Multimed Tools Appl (2010) 47:207–224
DOI 10.1007/s11042-009-0414-5

A scalable and adaptive video streaming framework over multiple paths

Ivan Lee · Jong Hyuk Park

Published online: 21 October 2009

Abstract In this paper, we examine the frame loss probabilities for multiple-description coded video transmitted over independent paths. We apply an efficient multiple description coding technique for the analysis, and we investigate the impact of drifting error in terms of the probability of receiving freeze frames for reconstructed video. In order to improve the video delivery, an adaptive video coding scheme by adjusting the length of group-of-pictures is investigated in this paper. In addition, a scalable video streaming framework from client-server, centralized peer-to-peer, and decentralized peer-to-peer network topologies are examined. Analytical and experimental results based on Gilbert model are used to evaluate the performance of the proposed adaptive and scalable video streaming framework.

Keywords Video · Streaming · Peer-to-peer · Multi-path

1 Introduction

Video streaming delivers video over the Internet (or intranet) to end-users who are playing back the video content at real time. The video can either be pre-recorded or live streamed. The major challenge posed in video streaming is its timing requirement, and a packet will be considered lost if the transmission fails to meet a

This paper is an extended version of [12] and [13].

I. Lee
School of Computer and Information Engineering, University of South Australia,
Adelaide, South Australia, Australia
e-mail: Ivan.Lee@unisa.edu.au

J. H. Park (✉)
Department of Computer Science and Engineering, Seoul National University of Technology,
Nowon-gu, Seoul, South Korea
e-mail: parkjonghyuk1@hotmail.com

real-time constraint. This differs significantly from a play-after-download approach where transmission delays are not considered as a factor of error. Another challenge for video streaming is the best-effort nature of today's Internet, where a successful data delivery is not guaranteed.

To resolve these problems, numerous techniques on transmission feedback control, adaptive source encoding algorithm, efficient packetization, resource allocation, and error control coding have been proposed to improve the quality of video communication [21, 22]. In [3], Chow et al. proposed a variation of Gilbert model [24] where the loss parameters of a path depend on an application's transmission rate. Using this model, the authors optimized the load distribution among multiple paths to achieve an improved streaming quality. An optimization framework was proposed to minimize the aggregate distortion for multiple video streams transmitted over a shared communication channel [2]. A distributed video streaming from multiple servers to a single receiver was also studied in [1]. The servers independently partitioned the media packets based on the bandwidth information, such that the resulting video quality at the receiver was maximized.

Among different approaches proposed for video streaming to resolve the challenges on packet-based and best-effort Internet today, distributed streaming over collaborative peer-to-peer (P2P) overlay networks has attracted increased attentions recently from both research and industrial communities. Unlike conventional client/server infrastructures commonly used for content distribution networks (CDN), in P2P networks each node acts both as a client and as a server. This approach yields a high throughput and a good tolerance to loss and delay caused by network congestion [15]. P2P multimedia streaming and caching services also reduce initial delays for playback, and hence minimize jitters during playback [7]. Tran et al. [19] investigated application-layer multicast tree, and proposed ZIGZAG which possessed features on short end-to-end delay, low control overhead, efficient join and failure recovery, and low maintenance overhead. Another study which utilized advantage of the strong buffering capabilities of end hosts was oStream [5], which is a tree based overlay that was specifically designed for one-to-many on-demand media distribution.

Due to the heterogeneous nature of today's Internet, data transmission over a massive number of channels is difficult to control by a centrally managed solution at a low cost. Multimedia content in general has a highly time varying bandwidth requirement since media data are variable bit rates (VBR) in nature using modern coding techniques [6, 8]. In addition, streaming applications demands guaranteed delivery in order to meet specified temporal and spatial constrains. The services which provide guaranteed delivery in Network Multimedia System (NMS) was improved in the control management level of the host and the underlying network architectures [14]. Physically copying data is expensive for the guaranteed services for network transmission. The integrated processing loops for performing manipulation functions over a single common unit instead of performing them serially with the concept of Application Level Framing [4]. Media synchronization is also need to guarantee jitter-free playback requirements [16, 18]. The high bandwidth requirement and a real time delivery constraint are the two major challenges for streaming video. Driven by the goal of improving long-term system performance, dynamic resource allocation schemes with application specific adaptation capabilities are integrated in the solution. Previous work enables complex adaptations by use of general models of target systems [17]. With large variations in bandwidth requirements of multimedia content, rate adaptation [20] adjusts the bandwidth used by a transmission channel according to the existing network conditions.

Our work in this paper focuses on overcoming the limitations of conventional client-server streaming applications using a multi-path streaming infrastructure and spatial domain MDC [10] technique. In this paper, we evaluate the performance of different real-time streaming systems in terms of the frame loss rate of the reconstructed video, which reflects the probably of a frozen playback video frame at the receiver device. Different scenarios are examined by adjusting three factors: (1) number of MDC sub-streams, (2) length of group of pictures, and (3) transmission with aligned and unaligned I-frame. Based on the multi-path video streaming framework, we also propose an adaptive system which dynamically adjusts GOP lengths for each sub-stream according to the network conditions. Frame loss rate under different scenarios are investigated: (1) sub-dividing a GOP into multiple sub-GOPs, (2) multiple MDC sub-streams with different GOP lengths, and (3) an adaptive streaming system based on time-series projection of the frame loss rate. In addition, this paper also presents different approaches to offload the bottleneck traffic by applying the MDC compressed video over the peer-to-peer network infrastructure. Three different streaming infrastructures: client/server, centralized P2P streaming, and decentralized P2P streaming, are examined. To extend from traditional network performance analysis in terms of packet loss rate, this paper further investigates the impacts of the loss traffic to the reconstructed video quality due to drifting error. The frame loss rate, which indicates the un-reconstructable video frames at the receiver, is analyzed in this paper.

The remainder of this paper is organized as follows. In Section 2, the adaptive multi-path video streaming scheme is introduced, followed by its frame loss pattern analysis in Section 3. The studies of aligned and unaligned I-frames are investigated in Section 4. In order to improve video delivery and reduce frame loss rates, an adaptive source coding based on channel conditions is investigated in Section 5. In Section 6, a scalable video streaming framework with different network topologies is studied. Experiments are presented in Section 7.

2 Adaptive multi-path video streaming

MDC techniques are designed for path diversity. They encode media content into multiple independent descriptions, and these descriptions are also known as sub-streams. Once a sub-stream is received at the client end, the granules within the stream can be decoded. The overall quality of recovered content is depending on the number of successfully delivered sub-streams. The more sub-streams are received, the higher the reconstructed video quality can be achieved. MDC provides loss tolerance, and it is therefore beneficial for delay-sensitive, real-time streaming applications where data losses are highly disruptive.

An efficient MDC technique proposed in [10] is applied in this paper. Figure 1 illustrates the concept of the MDC codec, which sub-samples each video frame into multiple sub-frames over the spatial domain before encoding with an H.264 video encoder. The corresponding MDC decoder applies cubic-spline interpolation to reconstruct the missing sub-frames. Thus, the system is capable of reconstructing

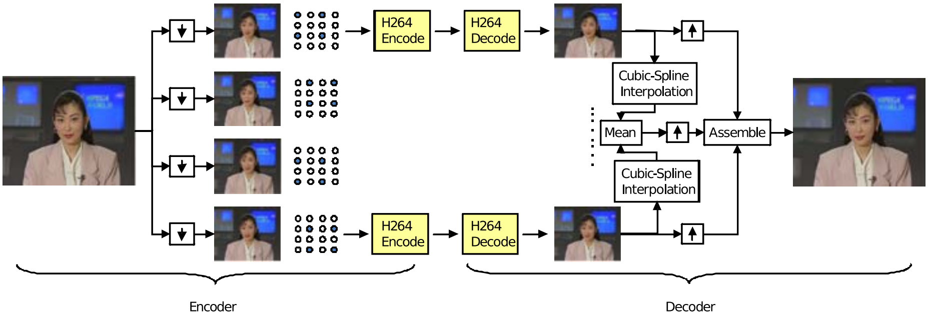


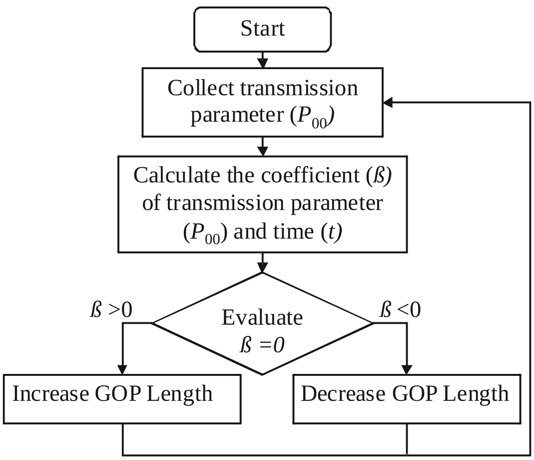
Fig. 1 An efficient MDC codec with spatial diversity

video with different quality levels depending on the number of MDC sub-streams received at the destination.

MDC is typically applied in scenarios where multiple nodes independently forward video content to the client node over physically connected networks. Prior research in [10] has proposed to apply MDC for a decentralized peer-to-peer streaming system, where the MDC sub-streams ($M_1 + M_2 + M_3 + \cdots + M_n$) received at server nodes are forwarded to the end user who request those sub-streams. Therefore, the more forwarding servers exist, the more paths can be used for transmission. The combination of the under utilized network bandwidth of multiple forwarding servers may give an overall broader bandwidth for MDC streaming, hence yielding a better reconstructed video quality at the receiver.

Prior research in [10] and [11] assume that MDC sub-streams are transmitted over channels with identical delay and throughput characteristics, and these parameters are unchanged for an entire streaming session. In this paper, we consider the nature of today's Internet where delay and throughput vary dynamically. An adaptive video streaming system is therefore required. We proposed an infrastructure which allows the receiver to report network conditions to the servers, and adaptively adjust the encoded bitstreams to improve the performance of video streaming. Figure 2

Fig. 2 Workflow of adaptive length of GOP



illustrates the work flow of the proposed infrastructure. The receiver monitors the network condition, and using time series projection to predict the loss pattern of subsequent video frames. The predicted value is delivered to the servers as a feedback to adjust the GOP lengths. In-depth analysis of the performance measure in terms of frame loss rate under different scenarios, as well as the details for the adaptive streaming infrastructure, are explained under the subsequent sections.

3 Frame loss analysis for reconstructed video

The video stream is consist of group of pictures. In this paper, we apply H264 baseline profile which uses two types of frames: Inter (I)-frame and Predictive (P)-frame. I-frames are encoded independent of prior frames. P-frames are encoded with respect to a prior I-frame or P-frame, where motion compensation techniques are applied for improving the compression efficiency. A group of frames that starts with an I-frame, followed by a set of P-frames and ends before the next I-frame is called a Group of Pictures (GOP). Losing an I-frame or a P-frame will result in distortion of the following frames within the same GOP. This is known as the drifting error.

In this paper, we simulate the packet delivery of the video bitstreams using Gilbert Model [24], which is a two-state Markov chain indicating a success state and a failure state for packet delivery. To simplify the analysis, we assume that each packet contains one encoded video sub-frame. Let S_0 (good) denote good state when the IP network packages are received correctly and timely, and S_1 (bad) denotes bad state when the packages are lost. The probabilities of the network transition from S_0 to S_1 and from S_1 to S_0 are denoted as P_{01} and P_{10}, respectively. The probabilities of staying in the same state are denoted as P_{00} for state S_0 and P_{11} for state S_1. Steady state analysis shows that the overall probabilities of good state P_{S0} and bad state P_{S1} are:

$$P_{S0} = \frac{P_{11} - 1}{P_{00} + P_{11} - 2} \tag{1}$$

$$P_{S1} = \frac{P_{00} - 1}{P_{00} + P_{11} - 2} \tag{2}$$

Let T denote the length of GOP, and M denote the number of sub-streams. At anytime i, the probability of a good frame transmission is $P_{S0}P_{00}^{i-1}$, where $i = 1, 2, \ldots, T$. The complement value, ρ_i, reflects the probability of all combinations with a frame loss before or at time i, and $\rho_i = 1 - P_{S0}P_{00}^{i-1}$,

$$\epsilon_{aligned} = \frac{1}{T}\sum_{i=1}^{T}\left(1 - P_{S0}P_{00}^{i-1}\right)^{M} \tag{3}$$

Substitute (1) into (3), the mean frame loss rate can be written in terms of transition probabilities P_{00} and P_{11}, as shown in (4).

$$\epsilon_{aligned} = \frac{1}{T}\sum_{i=1}^{T}\left(1 - \frac{(P_{11} - 1)\, P_{00}^{(i-1)}}{P_{00} + P_{11} - 2}\right)^{M} \tag{4}$$

3.1 MDC multi-path frame loss rate on different number of sub-streams

In this section, we study the frame loss rate of different number of MDC sub-streams. Let integers M_1 and M_2 denote the numbers of MDC sub-streams, where $1 \leq M_1 \leq M_2$. Since $0 \leq 1 - P_{S0}P_{00}^{i-1} \leq 1$ and $M_1 < M_2$, $0 \leq \left[1 - P_{S0}P_{00}^{i-1}\right]^{M_2} \leq \left[1 - P_{S0}P_{00}^{i-1}\right]^{M_1} \leq 1$. Let $\Delta\epsilon$ represent the difference of the frame loss rate between M_1 sub-streams and M_2 sub-streams, and

$$\Delta\epsilon = \epsilon_1 - \epsilon_2 \geq 0 \tag{5}$$

Thus, the frame loss rate is a decreasing function of number of MDC sub-streams. The more MDC sub-streams exist, the less frames will drop, and hence the better video quality.

3.2 MDC multi-path frame loss rate on different length of GOP

In this section, we study the frame loss rate of different lengths of MDC GOPs. Let $X_T = 1 - P_{S0}P_{00}^{T-1}$, then (3) can be represent as $\epsilon_{aligned} = \frac{1}{T}\left(x_1^M + x_2^M + \cdots + x_T^M\right)$. Let two integers T_1 and T_2 denote any two different lengths of GOP, and $1 \leq T_1 < T_2$. Then, $\Delta T = T_2 - T_1 > 0$ Since $M \geq 1$, $T \geq 1$, and $0 \leq x = 1 - P_{S0}P_{00}^{T-1} \leq 1$, the following relationship holds: $x_1^M \leq x_2^M \leq \cdots \leq x_T^M$. Let $\Delta\epsilon_T$ represent the difference of the frame loss rate between T_1 and T_2 GOP lengths. Let $\Delta w = w_2 - w_1$, where $w_1 = x_1^M + x_2^M + \cdots + x_{T_1}^M$, $w_2 = x_1^M + x_2^M + \cdots + x_{T_2}^M$, and

$$\Delta\epsilon_T = \epsilon_{T_2} - \epsilon_{T_1} = \frac{w_2}{T_2} - \frac{w_1}{T_1} = \frac{\frac{\Delta w}{\Delta T} - \frac{w_1}{T_1}}{\frac{(T_1+\Delta T)T_1}{T_1 \Delta T}} \tag{6}$$

To determine whether (6) is positive or negative, we can investigate the following relationship:

$$\frac{w_1}{T_1} \leq X_{T_1}^M \leq X_{T_1+1}^M \leq \frac{\Delta w}{\Delta T} \tag{7}$$

Taking the relationship expressed in (7) into (6), we observe that $\Delta\epsilon_T = \epsilon_{T_2} - \epsilon_{T_1} > 0$. Therefore, the frame loss rate is an increasing function of the GOP length: the longer the GOP length, the higher probability of frame loss will appear.

4 MDC streaming with aligned and unaligned I-frames

In the multi-path video streaming infrastructure proposed in this paper, the receiver can reconstruct the output video upon receiving any number of MDC substreams. Only if all sub-streams are lost, the video frame cannot be reconstructed. In this paper, we investigate the scenarios where the MDC sub-streams encode I-frames at different points of time.

To evaluate the performance of systems using aligned and unaligned I-frames, we need to determine the position of the I-frame in each channel. Let T denote the processing period, which is equal to the length of one GOP, let M denote the number of MDC sub-streams and d_n denote the offset in n-th path which equal to the period before the I-frame arrival. The GOP transmission time i_n can be presented as $i_n = (T - d_n + i) mod T$, where $i = 0, 1, 2, \ldots, T-1$. Then, the average frame loss rate of M paths during the T period can be represented as:

$$\epsilon_{unaligned} = \frac{1}{T} \sum_{i=0}^{T-1} \prod_{n=1}^{M} \left(1 - P_{S0} P_{00}^{(T-d_n+i) mod T}\right) \tag{8}$$

4.1 MDC multi-path frame loss rate on homogeneously distributed and aligned I-frame

In this section, the special scenario of homogeneous I-frame offset was studied. As the number of substreams is theoretically infinite large, and GOP length has a finite size, usually $M \geq T$. When the number of sub-streams M equals the GOP length T ($M = T$), and offset $d_n = n$, we can get a square matrix where I-frames are homogeneously scattered over MDC sub-streams. At any time i, we can always find a path has offset $n = i$, and the frame loss rate for this scenario is:

$$\epsilon_{homo} = \frac{M}{T} \left(1 - P_{S0} P_{00}^{0}\right) \ldots \left(1 - P_{S0} P_{00}^{M-1}\right) = \left(1 - P_{S0} P_{00}^{0}\right) \ldots \left(1 - P_{S0} P_{00}^{M-1}\right) \tag{9}$$

Let $\alpha_i = \left(1 - P_{S0} P_{00}^{0}\right)$. Equations (5) and (9) can be represented as

$$\epsilon_{aligned} = \frac{1}{M} \left[\alpha_0 + \alpha_1 + \cdots + \alpha_{M-1}\right] \tag{10}$$

$$\epsilon_{homo} = (\alpha_0 \alpha_1 \ldots \alpha_{M-1})^{1/M} \tag{11}$$

The values for $\alpha_0, \alpha_1, \ldots, \alpha_{M-1}$ is a set of positive real numbers, M is a positive integer. Since arithmetic mean of the set α_i is represented as $\epsilon_{aligned}$, and the geometric mean the set α_i is represented as ϵ_{homo}, as $\epsilon_{aligned} \geq \epsilon_{homo}$. Thus, distributing I-frames homogeneously over multiple sub-streams can reduce the frame loss rate for the reconstructed video at the receiver.

In the Internet environment, we can theoretically get infinity transmission channels. Meanwhile, the content that can be decoded depends on the active frames received from each channel at any point of time. As Fig. 3 shows, under the same network conditions, when MDC sub-streams consist of unaligned I-frames, the total number of reconstructable video frames will vary. Better video output can be obtained by properly adjusting the I-frame positions for each MDC substreams. The overall output is determined by network transmission rate, number of transmission paths, GOP length, as well as the I-frame position in each substream.

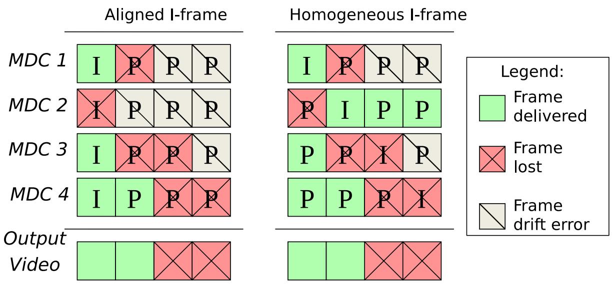


Fig. 3 Comparison of frame loss patterns for multi-path streaming with aligned and homogeneous unaligned I-frames

5 Adaptive compression according to channel condition

Dynamic resource allocation is a critical component in the system expected to deliver reliable performance while subject to the unpredictable workload. Instability and heterogeneous characteristics of MDC streaming channels may result in a severe quality impact for real-time media transmission. We propose a solution that integrates source coding techniques and network adaptation schemes to construct a distributed media streaming system. The video content processing components (encoder / decoder) are consisting of multiple control models. The FGS (fine granularity scalable) video coding [23] workflow control models can collaborate with transmission condition of any independent sub-channels. The MDC streaming frame loss rate depends on both the transmission conditions and encoding factors (length of GOP). The real time performance of video streaming is based on the overall consolidated result from all sub-channels. From (11), the network transmission rate (P_{00} and P_{S0}) and the number of sub-streams (n) are part of the network infrastructure performance metrics and they are mostly are based on the physical resources and devices. Adjustments over those devices involve more hardware and devices, which are expensive and difficult to give the prompt response according to the real time media transferring rate fluctuation. Therefore, the GOP length is the nominated component for the adaptive solutions. It can be adjusted according to real time streaming condition change.

5.1 Frame loss rate in single channel with multiple lengths of GOP

In this section, we study the scenario where each channel carrying an MDC sub-streams is encoded with a different GOP length (G_y, y is the number of GOP in that sub-stream) when the network condition P_{00} is constant. Let Y denote the total number of GOP for that single channel, in case there is an offset (d_n) of the first I-frame in the sub-stream transmission, from (3), frame loss rate of that sub-stream ϵ_n is:

$$\epsilon_n = \frac{1}{T}\sum_{i=1}^{T}\left(1 - P_{S0}P_{00}^{i-d_n-\sum_{y=1}^{Y-1}(G_{y-1}-1)}\right) \tag{12}$$

For the network condition change scenario, at any time i, we have (P_{00i}) to indicate the transmission states, (12) can be expressed in the following form, $\epsilon_n = \frac{1}{T}\sum_{i=1}^{T}\left(1 - P_{S0}\prod_{i=1}^{T-d_n-f(i)-1}P_{00i}\right)$, where $f(i)$ indicates the location of the closest I-frame occurred before or at frame i. As shown in Section 3.2, the frame loss rate is an increasing function of the GOP length. Let ϵ_y denote the frame loss rate of any GOP within one sub-stream, and $G_y \le T$, then $\epsilon_y \le \epsilon_n$ when the network condition remains the same.

$$\begin{aligned}\epsilon_y &= \frac{1}{G_y}\sum_{i=1}^{G_y}\left(1 - P_{S0}\prod_{i=1}^{G_y-d_n-f(i)-1}P_{00i}\right)\\ &\le \frac{1}{T}\sum_{i=1}^{T}\left(1 - P_{S0}\prod_{i=1}^{T-d_n-f(i)-1}P_{00i}\right) = \epsilon_n\end{aligned} \tag{13}$$

The overall sub-channel frame loss rate can be presented as:

$$\frac{1}{T}\left(G_1\epsilon_1 + G_2\epsilon_2 + \cdots + G_y\epsilon_y\right) \leq \frac{1}{T}\left(G_1 + G_2 + \cdots + G_y\right)\epsilon_n = \epsilon_n \tag{14}$$

During the total transmission period i, the frame loss rate of single channel with different GOP lengths (G_y) will be equal to or less than the identical GOP length across all sub-streams.

5.2 Dynamic resource allocation trigger by sub-channel transmission time series projection

The classic time series forecast model is used to reduce the computing complexity when people estimate each sub-channel transmission condition. According to the projected network transferring rate, the adjustment on the GOP length is determined for the upcoming video compression and transmission. Depending on the services level and the broadcasting environment, the GOP length is adjusted to meet the multimedia application requirement more effectively and to transfer video content more efficiently. We use projection of the network transferring rate P_{00} as the integrated factor of the overall network conditions, and the estimated sub-stream frame loss rate can be calculated from (12). By solving (12) with the projected value of transmission rate, we can determine the adjusted GOP length. The frame loss rate will be regarded as the overall streaming quality factor to drive the transmission. A linear forecast model is applied and the general form of the model can be expressed as $P_{00} = \beta t + c + e$, where P_{00} is a transition state probability, t represents the frame counter, β is the gradient of the linear function, c is an arbitrary constant, and e is the white noise.

To simplify the calculations on the adjustment of the GOP length, we analyze β to determine the adjustment. When $\beta < trigger$, which indicate the worse network transmission, we decrease the GOP length. When $\beta > trigger$, the GOP length will be increased. And when $\beta = trigger$, the GOP length will remain unchanged.

6 From centralized to decentralized video streaming

Conventional video streaming applications using the client/server architecture face the scalability constraints. In this section, we present different video streaming approaches, including client/server, centralized P2P streaming, and decentralized P2P streaming.

6.1 Client/server streaming

As illustrated in Fig. 4a, a centralized server is responsible for serving all the client requests. The aggregation link thus represents the bottleneck since the entire video traffic to all clients is carried on a single link. That is, the client/server model faces a scalability problem when the number of clients increases.

Two popular approaches are used for today's Internet to resolve scalability issues: (1) using cache engines, or (2) applying multicast. Cache engines are typically located at the edge of the network, and they temporally store duplicated data traffic designated to different clients. Thus, a download requests may be retrieved from a

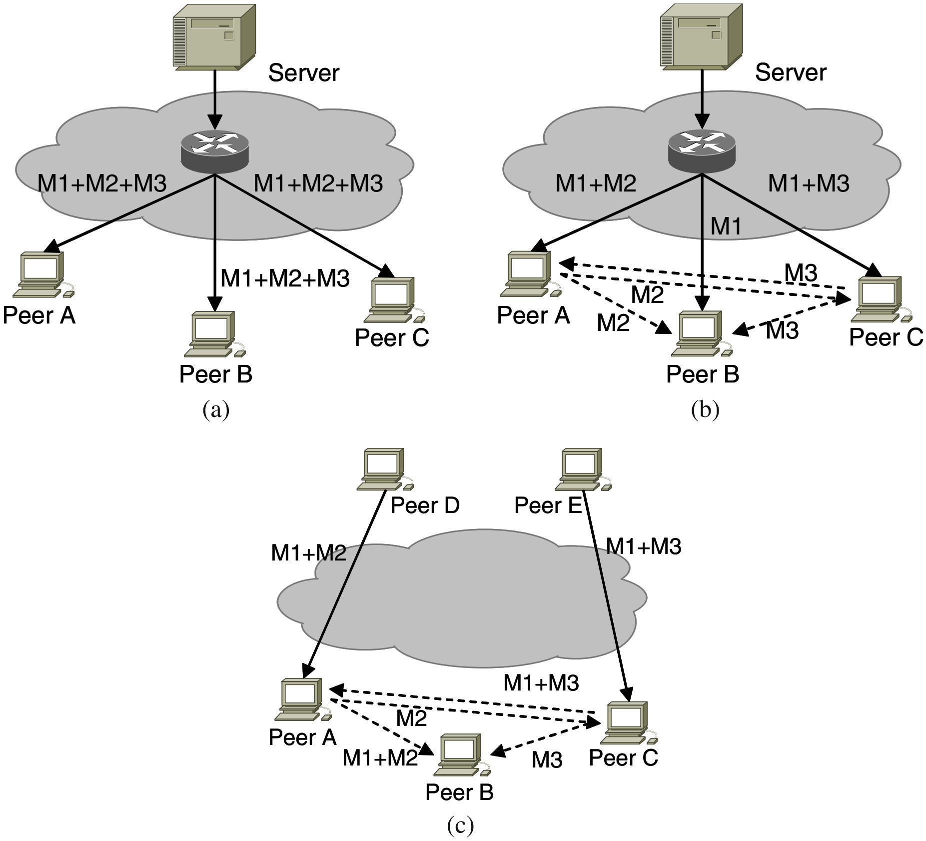


Fig. 4 Distributed video streaming with different network topologies. **a** Client/server streaming. **b** Centralized P2P streaming. **c** Decentralized P2P streaming

close-by cache engine instead of the far-away server, thus decreasing the download time and reducing the load on the bottleneck link. However, since the video traffic in general demands a much higher bandwidth than conventional text and image based traffic (such as the webpage applications), the storage space required on the cache engines will be substantially increased and making this approach impractical for video applications. Multicast eliminates duplicated traffics transmitted over network links, and it is best for broadcasting applications. However, the multicast may not fits the requirements for video on-demand applications where end-users may not download video simultaneously. In addition, multicast is not widely deployed on today's Internet backbone, and the cost for upgrading the Internet backbone also represents a barrier for its popularity.

In this paper, we choose multiple-description coding to compress video signals. Different P2P infrastructures will be examined. In the subsequent sections, an overview of centralized P2P streaming, and decentralized P2P streaming, will be discussed. The cache engine and the multicast techniques could both be applied to these P2P streaming models; however, the study is beyond the scope of this paper. This paper will focus on the comparisons without caching or multicast.

In general, the probability of receiving n frames followed by a lost frame is $P_{S0} \cdot P_{00}^{n-1} \cdot P_{01}$, and the frames loss rate for receiving n frames followed by a lost frame is $\frac{T-n}{T} \cdot P_{S0} \cdot P_{00}^{n-1} \cdot P_{01}$, where T is the length of the GOP. The mean frame loss rate, ε_{SDC} can be represented as

$$\varepsilon_{SDC} = 1 - \frac{(P_{11} - 1)\left(P_{00}^{T} - 1\right)}{T\,(P_{00} + P_{11} - 2)\,(P_{00} - 1)} \tag{15}$$

6.2 Centralized P2P streaming

Applications over the P2P framework offer promising alternatives to resolve many problems existing in the conventional client/server framework, with a cost of centralized manageability. Besides, P2P nodes do not serve with high availability, which may result in severe service degradation, especially for real-time applications such as video streaming.

In this paper, we propose a centralized P2P streaming framework with centralized manageability while offloading the traffic from the bottleneck bandwidth to under-utilized access networks. Centralized P2P streaming is controlled with a management server, which keeps track of the peer topology and manages the transmission of multimedia download sessions. The management server is in general a highly reliable source and it is used to deliver prioritized data; the P2P nodes are less reliable and therefore it is responsible to forwarding low prioritized data. Centralized P2P streaming was designed with layered video codec [9], but the same arrangement may be applied with MDC. This arrangement guarantees a minimum quality level with the multimedia data delivered from the reliable server, and a best-effort enhancement from the unreliable peers.

As illustrated in Fig. 4b, centralized P2P streaming offloads part of the video traffic from the bottleneck link to under-utilized P2P networks. In the client/server streaming framework illustrated in Fig. 4a, the entire MDC substreams (M1+M2+M3) are transmitted to each individual end-user. For centralized P2P streaming, multiple peers may forward the missing MDC substreams. The server may arrange to transmit different MDC-streams into multiple peer-nodes, and these peer-nodes are acting as the forwarding peers. Therefore, there are multiple forwarding peers within a P2P cluster. The proposed framework is illustrated in Fig. 4b.

Centralized P2P streaming offloads part of the MDC bitstreams to the P2P network. Let M denote the number of MDC substreams, and each substream with bitrate r_i where $i \in \{1 \ldots M\}$, $r_i \in R$ and $r_i \geq 0$. Let n denote the number of peers within a P2P cluster, $n \in Z^+$ where Z^+ is the set of all positive integers. There is one forwarding peer for each P2P cluster. The bottleneck bandwidth requirement for client/server streaming framework is $n \sum_{i=1}^{M} r_i$, and bottleneck bandwidth for centralized P2P streaming framework is $\sum_{i=1}^{M} r_i + (n-1)r_1$. Thus, the bandwidth reduction by upgrading from client/server streaming to centralized P2P streaming is $(n-1)\sum_{i=2}^{M} r_i$. Since $(n-1) \geq 0$ and $r_i \geq 0, \forall i$. The bandwidth requirement for centralized P2P streaming is always equal to or less then that of client/server streaming.

In centralized P2P streaming, the server and the peer nodes possess different loss attributes. The server is a reliable source and the peer-nodes are non-reliable sources. Hence the server should have a higher good state probability compares to the peer-nodes. To simplify the analysis, we assume that the peers have identical loss attributes, hence identical steady state probabilities will be used for all peers. In this section, we need to separate the steady state probabilities in (1) and (2) for servers and peers. For the server, the two states of Gilbert model are denoted as $(S0, s)$ (good) where the packets are received correctly and timely, and $(S1, s)$ (bad) where the packets are assumed to be lost. The state transition probability from $(S0, s)$-to-$(S1, s)$ and $(S1, s)$-to-$(S0, s)$ are denoted as $P_{01,s}$ and $P_{10,s}$, respectively. The probability of remaining in the same state are denoted as $P_{00,s}$ and $P_{11,s}$ where $P_{00,s} = 1 - P_{01,s}$ and $P_{11,s} = 1 - P_{10,s}$. $P_{x,s} \in R$ and $0 < P_{x,s} < 1, \forall x \in \{00, 01, 10, 11\}$. Similarly, the steady state probabilities and the transmission probabilities for the peers are denoted as $P_{S0,p}, P_{S1,p}, P_{x,p} \in \mathbf{R}^+$ and $0 < P_{x,p} < 1, \forall x \in \{00, 01, 10, 11\}$, respectively. The steady state probabilities of successful and unsuccessful packet transmissions from the server are $P_{S0,s} = \frac{P_{11,s}-1}{P_{00,s}+P_{11,s}-2}$ and $P_{S1,s} = \frac{P_{00,s}-1}{P_{00,s}+P_{11,s}-2}$. Similarly, the steady state probabilities of successful and unsuccessful packet transmissions from the peers are $P_{S0,p} = \frac{P_{11,p}-1}{P_{00,p}+P_{11,p}-2}$ and $P_{S1,p} = \frac{P_{00,p}-1}{P_{00,p}+P_{11,p}-2}$. For centralized P2P streaming with M MDC substreams transmitted over M independent paths, output video can be reconstructed upon receiving any substream. With MDC, the video frames cannot be reconstructed only if all substreams are lost. Let γ_i and ρ_i denote the probability of transmitting $i-1$ frames followed by a frame drop from the server and the peer nodes, respectively. Therefore, γ_i and ρ_i are

$$\gamma_i = \begin{cases} P_{S1,s} & \text{when } i = 1 \\ P_{S0,s} P_{00,s}^{i-2} P_{01,s} & \text{when } i > 1 \end{cases} \tag{16}$$

$$\rho_i = \begin{cases} P_{S1,p} & \text{when } i = 1 \\ P_{S0,p} P_{00,p}^{i-2} P_{01,p} & \text{when } i > 1 \end{cases} \tag{17}$$

For each MDC stream, a lost frame will make the remaining frames unreconstructable, which is known as the drifting error. The probability that at least one channel transmits the video at and before $i = k$, and all the channels have lost a frame at and before $i = k + 1$ is

$$\left(\sum_{i=1}^{k+1} \rho_i\right)^{M-1} \left(\sum_{i=1}^{k+1} \gamma_i\right) - \left(\sum_{i=1}^{k} \rho_i\right)^{M-1} \left(\sum_{i=1}^{k} \gamma_i\right) \tag{18}$$

Let $w(n) = \sum_{i=1}^{n} \rho_i$ and $v(n) = \sum_{i=1}^{n} \gamma_i$, the mean frame loss rate ε_{CP2P} for the multi-path transmission scheme can be expressed as

$$\varepsilon_{CP2P} = \frac{1}{T}\left[\sum_{k=1}^{T} x^{M-1} y\right] \tag{19}$$

where $x = P_{S1,p} - P_{S0,p}\left(P_{00,p}^{k-1} - 1\right)$ and $y = P_{S1,s} - P_{S0,s}\left(P_{00,s}^{k-1} - 1\right)$. By substituting steady state probabilities into (19), the frame loss rate for multi-path transmission can be simplified as

$$\varepsilon_{CP2P} = \frac{1}{T}\sum_{k=1}^{T} \alpha^{M-1}\beta \tag{20}$$

where $\alpha = 1 - \frac{(P_{11,p}-1)P_{00,p}^{k-1}}{P_{00,p}+P_{11,p}-2}$ and $\beta = 1 - \frac{(P_{11,s}-1)P_{00,s}^{k-1}}{P_{00,s}+P_{11,s}-2}$.

6.3 Decentralized P2P streaming

Centralized P2P streaming partially offloads video traffic from the bottleneck link; it extends the threshold of the client size but it is still constrained to a certain population. In other words, it is not fully scalable. Decentralized P2P streaming is the next phase after centralized P2P streaming, which aims to resolve the scalability issue when centralized P2P streaming faces the critical point where traffic congestion occur on the bottleneck traffic.

Figure 4c illustrates the scenario of the decentralized P2P streaming. Decentralized P2P streaming assumes that there are a significant number of peers receive the MDC substreams from the previous phase (i.e. centralized P2P streaming phase). When the bottleneck link observes traffic loss, any new peer requesting the video will be joining the decentralized P2P streaming framework, where the entire video substreams are transmitted from the P2P network. In decentralized P2P network, a central server is no longer required to guarantee the video delivery. Thus, decentralized P2P streaming is highly scalable with the trade-off in the guaranteed service. The reliability of the video stream, in terms of the frame loss probability, will be the main focus of discussion in this paper.

Consider a video sequence divided into M MDC substreams, and transmitted over M-independent paths. To simplify the analysis, assume each path has an identical loss attribute, and therefore contains identical state transition parameters in the Gilbert model.

For multiple-description coding with M independent substreams, output video can be reconstructed from any substream. Under MDC, the video frames cannot be reconstructed only if all streams are lost. Let T denotes the GOP period, and ρ_i denotes the probability of successfully transmitting $i-1$ frames followed by a frame drop. Then, ρ_i is represented by the following sequence:

$$\rho_i = \begin{cases} P_{S1} & \text{when } i = 1 \\ P_{S0} P_{00}^{i-2} P_{01} & \text{when } i > 1 \end{cases} \tag{21}$$

The probability that one channel loses a frame before or at $i = k$ is $\sum_{i=1}^{k} \rho_i$, and the probability that all the channels lose a frame before or at $i = k$ is $\left(\sum_{i=1}^{k} \rho_i\right)^M$. Similarly, the probability that all the channels have lost a frame before or at $i = k+1$ is $\left(\sum_{i=1}^{k+1} \rho_i\right)^M$. The probability that at least one channel transmits the video at and before $i = k$, and all the channels have lost a frame at $i = k+1$ is

$$\left(\sum_{i=1}^{k+1} \rho_i\right)^M - \left(\sum_{i=1}^{k} \rho_i\right)^M \tag{22}$$

The mean frame loss rate for the M multi-channel transmission scheme is therefore

$$\varepsilon_{MDC} = \frac{1}{T} \sum_{k=1}^{T} \left[P_{S1} - P_{S0} \left(P_{00}^{k-1} - 1 \right) \right]^M \tag{23}$$

Finally, by substituting (1) and (2) into (23), we obtain the frame loss rate for multi-path transmission ε_{MDC}

$$\varepsilon_{MDC} = \frac{1}{T}\sum_{k=1}^{T}\left[1 - \frac{(P_{11}-1)\,P_{00}^{k-1}}{P_{00}+P_{11}-2}\right]^{M} \tag{24}$$

7 Experiments

In this paper, MDC video streams consist of both I-frame and P-frames as described in Section 3. The transition state probability P_{00} decreases from 0.9 to 0.5 as illustrated in Figs. 5 and 6. The experiment is based the system discussed in Section 5.2 with initial GOP length equals 15, and at each transition the GOP length may be adjusted by a length of 5. We observe that the frame loss rate resets to 0.3 periodically, which indicates the beginning of GOPs (i.e. I-frames). For the fixed approach, we observed that the peak frame loss rates increase with P_{00} decreases. For the adaptive approach, GOP lengths are adjusted after changes of P_{00} are observed. We noticed that between each P_{00} transitions, multiple I-frames are observed. This is because the linear projection model is evaluated based on ten historical frames in this experiment. The proposed method resets GOP with different lengths multiple times before stabilization.

Figure 7 shows the experimental results for centralized P2P streaming, which examines the frame loss rate by evaluating (20) with $P_{00,p} = 0.9$, $P_{11,p} = 0.2$, $P_{00,s} = 0.95$, and $P_{11,s} = 0.15$. These parameters indicate that the server is more reliable than each individual peers. We observe that the frame loss rate increases with a larger GOP length due to drifting error.

The frame loss rate of the single video stream transmission and the MDC transmission using the Gilbert model are shown in (15) and (24), respectively. The frame loss

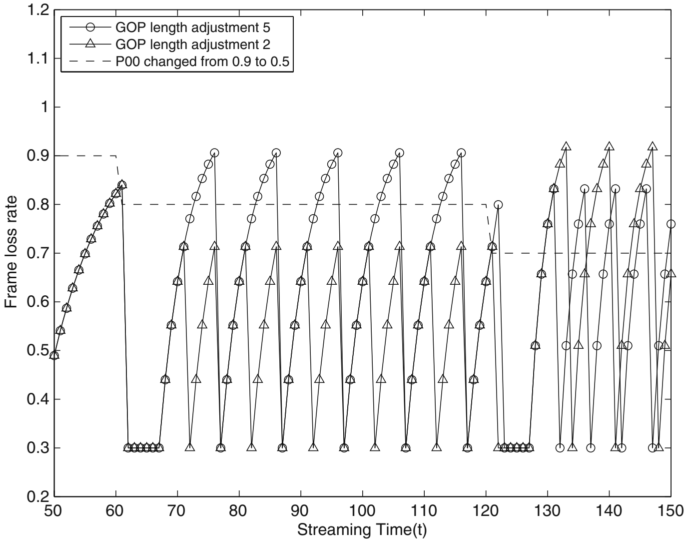


Fig. 5 Adaptive GOP length with different adjustment step sizes

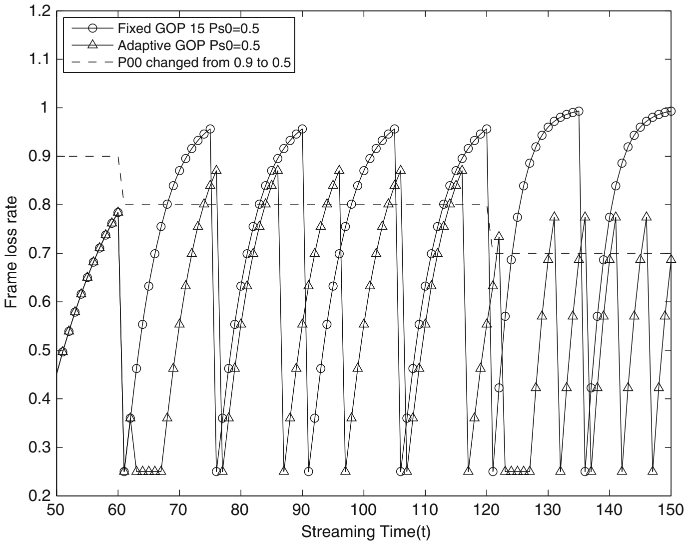


Fig. 6 Adaptive GOP length versus fixed GOP length in multi-channels

rate is a function of length of GOP (T), number of MDC substreams (M), and Gilbert model transition probabilities (P_{01} and P_{10}). To highlight the performance gain of multi-path transmission using MDC over SDC, identical Gilbert model transition probabilities are chosen for Fig. 8a to b, with $P_{00} = 0.9$ and $P_{11} = 0.2$. As shown in Fig. 8a, the frame loss rate increases with T, due to a higher drifting error associate with a larger T value. We observe that under all T values, MDC outperforms single stream transmission (the lower the frame loss rate, the higher perceived video

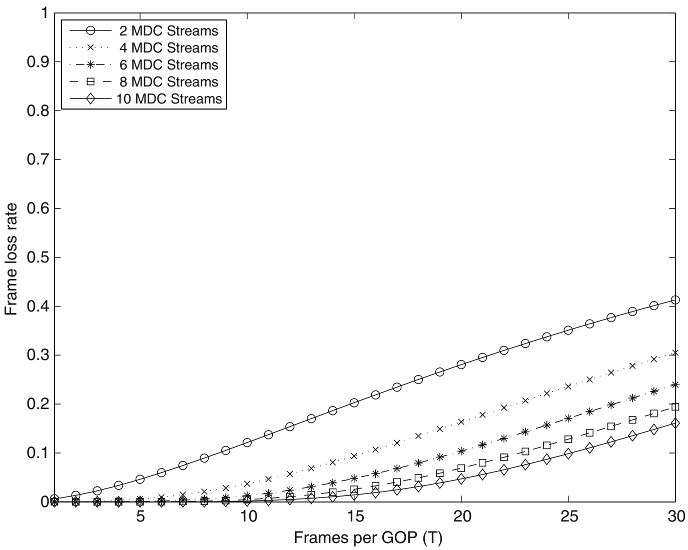


Fig. 7 Frame loss rates for centralized P2P streaming

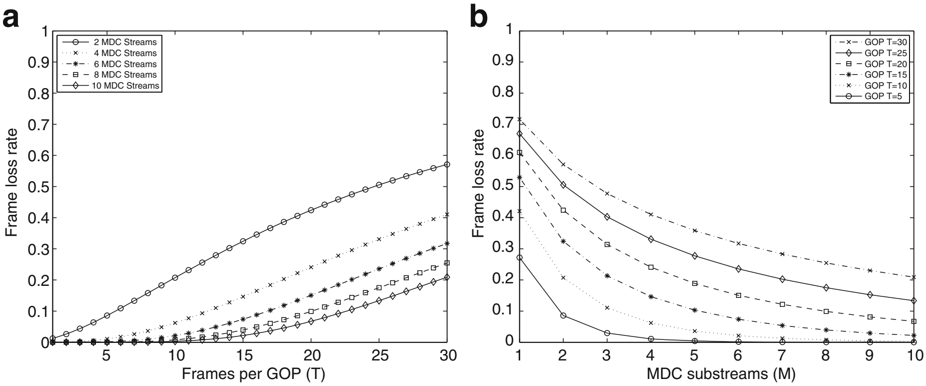


Fig. 8 Frame loss rates for decentralized P2P streaming (**a**, **b**)

quality). Figure 8b examines the number of MDC substreams and their impacts to the frame loss rate. We observed that the higher the number of MDC substreams, the lower the frame loss rate.

8 Conclusions

In this paper, we studied streamed MDC video over multiple paths. Adaptive GOP length in the sub-channel transmission was used to improve the reconstructed video quality over varying channel conditions. MDC with unaligned I-frames were also investigated. The frame loss rate analysis with experiments based on Gilbert Model is examined. Factors of MDC, such as different multiple description levels (GOP length, number of multiple sub-streams and offset of I-frame in sub-stream), and different Gilbert model transition state probabilities, were evaluated. We proved that the frame loss rate of MDC streaming over multiple paths is sensitive to the GOP lengths and the number of MDC streams. We observed that loss rates increase when the number of GOP increases, and loss rates decrease when the number of MDC sub-streams increases. We also observed that MDC with homogeneous unaligned I-frame distribution in transmissions performed better than the aligned cases in our analysis.

We investigated adaptive GOP lengths and their impacts on the frame loss rate. Scenarios under observation include deteriorated network conditions, different adjustment step sizes, as well as multi-path transmissions. We observed that the proposed adaptive solution outperforms the fixed solution in terms of lowering the frame loss rates.

The frame loss rates for different network topologies are also investigated in this paper. By adaptively escalating the network topology from client/server, to centralized P2P, and then to decentralized P2P phases, the bottleneck traffic can gradually be offloaded to under-utilized P2P networks. From the experimental results, we observed that increasing the MDC substreams reduced the frame loss rate; increasing the GOP length increased the frame loss rate; increasing the good state transition probabilities and decreasing the bad state transition probabilities also yielded a lower frame loss rate.

Acknowledgements This research was supported in part by the MKE (Ministry of Knowledge Economy), Korea, under the ITRC (Information Technology Research Center) Support program supervised by the IITA (Institute of Information Technology Advancement) (IITA-2008-C1090-0801-0020).

References

1. Chakareski J, Frossard P (2005) Distributed streaming via packet partitioning. In: Proc. of IEEE international conference on multimedia and expo, pp 1529–1532
2. Chakareski J, Frossard P (2006) Rate-distortion optimized distributed packet scheduling of multiple video streams over shared communication resources. IEEE Trans Multimedia 8(2): 207–218
3. Chow AL, Golubchik L, Lui JC, Lee W-J (2005) Multi-path streaming: optimization of load distribution. Perform Eval 62:417–438
4. Clark DD, Tennenhouse DL (1990) Architectural considerations for a new generation of proctocals. In: SIGCOMM'90, Philadelphia, pp 200–208
5. Cui Y, Li B, Nahrstedt K (2004) oStream: asynchronous streaming multicast in application-layer overlay networks. IEEE J Sel Areas Commun 22(1):91–106
6. Effelsberg W, Spaniol O, Danthine A, Ferrari D (eds) (1996) High-speed networking for multimedia applications. Kluwer Academic, Dordrecht
7. Jeon WJ, Nahrstedt K (2002) Peer-to-peer multimedia streaming and caching service. In: Proc. of IEEE international conference on multimedia and expo, pp 57–60
8. Kanakia H, Mishra P, Reibman A (1993) Adaptive congestion control scheme for real-time packet video transport. In: SIGCOMM symposium on communications architecture and protocols, San Francisco, pp 20–31
9. Lee I, Guan L (2002) A scalable video codec design for streaming over distributed peer-to-peer network. In: Proc. of IEEE global telecommunications conference, vol 1, pp 539–543
10. Lee I, Guan L (2005) Reliable video communication with multi-path streaming. In: Proc. of IEEE international conference on multimedia and expo
11. Lee I, He Y, Guan L (2005) Centralized P2P streaming with MDC. In: Proc. of IEEE multimedia signal processing workshop, Shanghai, pp 189–192
12. Lee I, He Y, Guan L (2008) From centralized to decentralized video streaming using multiple descriptions coding. In: Proc. of international symposium on ubiquitous multimedia computing (UMC), pp 112–117
13. Ling Z, Lee I (2006) Adaptive multi-path video streaming. In: Proc. of IEEE international symposium on multimedia, San Diego, pp 399–406
14. Nahrstedt K, Steinmetz R (1995) Resource management in networked multimedia systems. IEEE Comput 28:52–63
15. Nguyen T, Zakhor A (2002) Distributed video streaming over the internet. In: Proc. of SPIE conference on multimedia computing and networking, pp 186–195
16. Roberts J, Mocci U, Virtamo J (1996) Broadband network teletraffic final report of action cost 242. Springer, New York
17. Rosu D (1999) Dynamic resource allocation for adaptive real-time applications. PhD thesis, Georgia Institute of Technology
18. Steinmetz R (1996) Human perception of jitter and media synchronization. IEEE J Sel Areas Commun 14:61–72
19. Tran DA, Hua KA, Do TT (2002) Zigzag: an efficient peer-to-peer scheme for media streaming. Tech. rep., University of Central Florida
20. Wang X, Schulzrinne H (1999) Comparison of adaptive internet multimedia applications. IEICE Trans Commun E82-B(6):806–818
21. Wu D, Hou YT, Zhu W, Zhang Y-Q, Peha JM (2001) Streaming video over the internet: approaches and directions. IEEE Trans Circuits Syst Video Technol 11(3):282–300

22. Wu D, Hou YT, Zhu W, Lee H-J, Chiang T, Zhang Y-Q, Chao HJ (2000) On end-to-end architecture for transporting mpeg-4 video over the internet. IEEE Trans Circuits Syst Video Technol 18(6):977–995
23. Wu F, Li S, Zhang Y-Q (2001) A framework for efficient progressive fine granularity scalable video coding. IEEE Trans Circuits Syst Video Technol 11(3):332–344
24. Yee JR, Weldon EJ (1995) Evaluation of the performance of error-correcting codes on a Gilbert channel. IEEE Trans Commun 43:2316–2323

Ivan Lee received the B.E., M.Com., M.E.R., and Ph.D. degrees from the the University of Sydney, Sydney, Australia. He is currently a Faculty Member with the School of Computer and Information Science, University of South Australia, Adelaide, Australia. Prior to his current appointment, he was a Development Engineer at Cisco Systems, a Software Engineer at Remotek Corporation, and an Assistant Professor in the Department of Electrical and Computer Engineering at Ryerson University, Toronto, Canada. His research interests include multimedia networking and multimedia signal processing.

Jong Hyuk Park received his Ph.D. degree in the Graduate School of Information Security from Korea University, Korea. He is now a professor at the Department of Computer Science and Engineering, Seoul National University of Technology, Korea. He has published about 100 research papers in international journals and conferences. He has been serving as chairs, program committee, or organizing committee chair for many international conferences and workshops. He was editor-in-chief of the International Journal of Multimedia and Ubiquitous Engineering (IJMUE), the managing editor of the International Journal of Smart Home (IJSH) in 2007 and 2008. In addition, he has been serving as a guest editor for international journals by some publishers. His research interests include digital forensics, security, ubiquitous and pervasive computing, context awareness, multimedia services, etc. He got the best paper award in ISA-08 conference, April, 2008 and outstanding leadership award in IEEE HPCC-09, June, 2009.