i) *Live Harmonic Broadcasting* (LHB). Its bandwidth requirements include a channel for live broadcast C_L and multiple channels C_n for clients (Figure 4). Since the bandwidth of each channel is a harmonic series [8], the bandwidth requirement (β_h) is

$$\beta_h = \left(1 + \sum_{x=1}^{n-1} \frac{1}{x}\right) \times B, \quad \text{where } 1 \leqq n \leqq N + M. \tag{2}$$

Equation (2) represents the bandwidth requirement when the live video is not over:

$$\beta_h = \left(\sum_{x=1}^{N+M} \frac{1}{x}\right) \times B, \quad \text{where } N + M < n. \tag{3}$$

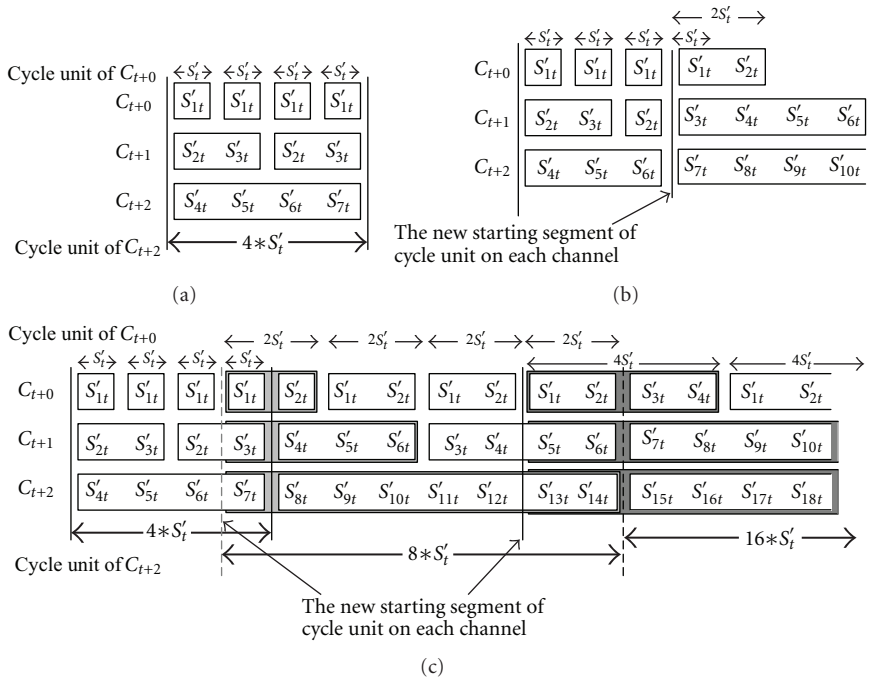


FIGURE 6: The scalable binomial broadcasting scheme.

The equation represents the bandwidth requirement when the live video is over, and the channel C_L is released.

(ii) *Live Staircase Broadcasting* (LSB). In comparison with the regular staircase broadcasting [7], this scheme requires an additional channel C_L for live broadcasting. Hence, the bandwidth requirement (β_s) is

$$\beta_s = \left(1 + c + m\left(\frac{1}{2^c}\right)\right) \times B, \tag{4}$$

where $c = \lfloor \log_2 n \rfloor$, $m = n - 2^c$, and $1 \leqq n \leqq N + M$:

$$\beta_s = \left(c + m\left(\frac{1}{2^c}\right)\right) \times B, \tag{5}$$

where $c = \lfloor \log_2 (N + M + 1) \rfloor$, $m = N + M - 2^c$, and $N + M < n$.

(iii) *Live Fast Broadcasting* (LFB. Its bandwidth requirement β_f can be obtained from [3]; it is

$$\beta_f = \left(1 + \left\lceil \log_2 n \right\rceil\right) \times B, \quad \text{where } 1 \leqq n \leqq N + M, \tag{6}$$

$$\beta_f = \left(1 + \left\lfloor \log_2 (N + M) \right\rfloor\right) \times B, \quad \text{where } N + M < n. \tag{7}$$

(iv) *Scalable Binomial Broadcasting*. Since the scalable binomial broadcasting scheme transfers a live video at a constant number of channels, its bandwidth requirement β_b is

$$\beta_b = \left(1 + \left\lfloor \log_2 N \right\rfloor\right) \times B. \tag{8}$$

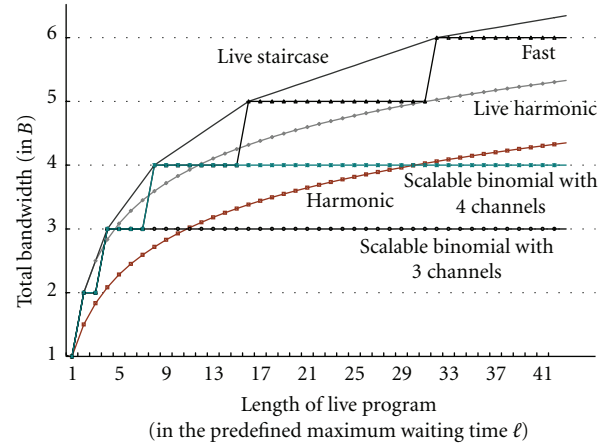


FIGURE 7: The required bandwidth versus the length of live videos.

Figure 7 depicts the bandwidth requirements of each broadcasting scheme. In [9], we proved that the optimal broadcasting scheme is harmonic scheme. However, in the study, we find that when the length of the live video is less than 16l, LFB becomes the optimal live broadcasting scheme, as shown in Figure 7. This is because LHB requires an additional channel, the live channel CL, to broadcast the live program. Furthermore, when the length exceeds 16l, LHB can be proved to be the optimal scheme by following the same deduction process revealed in [9]. Finally, Figure 7 displays that the scalable binomial broadcasting works with constant number of channels at the cost of doubling the maximum waiting time.

3.2. The Minimum Disk Transfer Rate Requirement. In the broadcasting schemes, the video segments are written to clients' disk as they need to be buffered. When clients need to consume the segments, they need to read them from disk. The disk transfer rate (Φ) is the sum of the read transfer rate (r_{read}) and write transfer rate (r_{write}). In order to ensure smooth playback, the minimum read transfer rate is the playback rate B. As well, the minimum write transfer rate must be large enough to save the need-to-be-buffered segments in time. It depends on which time slot a client receives video segments at. The following discusses the required disk transfer rate of each broadcasting schemes.

(i) *LHB*. From (2) and (3), we can find that the bandwidth requirement is maximum when $n = N + M$. The bandwidth requirement β at the time slot is $\beta = (1 + \sum_{x=1}^{N+M-1} 1/x) \times B$, and the server distributes maximum data at the time slot. In order to save video segments completely, the minimum write transfer rate at clients ($r_{\text{min\_write}}$) must equal the bandwidth requirement. Accordingly, the minimum disk transfer rate (Φmin) is the sum of $r_{\text{min\_read}}$ and $r_{\text{min\_write}}$. That is,

$$\Phi\text{min} = r_{\text{min\_read}} + r_{\text{min\_write}} = \left(2 + \sum_{x=1}^{N+M-1} \frac{1}{x}\right) \times B. \quad (9)$$

(ii) *LSB*. Like the LHB, when $n = N + M$, the bandwidth requirement is maximum. Thus,

$$\Phi\text{min} = \left(2 + c + m\left(\frac{1}{2^c}\right)\right) \times B, \quad (10)$$

where $c = \lfloor \log_2(N + M) \rfloor$, and $m = N + M - 2^c$.

(iii) *LFB*. From (6), we can derive the maximum bandwidth requirement when $n = N + M$. Thus,

$$\Phi\text{min} = \left(2 + \left\lceil \log_2(N + M) \right\rceil\right) \times B. \quad (11)$$

(iv) *Scalable Binomial Broadcasting*. Owing to the constant bandwidth requirement, the disk transfer rate is also constant and equal to

$$\Phi\text{min} = \left(2 + \left\lfloor \log_2 N \right\rfloor\right) \times B. \quad (12)$$

Figure 8 shows the maximum I/O load Φ of each broadcasting scheme.

3.3. Maximum Client Buffer Requirements. Based on the previous discussion, we find that the client's video incoming rate ($r_{\text{live}} + r_{\text{filling}}$) is greater than video playback rate (r_{playback}). The unconsumed video segments will be saved on the client's auxiliary storage. Because media playback rate is equal to the live broadcasting rate ($r_{\text{playback}} = r_{\text{live}} = B$), the rate to fill the missing segments mainly contributes to the buffer accumulation, as indicated in Figure 9.

Suppose that the filling process of missing segments ends at T_f and the live video ends at T_e. Thus the integration

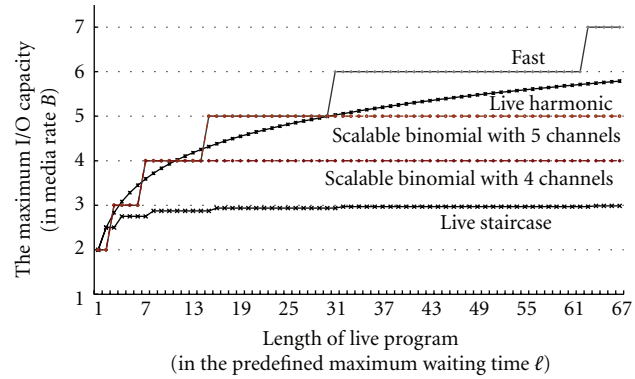


Figure 8: The maximum I/O load on clients versus the length of live videos.

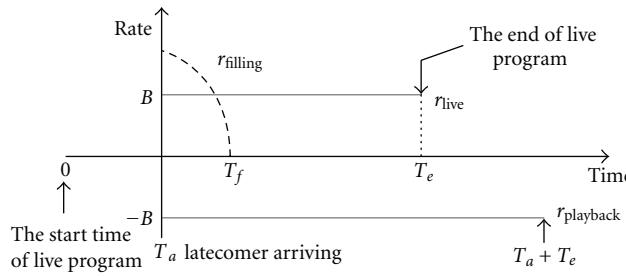


Figure 9: A media rate diagram of a client on LHB.

$\int_{T_a}^{T_f} r_{\text{filling}} dt$ is the size of missing video segments. If T_f is before T_e, as shown in Figure 9, the maximum buffer requirement (Z) is equal to the size of missing video segments $Z = \int_{T_a}^{T_f} r_{\text{filling}} dt$. In order to satisfy the continuous playback constraint, the time to fill the missing segments must be shorter than the time to play the missing segments. Therefore, if a client arrives in the first half of the live broadcasting, the buffered video size is always equal to the size of missing video segments. On the contrary, if a client arrives after the midpoint of a live show, the maximum buffer requirement varies with the adopted live broadcasting schemes. Suppose that the length of a live video is 1024l. Figure 10 illustrates the relationship between the buffer requirements and the client's arrival time.

Suppose that a client arrives at the nth time slot and the live video length is N. For the live broadcasting schemes, their maximum buffer (Z) requirements are listed as follows.

(i) *LHB*. Its maximum buffer requirement is

$$Z_h = \left(\sum_{i=n+1}^{N+n} \zeta^i\right) \times \ell B \quad \text{when } \zeta^i \geq 0, \quad (13)$$

where $\zeta^i = r_{\text{live},i} + \phi^i - r_{\text{playback},i}$. The ϕ^i is the sum of the $(i\text{-}n)$th column of filling rate matrix in (6) when $i \leq n + n$; others $\phi^i = 0$. (The $r_{\text{live},i} = 0$, when $i > N$; others $r_{\text{live},i} = r_{\text{playback},i} = B$.)

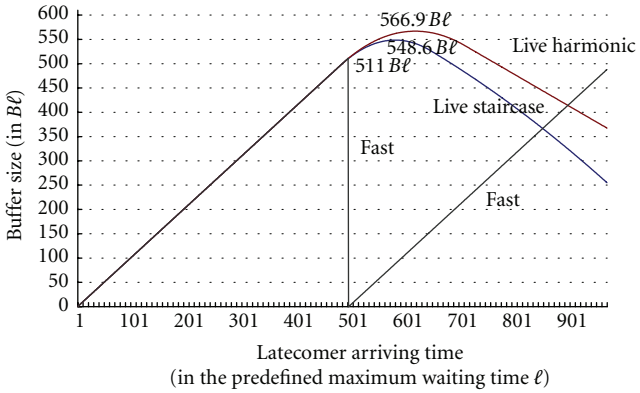


Figure 10: The relation between the buffer requirement and the client's arrival time.

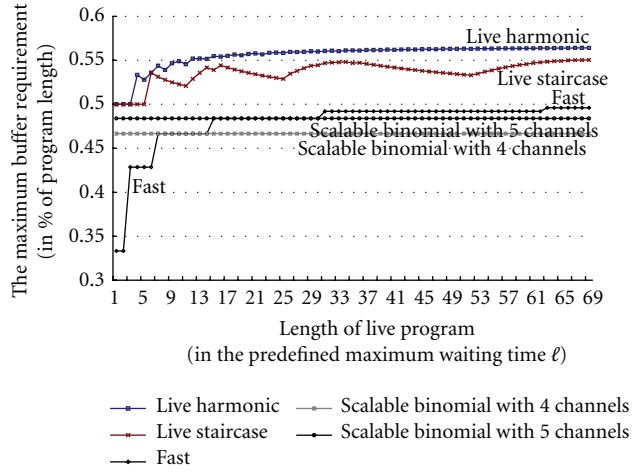


Figure 11: The maximum buffer requirements of each live broadcasting scheme.

(ii) *LSB*. Its maximum buffer requirement is

$$Z_s = \left(\sum_{i=n+1}^{N+n} \zeta^i \right) \times \ell B \quad \text{when } \zeta^i \geq 0, \tag{14}$$

where $\zeta^i = r_{\text{live},i} + \phi^i - r_{\text{playback},i}$. The ϕ^i is the sum of the $(i\text{-}n)$th column of filling rate matrix in (9) when $i \leq n + n$; others $\phi^i = 0$. (The $r_{\text{live},i} = 0$, when $i > N$; others $r_{\text{live},i} = r_{\text{playback},i} = B$.)

(iii) *LFB*. Its maximum buffer requirement can be obtained from [3]

$$Z_F = \left(\frac{(2^{c-1} - 1)}{(2^c - 1)} \right) \times \ell B, \tag{15}$$

where c is the number of channels, and $c = (1 + \lfloor \log_2 N \rfloor)$.

(iv) *Scalable Binomial Broadcasting*. Its maximum buffer requirement is identical to that of live fast broadcasting:

$$Z_B = \left(\frac{(2^{c-1} - 1)}{(2^c - 1)} \right) \times \ell B, \tag{16}$$

where C is a constant number of the preallocated channels.

According to (13) to (16), Figure 11 illustrates the maximum buffer requirements of the previous live broadcasting in the percentage of the video length.

4. Conclusions

Live program distribution is an important service on Internet. However, most current broadcasting schemes, such as PB, NPB, SB, and HB, cannot support live broadcasting. In this paper, we analyze the requirements for broadcasting live programs. In addition, the time skewing approach is proposed to allow traditional the VOD broadcasting schemes to distribute live programs. We also develop the scalable binomial broadcasting to distribute a live program with varying length in constant bandwidth consumption, however, at the cost of longer waiting time. The analyses and comparisons indicate that the scalable binomial broadcasting provides a reasonable performance than other live broadcasting schemes with respect to bandwidth requirements, I/O capacity, and receiver's buffer size.

References

[1] T. D. Little and D. Venkatesh, "Prospects for interactive video-on-demand," *IEEE Multimedia*, vol. 1, no. 3, pp. 14–24, 1994.

[2] T. C. Chiueh and C. H. Lu, "Periodic broadcasting approach to video-on-demand service," in *Integration Issues in Large Commercial Media Delivery Systems*, vol. 2615 of *Proceedings of SPIE*, pp. 162–169, October 1995.

[3] L. S. Juhn and L. M. Tseng, "Fast data broadcasting and receiving scheme for popular video service," *IEEE Transactions on Broadcasting*, vol. 44, no. 1, pp. 100–105, 1998.

[4] L. S. Juhn and L. M. Tseng, "Adaptive fast data broadcasting scheme for video-on-demand service," *IEEE Transactions on Broadcasting*, vol. 44, no. 2, pp. 182–185, 1998.

[5] J.- F. Paris, S. W. Carter, and D. E. Long, "A hybrid broadcasting protocol for video-on-demand," in *Proceedings of the Multimedia Computing and Networking Conference*, pp. 317–326, San Jose, Calif, USA, January 1999.

[6] Y. C. Tseng, M. H. Yang, and C. H. Chang, "A recursive frequency-splitting scheme for broadcasting hot videos in VOD service," *IEEE Transactions on Communications*, vol. 50, no. 8, pp. 1348–1355, 2002.

[7] L. S. Juhn and L. M. Tseng, "Staircase data broadcasting and receiving scheme for hot video service," *IEEE Transactions on Consumer Electronics*, vol. 43, no. 4, pp. 1110–1117, 1997.

[8] L. S. Juhn and L. M. Tseng, "Harmonic broadcasting for video-on-demand service," *IEEE Transactions on Broadcasting*, vol. 43, no. 3, pp. 268–271, 1997.

[9] Z. Y. Yang, L. S. Juhn, and L. M. Tseng, "On optimal broadcasting scheme for popular video service," *IEEE Transactions on Broadcasting*, vol. 45, no. 3, pp. 318–324, 1998.

[10] S. Viswanathan and T. Imielinski, "Metropolitan area video-on-demand service using pyramid broadcasting," *Multimedia Systems*, vol. 4, no. 4, pp. 197–208, 1996.

Hindawi Publishing Corporation
International Journal of Digital Multimedia Broadcasting
Volume 2012, Article ID 717538, 7 pages
doi:10.1155/2012/717538

Research Article

An Efficient Periodic Broadcasting with Small Latency and Buffer Demand for Near Video on Demand

Ying-Nan Chen and Li-Ming Tseng

Department of Computer Science and Information Engineering, National Central University, Chung-Li 32054, Taiwan

Correspondence should be addressed to Ying-Nan Chen, ynchen@dslab.csie.ncu.edu.tw

Received 31 January 2012; Accepted 3 February 2012

Academic Editor: Hsiang-Fu Yu

Broadcasting Protocols can efficiently transmit videos that simultaneously shared by clients with partitioning the videos into segments. Many studies focus on decreasing clients' waiting time, such as the fixed-delay pagoda broadcasting (FDPB) and the harmonic broadcasting schemes. However, limited-capability client devices such as PDAs and set-top boxes (STBs) suffer from storing a significant fraction of each video while it is being watched. How to reduce clients' buffer demands is thus an important issue. Related works include the staircase broadcasting (SB), the reverse fast broadcasting (RFB), and the hybrid broadcasting (HyB) schemes. This work improves FDPB to save client buffering space as well as waiting time. In comparison with SB, RFB, and HyB, the improved FDPB scheme can yield the smallest waiting time under the same buffer requirements.

1. Introduction

How to efficiently maintain the exhausted bandwidth with the growth in the number of clients is an important issue of VOD deployment. Dan et al. [1] presented that 80 percent of the demand is for a few number (10 or 20) of very popular videos. One way to broadcast a popular video is to partition the video into segments, which are transmitted on several channels currently and periodically. The approach (called periodic broadcasting [2]) lets multiple users share channels and thus obtains high bandwidth utilization. One of the channels only broadcasts the first segment in real time. The other channels transmit the remaining segments. When clients want to watch a video, they wait for the beginning of the first segment on the first channel. While clients start watching the video, their set-top boxes (STBs) or computers still download and buffer unreceived segments from the channels to enable them to play the video continuously.

The staggered broadcasting [3] scheme treats a complete video as a single segment and then transmits it on each channel at different start times. The fast broadcasting (FB) scheme [4] improves segment partitioning and arrangement to yield shorter service latency. The harmonic broadcasting (HB) scheme [5] initially partitions a video into equally sized segments, which are further divided into smaller subsegments according to the harmonic series. HB can yield the minimum waiting time [6]; however, its implementation is difficult due to the multitude of broadcasting channels [7]. The recursive frequency-splitting (RFS) [7] scheme broadcasts a segment as close to its frequency as possible to achieve a near-minimal waiting time. The study [8] focuses on reducing the computation complexity of RFS. Paris [9] proposed a fixed-delay pagoda broadcasting (FDPB) scheme that required clients to wait for a small-fixed delay before watching the selected videos.

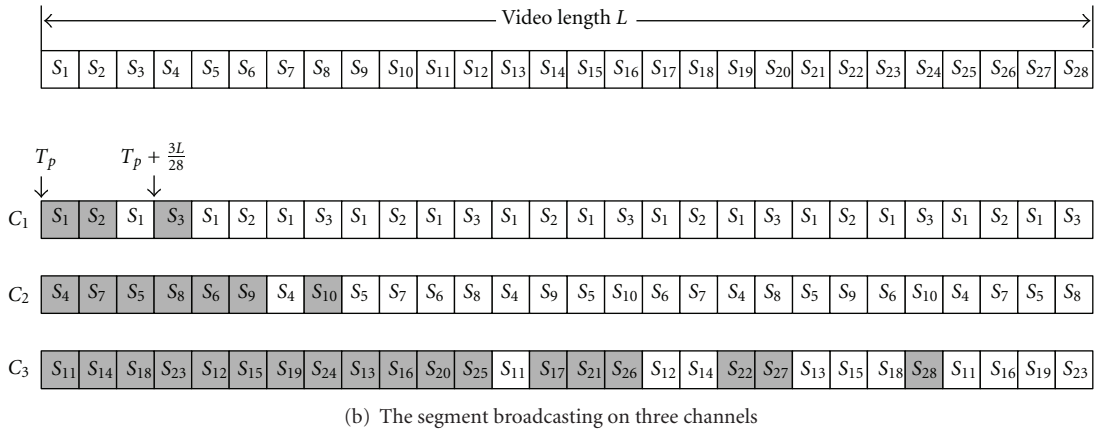
The staircase broadcasting (SB) scheme [10] requires a client to buffer only 25% of a playing video. In modifying the FB scheme, the reverse fast broadcasting (RFB) scheme [11] also buffers 25% of video size, merely half of what is required by the FB scheme. By combining RFS and RFB the hybrid broadcasting scheme (HyB) [12] yields small client buffering space and waiting time. The study in [13] proposed a generalized reverse sequence-based model to reduce their client buffer requirements. This work aims at improving FDPB to reduce required playback latency and buffering space. We prove the applicability of the improved FDPB, and compare

Stream 1		
Substream	1	2
First segment	1	2
Last segment	1	3

Stream 2		
Substream	1	2
First segment	4	7
Last segment	6	10

Stream 3				
Substream	1	2	3	4
First segment	11	14	18	23
Last segment	13	17	22	28

(a) The segment allocation on three channels



(b) The segment broadcasting on three channels

Figure 1: Illustration of channel allocation for the FDPB-3 scheme.

its performance with that of several existing approaches. Given a bandwidth of 5 channels, the new scheme reduces the broadcast latency by as much as 24% when compared to HyB and 34.5% when compared to RFB and SB. The buffer requirements for these schemes are about 25% of video size.

The rest of this paper is organized as follows. Section 2 introduces FDPB and proposes its improvement. Performance analysis and comparison are in Section 3. We make brief conclusions in Section 4.

2. Proposed Scheme

This section briefly reviews FDPB and then presents the improvement.

2.1. Fixed-Delay Pagoda Broadcasting (FDPB) Scheme. To help explain the improved FDPB, we first review the FDPB scheme in the literature [9]. Suppose that a video of duration L is broadcast over k channels. The scheme partitions each video into m equal-size segments of duration $l = L/m$. These segments will be broadcast at different frequencies over the k channels. The FDPB-p scheme requires customers to wait for a fixed time interval $w = pl$, where $p \in$ int, $p \geq 1$. Segment S_i is thus transmitted at least once every $p + i - 1$ slots. The FDPB-p scheme partitions each channel C_j into $c_j = \sqrt{p + i - 1}$ subchannels in such a way that slot j of channel C_j belongs to its subchannel j. FDPB then maps segments into subchannels in a strict sequential fashion.

Figure 1 illustrates the video partition and segment arrangement of FDPB-3. Channel C_1 is partitioned into two subchannels because $\sqrt{3 + 1 - 1}$ is close to 2. Let S_i be the ith segment. Since $p = 3$, segment S_1 needs to be broadcast at least once every three slots and is assigned to subchannel 1. The first segment of subchannel 2 is segment S_2, which appears at least once every $3 + 2 - 1 = 4$ slots. We thus map two segments into subchannel 2, and channel C_1 transmits three segments (i.e., S_1 to S_3). The first segment of channel C_2 is segment S_4, which is broadcast at least once every $3 + 4 - 1 = 6$ slots. Since 6 is close to $4 = 2^2$, channel C_2 is partitioned into two subchannels. The first subchannel periodically transmits segments S_4 to S_6, and the second subchannel broadcasts segments S_7 to S_{10}. The first segment of channel C_3 is segment S_{11}, which is broadcast at least once every $3 + 11 - 1 = 13$ slots. Similarly, since 13 approaches to $16 = 4^2$, channel C_3 is partitioned into four subchannels. The first subchannel sends segments S_{11} to S_{13}, the second subchannel sends segments S_{14} to S_{17}, the third subchannel sends segments S_{18} to S_{22}, and the fourth subchannel sends S_{23} to S_{28}. Hence, channel C_3 broadcasts 18 segments. Figure 1(b) demonstrates how to broadcast the segments on three channels and how to download the segments colored gray. Suppose that a client starts receiving the segments at time T_p. This figure shows that the client begins to watch the video at time $T_p + 3L/28$.

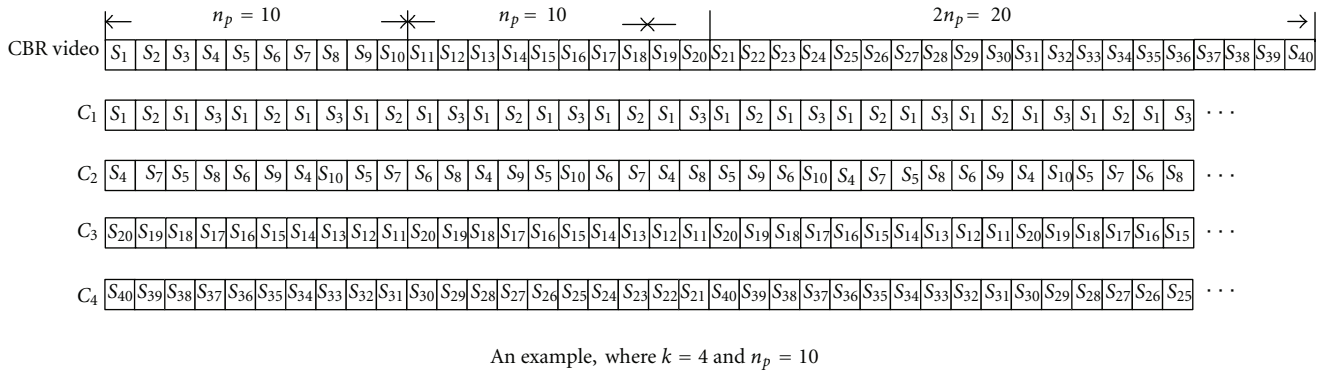


Figure 2: Segment transmission by RFDPB-3.

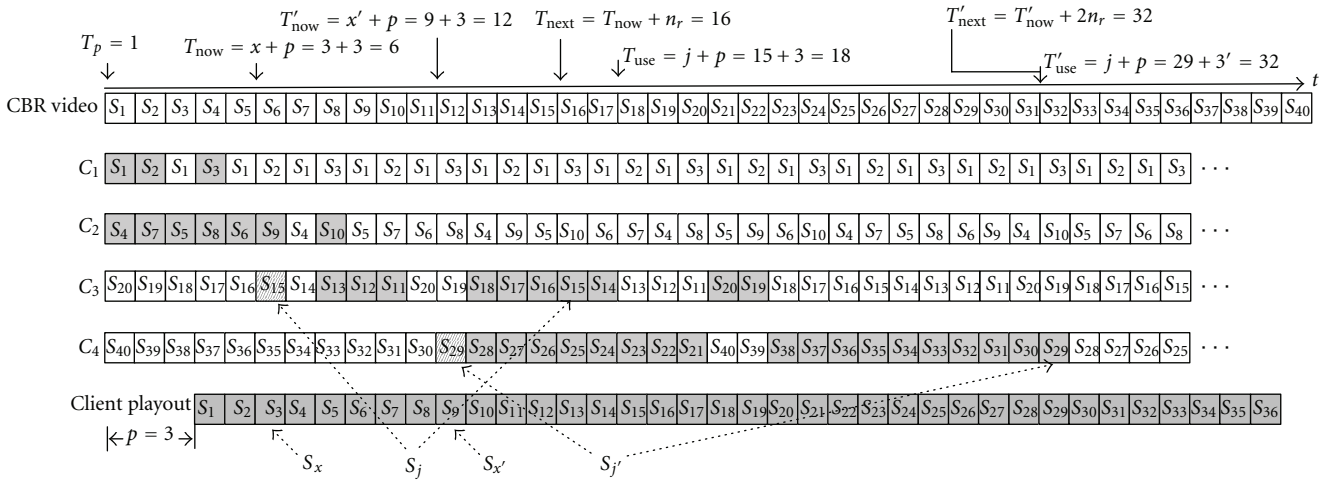


Figure 3: An illustration of segment downloading by RFDPB-3, where $k = 4$, $n_p = 10$, and $N = 40$.

2.2. Improvement of FDPB. By integrating the RFB with FDPB, this work designs the RFB-FDPB (RFDPB) scheme, which gains small low buffer requirements as well as waiting time.

2.2.1. Segment Delivery on the Server Side. Let k be the number of allocated channels. The data transmission rate of each channel is assumed to equal the playback rate b. An RFDPB server broadcasts segments according to the following steps.

(1) The largest number of segments that FDPB-p can transmit using $k-2$ channels is assumed to equal n_p. The RFDPB-p scheme then equally divides a constant-bit-rate (CBR) video into N segments, denoted by $S_1, S_2, \ldots, S_N$ in sequence, where $N = 4n_p$. Let L be the length of the video. The length of a segment, l, thus equals L/N.

(2) Channels C_1 to C_{k-2} periodically send segments S_1 to S_{n_p} according to FDPB-p.

(3) Segments S_{2n_p} to S_{n_p+1} are transmitted on channel C_{k-1}.

(4) Segments S_{4n_p} to S_{2n_p+1} are broadcast on channel C_k.

Figure 2 illustrates the segment delivery by RFDPB-3, where $k = 4$. The server broadcasts segments S_1 to S_{10} on channels C_1 to C_2 by FDPB-3 because FDPB-3 can transmit 10 segments using two channels, as indicated in Figure 1. Due to $n_p = 10$, RFDPB periodically broadcasts segments S_{20} to S_{11} and S_{40} to S_{21} on channels C_3 and C_4.

2.2.2. Segment Reception on the Client Side. Suppose that a client has enough buffers to store portions of a playing video. Let l be a basic time unit during video playing. We also let T_p be the time that the client starts downloading the video and be the origin (i.e., the first time unit) of the time axis throughout the paper. The following steps are involved in playing a video at the client.

(1) The client receives the segments on channels C_1 to C_{k-2} immediately when they are available on networks. Figure 3 further demonstrates the segment downloading, where the segments downloaded by a client are gray.

(2) The segment downloading on channel C_{k-1} is as follows. Suppose that the client first sees segment S_j on channel C_{k-1} at time T_{now} and next segment

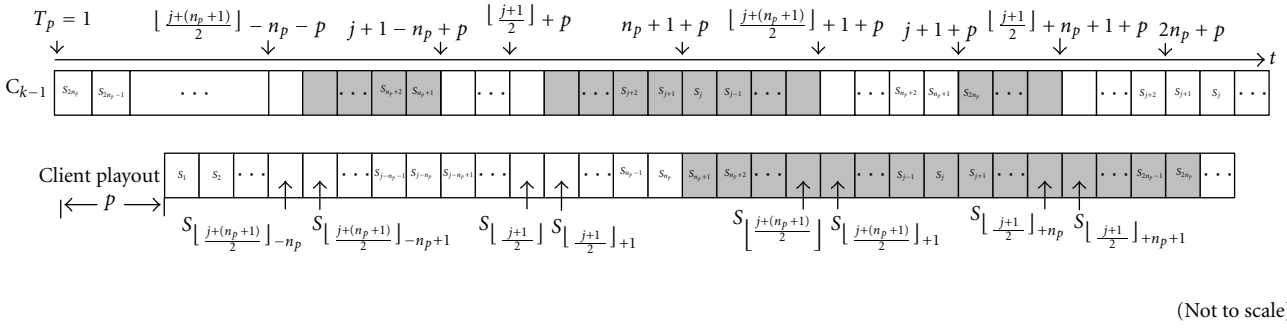


Figure 4: The segment downloading on channel C_{k-1} under RFDPB-p.

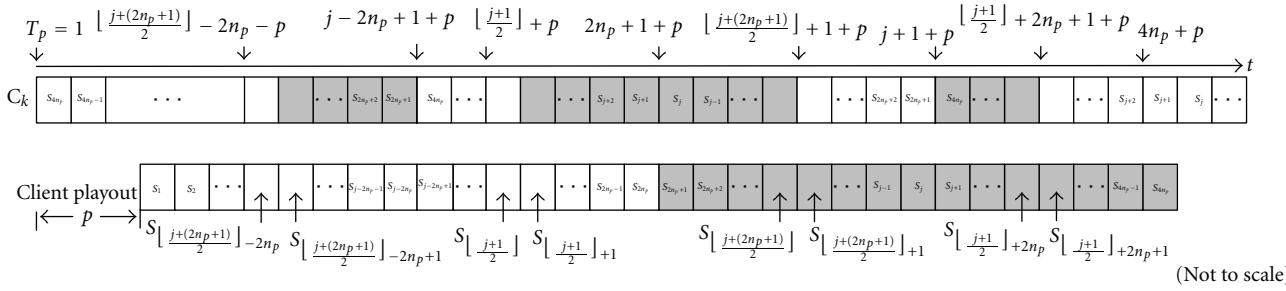


Figure 5: The segment downloading on channel C_k under RFDPB-p.

S_j at time T_{next}. The client is also assumed to play segments S_x and S_j at time T_{now} and T_{use}, respectively. The work in [11, 12] indicates that if $T_{\text{next}} \leq T_{\text{use}}$, the client does not need to receive segment S_j at time T_{now} and actually performs the downloading at time T_{next}, without causing playing interruption. FDPB further reveals that a client has to wait for a fix time unit p to play received segments. Thus, $T_{\text{use}} = j + p$th time unit, $T_{\text{now}} = x + p$th time unit, and $T_{\text{next}} = T_{\text{now}} + n_p$. Substituting these equations into $T_{\text{next}} \leq T_{\text{use}}$, we obtain

$$x + n_p \leq j. \tag{1}$$

If this inequality holds, a client does not receive segment S_j at time T_{now}; otherwise, it immediately downloads it. Suppose that the client first sees segment S_{15} with only diagonal lines on channel C_3 at the 6th time unit, as indicated in Figure 3. Due to $p = 3$, $x = 3$, $j = 15$, and $n_p = 10$, $T_{\text{now}} = x + p = 6$, $T_{\text{use}} = j + p = 18$, and $T_{\text{next}} = T_{\text{now}} + n_p = 16$. Clearly, it is unnecessary to download segment S_{15} at this time unit because inequality (1), $x + n_p = 13 \leq j = 15$, holds. The result also reflects that the client does not have to receive the segment when $T_{\text{next}} = 16 \leq T_{\text{use}} = 18$. When next segment S_{15} colored gray in Figure 3 arrives at the 16th time unit, the inequality is no more true, $13 + 10 > 15$, and the client thus downloads it.

(3) The segments on channel C_k are received in the similar way, as also indicated in Figure 3. Suppose that the client first sees segment $S_{j'}$ on channel C_k at time T'_{now}, and next segment $S_{j'}$ at time T'_{next}. The client is also assumed to play segments $S_{x'}$ and $S_{j'}$ at time T'_{now} and T'_{use}. Similarly, if $T'_{\text{next}} \leq T'_{\text{use}}$, the client does not receive segment $S_{j'}$. Substituting $T'_{use} = j' + p$th time unit, $T'_{\text{now}} = x' + p$th time unit, and $T'_{\text{next}} = T'_{\text{now}} + 2n_p$ into inequality $T'_{\text{next}} \leq T'_{\text{use}}$ obtains

$$x' + 2n_p \leq j'. \tag{2}$$

If the inequality is true, the client skips the downloading of segment $S_{j'}$. Suppose that the client first sees segment S_{29} with only diagonal lines on channel C_4 at the 12th time unit, as indicated in Figure 3. Due to $p = 3$, $x' = 9$, $j' = 29$, and $n_p = 10$, $T'_{\text{now}} = x' + p = 12$, $T'_{\text{use}} = j' + p = 32$, and $T'_{\text{next}} = T'_{\text{now}} + 2n_p = 32$. Clearly, it is unnecessary to download segment S_{29} at this time unit because inequality (2), $x' + 2n_p = 29 \leq j' = 29$, is true. The result also reflects that the client does not have to receive the segment when $T'_{\text{next}} = 32 \leq T'_{\text{use}} = 32$. When next segment S_{29} colored gray in Figure 3 arrives at the 32nd time unit, the inequality is no more true, $29 + 20 > 29$, and the client thus downloads it.

(4) The client plays the video in the order of $S_1, S_2, \ldots, S_N$ at time $T_p + p$.

(5) Stop loading data from networks when all the segments have been received.

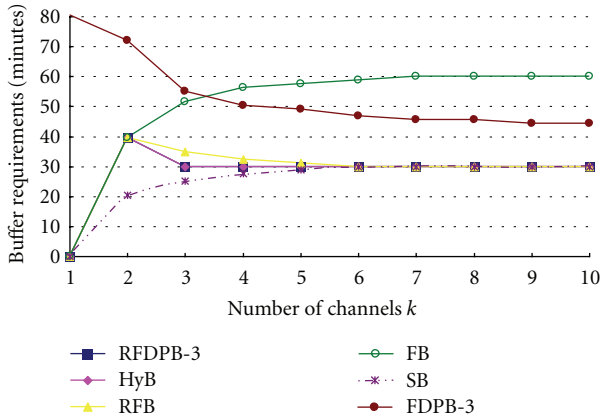


Figure 6: Comparison of the maximum buffer requirements in the number of minutes of a 120-minute video.

Table 1: The maximum buffering space required by different schemes in the percentage of video size using k channels.

k	1	2	3	4	5	6	7	8	9	10
RFDPB-3	0	33	25	25	25	25	25	25	25	25
HyB	0	33	25	25	25	25	25	25	25	25
RFB	0	33	29	27	26	25	25	25	25	25
FB	0	33	43	47	48	49	50	50	50	50
SB	0	17	21	23	24	25	25	25	25	25
FDPB-3	67	60	46	42	41	39	38	38	37	37

2.2.3. Workable Verification. This section describes that the RFDPB scheme guarantees continuous playing on the client side. Because the segment broadcasting on channels C_1 to C_{k-2} is based on FDPB [9], RFDPB is workable for these channels. We next prove that the segment arrangements on channels C_{k-1} and C_k also ensure a client to continuously play a video. FDPB further indicates that a video server must broadcast segment S_j at least once in every $j + p - 1$ time units to keep on-time data delivery on the client side. For the RFDPB scheme, a server broadcasts segment S_j on channel C_{k-1} in every n_p time units, where $n_p + 1 \le j \le 2n_p$. Because $j + p - 1 > n_p$, an RFDPB client can receive video segments on channel C_{k-1} on time. Similarly, the RFDPB scheme requires a server to transmit segment $S_{j'}$ on channel C_k in every $2n_p$ time units, where $2n_p+1 \le j' \le 4n_p$. Because $j' + p - 1 > 2n_p$, the segment delivery on this channel is also on time. Accordingly, RFDPB ensures continuous video playing at the client.

3. Analysis and Comparison

Before investigating entire buffer requirements for RFDPB, this paper analyzes the segment downloading on channel C_{k-1} first. For RFDPB-p, a client plays the segments received from this channel during time units n_p+1+p to $2n_p+p$. The possible time to download the segments is during the pth to $2n_p + p$th time units. Suppose that a client sees segment S_j at the $n_p + 1 + p$th time unit. Inequality (1) makes a

Table 2: The maximum numbers of segments, N, offered by different schemes.

k	1	2	3	4	5	6	7	8	9	10
FDPB-6	7	25	71	186	485	1286	3425	9195	24790	67054
RFDPB-6	1	3	28	100	284	744	1940	5144	13700	36780
HyB	1	3	4	12	36	100	292	804	2260	6088
FB (RFB)	1	3	7	15	31	63	127	255	511	1023
SB	1	3	7	15	31	63	127	255	511	1023

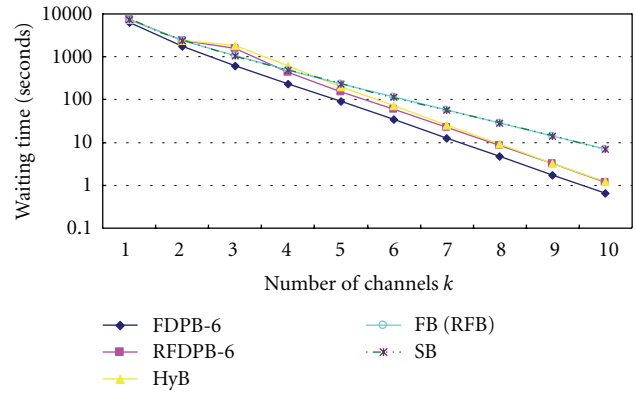


Figure 7: The maximum waiting time incurred on new clients at different numbers of channels (L = 120 minutes).

complete segment-downloading diagram for channel C_{k-1}, as indicated in Figure 4, where the segments downloaded by the client are colored gray. The figure shows that the client does not continuously download segments, and some segments are skipped. The segments are not downloaded during the first to $\lfloor (j + (n_p + 1))/2 \rfloor - n_p + p$th time units, during the $j + 1 - n_p + p$th to $\lfloor (j + 1)/2 \rfloor + p$th time units, during the $\lfloor (j+(n_p+1))/2 \rfloor+1+p$th to $j+p$th time units, and during the $\lfloor (j + 1)/2 \rfloor + n_p + 1 + p$th to $2n_p + p$th time units, respectively. Such results reflect that the client can delay downloading these segments by n_p time units, due to (1). Similarly, (2) also makes a complete segment-downloading diagram for channel C_k, as indicated in Figure 6, where a client is assumed to see segment S_j at the $2n_p + 1 + p$th time unit.

The RFDPB scheme uses the same methodology as HyB, combining a nonharmonic scheme and the RFB to reduce clients' waiting time and buffering spaces (Figure 5). For segment broadcasting on channels C_1 to C_{k-2} using the nonharmonic scheme, channels C_{k-1} and C_k broadcast the other segments according RFB. Yu's work [12] has proved HyB only buffer 25% of video size for $k \ge 3$, where k is the number of broadcasting channels. The proof of the maximum number of segments buffered by a client for RFDPB is similar to that of HyB, so this paper neglects it. Therefore, let $B(k)$ be the maximum number of segments buffered by a client, where k is the number of broadcasting channels. For RFDPB,

$$\begin{aligned} B(1) &= 0 \\ B(2) &= 1 \\ B(k) &= n_p, \quad k \ge 3. \end{aligned} \tag{3}$$

Table 3: The waiting time (seconds), offered by RFDPB-9, RFDPB-6, HyB, and RFDPB-3 schemes.

k	1	2	3	4	5	6	7	8	9	10
RFDPB-9	7200	2400	1350	385.71	139.66	52.6	19.9	7.47	2.79	1.04
RFDPB-6	7200	2400	1542.86	432	152.11	58.06	22.27	8.4	3.15	1.17
HyB	7200	2400	1800	600	200	72	24.66	8.96	3.19	1.18
RFDPB-3	7200	2400	1800	540	192.86	72.97	28.57	11.07	4.19	1.58

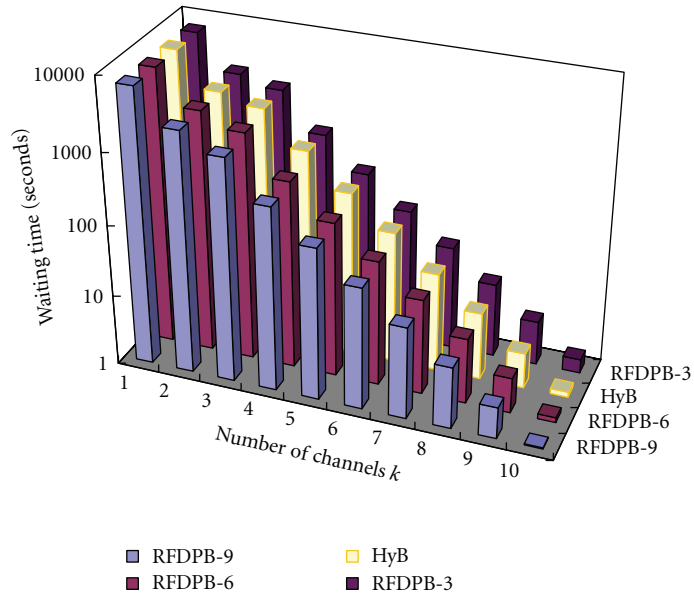


Figure 8: The maximum waiting time (RFDPB and HyB) incurred on new clients at different numbers of channels ($L = 120$ minutes).

The previous analysis indicates that RFDPB buffers at most $n_r/N = 25\%$ of video size. Table 1 lists the comparison of maximum buffering spaces required by RFDPB-3, FB, SB, RFB, FDPB-3, and HyB. RFDPB-3 reduces the buffering space up to 32.4% when compared to FDPB-3 and up to 50% when compared to FB and has the same good result with the other schemes. Given a video of 120 minutes, Figure 6 shows the buffer requirements at different channels. To understand how well the RFDPB scheme performs on clients waiting time, this work calculates the values of N offered by RFDPB-6, SB, FB, RFB, FDPB-6, and HyB given different numbers of server channels, as listed in Table 2. The inverse of N offered by each scheme reflects the waiting time for a new client to start his/her VOD service. Figure 7, which is drawn in a logarithmic scale, shows that RFDPB-6 performs close to FDPB-6 stably. For $k \geq 4$, the RFDPB-6 outperforms all the schemes, except the FDPB-6. To understand the relationship between RFDPB and HyB schemes, this work calculates clients' waiting time of RFDPB-9, RFDPB-6, HyB, and RFDPB-3, as listed in Table 3. Figure 8 shows that, for $p \geq 6$, RFDPB-p outperforms HyB. The previous comparisons clearly indicate that RFDPB exhibits a good tradeoff between client buffering spaces and waiting time.

4. Conclusions

This paper presents an improved version of FDPB for efficient periodic broadcasting of popular videos. The proposed scheme takes advantage of the FDPB and the RFB schemes to obtain small waiting time and low buffer demand. Through mathematical analysis, we prove the applicability of this scheme by demonstrating that client playback continuity is guaranteed. Given a bandwidth of 5 channels, the new scheme reduces the broadcast latency by as much as 24% when compared to HyB and 34.5% when compared to RFB and SB. The buffer requirements for these schemes are about 25% of video size.

References

[1] A. Dan, D. Sitaram, and P. Shahabuddin, "Dynamic batching policies for an on-demand video server," *Multimedia Systems*, vol. 4, no. 3, pp. 112–121, 1996.

[2] D. Saparilla, K. W. Ross, and M. Reisslein, "Periodic broadcasting with VBR-encoded video," in *Proceedings of the 18th Annual Joint Conference of the IEEE Computer and Communications Societie (INFOCOM '99)*, pp. 464–471, March 1999.

[3] K. C. Almeroth and M. H. Ammar, "The use of multicast delivery to provide a scalable and interactive video-on-demand service," *IEEE Journal on Selected Areas in Communications*, vol. 14, no. 6, pp. 1110–1122, 1996.

[4] L-S. Juhn and L. M. Tseng, "Fast data broadcasting and receiving scheme for popular video service," *IEEE Transactions on Broadcasting*, vol. 44, no. 1, pp. 100–105, 1998.

[5] L.-S. Juhn and L. M. Tseng, "Harmonic broadcasting for video-on-demand service," *IEEE Transactions on Broadcasting*, vol. 43, no. 3, pp. 268–271, 1997.

[6] Z.-Y. Yang, L. S. Juhn, and L. M. Tseng, "On optimal broadcasting scheme for popular video service," *IEEE Transactions on Broadcasting*, vol. 45, no. 3, pp. 318–322, 1999.

[7] Y.–C. Tseng, M. H. Yang, and C. H. Chang, "A recursive frequency-splitting scheme for broadcasting hot videos in VOD service," *IEEE Transactions on Communications*, vol. 50, no. 8, pp. 1348–1355, 2002.

[8] J.-P. Sheu, H. L. Wang, C. H. Chang, and Y. C. Tseng, "A fast video-on-demand broadcasting scheme for popular videos," *IEEE Transactions on Broadcasting*, vol. 50, no. 2, pp. 120–125, 2004.

[9] J. F. Paris, "A fixed-delay broadcasting protocol for video-on-demand," in *Proceedings of the 10th International Conference on Computer Communications and Networks (ICCCN '01)*, pp. 418–423, Scottsdale, Ariz, USA, October 2001.

[10] L.-S. Juhn and L. M. Tseng, "Staircase data broadcasting and receiving scheme for hot video service," *IEEE Transactions on Consumer Electronics*, vol. 43, no. 4, pp. 1110–1117, 1997.

[11] H.-F. Yu, H. C. Yang, and L. M. Tseng, "Reverse Fast Broadcasting (RFB) for video-on-demand applications," *IEEE Transactions on Broadcasting*, vol. 53, no. 1, pp. 103–110, 2007.

[12] H.-F. Yu, "Hybrid broadcasting with small buffer demand and waiting time for video-on-demand applications," *IEEE Transactions on Broadcasting*, vol. 54, no. 2, Article ID 4433980, pp. 304–311, 2008.

[13] H.-F. Yu, P. H. Ho, and H. C. Yang, "Generalized Sequence-Based and Reverse Sequence-Based Models for Broadcasting Hot Videos," *IEEE Transactions on Multimedia*, 2008.