Streaming Mobile Cloud Gaming Video Over TCP With Adaptive Source–FEC Coding

Jiyan Wu, *Member, IEEE*, Chau Yuen, *Senior Member, IEEE*, Ngai-Man Cheung, *Senior Member, IEEE*, Junliang Chen, and Chang Wen Chen, *Fellow, IEEE*

***Abstract*— Cloud gaming has emerged as a promising application to enable high-end game playing with thin clients. Transmission control protocol (TCP) is pervasively adopted as the transport-layer protocol in the mainstream cloud gaming systems for video communication. However, streaming mobile cloud gaming video using the TCP is challenged with several key technical barriers: 1) the performance limitations of wireless networks in bandwidth and reliability; 2) the high throughput demand and stringent delay constraint imposed by high-quality gaming video transmission; and 3) the deadline violations and throughput fluctuations caused by the packet retransmission and congestion control mechanisms in the TCP. To address these critical problems, this paper proposes an application-layer source–forward error correction (FEC) coding framework dubbed adaptive source–FEC coding over TCP (ESCOT). First, we analytically formulate the optimization problem of joint source–FEC coding to minimize the end-to-end distortion of real-time video communication over TCP. Second, we develop a heuristic solution for effective loss rate approximation, source rate control, and FEC coding adaptation. ESCOT is distinct from existing source–FEC coding schemes in proactively analyzing and leveraging the TCP characteristics. The proposed solution is able to effectively mitigate both consecutive and sporadic video frame drops caused by congestion and random packet losses. We conduct the performance evaluation through extensive emulations in the Exata platform using real-time gaming video encoded by the H.264 codec. Experimental results show that the ESCOT advances the state of the art with noticeable improvements in video peak signal-to-noise ratio, end-to-end delay, goodput, and frame success rate.**

***Index Terms*— Cloud gaming, forward error correction (FEC), mobile video streaming, rate control, stringent delay constraint, transmission control protocol (TCP).**

I. Introduction

Modern computer games are often hardware demanding (e.g., in terms of CPU and graphic card) and require the installation of complex game softwares. Cloud gaming is a promising solution to enable high-end video gaming with thin end devices (e.g., PCs, laptops, smartphones, and so on). By rendering graphics in the cloud and streaming encoded gaming videos to players, end users are relieved from downloading/updating game software [1]. The potential of cloud gaming has already attracted considerable attention from both industrial and research communities [2]–[6]. It is predicted that the global computer gaming market revenue will grow to U.S. $78 billion in 2017, among which cloud gaming is expected to be the main portion [7].

Despite the perceived advantages of cloud gaming applications, it is still challenging to develop an effective platform, and the key technical issue is in the gaming video transmission [8], [9]. In this paper, we present an investigation of the problem in streaming real-time gaming video to mobile devices, as shown in Fig. 1. Transmission control protocol (TCP) is commonly adopted as the transport protocol in the mainstream cloud gaming systems (e.g., GamingAnywhere [1], OnLive [5], CloudUnion [6], etc.) for video communication [3]. Furthermore, the TCP is also used in other real-time video applications (e.g., mobile Skype [10], Web Real-Time Communication (RTC) [11], and Hypertext Transfer Protocol (HTTP) live streaming [12]) due to its special features in firewall traversal and network friendliness. To deliver high-quality gaming video over TCP in lossy wireless environments, critical technical problems to be solved are summarized as follows.

1) *Delay Constraint:* Cloud gaming is a time-critical application with stringent delay constraints. The round trip time (RTT) is limited to be less than 130 ms to guarantee smooth gaming experience [8], [13].
2) *Throughput Demand:* High-quality gaming video streaming is bandwidth-intensive, and the throughput demand needs to maintain between 2 and 6 Mb/s [9], [13].
3) *Network Limitation:* This paper investigates mobile cloud gaming video delivery over wireless networks, which are characterized by limited radio resources and

Manuscript received August 24, 2015; revised December 19, 2015; accepted January 15, 2016. Date of publication February 8, 2016; date of current version January 5, 2017. This work was supported in part by the National Research Foundation, Prime Minister's Office, Singapore, under its IDM Futures Funding Initiative and administered by the Interactive and Digital Media Programme Office; in part by the U.S. National Science Foundation (NSF) under Grant 1405594; and in part by the National Natural Science Foundation of China under Grant 61550110244. This paper was recommended by Associate Editor P. Frossard.

J. Wu and C. Yuen are with the Engineering Product Development Pillar, Singapore University of Technology and Design, Singapore 487372 (e-mail: wujiyan@126.com; yuenchau@sutd.edu.sg).

N.-M. Cheung is with the Information Systems Technology and Design Pillar, Singapore University of Technology and Design, Singapore 487372 (e-mail: ngaiman_cheung@sutd.edu.sg).

J. Chen is with State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China (e-mail: chjl@bupt.edu.cn).

C. W. Chen is with the Department of Computer Science and Engineering, University at Buffalo, The State University of New York, Buffalo, NY 14228 USA (e-mail: chencw@buffalo.edu).

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

Digital Object Identifier 10.1109/TCSVT.2016.2527398

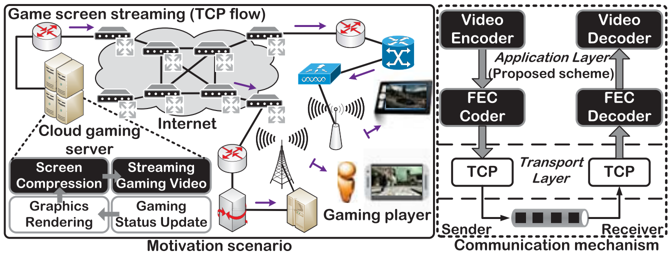


Fig. 1. Motivation scenario and communication mechanism of streaming cloud gaming video using TCP to mobile clients in wireless networks.

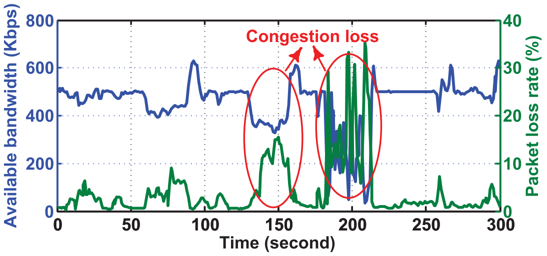


Fig. 2. Variations of available bandwidth and packet loss rates during client mobility in a wireless network.

time-varying status [14]. Besides, the congestion and random losses frequently occur in wired-cum-wireless packet switching networks.

4) *TCP Characteristics:* TCP has been pervasively adopted in mainstream cloud gaming systems for video communication. This transport protocol employs the packet retransmission for data protection and the additive-increase multiplicative-decrease (AIMD) algorithm for congestion control. Such mechanisms in TCP may induce frequent deadline violations and throughput fluctuations. Recent studies [15], [16] have reported the poor performance of TCP in wireless environments due to the misinterpretation of wireless errors as network congestion.

To address these challenging issues, we present, in this paper, an application-layer source–forward error correction (FEC) coding framework dubbed adaptive source–FEC coding over TCP (ESCOT). The ESCOT is distinct from the existing source and FEC coding schemes in proactively analyzing and leveraging the TCP-connection states. The proposed solution is able to deliver network-adaptive and distortion-minimized gaming video using TCP over error-prone wireless networks. Without the source rate control module, there may be severe quality degradations caused by congestion losses and bandwidth shrink. The FEC coding scheme is able to effectively mitigate sporadic frame drops induced by random packet losses. Section II-A will discuss the motivation for the joint source–FEC coding scheme in detail. In particular, main contributions of this paper can be summarized as follows.

1) Develop a mathematical framework to formulate the joint source–FEC coding adaptation to minimize the end-to-end distortion of TCP-based real-time video communication.
2) Propose an application-layer source–FEC coding framework that includes the following decision processes:
   a) an approximate analysis for effective loss rate estimation based on the Gilbert loss model and continuous-time Markov chain;
   b) a source rate control algorithm to minimize the end-to-end video distortion with respect to bandwidth limitation, delay constraint, and TCP friendliness;
   c) a FEC coding scheme striking an effective balance between delay and recoverability performance to minimize the effective loss rate.
3) Perform extensive emulations in Exata involving real-time gaming video (encoded with the H.264 codec) over TCP. Evaluation results show that the following holds.
   a) ESCOT improves the video peak signal-to-noise ratio (PSNR) by up to 8.5 (25.6%), 11.2 (33.2%), and 14.5 (45.9%) dB compared with the Adaptive High-Frame-Rate Video Streaming (APHIS) [9], joint source–FEC rate (JSFR) [17], and TCP New Reno schemes, respectively.
   b) ESCOT reduces the mean end-to-end delay by up to 19.2 (36.5%), 29.3 (45.2%), and 47.2 (55.3%) ms compared with the APHIS, JSFR, and New Reno schemes, respectively.
   c) ESCOT increases the average goodput by up to 0.24 (17.2%), 0.38 (22.3%), and 0.52 (28.2%) Mb/s compared with the APHIS, JSFR, and New Reno schemes, respectively.
   d) ESCOT increases the frame success rate by up to 8.55%, 16.2%, and 20.7% compared with the APHIS, JSFR, and New Reno, respectively.

The remainder of this paper is structured as follows. In Section II, we discuss the research motivation and briefly review existing studies related to the proposed research presented in this paper. Section III describes the system model and problem formulation. In Section IV, we present the scheduling algorithms for the proposed source–FEC coding framework. The performance evaluation is provided in Section V and the concluding remarks are given in Section VI.

II. Research Motivation and Related Work

A. Research Motivation

ESCOT is motivated by taking full advantage of the available network bandwidth to maximize real-time gaming video quality. To achieve this goal, a critical problem that degrades the network utilization is the congestion and random packet losses frequently occurred in packet switching networks. Fig. 2 profiles the evolutions of available bandwidth and packet loss rate during the client mobility in a Wi-Fi network. The video encoding rate is 500 kb/s. The moving speed of the client is 2 m/s and the emulation lasts for 300 s. Section V-A describes the detailed configurations of network environments and video coding parameters. We have the following observations from the network measurements: 1) the network congestion losses are often accompanied with substantial bandwidth shrink (e.g., during the intervals around 150 and 200 s) and such concurrence issues are also observed in [43] and 2) the random packet losses frequently

TABLE I
DIFFERENCES OF THE PROPOSED ESCOT WITH OUR EARLIER WORK [9]

Solution	TCP-awareness	Source rate control	Loss differentiation	Distortion model	Frame rate guarantee
ESCOT	✓	✓	✓	source-channel distortion	✓
APHIS [9]	×	×	×	frame-level channel distortion	×

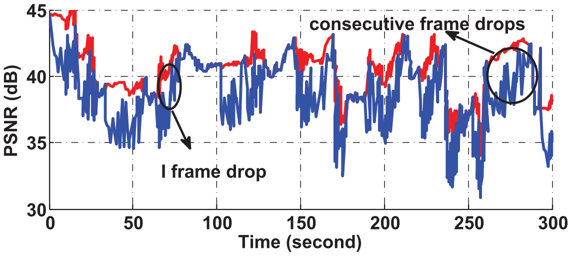


Fig. 3. Evolutions of instantaneous PSNR values during the interval of [0, 300] s.

occur during the mobile video transmission because of the user mobility, channel fading, link faults, and so on.

Fig. 3 shows the instantaneous PSNR values of the constant bit-rate video streaming received by the mobile terminal during the same interval (with the source video PSNR for comparison). The evolutions of PSNR values generally follow the variations of available bandwidth. Conventionally, the decrease in PSNR values caused by lost frames can be mitigated by error concealment methods [54]–[56]. Severe quality degradations can be perceived, if consecutive frames are lost or I frames encounter losses. The random packet losses can be mitigated by appending an appropriate number of FEC parity packets, since the retransmission mechanism in TCP may not be able to recover the lost packets within the delay constraint. However, recent studies [18], [19] reveal the important fact that FEC is ineffective to handle congestion losses, which are characterized by burstiness. As bandwidth shrink often occurs during congestion losses, injecting more FEC parity packets to the communication network will only increase the congestion level and possibly induce further losses. Therefore, we introduce the source rate control to ensure the congestion loss resilience and bandwidth adaptiveness of cloud gaming video.

In Section VI, the source rate control is motivated by the problems of bandwidth fluctuation and congestion loss. The FEC coding scheme aims at mitigating the sporadic frame drops caused by random packet losses.

B. Related Work

The related studies to this paper can be classified into three categories: 1) cloud gaming applications; 2) TCP-based multimedia communication; and 3) rate control and FEC coding schemes for real-time video transmission.

1) Cloud Gaming Applications: Cloud gaming is a typical multimedia application taking an advantage of the cloud computing technology. Chen *et al.* [13] conduct the measurement studies on the quality of service (QoS; delay performance, subjective video quality, and so on) of cloud gaming in both OnLive and StreamMyGame systems. The measurement results reveal that the OnLive gaming system is able to dynamically adapt the frame rate (in the gaming engine) according to different network conditions. Wu *et al.* [2] also conduct the extensive measurement studies in the CloudUnion gaming platform to identify the differences of queuing and response delays among gaming players. Claypool *et al.* [20] conduct a detailed measurement study on the traffic characteristics of OnLive and reveal the scheduling patterns of large/small packets. Huang *et al.* [1] propose the GamingAnywhere and adopt the measurement techniques proposed in the article to show that GamingAnywhere outperforms proprietary and closed cloud gaming systems. Wang and Dey [51] propose a rendering adaption technique to dynamically vary the richness and the complexity of graphic rendering depending on the network and cloud computing constraints. In the literature [52], a mobile gaming user experience (MGUE) model is developed and validated through subjective tests. Liu *et al.* [53] propose a content-aware model to measure user experience with regard to the impairments of rendering, video coding, and network.

In the literature [9], the authors propose a joint frame selection and the FEC coding scheme to deliver mobile gaming video. The proposed ESCOT is different from the APHIS in that we consider the TCP-based cloud gaming video transmission, while the APHIS assumes user datagram protocol as the transport-layer protocol. Table I summarizes the main differences of the proposed ESCOT and APHIS [9].

2) TCP-Based Multimedia Communication: Wu *et al.* [21] develope an application-layer FEC coding scheme to proactively leverage the delay friendliness of TCP in optimizing real-time video transmission. Shiang and van der Schaar [22] propose a media-TCP-friendly congestion control (MTCC) scheme for video streaming over wired IP networks. The MTCC is an application–transport-layer solution, which explicitly considers the distortion impacts, delay deadlines, and priority of different video packet classes. Habachi *et al.* [23] propose an AIMD-like media-aware congestion control that determines the optimal congestion window updating policy for video transmission. The authors formulate the congestion control problem as a partially observable Markov decision process to maximize the long-term expected multimedia quality. Brosh *et al.* [24] presente a discrete-time Markov model of the delay distribution for real-time TCP flows, and reveal the delay friendliness of TCP toward packet size through extensive measurements. The recent standards for MPEG-Dynamic Adaptive Streaming over HTTP and adaptive HTTP streaming are introduced in [57] and [58]. However, these standard methods cannot satisfy the stringent QoS requirements (e.g., end-to-end delay) of mobile cloud gaming video delivery using TCP in wireless networks.

3) FEC Coding and Rate Control: Ahmad *et al.* [26] propose a rateless coding scheme that keeps on sending the encoded symbols until receiving an acknowledgment or

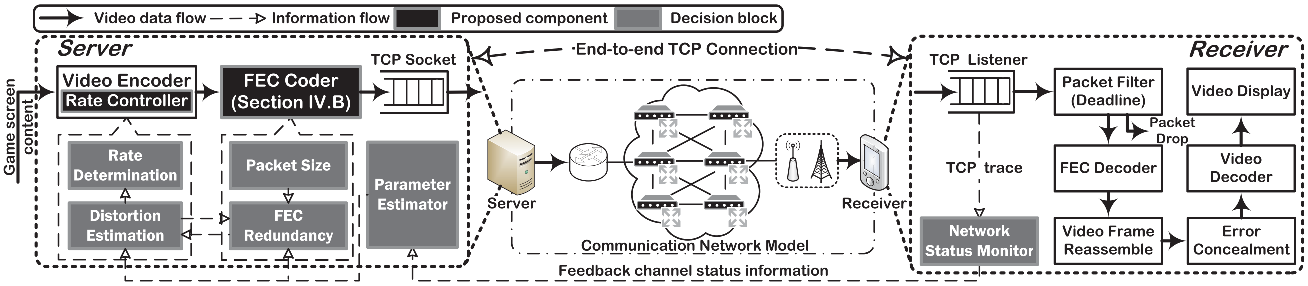


Fig. 4. System diagram and working components of the proposed ESCOT.

passing the deadline. Tournoux *et al.* [25] propose an on-the-fly erasure coding scheme called Tetrys that considers the feedback information during the FEC coding process. Frossard and Verscheure [17] use the recursive approach and a two-state Markov chain channel model to compute the post-FEC packet loss ratio and burst length for systematic FEC codes. In [27], a cross-layer FEC scheme using Raptor and rate compatible punctuated convolutional codes is proposed for video transmission over wireless channels. Xiao *et al.* [28] propose a subgroup of pictures (GoP) level FEC coding scheme to optimize the delay performance for real-time video applications. In [29], a randomized expanding FEC coding scheme is proposed to append parity packets for current and previous frames for enhanced data protection. Rate control has been widely investigated in recent studies [30]–[32] to optimize low-delay video communication.

In summary, the previous studies have not addressed the challenging problem of streaming mobile cloud gaming video using TCP, and this paper tackles this critical issue with joint source–FEC coding. To the best of our knowledge, the proposed ESCOT is the first application-layer source–FEC coding scheme that exploits the TCP characteristics to optimize real-time gaming video quality over wireless networks. The proposed algorithms in ESCOT can also be accommodated to other TCP-based real-time video applications, e.g., video conferencing, Web RTC, and so on.

III. System Model and Problem Formulation

A. System Overview

Fig. 4 presents the system overview of the proposed ESCOT scheme, which is a completely end-to-end solution. In this paper, we investigate the end-to-end transmission of a real-time gaming video flow using the TCP connection over wireless packet switching networks. The streaming protocols in the system involve the proposed ESCOT at the application layer and the TCP (New Reno) at the transport layer. New Reno is one of the most widely deployed TCP versions over the Internet due to the advantages in loss recovery and retransmission limitation [15]. The goal of the proposed framework is to achieve the optimal video quality with the rate control and FEC coding adaptation. The main control parameters in the proposed adaptation scheme involve the source (video) coding rate (V), redundancy value (R), and packet size (S).

The key working components at the sender side include the rate controller (in the video encoder), FEC coder, and parameter estimator. For each decision epoch (e.g., the duration of a GoP), the video encoding (source) rate is updated and the game screen content (uncompressed video) is encoded into video frames. This paper employs the H.264 codec [e.g., Joint Model (JM), ffmpeg, and ×264] for video compression, since it is widely used in existing cloud gaming systems [2]. At the FEC coder, the video frames are converted to FEC packets based on the packet size (S) and redundancy value (R) obtained with Algorithm 2 (Section IV-C). Then, these FEC packets are transmitted to the destination (mobile device) using the TCP socket.

After receiving the data packets at the receiver, the first step is to check whether they are past the decoding deadline. The overdue packets will be dropped, since they cannot contribute to the decoding process. The network status monitor is responsible for periodically estimating and sending back the network status information (i.e., bandwidth, packet loss rate, and RTT) to the sender side. It is important to collect the information, since these parameters are the input items in the rate control and FEC coding modules of the proposed ESCOT framework. The status information is estimated based on the TCP trace, which can be obtained at the application layer. To mitigate the quality degradations caused by frame drop, a basic error concealment method by frame copying is implemented at the receiver side. The other error resilience schemes in H.264/Advanced Video Coding (AVC) [63] can also be used in conjunction with the proposed ESCOT to further improve video quality.

The analytical framework of joint source–FEC coding adaptation for TCP-based real-time video communication involves the mathematical models of communication network [9], end-to-end video distortion [14], [33], systematic FEC coding [34], and TCP-connection delay [24]. For the sake of completeness and integration with system framework, these mathematical models are briefly introduced in this section. The basic notations used throughout this paper are summarized in Table II.

B. Model Description

1) Communication Network Model: The communication network represents the end-to-end connection link and includes wired-cum-wireless domains between end devices. From the perspective of gaming service provider at the application layer, we characterize the communication network with the properties of RTT, packet loss rate π_B, and available bandwidth μ. The detailed descriptions of these physical properties are presented in [9].

TABLE II
SUMMARY OF BASIC NOTATIONS

Symbol	Definition
$\mathbb{P}, \mathbb{E}, \mathbb{I}$	the probability, expectation value, indication function.
μ, π_B	the available bandwidth, packet loss rate.
$\mathcal{N}, \mathcal{F}$	the GoP length, encoding frame rate.
RTT	the round trip time.
G/B	the Good/Bad state of the network path.
ξ_B/ξ_G	the state transition probability from G/B to B/G.
Π	the effective loss rate of a video GoP.
$\pi_{\mathrm{t}}, \pi_{\mathrm{o}}$	the transmission, overdue loss rate.
$\mathsf{D}, \mathsf{D}_{\mathrm{src}}, \mathsf{D}_{\mathrm{chl}}$	the end-to-end, source, channel distortion.
$\mathrm{d}_{\mathrm{E}}, \mathrm{d}_{\mathrm{N}}, \mathrm{d}_{\mathrm{T}}$	the end-to-end, network-level, TCP-level delay.
V, ρ	the video source (encoding) rate, available source rate.
b, ω	the backlogged packets, congestion window size.
T	the delay constraint for each GoP.
n, k	the number of data, source packets in a FEC block.
$\mathsf{F}_{\langle i,j \rangle}(\theta)$	the transition probability from state i to j in time θ.
MSS	the maximum segment size.
R, S	the FEC redundancy value, FEC packet size.

In HTTP/TCP-based video streaming systems, the available bandwidth can be estimated according to the observed TCP throughput. The TCP throughput and packet loss rate can be acquired by analyzing the TCP traces via the software, e.g., the Wireshark, TCP trace, and so on. In the emulations of this paper, the TCP trace is enabled in Exata to analyze the throughput and packet loss rate. We model the burst packet losses over end-to-end communication path as a continuous-time stochastic process based on the Gilbert loss model [35]. This is a continuous-time Gilbert model, and the parameters are assumed to be independent of the sending traffic rate. Let the notations ξ_B and ξ_G denote the state transition probabilities from G to B and B to G, respectively. Two system-dependent parameters are adopted to specify the packet loss model: 1) the average loss rate π_B and 2) the average loss burst length $1/\xi_B$. Then, we can have $\pi_B = \xi_B/(\xi_B + \xi_G)$ and $\pi_G = \xi_G/(\xi_B + \xi_G)$.

2) End-to-End Video Distortion Model: To characterize the real-time gaming video quality, this paper employs a generic end-to-end video distortion model [33]. The user-perceived quality in video streaming environment is impacted by the end-to-end distortion (D). In particular, D is the sum of two main categories of distortion: source distortion ($\mathsf{D}_{\mathrm{src}}$) and channel distortion ($\mathsf{D}_{\mathrm{chl}}$), that is

$$\mathsf{D} = \mathsf{D}_{\mathrm{src}} + \mathsf{D}_{\mathrm{chl}}. \tag{1}$$

This analytic model indicates that the streaming video quality depends on both the distortion caused by the data compression of the media information and the distortion due to the transmission impairments in the communication network. The source distortion $\mathsf{D}_{\mathrm{src}}$ is mostly driven by the video encoding rate (V) and the video sequence content. These parameters largely impact the efficiency of the video codec (e.g., the larger distortion will be induced for a more complex video sequence under the same video encoding rate). The channel distortion $\mathsf{D}_{\mathrm{chl}}$ is mainly impacted by the effective loss rate (Π) defined as follows.

Definition 1 (Effective Loss Rate Π): The ratio of lost data in a video GoP after the FEC recovery process. This loss probability includes both the channel errors/losses and the expired packet arrivals.

$\mathsf{D}_{\mathrm{chl}}$ is generally proportional to the number of lost video frames. In particular, the end-to-end distortion can be expressed [in units of mean squared error (MSE)] as [14], [33]

$$\mathsf{D} = \underbrace{\frac{\alpha}{V - V_0}}_{\mathsf{D}_{\mathrm{src}}} + \underbrace{\beta \cdot \Pi}_{\mathsf{D}_{\mathrm{chl}}} \tag{2}$$

in which α, V_0, and β are the parameters depending on the specific video codec and sequence. These parameters can be online estimated by using trial encodings at the sender side [14], [36]. To enable the fast adaptation of the transmission scheduling to abrupt changes in the video content, these parameters can be updated for each GoP.

3) Systematic FEC Coding Model: The systematic Reed–Solomon (RS) [34] block erasure code is adopted for video data protection against channel losses. The main motivation to choose the RS code over the fountain codes (e.g., Luby transform code) is the stringent delay constraint of cloud gaming video, since the fountain code is often featured by large block size. It is also feasible to implement the latest systematic fountain codes (e.g., Raptor code [59], [60]) with limited block size.

An FEC block of n data packets contains k source packets and $n - k$ redundant packets. The receiver is able to recover all the k source packets, if any k packets of the FEC coding block are successfully received. In the FEC(n, k) coding, $(n - k)$ redundant data packets are introduced for every k source packets to make up a codeword (FEC block). If a random set of k out of the n coded packets is received by the client, a fraction of the k packets is systematic. In particular, the soft-decision decoding algorithms (e.g., the Koetter–Vardy decoder [37]) are implemented at the receiver side to achieve such goal.

The video data in each FEC block are converted into k source packets. For the last source packet, some zero bytes are often padded to keep the same size with other packets. Besides, the FEC packet size (S) also affects the tradeoff between loss recoverability and end-to-end delay. With a smaller FEC packet size, the value of n (FEC block size) will become larger, and there is higher possibility for the receiver to recover the lost packets. However, a larger FEC block size also delays the FEC decoding because the recovery process can only start after the client has received k FEC packets.

4) TCP-Connection Delay Model: The delay performance model in [24] is employed in the analytical framework to analyze the end-to-end TCP-connection delay, and this model is based on the following assumptions: 1) the New Reno is used as the TCP version to avoid frequent retransmission timeout (TO) events and 2) the Nagle algorithm [38] is disabled to reduce packet backlogging.

A high-level view of the TCP state transition is shown in Fig. 5 [24]. In order to develop an application-layer source–FEC coding scheme for real-time gaming video delivery using TCP, it is necessary to analyze the TCP state

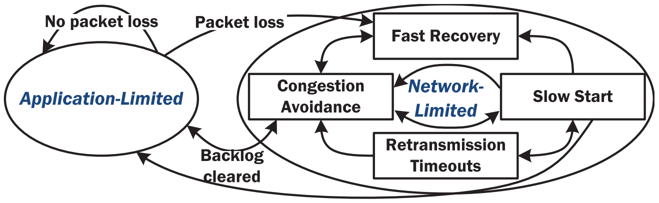


Fig. 5. State transitions of TCP connection.

and connection delay for the end-to-end distortion estimation. There are two main states for the TCP operation: application-limited (AL) and network-limited. The system transitions from an AL state to a network-limited state when a loss occurs. TCP-level delays are introduced only during network-limited states. The system transitions back to an AL state when the TCP sender matches its input and output rates (e.g., when packet backlog is cleared). While, in a network-limited state, the system moves among four states corresponding to TCPs congestion control phases: slow start (SS), congestion avoidance (CA), fast recovery (FR), and retransmission TOs.

Classically, the throughput and packet loss rate can be estimated based on network measurements (e.g., through running average) without considering the TCP states. The estimated values can be used to predict the throughput and loss rates in a short period. However, such solutions are unable to model/capture the dynamics (e.g., delay and throughput fluctuations) caused by the TCP mechanisms (e.g., congestion control and packet retransmission). Therefore, it is desirable to analyze the TCP states and characteristics in modeling end-to-end connection.

According to the literature [24], we model the end-to-end TCP-connection delay d_E with two main components: 1) network-level delay (d_N) and 2) TCP-level delay (d_T), that is

$$d_E = d_N + d_T. \tag{3}$$

In particular, the TCP-level delay includes the congestion control, retransmission, and head-of-line blocking time. The network-level latency consists of the packet transmission and path propagation delays. We employ the Markov chain model developed in [24] to analyze the TCP-level delay. According to (3) in [24], the steady-state TCP delay distribution is presented as

$$\mathbb{P}(d_T = d) = \frac{\sum_{s\in\Omega} \epsilon_s \cdot \sum_{s'\in\Omega} P_{\langle s,s'\rangle} \cdot \sum_{i=1}^{n_{\langle s,s'\rangle}} \mathbb{I}_{d^i_{\langle s,s'\rangle}=d}}{\sum_{s\in\Omega} \epsilon_s \cdot \sum_{s'\in\Omega}(P_{\langle s,s'\rangle} \cdot n_{\langle s,s'\rangle})} \tag{4}$$

in which $\mathbb{I}$ denotes the indicator function and ϵ_s denotes the steady-state distribution of the Markov chain. $n_{\langle s,s'\rangle}$ denotes the number of packets to be sent from state s to s'. The set (Ω) of the states (s) is defined as [24]

$$\Omega = \{\text{Application Limited (AL), Congestion Avoidance (CA),} \\ \text{Slow Start (SS), Fast Recovery (FR), Timeout (TO)}\}$$

$d^i_{\langle s,s'\rangle}$ in (4) denotes the TCP-level delay of the ith packet sent in a transition from state s to s'

$$d^i_{\langle s,s'\rangle} = \begin{cases} b/V + \text{RTT} + (3+i)/f, & \text{if } s' \in \text{FR} \\ b/V, & \text{otherwise} \end{cases} \tag{5}$$

where b denotes the backlogged packets and we consider the congestion-induced backlog in this paper. The packet backlogging can also be caused by Nagle's algorithm [38] that complements the TCP to limit the transmission of small segments. However, this algorithm is disabled in many delay-stringent multimedia applications to reduce the retransmission delay [24], and we also follow this practice in this paper. The backlog evolution for two consecutive states is modeled by

$$b = \begin{cases} \max\{0, b + V \cdot t_{\langle s,s'\rangle} - S\}, & \text{if } s == \text{AL} \,\&\&\, s' == \text{AL} \\ \max\{0, b + V \cdot t_{\langle s,s'\rangle} & \\ \quad - \omega \cdot \text{MSS}\}, & \text{if } s' \in \{\text{CA, SS}\} \\ \max\{0, b + V \cdot t_{\langle s,s'\rangle} & \\ \quad - (\omega + 3) \cdot \text{MSS}\}, & \text{if } s' == \text{FR} \\ \max\{0, b + V \cdot t_{\langle s,s'\rangle}\}, & \text{if } s' == \text{TO} \end{cases}$$

where $t_{\langle s,s'\rangle}$ denotes the time interval from the state s to s', and the congestion window size ω can be estimated with the formula $\omega = \mu \cdot \text{RTT}$. The number of packets ($n_{\langle s,s'\rangle}$) sent from s to s' in (4) is expressed as

$$n_{\langle s,s'\rangle} = \begin{cases} 1, & \text{if } s == \text{AL} \,\&\&\, s' == \upsilon\text{AL} \\ \lfloor \min\{b, \omega \cdot \text{MSS}\}/S \rfloor, & \text{if } s' \in \{\text{CA, SS}\} \\ \lfloor \min\{b, (\omega+3) \cdot \text{MSS}\}/S \rfloor, & \text{if } s' == \text{FR} \\ 0, & \text{if } s' == \text{TO}. \end{cases}$$

The TCP-level delay $d_T(n)$ for the mth frame can be estimated with the delay distribution presented in (4) and the above parameters.

The network-level delay d_N for n FEC packets can be estimated using

$$d_N(n) = \frac{\mu \cdot (1 - \pi_B)}{n \cdot S} + \frac{\rho}{\mu - V} \tag{6}$$

where $\mu \cdot (1 - \pi_B)$ denotes the loss-free bandwidth and S denotes the FEC packet size. The loss-free bandwidth is a good indicator of available capacity for end-to-end data transport over lossy paths [39]. ρ denotes the available source rate and is estimated using the latest historical values of path status information in conjunction with video traffic rate [14], [36]

$$\rho = \frac{(\bar{\mu} - V) \cdot \text{RTT}}{2}$$

in which $\bar{\mu}$ denotes the smoothed value of the measured available bandwidth with the confidence interval of 95%.

5) Effective Loss Rate Analysis: To accurately estimate the end-to-end distortion D, it is necessary to derive the effective loss rate Π that determines the channel distortion. As explained in Definition 1, Π is the combined probability of transmission (π_t) and overdue (π_o) loss rates. These terminologies are defined as follows.

Definition 2 (Transmission Loss Rate π_t): The ratio of FEC packets that encounter channel losses or buffer overflows in packet switching networks.

Definition 3 (Overdue Loss Rate π_o): The probability of expired arrivals of FEC packets at the destination out of the delay constraint imposed by the video applications.

With regard to the systematic property of FEC(n, k) coding, Π is expressed as

$$\Pi = \begin{cases} 0, & \text{if } \pi_t + (1-\pi_t)\cdot\pi_o < \dfrac{n-k}{n} \\ \pi_t + (1-\pi_t)\cdot\pi_o, & \text{otherwise.} \end{cases} \tag{7}$$

The expressions of π_t and π_o are presented as

$$\pi_t = \frac{1}{k}\cdot\sum_{\forall c}\left\{\mathcal{L}(c)\cdot\pi_{c_1}\cdot\prod_{i=1}^{n-1}[\mathsf{F}_{\langle c_i,c_{(i+1)}\rangle}(\theta_i)]\right\} \tag{8}$$

$$\pi_o \approx \exp\left[-\frac{T}{d_N(n)+d_T(n)}\right]. \tag{9}$$

The mathematical derivations are provided in the Appendix.

C. Problem Formulation

After introducing the above models and deriving the expression of the effective loss rate, we are ready to formulate the constrained optimization problem of end-to-end video distortion minimization. Given the feedback network status (RTT, μ, π_B, and ξ_B), the encoding frame rate $\mathcal{F}$ (in f/s), the video GoP length $\mathcal{N}$, and the delay constraint T, the goal of the adaption scheme is to find the optimal solution to minimize the end-to-end video distortion D. Mathematically, the joint source and FEC coding adaptation problem for each GoP can be formulated as

$$(\mathsf{P1}) : \{V, R, S\} = \arg\min\{\mathsf{D}\}$$

$$\text{s.t.}\begin{cases} d_N(n) + d_T(n) \le T & \text{(10a)} \\ \max\left\{V\cdot(1+R), \dfrac{n\cdot S}{T}\right\} < \mu & \text{(10b)} \end{cases}$$

where

$$\mathsf{D} = \mathsf{D}_{\text{src}} + \mathsf{D}_{\text{chl}}, \quad \mathsf{D}_{\text{src}} = \frac{\alpha}{V - V_0}, \quad \mathsf{D}_{\text{chl}} = \beta\cdot\Pi$$

$$k = \left\lceil\frac{V\cdot\mathcal{F}}{\mathcal{N}\cdot S}\right\rceil, \quad n = \lceil k\cdot(1+R)\rceil$$

$$\Pi = \begin{cases} 0, & \text{if } \pi_t + (1-\pi_t)\cdot\pi_o < \dfrac{n-k}{n} \\ \pi_t + (1-\pi_t)\cdot\pi_o, & \text{otherwise} \end{cases}$$

$$\pi_t = \frac{1}{k}\cdot\sum_{\forall c}\left\{\mathcal{L}(c)\cdot\pi_{c_1}\cdot\prod_{i=1}^{n-1}[F_{\langle c_i,c_{(i+1)}\rangle}(\theta_i)]\right\}$$

$$\pi_o \approx \exp\left[-\frac{T}{d_N(n)+d_T(n)}\right].$$

As stated in constraint (10a), the end-to-end delay for each video GoP is restricted to be less than the deadline T. In order to ensure the stable state of the communication system (10b), the total traffic rate after appending the parity packets should not exceed the available bandwidth (with regard to the video traffic variations [40]). It is infeasible to apply the brutal force (greedy search) algorithm to obtain the optimal solution for problem ($\mathsf{P1}$). First, the computational cost considers all the possible combinations of V, R, and S. Second, it is computationally prohibitive to obtain the expectation value of π_t by considering all the failure configurations (c) and inter-leaving levels (θ_i) in (12). Therefore, we develop heuristic algorithms for joint source–FEC coding adaption to achieve suboptimal performance with polynomial-time complexity. The detailed solution procedure will be presented in Section IV.

IV. Source–FEC Coding Algorithms

This section introduces the scheduling algorithms of the proposed source–FEC framework in detail. ESCOT is a joint adaptation scheme, since the source and FEC coding modules are inter-dependent. Given the bandwidth limitation, a larger source rate will decrease the FEC redundancy and vice versa.

As analyzed in Section III-C, it is difficult to directly solve the distortion minimization problem ($\mathsf{P1}$) in real-time manner. Therefore, we propose a suboptimal solution by decomposing ($\mathsf{P1}$) into the subproblems of source rate control ($\mathsf{P2}$) and FEC coding adaptation ($\mathsf{P3}$). In each decision epoch, the FEC coding algorithm (Algorithm 2) is invoked as the execution procedure in the source rate control scheme (i.e., Line 4 in Algorithm 1) to estimate the FEC redundancy value (R) and packet size (S). First, we provide an approximate analysis to approach the effective loss rate Π.

A. Effective Loss Rate Approximation

This section provides an approximate analysis to approach the effective loss rate based on the continuous-time Markov chain and the Gilbert loss model. In order to derive the value of Π, we first derive the closed-form expression for the transmission loss rate π_t. Note that (8) enables us to compute the transmission loss rate π_t, if all the possible combinations of c are considered. However, it is computationally prohibitive to obtain the expectation value of π_t when the FEC block size n becomes large, since there are 2^n possible combinations. In order to simplify the analysis and transmission scheduling, we propose to evenly spread the FEC packets, so that the interval is equal, i.e., $\theta_i = \theta_{i+1} = \theta, 1 \le i \le n-1$.

Let $\mathcal{J}$ and $\mathcal{L}$ denote the number of lost FEC and source packets. π_t can be rewritten as

$$\pi_t = \frac{1}{k}\sum_{j=n-k+1}^{n}\left\{\mathbb{P}(\mathcal{J}=j)\cdot\mathbb{E}[\mathcal{L}|\mathcal{J}=j]\right\}. \tag{11}$$

Let the notation $\left\langle {a \atop b} \right\rangle$ denote the event that any b out of a consecutive packets is lost, and the concatenation of events is allowed (e.g., $G\left\langle {k-1 \atop i} \right\rangle$ indicates that the event is preceded by a good state). We can derive the expression of $\mathbb{P}(\mathcal{J}=j)$ on the condition that the first state conforms to the stationary distribution of packet loss

$$\begin{aligned} \mathbb{P}(\mathcal{J}=j) &= \mathbb{P}\left(G\left\langle {n-1 \atop j} \right\rangle\right) + \mathbb{P}\left(B\left\langle {n-1 \atop j-1} \right\rangle\right) \\ &= \pi_G\cdot\mathbb{P}\left(\left\langle {n-1 \atop j} \right\rangle|G\right) + \pi_B\cdot\mathbb{P}\left(\left\langle {n-1 \atop j-1} \right\rangle|B\right) \end{aligned} \tag{12}$$

in which $\mathbb{P}\left(\left\langle {a \atop b} \right\rangle|q\right), q \in \{G, B\}$ represents the probability that the event $\left\langle {a \atop b} \right\rangle|q$ occurs. Although there is no closed form of $\mathbb{P}\left(\left\langle {a \atop b} \right\rangle|q\right), q \in \{G, B\}$ in the literature, it can be derived based on the recursive functions in [34]. In order to derive $\mathbb{E}[\mathcal{L}|\mathcal{J}=j]$, we first derive the expression of $\mathbb{P}(\mathcal{L}=i, \mathcal{J}=j)$. We consider the k source packets and the $n-k$

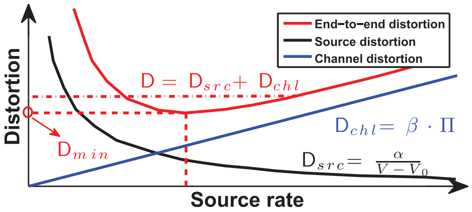


Fig. 6. Impact of source rate on end-to-end video distortion.

redundant packets separately. The state of the last FEC packet is

$$\begin{aligned}
&\mathbb{P}(\mathcal{L}=i,\mathcal{J}=j)\\
&=\mathbb{P}\left(\left\langle \genfrac{}{}{0pt}{}{k-1}{i}\right\rangle G\right)\cdot\mathbb{P}\left(G\left\langle \genfrac{}{}{0pt}{}{n-k}{j-i}\right\rangle\right)+\mathbb{P}\left(\left\langle \genfrac{}{}{0pt}{}{k-1}{i-1}\right\rangle B\right)\cdot\mathbb{P}\left(\left\langle \genfrac{}{}{0pt}{}{n-k}{j-i}\right\rangle|B\right)\\
&=\mathbb{P}\left(G\left\langle \genfrac{}{}{0pt}{}{k-1}{i}\right\rangle\right)\cdot\mathbb{P}\left(\left\langle \genfrac{}{}{0pt}{}{n-k}{j-i}\right\rangle|G\right)+\mathbb{P}\left(B\left\langle \genfrac{}{}{0pt}{}{k-1}{i-1}\right\rangle\right)\cdot\mathbb{P}\left(\left\langle \genfrac{}{}{0pt}{}{n-k}{j-i}\right\rangle|B\right)\\
&=\pi_G\cdot\mathbb{P}\left(\left\langle \genfrac{}{}{0pt}{}{k-1}{i}\right\rangle|G\right)\cdot\mathbb{P}\left(\left\langle \genfrac{}{}{0pt}{}{n-k}{j-i}\right\rangle|G\right)\\
&\quad+\pi_B\cdot\mathbb{P}\left(\left\langle \genfrac{}{}{0pt}{}{k-1}{i-1}\right\rangle|G\right)\cdot\mathbb{P}\left(\left\langle \genfrac{}{}{0pt}{}{n-k}{j-i}\right\rangle|G\right).
\end{aligned}$$

Remark: The above equation is based on the Markovian property of the Gilbert loss model, that is

$$\begin{aligned}
&\mathbb{P}(\mathcal{L}=i,\mathcal{J}=j|\text{the last state is } q)\\
&=\mathbb{P}(\mathcal{L}=i|\text{the last state is } q)\cdot\mathbb{P}(F=j|\text{the last state is } q).
\end{aligned}$$

Now, we have the expression of $\mathbb{E}[\mathcal{L}|\mathcal{J}=j]$ as

$$\mathbb{E}[\mathcal{L}|\mathcal{J}=j]=\sum_{i=1}^{k}\left[i\cdot\frac{\mathbb{P}(\mathcal{L}=i,\mathcal{J}=j)}{\mathbb{P}(\mathcal{J}=j)}\right]. \tag{13}$$

After plugging (12) and (13) into (11), we can obtain an approximate expression, i.e., (14), as shown at the bottom of this page, for π_t. The performance results (time complexity and approximation ratio) of this approximate analysis are presented in Section V-C.

B. Source Rate Control

In the source rate control process, there is an inherent tradeoff between the source and the channel distortion, as shown in Fig 6. For a given bandwidth limitation, improving the reliability through FEC coding comes at the expense of reducing source rate. Conversely, decreasing the FEC redundancy to support higher source rate reduces the error resilience. As analyzed in Section III-C, it is infeasible to derive the source rate value that achieves the minimal distortion ($\mathsf{D}_{\min}$) due to the limitation in time complexity. Note that there may be two possible values of source rate R to obtain a suboptimal end-to-end distortion value. It is desirable to calculate the smaller source rate for the bandwidth conservation.

Since ESCOT assumes TCP as the transport-layer protocol for video communication, it is necessary to consider the TCP friendliness [41] in the source rate calculation. According to the specifications in TCP-friendly rate control [42], the source rate should not exceed the TCP sending rate, that is

$$V_{\max}=\frac{\text{MSS}}{\text{RTT}\cdot\sqrt{\frac{2\times\pi_B}{3}}+12\times\sqrt{\frac{3\times\pi_B}{8}}\cdot\pi_B\cdot[1+32\times(\pi_B)^2]}.$$

With regard to constraint (10b), the upper limit of V is the smaller value of $V_{\max}$ and $(\mu/1+R)$. Therefore, the source rate control problem can be mathematically stated as follows:

$$(\mathsf{P2}):V=\arg\min\left\{\underbrace{\frac{\alpha}{V-V_0}}_{\mathsf{D}_{\text{src}}}+\underbrace{\beta\cdot\Pi(V,R,S)}_{\mathsf{D}_{\text{chl}}}\right\}$$

$$\text{s.t.}\begin{cases}V\le\min\left\{V_{\max},\dfrac{\mu}{1+R}\right\} & (15\text{a})\\ d_N(V,R,S)+d_T(R,S)\le T. & (15\text{b})\end{cases}$$

We develop a loop algorithm to determine the source rate that approaches the minimal end-to-end distortion. In particular, the estimated throughput of TCP New Reno is set as the initial value of the source rate [15], that is

$$\begin{aligned}
&V_{\text{int}}\\
&=\frac{\frac{1}{\pi_B}+\frac{\bar{\omega}^2}{1+\bar{\omega}^2}}{\omega'\cdot\text{RTT}+[1+2\pi_B+4(\pi_B)^2]\cdot T_0+\text{RTT}\cdot[1+\log(\omega/4)]}
\end{aligned}$$

where the values of $\bar{\omega}$, T_0, and ω' are specified in [15]. The step size (ΔV) of source rate can be dynamically adjusted and we set the value as 50 kb/s in this paper. In each iteration, the FEC coding algorithm is invoked to estimate the redundancy value R and packet size S. The search operation will terminate if the end-to-end distortion cannot be further reduced. Algorithm 1 outlines the sketch of the source rate control scheme, and Proposition 1 concludes the time complexity. The detailed explanations for Algorithm 1 are presented as follows.

1) *Line 1:* Initialize the source rate value, upper limit $V_{\max}$, and temporary variable D_{temp}.
2) *Line 2:* Estimate the distortion parameters V_0, α, and β using trial coding.
3) *Lines 3–14:* The main while loop to determine the source rate with the step size of ΔV.
4) *Line 4:* Estimate the FEC redundancy and packet size by invoking Algorithm 2.
5) *Lines 5–8:* Estimate the effective loss rate based on the FEC coding parameters and approximate analysis.
6) *Line 12:* Reduce the source rate by ΔV if the derivative of D at this point is negative.
7) *Line 13:* Or else, increase the source rate by ΔV.

$$\begin{aligned}
\pi_t=\frac{1}{k}\sum_{j=n-k+1}^{n}&\left\{\left[\sum_{i=1}^{k}i\cdot\frac{\pi_G\cdot\mathbb{P}(\left\langle \genfrac{}{}{0pt}{}{k-1}{i}\right\rangle|G)\cdot\mathbb{P}(\left\langle \genfrac{}{}{0pt}{}{n-k}{j-i}\right\rangle|G)+\pi_B\cdot\mathbb{P}(\left\langle \genfrac{}{}{0pt}{}{k-1}{i}\right\rangle|B)\cdot\mathbb{P}(\left\langle \genfrac{}{}{0pt}{}{n-k}{j-i}\right\rangle|B)}{\pi_G\cdot\mathbb{P}(\left\langle \genfrac{}{}{0pt}{}{n-1}{j}\right\rangle|G)+\pi_B\cdot\mathbb{P}(\left\langle \genfrac{}{}{0pt}{}{n-1}{j-1}\right\rangle|B)}\right]\right.\\
&\left.\cdot\left[\pi_G\cdot\mathbb{P}\left(\left\langle \genfrac{}{}{0pt}{}{n-1}{j}\right\rangle|G\right)+\pi_B\cdot\mathbb{P}\left(\left\langle \genfrac{}{}{0pt}{}{n-1}{j-1}\right\rangle|B\right)\right]\right\}
\end{aligned} \tag{14}$$

Algorithm 1 Source Rate Control

```
Input: $\{RTT, \mu, \pi_B, \xi_B\}$, T, $\mathcal{N}$, $\mathcal{F}$, $\Delta V = 50$ Kbps;
Output: Source coding rate $V$;
Initialize: $\mathsf{D}_{\text{temp}} = \infty$,
    $V_{\text{int}} = \frac{\frac{1}{\pi_B} + \frac{\bar{\omega}^2}{1+\bar{\omega}^2}}{\omega' \cdot RTT + [1+2\pi_B+4(\pi_B)^2] \cdot T_0 + RTT \cdot [1+\log(\omega/4)]}$,
    $V_{\max} = \frac{MSS}{RTT \cdot \sqrt{\frac{2\times\pi_B}{3}} + 12\times\sqrt{\frac{3\times\pi_B}{8}} \cdot \pi_B \cdot [1+32\times(\pi_B)^2]}$;
Estimate the distortion parameters $V_0, \alpha, \beta$ using trial coding;
while $0 < V < \min\{V_{\max}, \frac{\mu}{(1+R)}\}$ && $D < D_{temp}$ do
    Estimate $(R, S)$ using Algorithm 2;
    $k = \left\lceil \frac{V \cdot \mathcal{F}}{\mathcal{N}} \right\rceil, n = \lceil k \cdot (1+R) \rceil$;
    $\pi_{\text{t}}(n,k) =$ Equation (14), $\pi_{\text{o}}(n) = \exp\left[-\frac{\text{T}}{\text{d}_{\text{N}}(n)+\text{d}_{\text{T}}(n)}\right]$;
    $\Pi(n) = \pi_{\text{t}}(V,n,k) + [1-\pi_{\text{t}}(V,n,k)] \cdot \pi_m^{\text{o}}(V,n)$;
    $\mathsf{D}_{\text{src}} = \frac{\alpha}{V-V_0}$, $\mathsf{D}_{\text{chl}} = \beta \cdot \Pi$;
    $\mathsf{D} = \mathsf{D}_{\text{src}} + \mathsf{D}_{\text{chl}} = \frac{\alpha}{V-V_0} + \beta \cdot \Pi$;
    if $\frac{dD(V)}{dV} > 0$ then
        $V = V - \Delta V$, $\mathsf{D}_{\text{temp}} = \mathsf{D}$;
    end
    $V = V + \Delta V$, $\mathsf{D}_{\text{temp}} = \mathsf{D}$;
end
```

Proposition 1: The worst-case time complexity for Algorithm 1 is $O((V_{\max} - V_{\min}/\Delta V) \cdot ((\text{MSS}/\Delta S) + (1/\Delta R)))$, where $V_{\max}$ denotes the upper limit of the video encoding rate. ΔV, ΔS, and ΔR denotes the step size for the source rate control, packet size adjustment, and redundancy adaption, respectively.

Proof: There are at most $(V_{\max} - V_{\min}/\Delta V)$ iterations in the while loop to reduce the sum of total distortion. In each iteration, the time complexity for the FEC coding adaptation is $O((\text{MSS}/\Delta S) + (1/\Delta R))$ (Proposition 2). Thus, the worst case time complexity of Algorithm 1 is $O((V_{\max} - V_{\min}/\Delta V) \cdot ((\text{MSS}/\Delta S) + (1/\Delta R)))$. ■

C. FEC Coding Adaptation

It is a challenging issue to determine the FEC redundancy value due to the tradeoff between FEC recoverability and delay performance. On one side, increasing the number of parity packets is able to mitigate the transmission loss, but enlarges the end-to-end delay. If fewer parity packets are appended, the receiver could start the decoding process earlier, but at the sacrifice of recovery ability. Therefore, the goal of the proposed algorithm is to append just-enough parity packets to minimize the absolute difference between Π and $(n - k/n)$, that is

$$
\begin{aligned}
(\mathsf{P3}): \{S, R\} = \arg\min & \left| \Pi(V, S, R) - \frac{n-k}{n} \right| \\
\text{s.t. } & \frac{n \cdot S}{\mu \cdot (1-\pi_B)} + \frac{(\mu - \bar{V}) \cdot \text{RTT}}{2 \times (\mu - V)} + d_T(n, S) \leq T \\
& k = \left\lceil \frac{V \cdot \mathcal{F}}{\mathcal{N} \cdot S} \right\rceil, \quad n = \lceil k \cdot (1+R) \rceil.
\end{aligned}
$$

First, we consider the determination of FEC packet size. Assume that the effective loss rate of frame m will approximate the tolerable loss rate $(n - k/n)$ after the FEC redundancy adaption. In this case, the video frame data can be successfully recovered at the receiver side.

To determine the value of n, the expression of $\Pi(n)$ can be obtained with (7), (9), and (14). To determine the number of source packets in current GoP (FEC coding block), we choose the FEC packet size (S) among four values: 1000, 750, 550, and 250 B [39]. However, the minimum FEC block size is 10, and the expression of k_m is

$$
k = \max\left\{ \left\lceil \min_{S \in \{250,550,750,100\}} \left\{ \frac{S_m}{S} \right\} \right\rceil, 10 \right\}.
$$

Finally, we can obtain the value of n_m that enables $\Pi_m(n_m)$ to approximate $(n_m - k_m/k_m)$. In particular, the maximum value of n_m for I frame is expressed as

$$
n_{\max} = \left\lfloor \frac{\mu \cdot (1 - \pi_B) \cdot T}{S} \right\rfloor.
$$

We assume that R denotes the total number of the introduced FEC parity packets for the $(M-1)$ P frames and $R(m)$ denotes the sum of redundant packets for the mth frame. It can be obtained with $R = n - k$. As all the video frames in the same GoP are converted into the FEC packets of the same size (S), the value of k is $\lceil (V \cdot \mathcal{N}/S \cdot \mathcal{F}) \rceil$, and the upper limit of R can be determined with

$$
R = \left\lfloor \frac{\mu \cdot (1 - \pi_B) \cdot (M-1) \cdot T}{S} \right\rfloor - n_{\max}. \tag{16}
$$

Let the notation D represent the end-to-end distortion for all the P frames. The goal of the FEC coding is to appropriately allocate the R redundant packets to minimize the distortion D. Then, the parity packets can be appended to each video frame based on the results of $R(m), 2 \leq m \leq M$.

An important problem in designing FEC coding scheme is to distinguish the congestion losses from random losses [10], [43]. In the congested network status, the available bandwidth will shrink, and more data packets encounter channel losses. If we try to inject more FEC redundant packets, it will only add fuel to the fire and induce further packet losses. This vicious congestion circle is already observed in [43]. In the design of ESCOT, the identification of network congestions is based on the average $\overline{\text{RTT}}$, standard deviations σ_{RTT}, and the number of consecutive losses m [44]

$$
\overline{\text{RTT}} = (31/32) \times \overline{\text{RTT}} + (1/32) \times \text{RTT}
$$

and

$$
\sigma_{\text{RTT}} = (15/16) \times \sigma_{\text{RTT}} + (1/16) \times |\text{RTT} - \overline{\text{RTT}}|.
$$

Algorithm 2 summarizes the process of the FEC coding adaptation for redundancy value and packet size. The time complexity is concluded in Proposition 2.

Proposition 2: The FEC coding adaptation algorithm is a polynomial-time solution with the complexity of $O((\text{MSS}/\Delta S) + (1/\Delta R))$, in which MSS denotes the maximum segment size. ΔS and ΔR denote the adjustment step for packet size and FEC redundancy, respectively.

Algorithm 2 FEC Coding Adaptation

```
Input: $\{RTT, \mu, \pi_B, \xi_B\}$, $V$, T, $\Delta S = 50$ Bytes,
       $\Delta R = 0.5\%$, $m$ (number of consecutive lost
       packets);
Output: $R$, $S$;
Initialize: $R = 0, S = MSS$;
$\overline{RTT} = (31/32) \times \overline{RTT} + (1/32) \times RTT$
$\sigma_{RTT} = (15/16) \times \sigma_{RTT} + (1/16) \times |RTT - \overline{RTT}|$;
Con1 = $(m == 1)\ \&\&\ (RTT < \overline{RTT} - \sigma_{RTT})$;
Con2 = $(m == 2)\ \&\&\ (RTT < \overline{RTT} - \sigma_{RTT}/2)$;
Con3 = $(m == 3)\ \&\&\ (RTT < \overline{RTT})$;
Con4 = $(m > 3)\ \&\&\ (RTT < \overline{RTT} - \sigma_{RTT}/2)$;
if (Con1||Con2||Con3||Con4 == $False$) then
    Update loss parameters $\pi_B$ and $1/\xi_B$;
end
while $\{(S > 0)\ \&\&\ (\mathbb{E}(\mathrm{d_E}) < \mathrm{T})\}$ do
    $S' = S,\ S = S - \Delta S,\ k = \left\lceil \frac{V\cdot\mathcal{N}}{S\cdot\mathcal{F}} \right\rceil$;
    $\pi_{\mathrm{o}}(R, S) = \exp\left\{-\frac{\mathrm{T}\cdot n\cdot S\cdot(\mu-\lambda)}{n\cdot S\cdot[\rho+(\mu-\lambda)\cdot\mathbb{E}(d_{\mathrm{T}})]+\mu\cdot(1-\pi_B)\cdot(\mu-\lambda)}\right\}$;
    $\pi_{\mathrm{t}}$ = Equation (14);
    $\Pi(R, S) = \pi_t(n, k, S) + [1 - \pi_t(n, k, S)] \cdot \pi_o(n, k, S)$;
    Update the end-to-end distortion $\mathsf{D}(R, S)$;
    Procedure: $R$ = FEC_Redundancy_Adaptation ($V$, $S$);
    if $\{\Pi(R, S) > \Pi(R, S')\}$ then
        break;
    end
end
Procedure: FEC_redundancy_adaptation
while $\left\{\max\left\{\lambda, \frac{n\cdot S}{\mathrm{T}}\right\} \cdot (1 + \Delta R) < \mu\right\}$ do
    $R = R + \Delta R$;
    Update the effective loss rate $\Pi(R)$;
    $\mathsf{D} = \mathsf{D}_{\mathrm{src}}(V, R, S) + \mathsf{D}_{\mathrm{chl}}(V, R, S)$;
end
Return $R$, $S$;
```

Proof: In the worst case, there are $(\mathrm{MSS}/\Delta S)$ times of operations in the while loop to search for the FEC packet size. The maximum number of calculations to obtain the FEC redundancy value is $(1/\Delta R)$. Therefore, the time complexity of Algorithm 2 is $O((\mathrm{MSS}/\Delta S) + (1/\Delta R))$. ■

V. Performance Evaluation

In this section, we present the performance results obtained through extensive emulations over the Exata [45] platform to validate the efficacy of the ESCOT. Exata is an advanced edition of QualNet [46], and we conduct the emulations involving real-time H.264 video streaming. The semiphysical emulation differs from the traditional trace-driven simulations and is able to mimic real data transfer with high fidelity.

First, we describe the evaluation methodology that includes the emulation setup, reference schemes, and performance metrics. Then, we present and analyze the emulation results in wired networks. The componentwise validation is shown in Section V-C.

TABLE III

Parameter Configurations of Wireless Networks

WLAN parameter (Trajectory I, II)	Value
Average channel bit rate	8 Mbps
Slot time	10 μs
Maximum contention window	32
available capacity	2.5 (I), 3 (II) Mbps
average loss rate	6%
average burst loss length	7.5 ms
Cellular parameter (Trajectory III, IV)	**Value**
Target SIR value	10 dB
Orthogonality factor	0.4
Common control channel power	33 dB
Maximum power of BS	43 dB
Total cell bandwidth	3.84 Mc/s
Inter/intra cell interference ratio	0.55
Background noise power	-106 dB
available capacity	3.5 (III), 4 (IV) Mbps
average loss rate	3%
average burst loss length	5 ms

A. Evaluation Methodology

1) Network Emulator: Exata 2.1 [45] is adopted as the network emulator, and Fig. 7 shows the system architecture for performance evaluation in wireless networks. The emulation topology consists of two communication terminals: cross traffic generators and the core networks. The sender and the receiver are mapped to real local-network computers, which are connected to the emulation server via the Exata 2.1 Connection Manager. As shown in Fig. 7, each router in the core networks is connected to an edge node, which is used for generating cross traffic to emulate network dynamics. The edge nodes are linked to four traffic generators that produce cross traffic with a Pareto distribution. The packet sizes of background traffic are varied to mimic the real traces collected on the Internet: 43% are large (>1400 B), 17% are medium [(144, 1400) B], and 40% are small (<144 B) [47]. The aggregate cross traffic loads imposed on the network paths are similar and vary randomly between 20% and 40% of the bottleneck links' bandwidth to create the traffic variations. We conduct the semiphysical emulations in both cellular and Wi-Fi networks, which represent the most commonly used access options for mobile users. The main parameter configurations of cellular and Wi-Fi networks are listed in Table III [9], [61].

The mobile terminal is set to move along four predefined trajectories (at the moving speed of 2, 4, and 8 m/s) shown in Fig. 7. These trajectories represent the commonly used cellular and Wi-Fi networks. The available capacities of both wireless networks with regard to different mobile trajectories are also specified in Table III.

Since this paper employs the delay performance model in [24], we follow the same TCP settings to use the New Reno and disable the Nagle algorithm [38]. Table IV summarizes the configurations of TCP parameters.

2) Video Codec: We use the H.264/AVC standard reference software JM 18.2 [48] as the video codec. In order to implement the emulations involving real-time encoded video streaming, we integrate the source code of JM with Exata and develop an application-layer protocol named Video Communication. This application adopts TCP as the transport-layer protocol. The development steps can be referred to Exata Programmers' Guide [45]. The video test sequences are

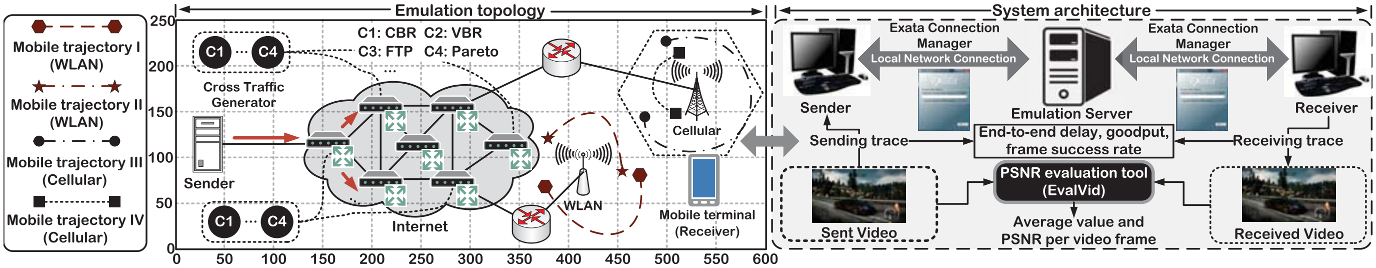


Fig. 7. System architecture and emulation topology for performance evaluation.

TABLE IV
CONFIGURATION OF TCP PARAMETER

TCP parameter	Value
TCP variant	New Reno
Use Nagle algorithm	No
Use RFC 1323	No
Enable TCP trace	Yes
Send/Receive buffer size	64 KBytes

(a) (b) (c) (d)

Fig. 8. Gaming video sequences rendered from the Unity engine. (a) *Shooter*. (b) *Robot Lab*. (c) *Race*. (d) *Viking*.

Shooter, Robot Lab, Race, and *Viking* rendered from the Unity engine, as shown in Fig. 8. Each of the selected sequences features a different pattern of temporal motion and spatial characteristic. We concatenate the video sequences to be 30 000 frame-long in order to obtain statistically meaningful results. The video streaming is encoded at 60 f/s and a GoP includes 30 frames with the structure of IPP...P. Since the B (bidirectional) frame needs the prediction from the subsequent frames, it often incurs a delay larger than the playout interval between two consecutive frames [62]. Therefore, the bidirectional prediction mode should be disabled for real-time applications to reduce video coding latency [62]. We consider the low-delay encoded video stream exclusively consists of I and P frames (without B frames).

The delay constraint T is set to be 500 ms for each FEC block to prevent the playback buffer starvation [14]. The video encoding rates are configured as 2 and 3 Mb/s in the two trajectories for the reference schemes, while ESCOT dynamically adjusts the source rate during the video delivery.

3) Reference Schemes: We compare the performance of ESCOT with the following representative transmission schemes for real-time video applications.

1) *APHIS [9]:* This framework adjusts the traffic load with the priority-aware frame selection. A sub-GoP level FEC coding scheme is employed to provide unequal protection for the I frames.
2) *JSFR [17]:* The JSFR selection scheme dynamically adjusts the source and FEC coding rates to minimize the end-to-end video distortion.
3) *New Reno:* TCP New Reno has the ability to recover from multiple losses within the same loss window, and thus avoids frequent retransmission TO events. This reference scheme is to the video transmission performance without introducing the source–FEC coding.

4) Benchmarks: The following benchmarks are used for the performance comparison of the evaluated schemes.

1) PSNR is the standard metric to measure objective video quality. This parameter is expressed as a function of the MSE between the original and the reconstructed video frames. If a video frame is dropped during transmission or past the deadline, it is considered lost and is concealed by copying from the last received frame before it.
2) Game mean opinion score (GMOS) [52] is a subjective quality metric that reflects the MGUEs. This parameter is based on the MGUE model introduced in [52].
3) The end-to-end delay of a video frame consists of the transmission delay and the holding time at both the server and client sides. This duration is counted from the generation time of a video frame to the time when it can be decoded.
4) Goodput [49] represents the amount of useful information bits successfully received by the destinations within the imposed deadline.
5) Frame success rate [50] represents the percentage of video frames successfully received by the destination within the delay constraint. This metric reflects the level of playback fluency perceived by end users.

B. Evaluation Results

1) PSNR: Fig. 9(a) shows the average PSNR values and confidence intervals for all the transmission schemes in the mobile emulation scenarios. The proposed ESCOT achieves higher values with lower variations than the reference schemes in the trajectories of both Wireless Local Area Network (WLAN) and cellular networks. In particular, the video quality received in cellular networks is obviously higher than that in Wi-Fi networks, because the cellular link is able to better

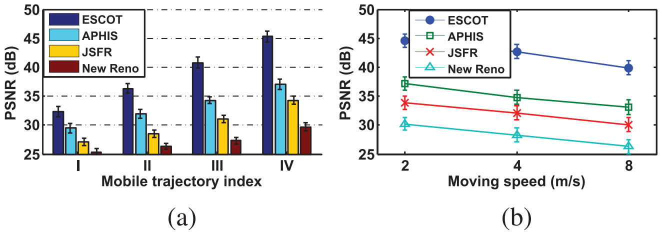


Fig. 9. Comparison of average PSNR results in different emulation scenarios. (a) Different trajectories. (b) Different moving speeds.

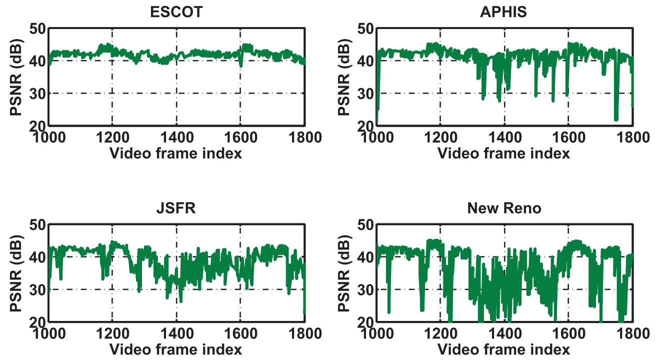


Fig. 10. Comparison of PSNR per video frame indexed from 1000 to 1800 measured from the *Shooter* sequence.

sustain user mobility [14], [61]. The higher bandwidth in cellular networks enables the proposed solution to enlarge the source coding rate to improve video PSNR. Fig. 9(b) presents the video quality results of different moving speeds along mobile trajectory III. The ESCOT outperforms the reference FEC coding schemes, and the performance gaps become larger while the moving speed increases. These results are expected because the source rate control algorithm in ESCOT is able to dynamically adjust video coding rate according to the bandwidth variations during client mobility.

In order to have a close-up view of the measured objective video quality during the playback process, Fig. 10 shows the instantaneous PSNR values for the frames indexed from 1000 to 1800. We can observe that a more fluent video streaming is received with the proposed ESCOT. Although the PSNR values of ESCOT may be lower for some frames, most of them are caused by the frame drop for bandwidth conservation to protect higher priority frames. The presented microscopic results in Fig. 10 significantly indicate fewer video stalls and glitches for the proposed FEC coding scheme.

To compare the subjective video quality, Fig. 11 shows the typical results of the received video frames with the competing transmission schemes. It can be observed that the video frame received with the ESCOT is well protected. The images received with the reference schemes are damaged to different levels due to the transmission losses and error propagations.

2) GMOS: To depict the user experience in mobile cloud gaming environment, Fig. 12(a) presents the average GMOS values in different trajectories. The factors include the game type, resolution, frame rate, delay, PSNR, and packet loss rate [52]. As the available network bandwidth increases (in trajectories III and IV), the user perceives higher gaming experiences. The instantaneous GMOS values during the interval of [100, 400] s, while the mobile terminal is moving along trajectory I, are shown in Fig. 12(b). The results indicate that the ESCOT is able to achieve better gaming experiences than the reference transmission schemes.

3) Delay: Fig. 13(a) shows the average end-to-end delay of the video frames measured from different emulation scenarios. The results' trend is almost opposite to that shown in Fig. 9(a). This is because the real-time video quality is inversely proportional to the end-to-end delay. ESCOT achieves appreciable improvement in the delay performance than the competing schemes as we analyze the TCP-connection state in the FEC coding adaption. This strategy helps to mitigate throughput fluctuations and reduce TCP-level latency. During the video playback, the receiver side is plagued with disruptions while using the three reference schemes. Fig. 13(b) shows the instantaneous values of end-to-end delay during the interval of $[100, 350]$ s. ESCOT substantially mitigates the delay jitter than other transmission schemes, and this guarantees the seamless streaming video received by the mobile terminal.

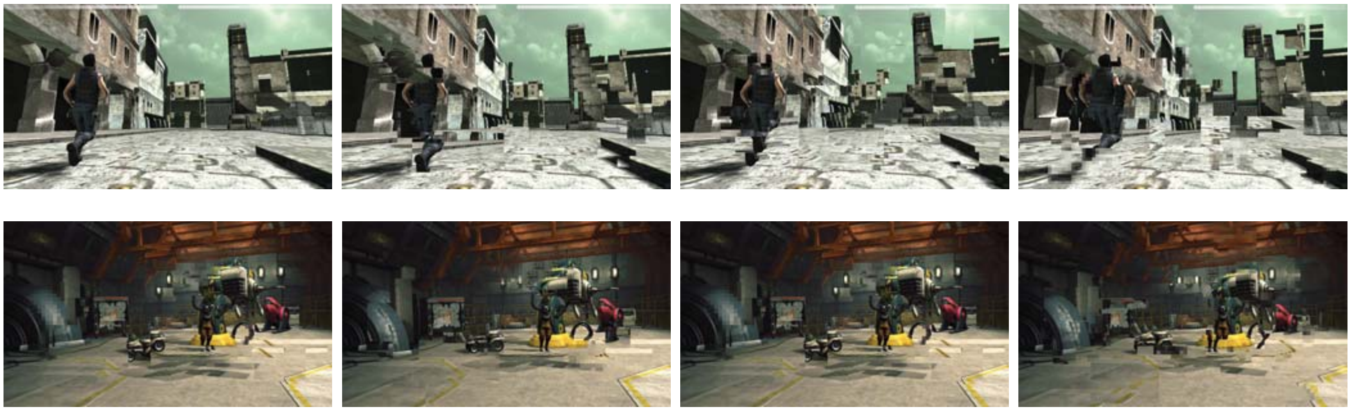
In order to compare the microscopic results of the delay performance, Fig. 14 shows the cumulative distribution functions (CDFs) of the end-to-end delay with regard to different mobile trajectories. During the video transmission process, the receiver side is often plagued with playback disruptions and stalls while using the reference schemes. A significantly smoother video streaming can be obtained with the proposed ESCOT scheme.

4) Goodput: Fig. 15(a) shows the average results of goodput measured from the different emulation scenarios. Expectedly, the results' pattern is similar to the average PSNR results shown in Fig. 9(a). According to the literature [49], the goodput performance is a key parameter in guaranteeing and optimizing the quality of real-time multimedia traffic.

Fig. 15(b) sketches the evolutions of the instantaneous goodput values during the interval of $[100, 350]$ s, while the video encoding rate is 2.5 Mb/s. ESCOT achieves higher and smoother goodput than other FEC coding schemes as it adapts the FEC redundancy and packet size according to the estimated TCP-connection state and time-varying network status. The EEP scheme performs the FEC coding at the GoP level, which leads to larger end-to-end delay and lower goodput than other transmission schemes.

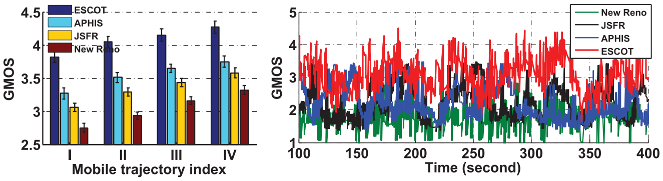
The bandwidth consumption of the evaluated schemes is presented in Fig. 16. It can be observed that the bandwidth consumptions of ESCOT, APHIS, and JSFR are close to each other. New Reno consumes less bandwidth as it does not use FEC coding. We can arrive at the conclusion that the superiority of ESCOT is not due to the higher bandwidth usage, but on the effectiveness of the source–FEC coding algorithms.

5) Frame Success Rate: Fig. 17(a) shows the frame success rate in the four mobile trajectories with the encoding frame rate of 60 f/s. This metric is of vital significance to the real-time gaming video quality, since it guarantees the received frame rate and playback fluency perceived by end users. ESCOT significantly achieves higher frame success rate than the reference schemes as it adjusts the source rate



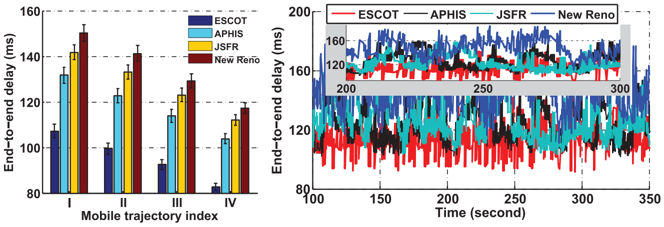
(a) (b) (c) (d)

Fig. 11. Comparison of subjective video quality measured from the *Shooter* and *Robot Lab* sequences. (a) ESCOT. (b) APHIS. (c) JSFR. (d) New Reno.



(a) (b)

Fig. 12. Evaluation results of GMOS. (a) Average values. (b) Instantaneous values.



(a) (b)

Fig. 13. Comparison of end-to-end delay performance. (a) Average delay. (b) Instantaneous values.

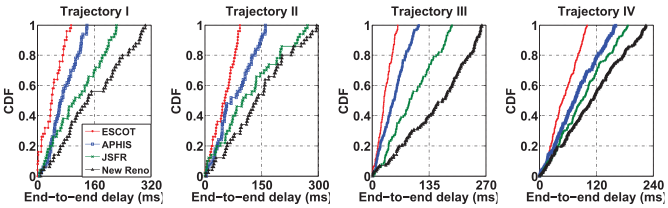
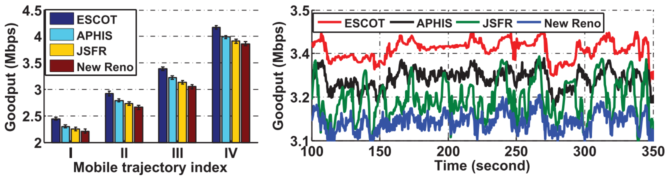
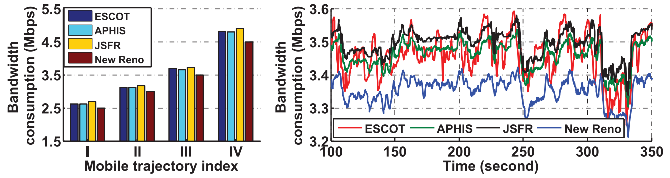


Fig. 14. CDF of end-to-end delay.



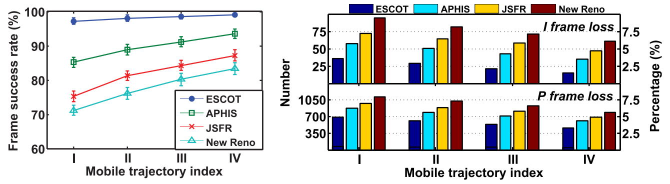
(a) (b)

Fig. 15. Comparison of goodput performance. (a) Average values. (b) Instantaneous values.



(a) (b)

Fig. 16. Comparison of bandwidth consumption. (a) Average values. (b) Instantaneous values.



(a) (b)

Fig. 17. Comparison of frame success rate and loss number. (a) Average ratio. (b) Number and ratio of lost frames.

and FEC redundancy according to the network status and TCP-connection state. APHIS proactively drops some less important frames, while network congestion and bandwidth shrink occur to protect the more important frames.

In Fig. 17(b), the number of lost frames measured in the four mobile trajectories is presented. ESCOT mitigates the frame drop problems by dynamically adjusting the source rate and FEC redundancy in case of bandwidth shrink and random packet losses. Due to the large size of the I frames, it is more likely for these important frames to encounter losses if using the reference schemes due to the lack of rate control. APHIS adjusts the video traffic load by selectively dropping the lower-priority frames, and thus degrades the fluency level of gaming video. However, APHIS delivers more I frames than the JSFR and New Reno as it considers the frame priority in code rate adaption. Table V compares the distribution of frame loss events, while the terminal is moving along the fourth mobile trajectory. The consecutive frame drops also induce noticeable quality degradations, as shown in Fig. 3.

TABLE V
DISTRIBUTION OF FRAME LOSS EVENTS IN TRAJECTORY IV

Scheme	Number of consecutive losses				Total
	1	2	3	> 3	
ESCOT	260(56%)	102(22%)	60(13%)	42(9%)	465
APHIS	263(43%)	177(29%)	98(16%)	74(12%)	612
JSFR	214(31%)	241(35%)	138(20%)	96(14%)	690
New Reno	203(26%)	304(39%)	157(20%)	118(15%)	782

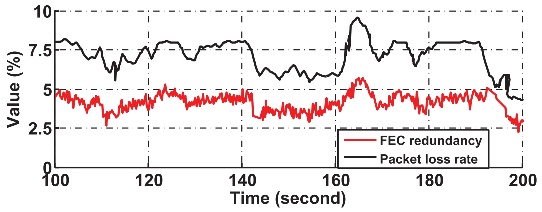


Fig. 18. Evolutions of instantaneous packet loss rate and FEC redundancy value during the interval of [100, 200] s.

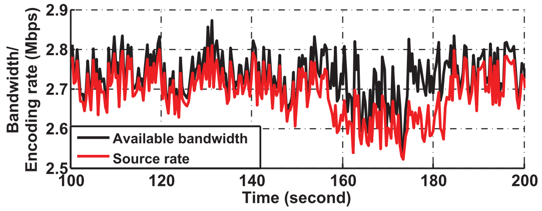
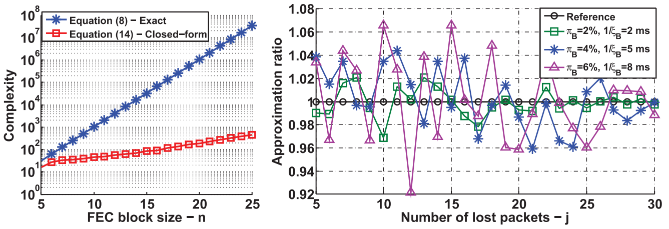


Fig. 19. Evolutions of available bandwidth and video encoding rate during the interval of [100, 200] s.



(a) (b)

Fig. 20. Performance of the approximate analysis. (a) Time complexity. (b) Approximation ratio.

C. Componentwise Validation

Fig. 18 presents the instantaneous values of packet loss rate and FEC redundancy during the interval of [100, 200] s. It can be observed the proposed FEC coding scheme is able to dynamically adjust the redundancy value according to the time-varying loss rate. Besides, the FEC redundancy is less than the estimated packet loss rate, since it distinguishes congestion losses and random losses.

Fig. 19 sketches the evolutions of available bandwidth and video encoding rate during the same interval. ESCOT is able to take full advantage of the available network resources while respecting the TCP friendliness.

Fig. 20(a) compares the time complexity of the exact and close-form equations for transmission loss rate π_t with regard to different FEC block sizes n. As expected, the approximate analysis is able to significantly achieve lower complexity, and the superiority enlarges as the coding block size increases. The approximation ratios with regard to different numbers of lost packets j are presented in Fig. 20(b). We can observe that such analysis achieves satisfactory estimation accuracy with regard to different packet loss rates and burst lengths.

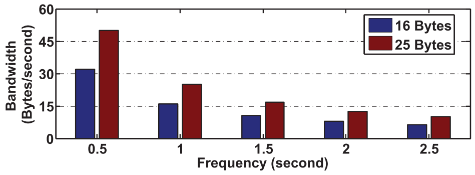


Fig. 21. Bandwidth required for feedback information.

The required bandwidth with regard to different feedback frequencies is presented in Fig. 21. In our emulations, the feedback information is sent back to the client for every 0.5 s, and the feedback packet size is 16 B. Therefore, the bandwidth required for the feedback information is 32 B/s. In general, the estimation accuracy can be improved by increasing the feedback frequency.

VI. CONCLUSION AND DISCUSSION

Recent years have witnessed the significant growth of cloud gaming applications in the computer gaming industry. Video transmission is a key technical problem in developing an effective cloud gaming platform. Because of the special features in firewall traversal and network friendliness, TCP is commonly adopted as the transport-layer protocol in popular cloud gaming systems to stream gaming video. It is extremely challenging to deliver mobile cloud gaming video over TCP in wireless networks with regard to the stringent QoS requirements, time-varying channel status, and special TCP features.

This paper tackles the critical problem by developing an application-layer source–FEC coding scheme for mobile cloud gaming video over TCP. The proposed ESCOT is distinct from the existing FEC coding schemes in which we analyze and leverage the TCP characteristics to optimize the overall quality of the real-time gaming video. Through modeling and analysis, we develop solutions for effective loss rate estimation, source rate control, and FEC coding adaptation. Emulation results show that the ESCOT is able to substantially reduce the end-to-end delay and improve video PSNR over the reference schemes.

As future work, we will consider the following aspects.

1) Improve the adaptive source–FEC coding scheme to jointly optimize distortion and experience (e.g., video stall [51]) of real-time gaming video over wireless networks.
2) Introduce the speculation-based technology [8] to further improve the delay performance based on the user inputs and past gaming events.

APPENDIX
DERIVATIONS FOR EFFECTIVE LOSS RATE

First, we provide the theoretical analysis on the transmission loss rate based on the Gilbert model and the continuous-time Markov chain. Assume that the notation c denotes a $n-$tuple that represents a particular failure configuration during the transmission. If the ith FEC packet in a coding block is lost, then $c_i = B,\ 1 \leq i \leq n$ and vice versa. By considering all the possible configurations of c, we can compute the transmission loss rate π_t as

$$\pi_t = \frac{1}{k} \sum_{\forall c} \mathcal{L}(c) \cdot \mathbb{P}(c) \tag{17}$$

where $\forall c$ represents for all the possible combinations of c. $\mathcal{L}(c)$ represents the number of lost FEC packets for a given c. For the systematic FEC(n, k) coding, $\mathcal{L}(c)$ is expressed as

$$\mathcal{L}(c) = \begin{cases} 0, & \text{if } \sum\limits_{i=1}^{n} 1_{\{c_i = B\}} \leq n - k \\ \sum\limits_{i=1}^{k} 1_{\{c_i = B\}}, & \text{otherwise.} \end{cases}$$

Let $\mathbb{P}(c)$ denote the probability of the path failure configuration in the transmission of the n packets. The derivation of $\mathbb{P}(c)$ for the continuous-time Gilbert loss model is the product of the state transition probabilities. We use the symbol $\mathsf{F}_{\langle i,j \rangle}(\theta)$ to express the probability of transition from state i to j in time interval θ

$$\mathsf{F}_{\langle i,j \rangle}(\theta) = \mathbb{P}[\mathcal{X}(\theta) = j | \mathcal{X}(0) = i]. \tag{18}$$

According to the property of continuous-time Markov chain, we have the following state transition matrix:

$$\begin{bmatrix} \mathsf{F}_{\langle G,G \rangle}(\theta) & \mathsf{F}_{\langle G,B \rangle}(\theta) \\ \mathsf{F}_{\langle B,G \rangle}(\theta) & \mathsf{F}_{\langle B,B \rangle}(\theta) \end{bmatrix} = \begin{bmatrix} \pi_G + \pi_B \cdot \kappa & \pi_B - \pi_B \cdot \kappa \\ \pi_G - \pi_G \cdot \kappa & \pi_B + \pi_G \cdot \kappa \end{bmatrix}$$

in which $\kappa = \exp[-(\xi_B + \xi_G) \cdot \theta]$. Now, the expression of $\mathbb{P}(c_i)$ can be obtained, e.g., for $n = 3$ and $c_3 = B | c_2 = B | c_1 = G$, we have

$$\mathbb{P}[c_3 = B | c_2 = B | c_1 = G] = \pi_G \cdot \mathsf{F}_{\langle G,B \rangle}(\theta_1) \cdot \mathsf{F}_{\langle B,B \rangle}(\theta_2)$$

where θ_i represents the departure interval between the ith and the $(i + 1)$th FEC packets. Therefore, $\mathbb{P}(c)$ is calculated with

$$\mathbb{P}(c) = \pi_{c_1} \cdot \prod_{i=1}^{n-1} [\mathsf{F}_{\langle c_i, c_{(i+1)} \rangle}(\theta_i)]. \tag{19}$$

After a sequence of algebraic manipulations, the expression of transmission loss rate is presented as

$$\pi_t = \frac{1}{k} \cdot \sum_{\forall c} \left\{ \mathcal{L}(c) \cdot \pi_{c_1} \cdot \prod_{i=1}^{n-1} [\mathsf{F}_{\langle c_i, c_{(i+1)} \rangle}(\theta_i)] \right\}. \tag{20}$$

Equation (20) enables us to obtain the expectation value of π_t for the given transition probabilities. In Section IV-A, an approximate analysis is presented to derive the closed-form transmission loss rate based on (20).

To model the overdue loss rate (π_o), we consider the communication network as a queuing system with a single server. Recent measurement studies reveal that the video traffic pattern follows the Markov-modulated process [40]. Therefore, we model the queuing delay with the $M/G/1$ model, and the overdue loss probability of packets over the communication network is expressed as [14], [36]

$$\pi_o = \mathbb{P}[d_E(n) > T] \approx \exp\left\{ -\frac{T}{d_E(n)} \right\} \tag{21}$$

where $d_E(n)$ denotes the expected value of end-to-end delay and can be obtained with (3). Therefore, the probability of expired arrival of the FEC packets is expressed with

$$\pi_o \approx \exp\left[-\frac{T}{d_N(n) + d_T(n)} \right]. \tag{22}$$

Based on (20) and (22), the expectation value of Π can be estimated.

Acknowledgment

The authors would like to thank the anonymous reviewers for their valuable comments and the Associate Editor for efforts to coordinate this peer review.

References

[1] C.-Y. Huang, C.-H. Hsu, Y.-C. Chang, and K.-T. Chen, "GamingAnywhere: An open cloud gaming system," in *Proc. 4th ACM MMSys*, 2013, pp. 36–47.

[2] D. Wu, Z. Xue, and J. He, "iCloudAccess: Cost-effective streaming of video games from the cloud with low latency," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 24, no. 8, pp. 1405–1416, Aug. 2014.

[3] Z. Xue, D. Wu, J. He, X. Hei, and Y. Liu, "Playing high-end video games in the cloud: A measurement study," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 25, no. 12, pp. 2013–2025, Dec. 2015.

[4] S.-P. Chuah, C. Yuen, and N.-M. Cheung, "Cloud gaming: A green solution to massive multiplayer online games," *IEEE Wireless Commun.*, vol. 21, no. 4, pp. 78–87, Aug. 2014.

[5] OnLive. [Online]. Available: http://www.onlive.com/, accessed 2015.

[6] CloudUnion. [Online]. Available: http://www.cloudunion.cn/, accessed 2015.

[7] *Distribution and Monetization Strategies to Increase Revenues From Cloud Gaming*. [Online]. Available: http://www.cgconfusa.com/report/documents/Content-5minCloudGamingReportHighlights.pdf, accessed 2015.

[8] K. Lee *et al.*, "Outatime: Using speculation to enable low-latency continuous interaction for mobile cloud gaming," in *Proc. 13th Annu. Int. Conf. MobiSys*, 2015, pp. 151–165.

[9] J. Wu, C. Yuen, N.-M. Cheung, J. Chen, and C. W. Chen, "Enabling adaptive high-frame-rate video streaming in mobile cloud gaming applications," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 25, no. 12, pp. 1988–2001, Dec. 2015.

[10] C. Yu, Y. Xu, B. Liu, and Y. Liu, "'Can you SEE me now?' A measurement study of mobile video calls," in *Proc. IEEE INFOCOM*, Apr./May 2014, pp. 1456–1464.

[11] *Web Real-Time Communication*. [Online]. Available: http://www.webrtc.org/, accessed 2015.

[12] *HTTP Live Streaming*. [Online]. Available: https://developer.apple.com/streaming/, accessed 2015.

[13] K.-T. Chen, Y.-C. Chang, H.-J. Hsu, D.-Y. Chen, C.-Y. Huang, and C.-H. Hsu, "On the quality of service of cloud gaming systems," *IEEE Trans. Multimedia*, vol. 16, no. 2, pp. 480–495, Feb. 2014.

[14] J. Wu, B. Cheng, C. Yuen, Y. Shang, and J. Chen, "Distortion-aware concurrent multipath transfer for mobile video streaming in heterogeneous wireless networks," *IEEE Trans. Mobile Comput.*, vol. 14, no. 4, pp. 688–701, Apr. 2015.

[15] M. Panda, H. L. Vu, M. Mandjes, and S. R. Pokhrel, "Performance analysis of TCP NewReno over a cellular last-mile: Buffer and channel losses," *IEEE Trans. Mobile Comput.*, vol. 14, no. 8, pp. 1629–1643, Aug. 2015.

[16] D. Moltchanov, "A study of TCP performance in wireless environment using fixed-point approximation," *Comput. Netw.*, vol. 56, no. 4, pp. 1263–1285, 2012.

[17] P. Frossard and O. Verscheure, "Joint source/FEC rate selection for quality-optimal MPEG-2 video delivery," *IEEE Trans. Image Process.*, vol. 10, no. 12, pp. 1815–1825, Dec. 2001.

[18] X. Yu, J. W. Modestino, R. Kurceren, and Y. S. Chan, "A model-based approach to evaluation of the efficacy of FEC coding in combating network packet losses," *IEEE/ACM Trans. Netw.*, vol. 16, no. 3, pp. 628–641, Jun. 2008.

[19] Y. Li, Y. Zhang, L. Qiu, and S. Lam, "SmartTunnel: Achieving reliability in the Internet," in *Proc. IEEE INFOCOM*, May 2007, pp. 830–838.

[20] M. Claypool, D. Finkel, A. Grant, and M. Solano, "On the performance of OnLive thin client games," *Multimedia Syst.*, vol. 20, no. 5, pp. 471–484, 2014.

[21] J. Wu, C. Yuen, and J. Chen, "Leveraging the delay-friendliness of TCP with FEC coding in real-time video communication," *IEEE Trans. Commun.*, vol. 63, no. 10, pp. 3584–3599, Oct. 2015.

[22] H.-P. Shiang and M. van der Schaar, "A quality-centric TCP-friendly congestion control for multimedia transmission," *IEEE Trans. Multimedia*, vol. 14, no. 3, pp. 896–909, Jun. 2012.

[23] O. Habachi, H.-P. Shiang, M. van der Schaar, and Y. Hayel, "Online learning based congestion control for adaptive multimedia transmission," *IEEE Trans. Signal Process.*, vol. 61, no. 6, pp. 1460–1469, Mar. 2013.

[24] E. Brosh, S. A. Baset, V. Misra, D. Rubenstein, and H. Schulzrinne, "The delay-friendliness of TCP for real-time traffic," *IEEE/ACM Trans. Netw.*, vol. 18, no. 5, pp. 1478–1491, Oct. 2010.

[25] P. U. Tournoux, E. Lochin, J. Lacan, A. Bouabdallah, and V. Roca, "On-the-fly erasure coding for real-time video applications," *IEEE Trans. Multimedia*, vol. 13, no. 4, pp. 797–812, Aug. 2011.

[26] S. Ahmad, R. Hamzaoui, and M. Al-Akaidi, "Adaptive unicast video streaming with rateless codes and feedback," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 20, no. 2, pp. 275–285, Feb. 2010.

[27] Y. Wu, S. Kumar, F. Hu, Y. Zhu, and J. D. Matyjas, "Cross-layer forward error correction scheme using raptor and RCPC codes for prioritized video transmission over wireless channels," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 24, no. 6, pp. 1047–1060, Jun. 2014.

[28] J. Xiao, T. Tillo, C. Lin, and Y. Zhao, "Dynamic sub-GOP forward error correction code for real-time video applications," *IEEE Trans. Multimedia*, vol. 14, no. 4, pp. 1298–1308, Aug. 2012.

[29] J. Xiao, T. Tillo, and Y. Zhao, "Real-time video streaming using randomized expanding Reed–Solomon code," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 23, no. 11, pp. 1825–1836, Nov. 2013.

[30] Y. Liu, Z. G. Li, and Y. C. Soh, "A novel rate control scheme for low delay video communication of H.264/AVC standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 1, pp. 68–78, Jan. 2007.

[31] J. Wu, Y. Shang, J. Huang, X. Zhang, B. Cheng, and J. Chen, "Joint source-channel coding and optimization for mobile video streaming in heterogeneous wireless networks," *EURASIP J. Wireless Commun. Netw.*, vol. 2013, p. 283, Dec. 2013.

[32] P. Nunes and F. Pereira, "Joint rate control algorithm for low-delay MPEG-4 object-based video encoding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 19, no. 9, pp. 1274–1288, Sep. 2009.

[33] K. Stuhlmuller, N. Farber, M. Link, and B. Girod, "Analysis of video transmission over lossy channels," *IEEE J. Sel. Areas Commun.*, vol. 18, no. 6, pp. 1012–1032, Jun. 2000.

[34] P. Frossard, "FEC performances in multimedia streaming," *IEEE Commun. Lett.*, vol. 5, no. 3, pp. 122–124, Mar. 2001.

[35] E. N. Gilbert, "Capacity of a burst-noise channel," *Bell Syst. Tech. J.*, vol. 39, no. 5, pp. 1253–1265, 1960.

[36] J. Wu, B. Cheng, C. Yuen, N.-M. Cheung, and J. Chen, "Trading delay for distortion in one-way video communication over the Internet," *IEEE Trans. Circuits Syst. Video Technol.*, to be published.

[37] R. Koetter and A. Vardy, "Algebraic soft-decision decoding of Reed–Solomon codes," *IEEE Trans. Inf. Theory*, vol. 49, no. 11, pp. 2809–2825, Nov. 2003.

[38] J. Nagle, *Congestion Control in IP/TCP Internetworks*, document RFC 896, 1984.

[39] V. Sharma, K. Kar, K. K. Ramakrishnan, and S. Kalyanaraman, "A transport protocol to exploit multipath diversity in wireless networks," *IEEE/ACM Trans. Netw.*, vol. 20, no. 4, pp. 1024–1039, Aug. 2012.

[40] G. Van der Auwera and M. Reisslein, "Implications of smoothing on statistical multiplexing of H.264/AVC and SVC video streams," *IEEE Trans. Broadcast.*, vol. 55, no. 3, pp. 541–558, Sep. 2009.

[41] S. Floyd and K. Fall, "Promoting the use of end-to-end congestion control in the Internet," *IEEE/ACM Trans. Netw.*, vol. 7, no. 4, pp. 458–472, Aug. 1999.

[42] S. Floyd, J. Padhye, and J. Widmer, *TCP Friendly Rate Control (TFRC): Protocol Specification*, document IETF RFC 5438, 2008.

[43] Y. Xu, C. Yu, J. Li, and Y. Liu, "Video telephony for end-consumers: Measurement study of Google+, iChat, and Skype," *IEEE/ACM Trans. Netw.*, vol. 22, no. 3, pp. 826–839, Jun. 2014.

[44] S. Cen, P. C. Cosman, and G. M. Voelker, "End-to-end differentiation of congestion and wireless losses," *IEEE/ACM Trans. Netw.*, vol. 11, no. 5, pp. 703–717, Oct. 2003.

[45] *EXata*. [Online]. Available: http://www.scalable-networks.com/exata, accessed 2015.

[46] *QualNet*. [Online]. Available: http://www.scalable-networks.com/qualnet, accessed 2015.

[47] J. Wu, J. Yang, Y. Shang, B. Cheng, and J. Chen, "SPMLD: Sub-packet based multipath load distribution for real-time multimedia traffic," *J. Commun. Netw.*, vol. 16, no. 5, pp. 548–558, Oct. 2014.

[48] *JM Software*. [Online]. Available: http://iphome.hhi.de/suehring/tml/, accessed 2015.

[49] J. Wu, C. Yuen, B. Cheng, Y. Shang, and J. Chen, "Goodput-aware load distribution for real-time traffic over multipath networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 26, no. 8, pp. 2286–2299, Aug. 2015.

[50] Z. Feng, G. Papageorgiou, S. V. Krishnamurthy, R. Govindan, and T. La Porta, "Trading off distortion for delay for video transmissions in wireless networks," in *Proc. IEEE INFOCOM*, Apr. 2013, pp. 1878–1886.

[51] S. Wang and S. Dey, "Adaptive mobile cloud computing to enable rich mobile multimedia applications," *IEEE Trans. Multimedia*, vol. 15, no. 4, pp. 870–883, Jun. 2013.

[52] S. Wang and S. Dey, "Cloud mobile gaming: Modeling and measuring user experience in mobile wireless networks," *ACM Trans. Mobile Comput. Commun. Rev.*, vol. 16, no. 1, pp. 10–21, Jan. 2012.

[53] Y. Liu, S. Wang, and S. Dey, "Content-aware modeling and enhancing user experience in cloud mobile rendering and streaming," *IEEE J. Emerg. Sel. Topics Circuits Syst.*, vol. 4, no. 1, pp. 43–56, Mar. 2014.

[54] J. Zhang and H. Li, "A novel error concealment algorithm for H.264/AVC," in *MultiMedia Modeling*. Springer, 2015, pp. 466–476.

[55] Z. Wu and J. M. Boyce, "An error concealment scheme for entire frame losses based on H.264/AVC," in *Proc. IEEE ISCAS*, May 2006, pp. 1–5.

[56] S. Belfiore, M. Grangetto, E. Magli, and G. Olmo, "Concealment of whole-frame losses for wireless low bit-rate video based on multiframe optical flow estimation," *IEEE Trans. Multimedia*, vol. 7, no. 2, pp. 316–329, Apr. 2005.

[57] I. Sodagar, "The MPEG-DASH standard for multimedia streaming over the Internet," *IEEE MultiMedia*, vol. 18, no. 4, pp. 62–67, Apr. 2011.

[58] T. Stockhammer, "Dynamic adaptive streaming over HTTP—Standards and design principles," in *Proc. 2nd Annu. ACM Conf. MMSys*, 2011, pp. 133–144.

[59] M. Luby, M. Watson, T. Gasiba, T. Stockhammer, and W. Xu, "Raptor codes for reliable download delivery in wireless broadcast systems," in *Proc. 3rd IEEE CCNC*, Jan. 2006, pp. 192–197.

[60] A. Fiandrotti, V. Bioglio, M. Grangetto, R. Gaeta, and E. Magli, "Band codes for energy-efficient network coding with application to P2P mobile streaming," *IEEE Trans. Multimedia*, vol. 16, no. 2, pp. 521–532, Feb. 2014.

[61] J. Wu, C. Yuen, B. Cheng, M. Wang, and J. Chen, "Streaming high-quality mobile video with multipath TCP in heterogeneous wireless networks," *IEEE Trans. Mobile Comput.*, to be published.

[62] J. Wu, C. Yuen, N.-M. Cheung, and J. Chen, "Delay-constrained high definition video transmission in heterogeneous wireless networks with multi-homed terminals," *IEEE Trans. Mobile Comput.*, vol. 15, no. 3, pp. 641–655, Mar. 2016.

[63] S. Kumar, L. Xu, M. K. Mandal, and S. Panchanathan, "Error resiliency schemes in H.264/AVC standard," *J. Vis. Commun. Image Represent.*, vol. 17, no. 2, pp. 425–450, 2006.

Jiyan Wu (S'12–M'14) received the bachelor's degree from North China Institute of Science and Technology, Hebei, China, in 2008; the master's degree from China University of Mining and Technology, Beijing, China, in 2011; and the Ph.D. degree in computer science and technology from Beijing University of Posts and Telecommunications, Beijing, in 2014, under the supervision of Prof. J. Chen.

He was a Post-Doctoral Research Fellow with Singapore University of Technology and Design, Singapore, from 2014 to 2016, and a Software Developer (C^{++}) with Sinosoft Technologies Co. Ltd., Beijing, from 2010 to 2014. He is currently a Senior Software Engineer with the Department of Research and Development, OmniVision Technologies Singapore Pte. Ltd., Singapore. His programming experience/interests include data management and visualization for large-scale power system analysis. His research interests include video communication, forward error correction, heterogeneous wireless networks, and multipath transmission.

Chau Yuen (SM'12) received the B.Eng. and Ph.D. degrees from Nanyang Technological University, Singapore, in 2000 and 2004, respectively.

He was a Post-Doctoral Fellow with Lucent Technologies Bell Laboratories, Murray Hill, NJ, USA, in 2005. He was with the Institute for Infocomm Research, Singapore, as a Senior Research Engineer from 2006 to 2010. He was a Visiting Assistant Professor with The Hong Kong Polytechnic University, Hong Kong, in 2008. He has been an Assistant Professor with Singapore University of Technology and Design, Singapore, since 2010.

Dr. Yuen received the IEEE Asia-Pacific Outstanding Young Researcher Award in 2012. He serves as an Associate Editor of IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY.

Ngai-Man Cheung (SM'13) received the Ph.D. degree in electrical engineering from University of Southern California, Los Angeles, CA, USA, in 2008.

He was a Post-Doctoral Researcher with the Image, Video and Multimedia Systems Group, Stanford University, Stanford, CA, USA, from 2009 to 2011. He was with the Texas Instruments Research Center Japan; Nokia Research Center, Sunnyvale, CA, USA; IBM T. J. Watson Research Center, Yorktown Heights, NY, USA; HP Laboratories, Tokyo, Japan; The Hong Kong University of Science and Technology, Hong Kong; and Mitsubishi Electric Research Laboratories, Cambridge, MA, USA. He is currently an Assistant Professor with the Singapore University of Technology and Design, Singapore.

Junliang Chen received the B.S. degree in electrical engineering from Shanghai Jiao Tong University, Shanghai, China, in 1955 and the Ph.D. degree in electrical engineering from Moscow Institute of Radio Engineering, Moscow, Russia, in 1961.

He has been with Beijing University of Posts and Telecommunications, Beijing, China, since 1955, where he is the Chairman and a Professor with the Research Institute of Networking and Switching Technology. His research interests include communication networks and next-generation service creation technology.

Prof. Chen was elected as a member of the Chinese Academy of Sciences, Beijing, in 1991, and the Chinese Academy of Engineering, Beijing, in 1994, for his contributions to fault diagnosis in stored program control exchange. He received the first, second, and third prizes of the National Scientific and Technological Progress Award in 1988, 2004, and 1999, respectively.

Chang Wen Chen (F'04) received the B.S. degree from University of Science and Technology of China, Hefei, China, in 1983; the M.S.E.E. degree from University of Southern California, Los Angeles, CA, USA, in 1986; and the Ph.D. degree from University of Illinois at Urbana–Champaign, Champaign, IL, USA, in 1992.

He was with the Faculty of Electrical and Computer Engineering, University of Rochester, Rochester, NY, USA, from 1992 to 1996, and the Faculty of Electrical and Computer Engineering, University of Missouri–Columbia, Columbia, MO, USA, from 1996 to 2003. He was the Allen Henry Endow Chair Professor with the Florida Institute of Technology, Melbourne, FL, USA, from 2003 to 2007. He is currently an Empire Innovation Professor of Computer Science and Engineering with University at Buffalo, The State University of New York, Buffalo, NY, USA.

Dr. Chen is an International Society for Optics and Photonics (SPIE) Fellow. He and his students have received eight best paper awards or best student paper awards over the past two decades. He also received several research and professional achievement awards, including the Sigma Xi Excellence in Graduate Research Mentoring Award in 2003, the Alexander von Humboldt Research Award in 2010, and The State University of New York at Buffalo Exceptional Scholar-Sustained Achievement Award in 2012. He served as the Editor-in-Chief of IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY from 2006 to 2009. He has been the Editor-in-Chief of IEEE TRANSACTIONS ON MULTIMEDIA since 2014. He has been an Editor of several other major IEEE Transactions and IEEE Journals, including PROCEEDINGS OF IEEE, IEEE JOURNAL OF SELECTED AREAS IN COMMUNICATIONS, and IEEE JOURNAL ON EMERGING AND SELECTED TOPICS IN CIRCUITS AND SYSTEMS. He has served as the Conference Chair of several major IEEE, Association for Computing Machinery, and SPIE conferences related to multimedia, video communications, and signal processing.