

An Implementation of End-to-End Controlled Streaming System Using Similarity-based Frame Discarding Approach on DiffServ

Jau-Yuan Chen, *Student Member, IEEE*, Min-Way Hsu, and Cheng-Fu Chou

Abstract —*Video streaming technology has been widely used for various Internet services. In this paper, we present an implementation strategy of video streaming systems under end-to-end traffic controls, and we adapt this system to fit into the differentiated services (DiffServ) architecture. Unlike other previous works, we suggest that an end-to-end controlled streaming system is viable and easily deployed without any adaptation to core network routers. We will present experimental results to illustrate the effectiveness of our proposed system.*¹

Index Terms —*DiffServ, End-to-End, Streaming*

I. INTRODUCTION

Video streaming technology has been widely used for various Internet services, such as Internet television programs, and video-on-demand services. Rather than downloading the whole video content, the video player at the end host will only play the received parts of the video that are able to be decoded. However, streaming services are in essence highly sensitive to network congestion, packet loss and delay, which may arise from an overloaded core network; only if any of the cases occurs, some of the received frames may not be capable to be decoded and end users will soon and easily detect such an annoying situation. Some previous works [1], [2], [3], [4], [5] have been proposed to smooth the video streaming by sending parts of the video in advance when the available bandwidth is larger than the video bit rate, and clients have to pre-fetch pre-sending parts in their own buffer. Nevertheless, such an approach works only under the circumstances that the average bandwidth is larger than the average video bit rate. In the current best-effort Internet, to provide guarantees of any amount of bandwidth requires several upgrades to the routers in the current core networks, and which makes it impossible to be implemented.

Integrated services (IntServ) [6] and differentiated services (DiffServ) [7] are the most prominent architectures to support quality of service (QoS) in Internet. The guaranteed service model of IntServ [8] provides per-packet delay guarantees and has to maintain per-flow states in core

routers, which makes the IntServ model not scalable. DiffServ provides only per-flow bandwidth guarantees by keeping per-class states instead of per-flow states in core routers, and is more scalable [9]. To achieve QoS guarantees, admission control is required so that network resources are not over-allocated and can be utilized more efficiently.

In this paper, we present an implementation strategy of video streaming systems under end-to-end traffic controls, and we adapt this system to fit into the differentiated services architecture. Unlike other previous works, we suggest that an end-to-end controlled streaming system is viable and easily deployed without any adaptation to core network routers. In our system, each streaming video will be pre-processed to get the information about the degree of similarity of any consecutive frames, and a frame similarity report is generated. Instead of sending all the frames in the streaming video like before, our system dynamically adjusts the proper bit rate according to the current network status and discards parts of the frames of the video in the order of frame priorities stated by the frame similarity report. Not only does our system utilize the bandwidth efficiently, but we also avoid overloaded traffic on the Internet. In addition, we could provide more smooth video streaming services than others when the network resources is highly constraint by reason of the similarity-based frame dropping mechanism. Under the DiffServ architecture, we provide a differentiated service with the properties of consistency and controllability. In other words, the higher classes always receive better services, and the server can take control of performance of each class and admissions to services based on some policies. We will present experimental results to illustrate the effectiveness of our proposed system.

II. RELATED WORKS

Packet loss during a streaming service usually results from insufficient available bandwidth and congestions. In both cases, if the streaming server tries to transmit data with the original sending rate, many packets will be dropped during transmission paths and the quality of video received declines.

A. Video Smoothing Approaches

Videos are usually compressed to reduce the size of necessary storage and transmission time. Whereas compressed videos, especially variable-bit-rate (VBR)

¹ This work was partially supported by the National Science Council and the Ministry of Education of ROC under the contract No. NSC92-2622-E-002-002 and 89E-FA06-2-4-8.

Jau-Yuan Chen, Min-Way Hsu and Cheng-Fu Chou are with the Department of Computer Science and Information Engineering, National Taiwan University, Taipei, Taiwan.
(Email: {JyChen, minway, ccf}@cmlab.csie.ntu.edu.tw)

videos, typically exhibits significant bursts because of encoding schemes and content variations. In order to lower down bandwidth requirement when transmitting bursty videos, some video smoothing approaches [1], [2], [3], [4], [5] have been proposed. When available bandwidth is larger than the video bit rate, the streaming server transmits some bursty parts of the video in advance as well as the normal video stream, and the pre-fetched video will be temporarily stored in the client-side buffer. Therefore, the bandwidth required is averaged to a smoother and lower value, as shown in Figure 1. However, the video smoothing approach is only suitable for the condition that the average available bandwidth is larger than the average video bit rate. If the average bandwidth is less than the average video bit rate, it is impossible to deliver the video stream without any information loss.

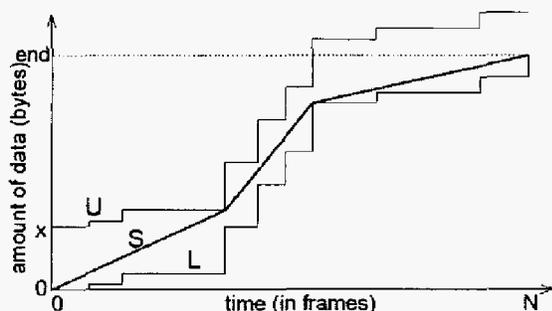


Fig. 1. Constraint Curves of Video Smoothing Transmission: The transmission plan S stays between the upper (U) and the lower (L) smoothing constraints.

B. Selective Frame Discarding Approaches

A simple and naïve way to deliver a video stream is to transmit all frames with required rates. However, being in a bandwidth-constrained network, such a simple approach will lead to packet losses or being late for its playback time because of network jitter, and in both cases, many frames may not be decoded at the receiver and should be dropped. Zhang [10] introduced the concept of selective frame discarding (SFD) algorithm at the server, which preemptively discards frames by taking network constraints, QoS requirements, and some application-specific information, such as content information of a frame, into consideration, so that the server can make the best of network resources and reduce the probability of frame discarding. Ramanathan [11] proposed a simple and efficient scheme, which exploits the inevitability of frame losses, and he called the subset of packets of a frame that will invalidate the frame once any of the packets are lost as “packet resiliency.” In Ramanathan scheme, whenever packet losses threshold has reached to invalidate a frame, all other packets that constitute this frame are forced to be discarded, and thus, network bandwidth and buffer spaces are released and can be reallocated to other video channels;

thereby, more services can be admitted and served. Hemy [12] proposed an integrated MPEG streaming system with a MPEG-aware filter. The main function of this filter is to adjust the video bit rate by discarding frames according to the current network condition. The policy always discards B-frames first, then P-frames, and I-frames last.

The above SFD mechanisms consider frame type, frame dependency and cost, whereas they do not mention other factors affecting human vision, such as content relation between two consecutive frames. Furthermore, the continuity of discarded frames is becoming less important in the video schemes with inter-frame dependency like MPEG. In such video scheme, each frame has different importance and those with higher priority, for example, I-frames in MPEG video, are distributed evenly. In our frame discarding scheme, we not only consider the frame type and dependency, but also take the similarity between frames into account. A frame that is less similar with its previous frame is more important because it is likely to be the beginning of a scene. In addition, we adjust the decision window size dynamically to accommodate the network dynamics.

III. SYSTEM ARCHITECTURE

There are two main parts of our streaming system. The first one is the similarity-based frame discarding module, which can do the dynamic adaptation to streaming bit rate; the other is the service control module, which controls service differentiation. In this section, we explain how the proposed system works, and we use two different strategies to deal with the situation that the available resources is rare.

A. System Overview

The architecture of our proposed system is shown as Figure 2. Most of the video streaming services establish connections over user datagram protocol (UDP) or real-time transport protocol (RTP), and thus, it is unfair to compete for bandwidth resources with connections established over transmission control protocol (TCP). Therefore, we adopt the widely accepted TCP-friendly rate control (TFRC) protocol [15], which is made to break the unfairness, to get the network status information. In the server side, frame similarity report is pre-generated from calculating the similarity degrees between consecutive frames, and the similarity-based frame discarding (SimFD) module determines whether a frame will be sent to the client based on the TFRC report sent back from the client and the frame priority, which is decided from the similarity degree. In addition, SimFD module should consider the policy running on the service control module to achieve DiffServ provisioning.

In the DiffServ architecture, all clients will be divided into several classes, and we assume the number of classes is fixed. Besides, all clients have their own service level agreements (SLA) reporting their service class. The aim of

service control module is to manage the policy to provide differentiated services. Under general situations, the service control module reports the user SLA information to the SimFD module, and the SimFD module will dynamically adjust the sending bit rate to the user in accordance with the service class. However, the best-effort Internet does not guarantee that the resource is always available for services, and on the other hand, the system operator would like to keep services running for the most benefits; to balance both of these, the service admission control policy should be carried out on the service control module.

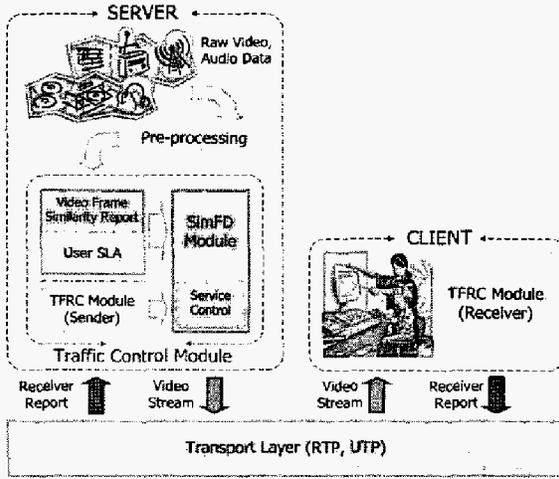


Fig. 2. System Architecture

B. Similarity-based Frame Discarding

In this section, we introduce the proposed method to determine the similarity degree of each frame and the frame discarding policy taken by the SimFD module.

1) Video Similarity

During a streaming service, frames may arrive too late to be played or even be dropped in the core networks. When either of these situations occurs, clients will suffer an unsmooth video quality, and we found that the decreased degree of QoS is correlative to the similarity between the absent frame and the previous one. The more dissimilar they are, the more the QoS is hurt. A video is indeed a sequence of pictures, so-called frames, played more than thirty pictures per second. Therefore, we define the frame similarity with the peak signal noise ratio (PSNR), which is usually used to estimate the difference between a reconstructed picture and the original one, as follows:

$$\text{Similarity}(f(i), f'(i)) = \text{PSNR}(f(i), f'(i))$$

where $f(i)$ and $f'(i)$ are the original frame and the received frame respectively. We denote the video similarity as follows:

$$\text{Similarity}(V, V') = \frac{\sum \text{Similarity}(V(i), V'(i))}{\text{number of frames}}$$

where V and $V(i)$ represents the original frame sequence and the i -th frame; V' and $V'(i)$ represents the received frame sequence and the i -th frame. In order to lighten loads on the server, the similarity report can be pre-generated at other host instead of on-line calculation. The size of the frame sequence can be adjusted dynamically. A larger frame sequence could have a better QoS, whereas a smaller frame sequence can react to network changes directly.

2) Frame Discarding Policy

A frame-based MPEG video, such as MPEG-1/2/4, has three types of frames: I-frame, B-frame, and P-frame; a group-of-pictures (GOP) is defined as a frame sequence starting with an I-frame. According to the inter-frame dependency, if an I-frame is discarded or lost, the entire GOP will be lost. Similarly, discarding a P-frame will lead to the loss of the P-frames and B frames that depend on it, and there is no frame depends on B-frames. Hence, while the network resources are not enough during streaming, the B-frame is always the first consideration in being discarded due to the inter-frame dependency, and then the P-frame, and the I-frame at last. In discarding I-frames and B-frames, the SimFD module discards frames by their similarity. As to discarding P-frames, the SimFD module discards frames from the last one and continues in the inverse order because of the dependency between P-frames. The action of discarding frames continues until there is enough bandwidth to transmit this frame sequence.

C. Service Control

Our goal focuses on the implementation of the streaming system with end-to-end frame-based rate control, and applies this system to the DiffServ architecture. In order to achieve DiffServ provisioning, we have to control the service in terms of the client amount, client classes, etc.

The concept of proportional differentiation [13], [14] is widely used in various DiffServ services. The main idea is to keep the performance spacing between classes in a fixed ratio, that is, the higher class will receive better services for all time. However, this policy does not guarantee QoS of any class. While the network resources are highly constraint, QoS of all classes will diminish but still keep the spacing.

Assume there are n classes in our service system, where n is a finite and fixed number; $R(i)$ denotes the service bit rate of class i , and should be a function of the amount of the clients. Then, the proportional differentiation is defined as follows:

$$\frac{R(1)}{R(2)} = \frac{R(2)}{R(3)} = \frac{R(3)}{R(4)} = \dots = \frac{R(n-1)}{R(n)} = r > 1 \text{ (constant)}$$

($R(1)$ is the highest class)

IV. PERFORMANCE EVALUATION

There are two parts of our experiments. First, we compare our SimFD scheme with other previous SFD scheme with different bandwidth capacities in terms of video similarity; the second experiments, we show the viability of our integrated end-to-end controlled DiffServ streaming system.

A. Comparison of Different Frame Discarding Schemes

We perform our experiments on a three-node topology as shown in Figure 3. The bandwidth of these two links is the same and varies as parameter in our simulation. The server node simulates as a real streaming server by reading a pre-processed trace file and then sends packets to the client node. The trace file format shows in Table I. We parse the video file and record essential information for video streaming including frame type, frame size and video bit rate into the trace file. Therefore, a simulated node can act as a real streaming server by reading this trace file. Then, the client in the simulation records received frames into another receiver file for us to calculate the video similarity between the original video and the received video.

In this simulation, we compare our SimFD scheme with the following two schemes: one is the scheme that considers only the frame type, so called "Type" scheme; the other considers not only the frame type but the inter-frame dependency as well, so called "TypeDepend" scheme. The streaming video bit rate is 300Kbits/sec, about 38 KB/sec. The bandwidth capacity varies from 10KB/sec to 40KB/sec. According to the results shown in Figure 4, we can see that our SimFD scheme always has better performance than the other two schemes, and has significant improvements under circumstances that network resources are constraint.



Fig. 3. SimFD Scheme Simulation Topology

B. Integrated End-to-End DiffServ System

In this experiment, we integrate our SimFD module with modified DiffServ module into an end-to-end controlled streaming system. In the original DiffServ module, there are edge-router functions and core-router functions, and has to keep per-flow states to achieve the DiffServ goal. In our system, both edge and core routers only need to do packet forwarding and not any information will be kept in routers. We use application-specific information and the TFRC report to dynamic control our traffic to carry out the

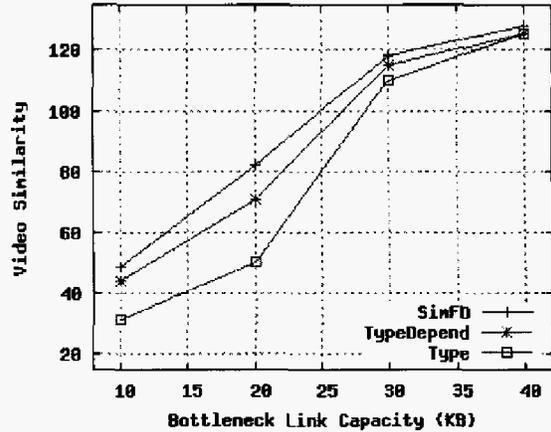


Fig. 4. Comparison of SimFD, TypeDepend, and Type Schemes

DiffServ goal. Figure 5 illustrates our simulation topology. There is one server, three clients with three different classes and one bottleneck links. All edge links have a capacity of 10Mb/sec and the bottleneck link has 1Mb/sec. Propagation delay is 1ms on edge links and 10ms on the bottlenecks link. The end-to-end controlled mechanism is based on the TFRC protocol and the class information. Total simulation time is 1200 seconds. Initially, one flow 0 (from S to 0 in Figure 5) is active, and the other flows get active with 400 seconds intervals. Figure 6 shows the instantaneous rate of flows. As a new flow joins in, the control module will dynamically adjust the rates according to TFRC reports and class information of each flow.

In order to show the viability of our system, we implement a streaming system shown in Figure 7. We use frame-based MPEG-4 video as the source, and each connection has two streams (part 1): one is the video stream and the other is the audio stream. The SdRate (part 2) shows the current sending bit rate of each streaming connection; this system also shows information of each client (part 3), including file streamed and service level, etc. Figure 7 and Figure 8 present the real video screens received by clients of lower class and higher class respectively. We can observe that higher-class clients have a much better quality than those in lower class.

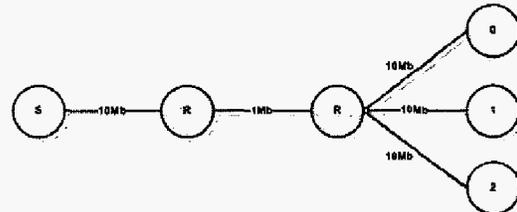


Fig. 5. DiffServ Simulation Topology

TABLE I
TRACE FILE FORMAT

Video Name	Length	Rate	# of Frames	# of I-Frames	# of P-Frames	# of B-Frames
------------	--------	------	-------------	---------------	---------------	---------------

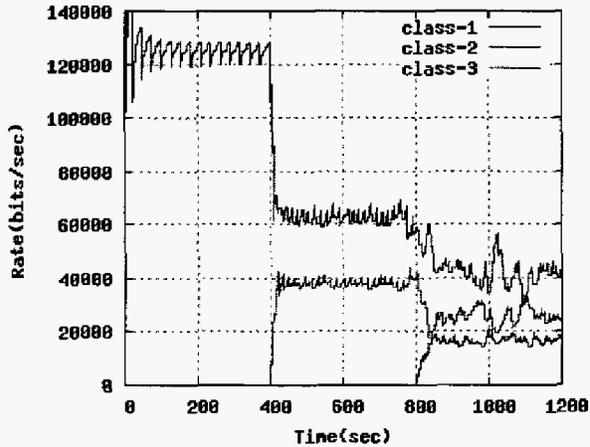


Fig. 6. Rate of Different Classes of Flows

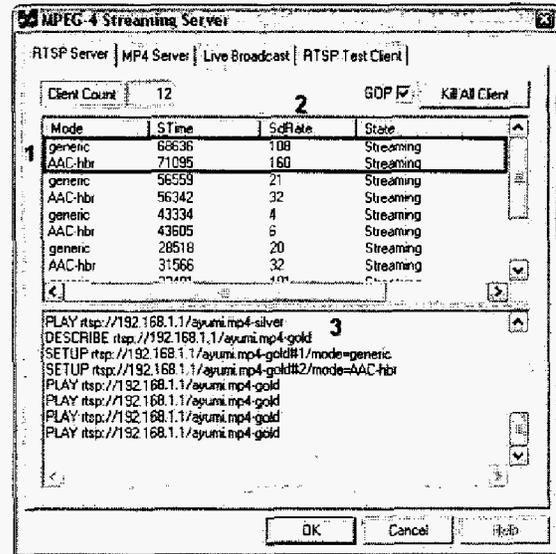


Fig. 7. Streaming Server

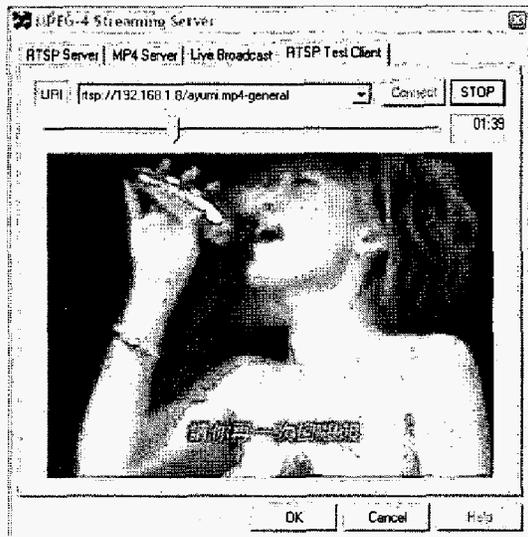


Fig. 8. Streaming Client (Lower Class)

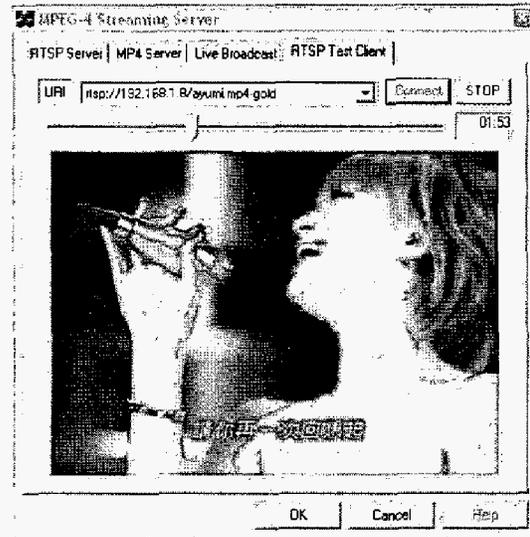


Fig. 9. Streaming Client (Higher Class)

V. CONCLUSION

Streaming system is very congestion-sensitive, and most of current proposed approaches have focused on controlling or monitoring the core networks, which usually requires significant upgrades on routers. We proposed a novel

approach to implement DiffServ-enabled streaming system by using similarity-based frame discarding algorithm and end-to-end controlled mechanism without any core state information. In addition, we have shown its viability in the real world.

REFERENCES

- [1]. J. Rexford and D. Towsley, "Smoothing Variable-Bit-Rate Video in an Internetwork," in *IEEE/ACM Transactions on Networking* 1999, p.202 – 215.
- [2]. W. Feng and J. Rexford, "A Comparison of Bandwidth Smoothing Techniques for the Transmission of Pre-recorded Compressed Video," in *Proc. IEEE INFOCOM*, pp 58-66, April 1997.
- [3]. M. Krunz and S.K. Tripathi, "On the Characteristics of VBR MPEG Streams," in *Proc. ACM SIGMETRICS*, pp. 192-202, June 1997.
- [4]. J. Rexford, S. Sen, J. Dey, W. Feng, J. Kurose, J. Stankovic, and D. Towsley, "Online Smoothing of Live, Variable-bit-rate Video," in *Proc. Workshop on Network and Operating System Support for Digital Audio and Video*, pp. 249-257, May 1997.
- [5]. J. Rexford and D. Towsley, "Smoothing Variable-Bit-Rate Video in an Internetwork," in *Proc. SPIE Symposium on Voice, Video, and Data Communications: Multimedia Networks: Security, Displays, Terminals, and Gateways*, November 1997.
- [6]. R. Braden, L. Zhang, S. Berson, S. Herzog, and S. Jamin, "Resource reSerVation Protocol (RSVP) version 1, Functional Specification," IETF, RFC 2205, September, 1997.
- [7]. S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang and W. Weiss, "An Architecture for Differentiated Services," RFC 2475, December 1998.
- [8]. S. Shenker, C. Partridge, and R. Guerin, "Specification of Guaranteed Quality of Service," IETF, RFC 2212, September 1997.
- [9]. B. Davie, A. Chamy, J.C.R. Bennett, K. Benson, J.Y. Le Boudec, et al., "An Expedited Forwarding PHB (Per-Hop Behavior)," IETF, RFC 3246, March 2002.
- [10]. Z.-L. Zhang, S. Nelakuditi, R. Aggarwal, and R.-P. Tsang, "Efficient Selective Frame Discard Algorithms for Stored Video Delivery across Resource Constrained Networks," *Real-Time Imaging, Special Issue on Adaptive Real Time Multimedia Transmission over Packet Switching Networks*, Vol. 7, No. 3, June 2001, pp. 255-273.
- [11]. S. Ramanathan, P.V. Rangan, and H.M. Vin, "Frame-Induced Packet Discarding: An Efficient Strategy for Video Networking," in *Proc. of 4th NOSSDAV*, Lancaster, England, November 1993.
- [12]. M. Hemy, U. Hengartner, P. Steenkiste, and T. Gross, "MPEG System Streams in Best-Effort Networks," *Proc. IEEE Packet Video 1999*, New York, April, 1999.
- [13]. C. Dovrolis and D. Stiliadis, "Relative Differentiated Services in the Internet: Issues and Mechanisms," in *ACM SIGMETRICS*, May 1999. Short Paper.
- [14]. C. Dovrolis, D. Stiliadis, and P. Ramanathan, "Proportional Differentiated Services: Delay Differentiation and Packet Scheduling," *IEEE/ACM Trans. on Networking*, 10(1):12-26, 2002.
- [15]. M. Handley, S. Floyd, J. Padhye, and J. Widmer, "TCP-friendly Rate Control (TFRC): Protocol Specification," IETF, RFC 3448, January 2003.



Jau-Yuan Chen received the B.S. degree in computer science from National Taiwan University (NTU) in 2003. He is currently a master student at NTU and is a member of Network Group in Communication and Multimedia Laboratory. His research interests include applied topics in streaming systems, differentiated services, and overlay networks.



Min-Way Hsu received the B.S. degree in Department of Computer Science and Information Engineering, National Taiwan University in 2004. He is currently working at CyberLink Corp. His research interests are streaming systems and wireless multimedia networks.



Cheng-Fu Chou received the M.S. and Ph.D. degree from University of Maryland, College Park, in 1999 and 2002, respectively. In 2002, he joined the Department Computer Science and Information Engineering, National Taiwan University. His current research interests are in wide-area network applications, distributed multimedia systems, heterogeneous wireless communication systems, and wireless sensor network and performance evaluation.