IEEE Access
Multidisciplinary ⁝ Rapid Review ⁝ Open Access Journal

Received August 1, 2017, accepted August 28, 2017, date of publication September 11, 2017,
date of current version December 22, 2017.

Digital Object Identifier 10.1109/ACCESS.2017.2748138

ROI-Based Video Transmission in Heterogeneous Wireless Networks With Multi-Homed Terminals

ZHEWEI ZHANG[1], TAO JING[1], (Member, IEEE), JINGNING HAN[2], (Member, IEEE), YAOWU XU[2], (Member, IEEE), XUEJING LI[1], AND MEILIN GAO[1]

[1]School of Electronics and Information Engineering, Beijing Jiaotong University, Beijing 100044, China
[2]Google Inc., Mountain View, CA 94043 USA

Corresponding author: Zhewei Zhang (zhwzhang@bjtu.edu.cn)

ABSTRACT We consider the problem of delivering region of interest (ROI)-coded mobile video streams using limited radio resources. Under the conditions of limited bandwidth and time-varying channel status, the goal is to optimize the transmission latency, while ensuring the quality of the ROI parts. Multi-homing support enables the terminals to establish multiple connections for transmission performance improvement. In this paper, we propose a novel framework for ROI-based video transmission in heterogeneous wireless networks with multi-homed terminals. The framework contains the modules of ROI detector and frame splitter, where macroblocks are categorized based on ROI detection and encapsulated into transforming units. It also includes a channel monitor that keeps track of the status of each communication path and sends feedback signals to the streaming controller for packet-scheduling control; a deep learning method is proposed for channel status prediction. To address the delivery problem, we propose a scheduling approach based on the formulated network model and the rate-distortion model. The scheduling method makes a tradeoff between the transmission delay and the distortion. It also guarantees that packets with ROI content are delivered on paths with sufficient bandwidths and low loss rates. Through comparisons with other scheduling methods, we find that the proposed scheme outperforms the other scheduling methods in terms of improving the quality (peak signal-to-noise ratio), balancing the end-to-end delay, and maintaining the playback fluency.

INDEX TERMS Heterogeneous wireless networks, multi-homed communication, region of interest (ROI)-based video coding, deep learning, video transmission.

I. INTRODUCTION

In the recent years, the booming popularity of terminals such as smart phones has enabled mobile users to access their networks and watch videos at any place. The proliferating wireless infrastructure offers a variety of broadband access options, e.g., IEEE 802.11 wireless local area networks (WLANs), IEEE 802.15 wireless personal area networks (WPANs), WiMAX [1], LTE [2], etc. With the rapid breakthrough of these wireless technologies, mobile videos will most probably generate most of the mobile traffic growth by 2021, as predicted by Cisco [3]. The current single wireless network cannot provide satisfactory quality for video streaming services, owing to their small coverage and limited bandwidth. It is reported in [4] that sometimes WLANs fail to sustain the users' mobility, as they lack robustness. Some cellular networks (e.g., UMTS) can provide robust connections; however, they cannot guarantee the quality of service (QoS) because of bandwidth constraints [5]. Although LTE and WiMAX can offer higher data rates and broader coverage, they are not widely deployed yet [6]. Based on the discussions above, there is a tendency for mobile clients to equip themselves with multiple interfaces for accessing different networks simultaneously, which enables them to have the capability of multi-homing access. To deliver high-quality video services, studies on video-streaming transmissions in heterogeneous wireless infrastructures with multi-homed clients have become vital and popular.

Despite some innovations in network infrastructures, which enhance the performance, internet-media streaming applications still suffer from limited bandwidths and packet losses. Although the idea of multipath video streaming has been proposed as a solution to overcome network limitations, it still needs improvements for balancing the load over disjoint paths between the server and the client, and the trade-off between distortion and delay ought to be considered as well. For reducing the transmission bits and saving

2169-3536 © 2017 IEEE. Translations and content mining are permitted for academic research only.
Personal use is also permitted, but republication/redistribution requires IEEE permission.
See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

bandwidth, we propose a region of interest (ROI)-based transmission mode for video streaming, in which the pixels (blocks) in each frame are categorized into ROI and non-ROI groups [7]. Usually, blocks belonging to the non-ROI group are treated as background blocks, and they are assigned a higher quantization parameter (QP) in order to reduce the bit output [8], [9]. The technology of ROI-based video coding can be used for low-bitrate video transmissions, and it guarantees an acceptable video quality under a limited target bitrate [10]. Typical codecs such as HEVC [11] and VP9 [12] induce ROI-mode coding for low-bitrate compression. In addition, some hardware architectures [13]–[15] for foreground detection and motion estimation provide capabilities of real-time processing for ROI-based coding. Therefore, studies on ROI-based video transmission in heterogeneous wireless networks focus on bandwidth saving and quality improvements.

A. RELATED WORK

Some of the related works focused on multi-homed video transmission in heterogeneous wireless networks. Song and Zhuang [16] proposed a framework to analytically evaluate the video streaming performance with flow splitting and multipath transmission, in which a probability generation function (PGF) and z-transform method was applied to derive the packet delay. Wu *et al.* [17] proposed a novel scheduling framework, named ASCOT, which featured frame-level data protection and allocation over multiple paths. They also controlled the frame protection level using forward error correction (FEC) coding to achieve the target quality. Han *et al.* [4] designed an end-to-end virtual path construction system based on Luby transform (LT) code and JM software [18], the goal of which was to provide a high-quality live-video streaming service over heterogeneous wireless networks. The earliest delivery path first (EDPF) algorithm [19] estimates the packets' arrival times based on the bandwidth and propagation delay, to find the earliest path to deliver the packet. The load balancing algorithm (LBA) [20] sorts the streaming packets according to their priority weights, to ensure that the packets with higher priority weights are delivered successfully. This algorithm also takes the correlations among packets into account, and it automatically drops a packet if one of its ancestors were not scheduled. This strategy ensures that the algorithm does not waste network resources. In [21], an analytical framework for optimal rate allocation based on the observed available bit rate (ABR) and round-trip time (RTT) was designed. The authors declared that their allocation policy outperformed those of the heuristic schemes, in terms of the quality. Many multipath streaming methods for wireless networks focused on the trade-off between throughput and delay [22]–[25]. In addition, many papers formulated video streaming as a Markov decision process (MDP), and they introduced reward functions to consider the QoS requirements [26]–[28].

Some of the other related works focused on the research on ROI-based coding, which targeted two major issues: studies of rate control and detection methods. Earlier works [7], [29] presented ROI-based rate-control schemes for the H.263 standard, and the authors developed a block-based ROI segmentation method to implement the region-based codec. In [8], a quality adjustable rate-control method for ROI-based coding was proposed, and the possible visual quality range of ROI was defined according to the range of the ROI QP, which was predicted by the rate-control algorithm. An improved ROI-based rate-control algorithm for H.264/AVC was proposed in [30], which exploited the features of the human visual system (HVS), with video content taken into account. Lee *et al.* [31], [32] studied the texture and non-texture rate models and proposed a novel frame-level rate-control scheme for HEVC. In our previous work [33], we presented a new rate-control scheme for ROI-mode coding based on the discrete cosine transform (DCT) coefficient model and neural networks. As for the ROI detection methods, many papers discussed solutions for the foreground and background separation in video frames, in which low-rank decompositions with principle component analysis (PCA) method [34]–[37], Gaussian mixture model (GMM)-based method [38], [39], or other color/textured model-based methods [40], [41] were included. The development of ROI detection methods provided fast, robust, and real-time benefits for ROI-based coding.

B. OVERVIEW AND CONTRIBUTIONS

In this paper, we address the problem of ROI-based video transmission in heterogeneous wireless networks with limited bandwidths and tight delay constraints, by designing a novel transmission framework. When a frame is available from the source, its blocks are categorized into ROI and non-ROI groups by the ROI detector. A frame-split strategy is adopted to split the entire frame into slices. Then, the slices are encapsulated into the maximum transmission unit (MTU). The FEC module is used in the framework to mitigate the channel losses at the expense of bringing the redundant bits. In low-delay video coding, it is not recommended to use bidirectional prediction, so as to keep the casual encoding order, according to [26]. Hence, the video bitstream consists of only I and P frames in each group of pictures (GoP). We formulate a rate-distortion (RD) model for ROI-based video coding, which approximately estimates the generated bits in a frame. Moreover, a model for heterogeneous wireless networks, which considers the end-to-end delay, channel loss rate, and channel status is established. Meanwhile, a deep learning approach for forecasting the channel status according to its previous values is proposed. Then, the channel loss rate for each path can be computed based on the channel status. To deliver the stream packets with low end-to-end delay and distortion, we propose a balanced delay/distortion scheduling approach for ROI-based coding. It assures that the MTUs containing ROI content are delivered over paths with good channel conditions and sufficient bandwidths. Each communication path is assigned a performance score, which takes the bandwidth, channel loss rate, and fixed delay into account. From the experimental results, we can conclude that

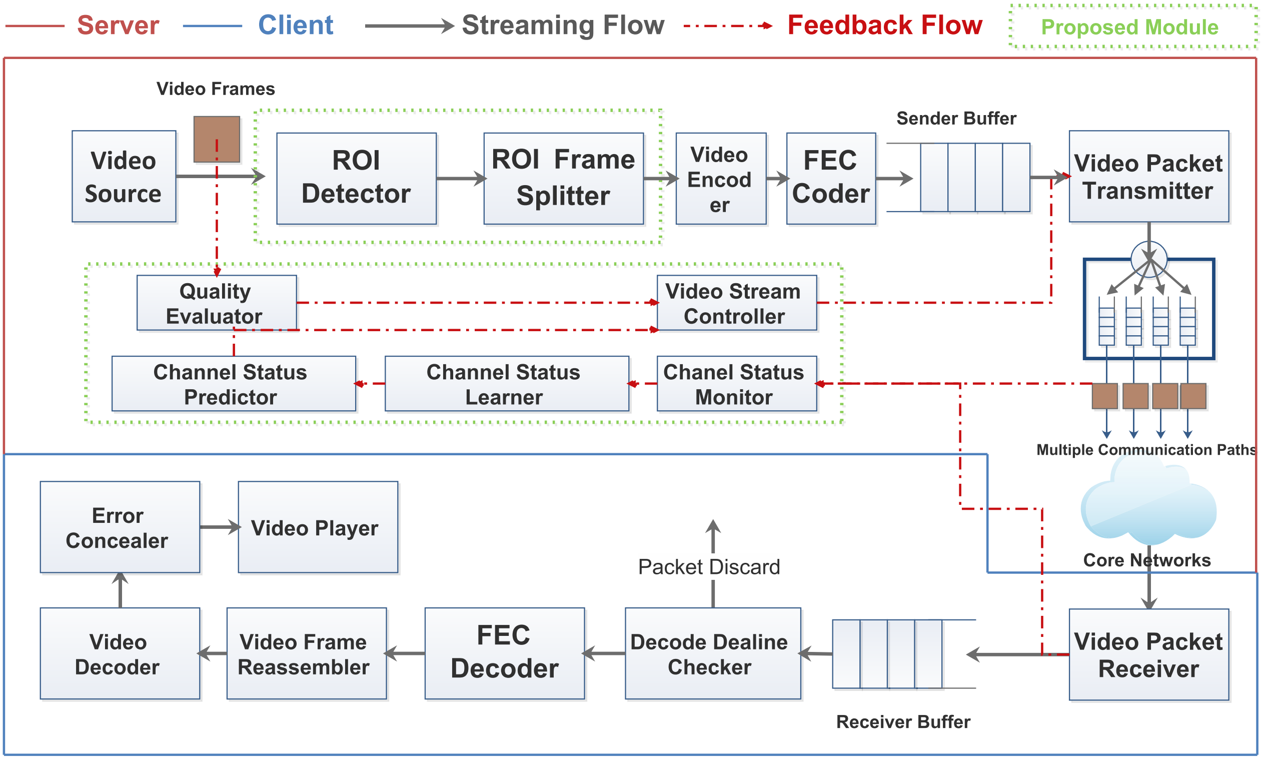


FIGURE 1. Overview of the proposed ROI-based video transmission network.

the proposed scheduling method can to keep the delay and distortion within acceptable bounds, and that the ROI part of each frame can be displayed clearly even if the network environment is poor.

The remainder of this paper is organized as follows. Section. II depicts the proposed ROI-based framework for video transmission and the scheduling approach. We also discuss the formulated RD model and network model in this section. In Section. III, we provide the performance evaluation and the experimental results. The concluding remarks and plans for future work are given in Section. IV.

II. SYSTEM MODEL

A. SYSTEM OVERVIEW

Fig. 1 displays the proposed system framework for ROI-based video transmission, which includes the system components at both the server and the receiver. The server side is in charge of the video streaming allocation from the encoder's output, based on ROI frame splitting. The split packets are sent to the FEC encoder for forward error correction coding (i.e., the systematic Reed–Solomon code). Next, the packet transmitter delivers the packets through multiple communication paths, based on the scheduling strategy. The client side is responsible for the video stream decoding and reassembling. When packets arrive at the receiver buffer, the decoder deadline checker checks whether the packet is past the decoding deadline, and it will discard the packet if it is overdue. Then, the information bytes of the packets are error corrected by the FEC decoder. In order to restore the original video stream, the video frame reassembler readjusts the split frame data into the original frame based on each packet header's absolute offsets. These frames are re-sequenced in the correct order for the decoding process, and the frames discarded by the deadline checker are concealed by copying the last received ones, before decoding. Finally, the decoded frames are displayed in the video player.

The two proposed modules are marked with dashed green rectangles in Fig. 1. The ROI module consists of the detector and the frame splitter, which are placed before the FEC coder. ROI blocks are detected using some detection methods by the detector, and the frame splitter cuts each frame into slices. Next, the slices are classified into two groups: ROI and non-ROI slices. Following [6], each slice is assigned an extra header that includes its size, the origin frame's size, and its absolute offsets. Then, the FEC-coded slices are delivered through different paths by the transmitter. The feedback module is responsible for channel-status monitoring and video-stream controlling. The channel-status monitor acquires the path status information and sends it to the channel-status learner, in which the properties of each path can be learned using deep-learning approaches. Then, the predictor forecasts the future channel-status information and hands it to the video-stream controller together with the quality information from the quality evaluator. The video-stream controller figures out control strategies for ROI frame splitting and packet transmission.

B. VIDEO RATE/DISTORTION MODEL

We consider a single HD video stream encoded using the H.264/AVC standard for end-to-end transmission. According to our previous work [33], we assume that the DCT coefficients are approximately uncorrelated and that they are Laplacian distributed with variance σ^2. Note that σ^2 is also the variance of the difference frame pixels, since DCT is an orthogonal transformation. Let us assume that the DCT coefficients are quantized with a uniform scalar step size q_s, and that the frame rate (in bits/pixel) $R(q_s) \approx H(q_s)$, where $H(q_s)$ is the empirical entropy function of the q_s-quantized coefficients. In [42], $H(q_s)$ has the following expression:

$$H(q_s) = \begin{cases} \frac{1}{2}\log_2(2e^2\frac{\sigma^2}{q_s^2}), & \frac{\sigma^2}{q_s^2} > \frac{1}{2e} \\ \frac{e}{\ln 2}\frac{\sigma^2}{q_s^2}, & \frac{\sigma^2}{q_s^2} \leq \frac{1}{2e}. \end{cases} \quad (1)$$

Observe that $\frac{\sigma^2}{q_s^2}$ is larger than $1/(2e)$ for small values of q_s (high-rate case) and smaller for large values of q_s (low-rate case). For video transmission in heterogeneous wireless networks, we focus on the more interesting low-rate case for low-delay transmission, which indicates that the peak signal-to-noise ratio (PSNR) is below 40 dB. Specifically, let S_V be the set of macroblocks whose standard deviation σ is in the interval $(V - \varsigma, V + \varsigma]$, where ς is a small value that can be set as 0.5 and V is an integer. For ROI-based video encoding, $S_V = S_V^R \bigcup S_V^{NR}$, where S_V^R is the set of ROI macroblocks whose $\sigma \in (V - \varsigma, V + \varsigma]$ and S_V^{NR} is the set of non-ROI macroblocks. Using (1), the average encoding rate R_V for DCT coefficients in S_V is given as:

$$R_V = \frac{1}{AN_V}(\sum_{n=1}^{N_V^R} B_{V,n}^R + \sum_{n=1}^{N_V^{NR}} B_{V,n}^{NR}), \quad (2)$$

where N_V is the number of macroblocks in S_V, and N_V^R and N_V^{NR} denote the number of ROI and non-ROI macroblocks, respectively, in S_V. Obviously, $N_V = N_V^R + N_V^{NR}$. $B_{V,n}$ is the number of DCT bits produced by the nth macroblock (both ROI and non-ROI), and A is the number of pixels in a macroblock (i.e., $A = 16^2$). Next, the expected number of bits produced by the ith macroblock is given by:

$$B_i = A(\frac{e\sigma_i^2}{\ln 2 \cdot q_{s_{V_i}}^2} + C) \approx A(\frac{eV_i^2}{\ln 2 \cdot q_{s_{V_i}}^2} + C), \quad (3)$$

where σ_i is the empirical standard deviation whose value belongs to $(V_i - \varsigma, V_i + \varsigma]$, and $q_{s_{V_i}}$ is the quantization step size used for that macroblock. C is a constant defined as the *overhead* rate, which models the average rate to encode the motion vectors and the coder's syntax and header. Therefore, the total estimated bits produced by a frame can be written as:

$$B = \sum_V N_V R_V = \frac{1}{A}\sum_V(\sum_{n=1}^{N_V^R} \frac{AeV^2}{\ln 2 q_{s_{V_n}}^2} + \sum_{n=1}^{N_V^{NR}} \frac{AeV^2}{\ln 2(q_{s_{V_n}} + \Delta_q)^2}) + C\sum_V N_V, \quad (4)$$

where Δ_q is the quantization step offset between ROI and non-ROI blocks.

Now, we consider the distortion model for ROI-based video transmission. Assume each GoP consists of G frames and that each of them is identified by a frame number g $(1 \leq g \leq G)$. According to [21], the total distortion of frame g can be expressed as:

$$d_g = e_g + y_g, \quad (5)$$

in which e_g denotes the truncation distortion and y_g is the drifting distortion. Note that $e_g = D_g + \Pi_g \cdot \delta_g$, in which D_g denotes the full-quality distortion of frame g, Π_g is the effective loss rate, and δ_g represents the extra distortion introduced by dropping packets. D_g is considered as the case where all the packets belonging to this frame are received, and its formula is given by:

$$D_g = \frac{1}{\nu}\sum_V(\sum_{n=1}^{N_V^R} q_{s_{V_n}}^2 + \sum_{n=1}^{N_V^{NR}}(q_{s_{V_n}} + \Delta_q)^2), \quad (6)$$

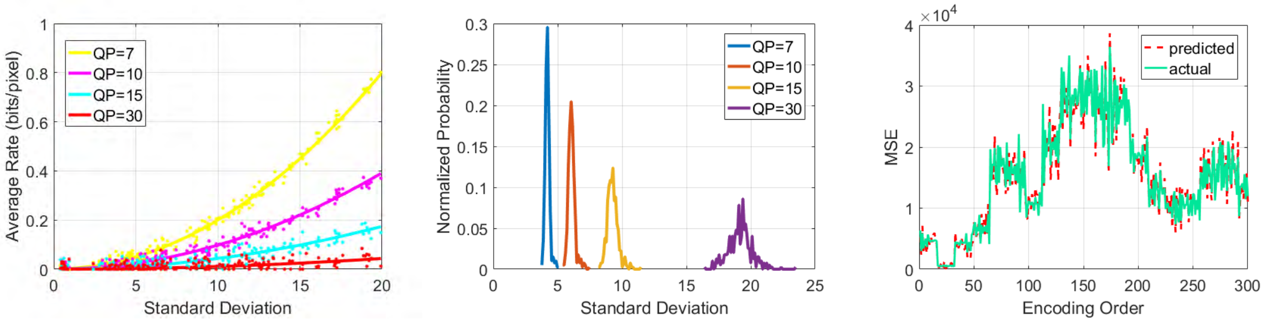
where ν is a factor set as 12, from [43]. Note that q_s^2/ν gives an approximate the mean square error (MSE) estimation for each macroblock. The drifting distortion is caused by the imperfect reconstruction of previous frames used for inter prediction. We use the *IPPP* $\cdots$ GoP structure in this paper; thus, only the previous frames within a GoP will cause drifting distortion for the latter frames. Based on the models in [42], y_g can be written as:

$$y_g = \begin{cases} \alpha_g, & \text{if } g = 1 \\ \alpha_g + \sum_{1 \leq n \leq g} \beta_{g,k} \cdot e_n, & \text{if } 1 < g \leq G, \end{cases} \quad (7)$$

where α_g and β_g are nonnegative estimated parameters. In [44], the authors used a multinomial regression method to estimate them with polynomials. To verify the proposed ROI-based RD model, we provide the results in Fig. 2. These results are obtained by encoding 6 s of the *Flower* (CIF) sequence at 15 frames/s using the H.264 encoder with a fixed QP. Observe that the proposed RD model effectively formulates the relationship between the RD and the q_s, σ. Besides, for higher QP, the increased quantization noise makes the histogram peaks move to the right, as shown in Fig. 2(b).

C. WIRELESS ACCESS NETWORK MODEL

Consider a heterogeneous wireless network with P communication paths between two transmission ends. We model the burst losses on each path using the Gilbert model [45].



(a)

(b)

(c)

FIGURE 2. (a) Verification of the proposed ROI-based rate model for the encoding rate as a function of standard deviation σ. (b) Normalized histograms of σ for different QP (QP = $1/2q_S$). (c) Actual distortion of each macroblock versus the predicted values of D_g based on the distortion model. Sequences: *Flower* ($g = 10$).

This model assumes the path state $\chi_p(t)$ at time t to be one of two values: $\mathcal{G}$ (Good) or $\mathcal{B}$ (Bad). The packet can be successfully delivered if $\chi_p(t) = \mathcal{G}$ and dropped if $\chi_p(t) = \mathcal{B}$. Let us assume that B_g are the output bits for the frame g and MTU is the maximum transmission unit. The number of packets belonging to frame g is $\lceil B_g/MTU \rceil$. We define c_p as the status tuple for path p with the size $\lceil B_{g,p}/MTU \rceil$ (i.e., $c_p = (\mathcal{G}, \mathcal{B}, \cdots, \mathcal{G})$), in which $B_{g,p}$ denotes the allocated bits on path p. Then, the transmission loss rate on path p for frame g can be written as:

$$\pi_{g,p} = \frac{1}{\lceil B_{g,p}/MTU \rceil} \sum_{i=1}^{\lceil \frac{B_{g,p}}{MTU} \rceil} \delta(c_p^i = \mathcal{B}), \tag{8}$$

where $\delta(\bullet)$ is an indicator function whose value is 1 if $\bullet$ is true, and 0 otherwise. In capacity-limited networks, the loss probability of packets over path p out of the deadline T can be obtained from the exponential distribution [42]:

$$\begin{aligned} \pi^{\S}_{g,p} &= P(\mathfrak{D}_{g,p} > T) \\ &= \frac{B_{g,p}}{\sum_{g=1}^{G} B_{g,p}} \exp(-\frac{\mu_p \cdot T}{\sum_{g=1}^{G} B_{g,p} + \mu_p \cdot f_p}), \end{aligned} \tag{9}$$

in which $\mathfrak{D}_{g,p}$ is the end-to-end delay for frame g, μ_p is the available bandwidth, and f_p is the fixed delay over path p. Note that the fixed delay includes the latency of delivery, processing, and propagation. Therefore, Π_g can be expressed as follows:

$$\Pi_g = \sum_p \Pi_{g,p} = \sum_p [\pi_{g,p} + (1 - \pi_{g,p})\pi^{\S}_{g,p}]. \tag{10}$$

Next, we consider the delay for each path. According to [6], the video encoding data is generated in bursts, and each path carries a substream of the video streaming flow. We employ the envelop process for each path as follows:

$$\hat{A}_p(t) = \lambda_p \cdot t + B_{g,p}, \tag{11}$$

where λ_p denotes the long-term average video streaming rate and $\hat{A}_p(t)$ is the size of the cumulative substreaming flow in $[0, t)$ over path p. Obviously, $\sum_{p=1}^{P} \lambda_p = \lambda$ and $\sum_{p=1}^{P} B_{g,p} = B$. Similar to [20], we model each path as a work-conserving queueing system. Assuming that $\lambda > \mu_p$, the ROI frame splitter separates the streaming flow into subflows that satisfy $\lambda_p < \mu_p$. Then, the end-to-end delay $\mathfrak{D}_{g,p}$ for frame g over path p is the sum of the queueing delay and the fixed delay, as follows:

$$\mathfrak{D}_{g,p} = \frac{B_{g,p}}{\mu_p - \lambda_p} + f_p, \tag{12}$$

To estimate the path status of each transmission interval, we propose a recurrent neural network (RNN) model. Previous works introduced continuous-time Markov chains [17], [45] to model $\chi_p(t)$ in wireless networks. However, such models worked only in approximate manners, since both the stationary probabilities of paths and the state transition probabilities vary with time. In addition, the occurrence of consecutive burst losses decreases the evaluation accuracy of the Markov model. The RNN model, on the other hand, can overcome these shortcomings. As shown in Fig. 3, the output of the network is the current channel status $\chi_p(t)$ and the input is the historical channel status vector $\boldsymbol{\chi_p} = [\chi_p(t-K\tau), \chi_p(t-(K-1)\tau), \cdots, \chi_p(t-\tau)]$, in which K is the window size and τ is the unit transmission slot. Further, we denote $\chi_p(t)$ as a binary value where $\mathcal{G} = 1$ and $\mathcal{B} = 0$. To save the computing costs for forward and backward propagation, only one hidden layer is used in our work. Forward propagation begins with a specification of the initial hidden state $\boldsymbol{h}^{(0)}$. Then, for each time step, the RNN is updated by the following rules:

$$\begin{cases} \boldsymbol{h}^{(t)} = \tanh(\boldsymbol{b} + \boldsymbol{W}\boldsymbol{h}^{(t-\tau)} + \boldsymbol{U}\boldsymbol{x}^{(t)}) \\ \boldsymbol{o}^{(t)} = \boldsymbol{c} + \boldsymbol{V}\boldsymbol{h}^{(t)} \\ \hat{\boldsymbol{y}}^{(t)} = \mathrm{softmax}(\boldsymbol{o}^{(t)}) \end{cases} \tag{13}$$

where the parameters are the bias vectors $\boldsymbol{b}$ and $\boldsymbol{c}$ along with the weight matrices $\boldsymbol{U}$, $\boldsymbol{V}$, and $\boldsymbol{W}$, respectively, for input–hidden, hidden–output, and hidden–hidden connections. Note that $\boldsymbol{x}^{(t)}$ is updated by shifting the sliding window;for example, $\boldsymbol{x}^{(t+\tau)} = [\chi_p(t-(K-1)\tau), \cdots, \chi_p(t)]$, in which the oldest value is removed. Given a fixed time step T_s, the loss

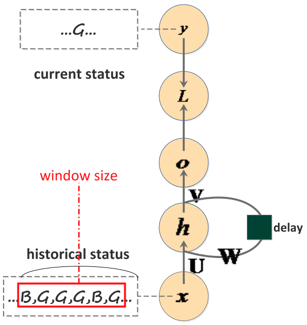


FIGURE 3. Proposed recurrent neural network (RNN) framework for path status prediction.

function L can be defined as:

$$L=\sum_{k=1}^{T_s} L^{(k)} = -\sum_{k=1}^{T_s}\log p_{model}(\boldsymbol{y}^{(t+k\tau)}|\{\boldsymbol{x}^{(t)},\cdots,\boldsymbol{x}^{(t+k\tau)}\}), \tag{14}$$

in which p_{model} is given by a softmax function as:

$$\begin{aligned} &p_{model}(\boldsymbol{y}^{(t+k\tau)}=i|\{\boldsymbol{x}^{(t)},\cdots,\boldsymbol{x}^{(t+k\tau)}\}) \\ &= \frac{e^{\boldsymbol{V}_i\boldsymbol{h}^{(t+k\tau)}+\boldsymbol{c}_i}}{\sum_j e^{\boldsymbol{V}_j\boldsymbol{h}^{(t+k\tau)}+\boldsymbol{c}_j}} = \frac{e^{\boldsymbol{V}_i \tanh(\boldsymbol{b}+\boldsymbol{W}\boldsymbol{h}^{(t+(k-1)\tau)}+\boldsymbol{U}\boldsymbol{x}^{(t+k\tau)})+\boldsymbol{c}_i}}{\sum_j e^{\boldsymbol{V}_j \tanh(\boldsymbol{b}+\boldsymbol{W}\boldsymbol{h}^{(t+(k-1)\tau)}+\boldsymbol{U}\boldsymbol{x}^{(t+k\tau)})+\boldsymbol{c}_j}}, \end{aligned} \tag{15}$$

and $\boldsymbol{V}_i$ denotes the ith row of $\boldsymbol{V}$. For backward propagation, we start the recursion from $t+T_s\tau$ down to t. The gradient $\nabla_{\boldsymbol{o}^{(t+k\tau)}}L$ on the outputs at time step $t+k\tau$ $(1\leq k\leq T_s)$, for all components i, is as follows:

$$(\nabla_{\boldsymbol{o}^{(t+k\tau)}}L)_i = \frac{\partial L}{\partial L^{(t+k\tau)}}\frac{\partial L^{(t+k\tau)}}{\partial o_i^{(t+k\tau)}} = \hat{\boldsymbol{y}}_i^{(t+k\tau)} - \boldsymbol{y_i}^{(t+k\tau)}. \tag{16}$$

For $\nabla_{\boldsymbol{h}^{(t+k\tau)}}L$, when $k=T_s$, $\boldsymbol{h}^{(t+T_s\tau)}$ has only $\boldsymbol{o}^{(t+T_s\tau)}$ as a descendant. Hence, its gradient is given by:

$$\nabla_{\boldsymbol{h}^{(t+T_s\tau)}}L = (\nabla_{\boldsymbol{o}^{(t+T_s\tau)}}L)\frac{\partial \boldsymbol{o}^{(t+T_s\tau)}}{\partial \boldsymbol{h}^{(t+T_s\tau)}} = (\nabla_{\boldsymbol{o}^{(t+T_s\tau)}}L)\boldsymbol{V}. \tag{17}$$

Then, from $t+(T_s-1)\tau$ to t, $\boldsymbol{h}^{(t+k\tau)}$ has descendants depending on $\boldsymbol{o}^{(t+k\tau)}$ and $\boldsymbol{h}^{(t+(k+1)\tau)}$. Thus, its gradient is given by:

$$\nabla_{\boldsymbol{h}^{(t+k\tau)}}L = \nabla_{\boldsymbol{h}^{(t+(k+1)\tau)}}L\frac{\partial \boldsymbol{h}^{(t+(k+1)\tau)}}{\partial \boldsymbol{h}^{(t+k\tau)}} + \nabla_{\boldsymbol{o}^{(t+k\tau)}}L\frac{\partial \boldsymbol{o}^{(t+k\tau)}}{\partial \boldsymbol{h}^{(t+k\tau)}}. \tag{18}$$

Once $\nabla_{\boldsymbol{h}}L$ and $\nabla_{\boldsymbol{o}}L$ are computed, we can obtain the gradients for each parameter, such as $\nabla_{\boldsymbol{c}}L$, $\nabla_{\boldsymbol{b}}L$, $\nabla_{\boldsymbol{V}}L$, $\nabla_{\boldsymbol{W}}L$, and $\nabla_{\boldsymbol{U}}L$. Details are given in [46]. The RNN model is embedded into the channel-status learner, and the probe packets are pre-sent through each path to learn the available bandwidth μ_p and the channel status $\chi_p(t)$. Then, the previous statuses of each path are sent to the RNN as the training samples. During the packet transmission process, the channel monitor evaluates the prediction error of the channel status and retrains the RNN using the historical statuses, if necessary. Meanwhile, the channel-status predictor is able to compute $\pi_{g,p}$ based on the predicted data. To obtain a brief overview of the learning performance, we compare the learning efficiencies of the RNN and the common artificial neural network (ANN). As shown in Fig. 4, the proposed RNN model can learn the samples of channel status effectively as the training iteration grows, while ANN fails to reduce the error. It means that the correlation information between the previous and the current statuses is captured by the RNN. Furthermore, Fig. 4(b) displays the prediction results of different models during a period. One can see that the RNN model achieves a higher forecast precision than other models. More details are given in Sec. III-D

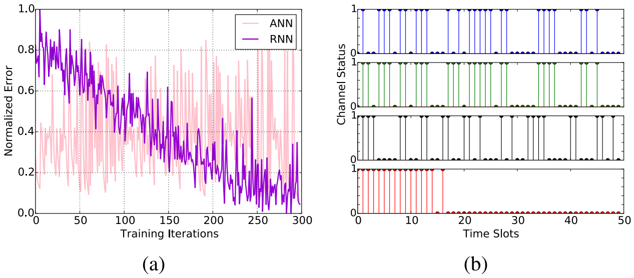


FIGURE 4. (a) Learning efficiency performance for RNN and artificial neural network (ANN). (b) Channel status prediction performance for different models. (From top to bottom: The ground truth statuses, RNN's prediction, Markov model's prediction, ANN's prediction)

D. PROBLEM STATEMENT AND SOLUTION

The goal of the video stream controller is to achieve a trade-off between delay and distortion. Given the weight factor γ and the deadline delay constraint T, we state the optimization problem as follows:

$$\begin{aligned} &\text{Minimize}: \gamma d_g + (1-\gamma)\max_{p\in P}\{\mathfrak{D}_{g,p}\} \\ &s.t.: \begin{cases} d_g = \text{Equation (5)}, \\ \mathfrak{D}_{g,p} = \text{Equation (12)}, \\ \dfrac{\sum_{g=1}^{G} B_{g,p}}{\mu_p\cdot G} + f_p \leq T, \\ \sum_{p=1}^{P}\lambda_p = \lambda, \quad \sum_{p=1}^{P} B_{g,p} = B, \\ \lambda > \mu_p, \quad \lambda_p < \mu_p, \ p\in P. \end{cases} \end{aligned} \tag{19}$$

For each frame in a GoP, the above optimization problem minimizes the combination function of the delay and distortion. Note that γ is set a constraint positive value close to 0, to keep d_g and $\max_{p\in P}\{\mathfrak{D}_{g,p}\}$ in the same order of magnitude. Previous solutions to this problem consisted of the greedy algorithm and the water-filling method [6], and it was stated that the problem had no optimal solution with polynomial time-complexity (NP-Hard). The water-filling method was able to dynamically allocate the "good quality data over networks to reduce the distortion, with $o(N)$ time-complexity; however, it only minimized the end-to-end delay, without balancing the transmission distortion. Moreover, the ROI-based encoding/transmission mode was not taken into consideration. We propose an improved water-filling algorithm based on the ROI mode, which compromises between delay and distortion, namely, the D&D water-filling algorithm. First, we assign each path with a delay–distortion (D&D) weight ω_p:

$$\omega_p = \frac{1-\frac{1}{2}[\frac{(\mu_p-\lambda_p)f_p}{\sum_p(\mu_p-\lambda_p)f_p}+\frac{\pi_{g,p}}{\pi_g}]}{\sum_p\{1-\frac{1}{2}[\frac{(\mu_p-\lambda_p)f_p}{\sum_p(\mu_p-\lambda_p)f_p}+\frac{\pi_{g,p}}{\pi_g}]\}} = \frac{1-\frac{1}{2}[\frac{(\mu_p-\lambda_p)f_p}{\sum_p(\mu_p-\lambda_p)f_p}+\frac{\pi_{g,p}}{\pi_g}]}{P+\frac{1}{P}-\frac{\sum_p(\mu_p-\lambda_p)f_p}{2P\sum_p(\mu_p-\lambda_p)f_p}}. \quad (20)$$

Note that ω_p is computed based on the mean of the normalized preloaded water $(\mu_p-\lambda_p)f_p$ and the path loss probability $\Pi_{g,p}$. The path with lower preloaded water size and loss probability is given a higher weight, and the task of the optimization problem is to fill B_g units of the frame g into P paths (buckets) while keeping the highest level (delay) as low as possible, together with packet loss optimization. Let us assume that the optimal water level after the paths being filled is $\mathfrak{D}_l^*$. When $\mathfrak{D}_l^* > \max_{p\in P}\{f_p\}$, B_g can be expressed as follows:

$$B_g = \sum_{p\in P}(\mu_p-\lambda_p)(\mathfrak{D}_l^* - f_p). \quad (21)$$

By introducing the weight factor ω_p, the balanced optimal water level is given as:

$$\mathfrak{D}_l^* = \frac{B_g + P\sum_p \omega_p(\mu_p-\lambda_p)f_p}{P\sum_p \omega_p(\mu_p-\lambda_p)}. \quad (22)$$

Then, the optimal value for $B_{g,p}$ is given by:

$$B_{g,p} = (\mu_p-\lambda_p)[\frac{B_g + P\sum_p \omega_p(\mu_p-\lambda_p)f_p}{P\sum_p \omega_p(\mu_p-\lambda_p)} - f_p]. \quad (23)$$

On the other hand, when $\mathfrak{D}_l^* < \max_{p\in P}\{f_p\}$, it means that $B_g < \sum_p(\mu_p-\lambda_p)(\max_{n\in P}(f_n)-f_p)$. In this case, $\max_{p\in P}\{\mathfrak{D}_{g,p}\} = \max_{p\in P}(f_p)$. Then, we only consider the distortion optimization in which the ROI slices of packets are allocated to the paths with lower loss probabilities. Therefore, we have the proposed D&D water-filling algorithm for ROI slice packet assignment and transmission over P paths, as depicted in Alg. 1. As for the MTU's encapsulation, each MTU contains the transformed DCT coefficients of the macroblocks together with the additional header syntax, and several MTUs form the slices. As stated earlier, the slices are assigned with additional headers that contain their absolute offsets. The MTUs have similar sizes; however, they are allocated to the paths under a priority strategy: MTUs that contain ROI macroblocks with lower standard deviations for DCT coefficients are allocated to the paths that have higher ω_p.

Algorithm 1 D&D Water-Filling Algorithm

```
Given the long-term streaming rate $\lambda$, initiate the sub-streaming rate for each path $\lambda_p = \frac{\mu_p\cdot\lambda}{\sum_p \mu_p}$.
Compute $\omega_p$ by (20) and sort the paths based on $\omega_p$ in descending order.
if $\mathfrak{D}_l^* < T$ then
    if $\mathfrak{D}_l^* > \max_{p\in P}\{f_p\}$ Compute $\mathfrak{D}_l^*$ using (22). else, $\mathfrak{D}_l^* = \max_{p\in P}\{f_p\}$.
else
    $\mathfrak{D}_l^* = T$.
end if
Compute $B_g$ using (4), and assign each path with $B_{g,p}$ size of data by (23).
for each path sorted by $\omega_p$ in descending order do
    if $\lceil\frac{B_{g,p}}{MTU}\rceil$ ROI $MTU$s are available, fill path $p$ with $\lceil\frac{B_{g,p}}{MTU}\rceil$ ROI $MTU$s for transmission.
    else if $\lceil\frac{B_{g,p}}{MTU}\rceil$ non-ROI $MTU$s are available, fill path $p$ with $\lceil\frac{B_{g,p}}{MTU}\rceil$ non-ROI $MTU$s for transmission.
    else fill path $p$ with the remaining ROI or non-ROI $MTU$s for transmission.
end for
```

For example, consider the path with the highest ω_p; when $B_{g,p}$ is acquired, we have:

$$\frac{1}{A}\sum_{V=0}^{V_1}\sum_{n=1}^{N_V^R}\frac{AeV^2}{\ln 2q_{s_{V_n}}^2} + C\sum_{V=0}^{V_1} N_V^R \simeq B_{g,p}. \quad (24)$$

Then, the ROI slice for path p is composed of $\sum_{V=0}^{V_1} N_V^R$ ROI macroblocks, and these macroblocks are encapsulated into $\lceil\frac{B_{g,p}}{MTU}\rceil$ MTUs. For the next path, we change $V = V_1$ and $V_1 = V_2$, and form the next ROI slice with $\sum_{V=V_1}^{V_2} N_V^R$ ROI macroblocks, and so on. Note that the enclosure of non-ROI slices follows the same rules as that of ROI slices. A brief illustration of the proposed algorithm is displayed in Fig. 5.

III. EXPERIMENTS

A. EXPERIMENT SETUP

We use the OMNeT++5.1 (INET) [47] and the JM 18.6 [18] as the network emulator and video codec, respectively. The INET framework can be considered the standard protocol model library of OMNeT++. INET contains models for the Internet stack (TCP, UDP, IPv4, IPv6, OSPF, BGP, etc.) and wired and wireless link layer protocols (Ethernet, PPP, IEEE 802.11, etc.), which are useful for emulating heterogeneous wireless networks. JM 18.6 is the reference software for H.264/AVC, in which we choose JM owing to the source code integration, as both OMNeT++ and JM are developed using C++. Fig. 6 shows the designed network architecture used in our experiment. In this network topology, the server has one wired network interface and the client has WLAN and cellular interfaces. An end-to-end connection path is established by binding a pair of IP addresses from the server to the client. For example, p_1 has the ports {PS, PD1}, and four paths

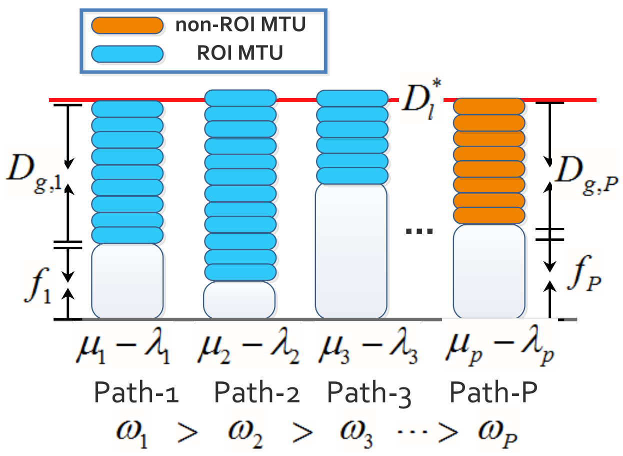


FIGURE 5. Proposed solution structure for the D&D water-filling algorithm.

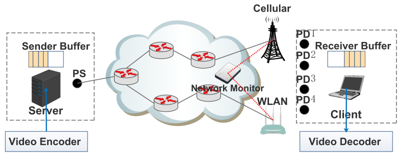


FIGURE 6. The designed network architecture for experiments.

are included in our experiment. Referring to the previous works [6], [17], we use the parameter configurations of wireless networks shown in Table. 1. For the wired networks connected to the server, we implement the static routing strategy for each pair of interfaces. In addition, each link between two routers is set with a fixed delay of 5 ms and a uniformly distributed channel loss rate of 2% − 5%. In order to distinguish the path bandwidth, we set limited bandwidths for the paths $p_1 \cdots p_4$, as {600, 1000, 800, 200} kbps, respectively, as in [20]. When the transient state begins, the network monitor sends some probe packets with exponentially distributed intervals to the server and the client. Based on the probe ack packets, the monitor evaluates the actual packets loss rate for each path using the proposed RNN model. It also checks the video packets arrival intervals for each path and computes the actual bandwidth. For the configuration of video sources, each stream is encoded at 25 fps, and a GoP consists of 6 frames. The key frame interval is set to 5 and the delay deadline for each frame (T) is 200 ms [19], [20]. In addition, an error concealment strategy is considered in our work. If a frame cannot be decoded owing to transmission or overdue losses, it will be concealed with the frame-coping strategy.

For comparison analysis, we compare the proposed method with three typical packet-scheduling approaches over heterogeneous wireless networks. They are described as follows:

- Earliest delivery path first (EDPF) [19]. This method ensures that when a new video packet arrives, it is scheduled in the path with the shortest transmission delay.

TABLE 1. Parameter configurations of wireless network.

Cellular	value	WLAN	value
Average SNR	15 dB	Average SNR	15 dB
Total cell bandwidth	3.84 Mc/s	Average channel bit rate	2 Mbps
Average loss rate	3%	Average loss rate	6%
Common channel power	43 dB	Max connection window	32
Average burst length	10 ms	Average burst length	20 ms
Available capacity	350 Kbps	Available capacity	500 Kbps

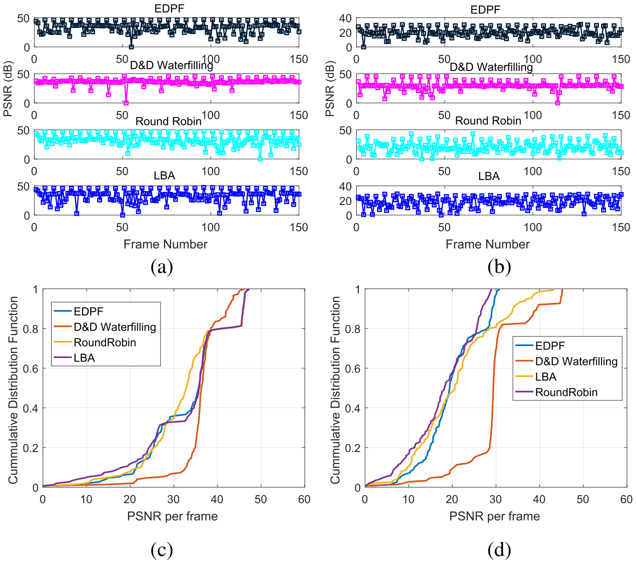
- Round robin [48]. This method randomly schedules the newly arriving packets among the paths based on a probability. We set the probability of each path in direct proportion to its bandwidth.
- Load balancing algorithm (LBA) [20]. This method focuses on the interdependencies between packets. It also considers the network resources, based on a policy that, if one of the ancestor packets could not be scheduled, the algorithm automatically drops the current packet. No enhancement layer packets are used in our experiment; the packets weight ω_i in [20] is only set for the base layer.

B. QUALITY EVALUATIONS

We adopt the PSNR [49] as the standard metric for the received video quality. Since the video stream is generated by the ROI-based codec, the weighted mean measurement for PSNR is used in our experiment:

$$\mathrm{PSNR} = \kappa 10 \lg \frac{255^2}{\mathrm{MSE}_R} + (1-\kappa) 10 \lg \frac{255^2}{\mathrm{MSE}_{NR}}, \qquad (25)$$

where MSE_R and MSE_{NR} denote the mean squared errors of the ROI and non-ROI pixel sets, respectively. κ is the weight factor, which is set as 0.6 for emphasizing the ROI group. Fig. 7 shows the instantaneous PSNR values and the cumulative distribution functions (CDFs) of *Habour* and *City*. Their target bitrates are set to 400 kbps and 600 kbps respectively, and the quantization step Δ_q between the ROI and non-ROI blocks is set to 10. From Fig. 7(a) and 7(b), we see that the proposed scheduling method achieves a better average PSNR performance than other methods. LBA shows higher PSNR values on I frames, but it reduces the PSNR values on P frames, for optimizing the network resources by considering the packet weights and their interdependencies. According to Fig. 7(c) and 7(d), we see that the PSNR values of our method are mainly distributed from 30 to 40, which indicates that our method achieves an acceptable received video quality. In order to verify each method's performance under limited bandwidths with poor channel statuses, we revise the bandwidth for each path as {300, 500, 400, 100} kbps, and increase the loss rates of the cellular and WLAN networks to 8%. The received frames, based on different methods, are subjectively displayed in Fig. 7. Compared to the raw frames, the received frames of our method have good visual qualities with less distortion. Even though some block areas belonging to the non-ROI group may show some distortion, the block areas in the ROI group are displayed clearly. Without loss of



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

FIGURE 7. **(a), (b): Comparisons of instantaneous PSNR values of video frames indexed from 1 to 150. (c), (d): CDFs of PSNR per frame comparisons for different methods. (a) *Habour* (1-150). (b) *City* (1-150). (c) *Habour* (1-150). (d) *City* (1-150).**

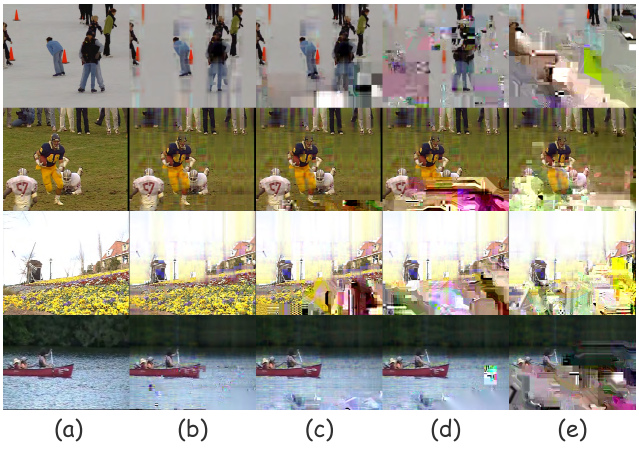
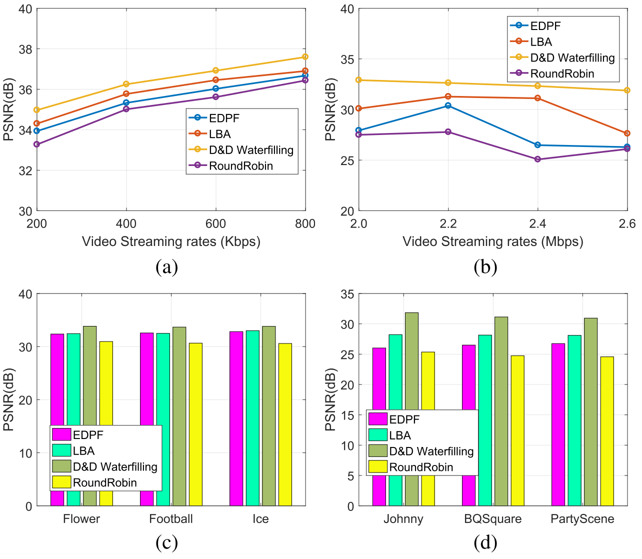
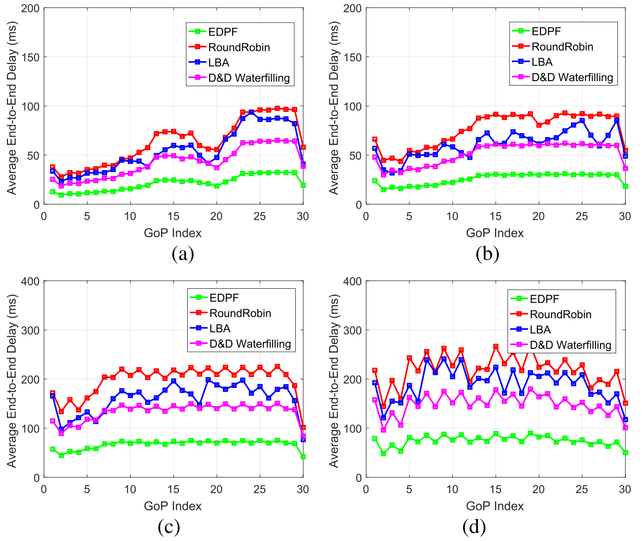


FIGURE 8. **Comparisons of subjective received video qualities under limited bandwidths with poor channel statuses. (a) Raw frame (top to bottom: *Ice*, *Football*, *Flower*, *Canoe*). (b) D&D Water-filling. (c) LBA. (d) EDPF. (e) Round robin.**



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

FIGURE 9. **(a), (b): Average PSNR results for different streaming rates. (c), (d): Average PSNR results for different sequences. (a) Low bitrates sequences. (b) High bitrates sequences. (c) Low bitrates sequences. (d) High bitrates sequences.**



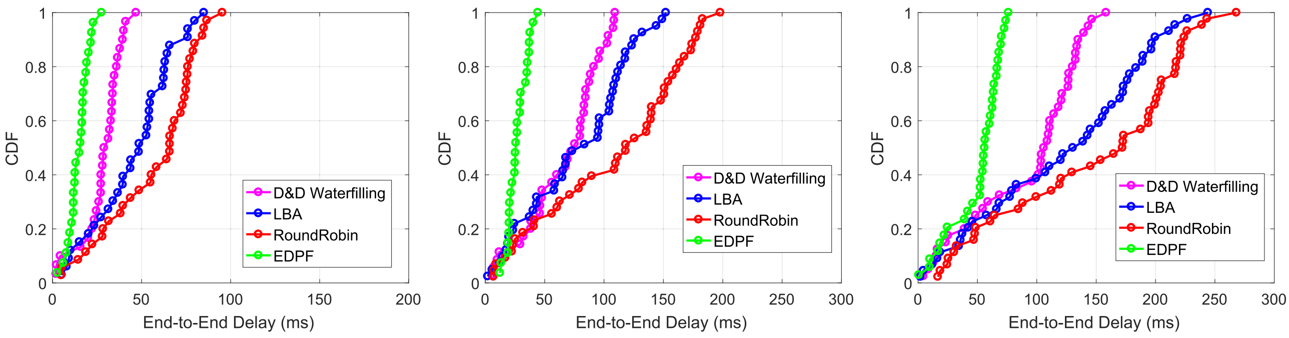
(a) (b) (c) (d)

FIGURE 10. **Comparison of the average end-to-end delays of GoPs indexed from 1 to 30. (a) *City* (600kbps). (b) *Football* (600kbps). (c) *Johnny* (1.8Mbps). (d) *PartyScene* (2.0Mbps).**

generality, we provide the average PSNR results for different video streaming rates and video sequences, the details of which are shown in Fig. 9. For low-bitrate transmissions, the PSNR value shows an increasing trend when the bitrates increase, while it shows an opposite trend under the condition of high bitrates. For high bitrates, the increase of bitrates increases the dropping rate, and thus, reduces the received quality. In summary, the proposed method performs well in terms of the received video quality.

C. DELAY ANALYSES

Fig. 12 plots the average end-to-end delays in the GoP units of different sequences. For each GoP unit, the delay is computed as the average value of all frames in it. Fig. 10(a), 10(b) plot the delay values for low-bitrate sequences and Fig. 10(c), 10(d) plot the delay values for high-bitrate sequences. It can be observed that when the streaming bitrate is low, the delay values do not exceed 100 ms for the four scheduling methods. However, when the streaming bitrate is high, some methods exceed the delay deadline T. We can see that EDPF achieves the lowest end-to-end delay, followed by our method, as EDPF always guarantees that a newly arrived packet is scheduled over the path with the shortest transmission delay. Obviously, our method considers both the transmission delay and the distortion by making a trade-off between them. Hence, the received video quality is improved at the expense of extra end-to-end delay. To have a close-up view of the delay performance results, we plot the CDFs



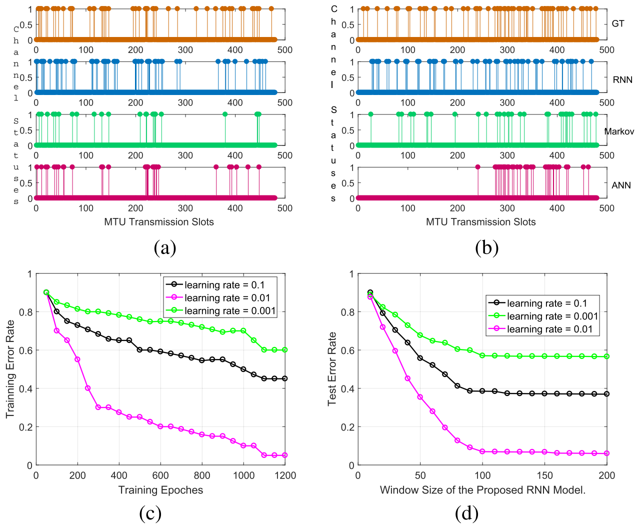
(a) (b) (c)

FIGURE 11. CDF of end-to-end transmission delay for different streaming rates. (a) 600kbps. (b) 1.5Mbps. (c) 2.0Mbps.

of the end-to-end frame delays for different video streaming rates, for comparison. Without loss of generality, all the selected low-bitrate and high-bitrate sequences are tested to collect the delay data, and each sequence is tested several times to acquire the average data. As given in Fig. 11, the proposed D&D water-filling algorithm guarantees that more than 50 percent of video frames are delivered in 30 ms for low-bitrate streams and in 100 ms for high-bitrate streams. As the bitrates increase, the CDF curves expand from left to right, which indicates that the frame delay increases owing to the limited transmission bandwidth.

D. PATH STATUS PERFORMANCE ANALYSES

We provide explicit analyses of the communication path statuses by inspecting the outputs of the network monitor. As stated earlier, the network monitor collects the packet's statistical information from the sender and the receiver. It also sends some probe packets periodically to acquire the real-time bandwidth of each path. Since the packets are encapsulated by the *MTU* units, during each *MTU* transmission time slot, any bit-error or packet loss circumstance is regarded as a bad channel status ($\chi_p(t) = \mathcal{B}$). Note that some error bits can be corrected using the FEC decoder. However, we still treat the channel status as bad since some bits are mis-demodulated because of the channel distortion. As depicted in Sec. II-C, we test the performance of the path status estimation by comparing the proposed RNN model with the Markov model [45] and the ANN model. In the Markov model, the transition probability from state i to state j in time $K\tau$: $P[\chi_p(t+K\tau) = j|\chi_p(t) = i]$ is computed using the statistical expectation of the stationary probabilities $\pi^{\mathcal{G}}$ and $\pi^{\mathcal{B}}$, in which they stand for the stationary probabilities of the Good and Bad states. In addition, $\pi^{\mathcal{G}}$ and $\pi^{\mathcal{B}}$ are updated from the historical data, and K denotes the window size for the transmission slots. In the ANN model, we use a neural network with two hidden layers. The Radbas function is used as the inner function linked between the input and hidden layers for nonlinear prediction, and the output layer contains a pure linear function followed by the sigmoid function. Fig. 12(a) and 12(b) plot a path's real-time status and the estimation results for different channel SNR values of 13 and 10 dB, respectively.



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

FIGURE 12. (a), (b): Path statuses monitoring results and their evaluations based on different approaches. (From top to bottom: The ground truth statuses, RNN's prediction, Markov model's prediction, ANN's prediction) (c), (d): Performance analyses of the proposed RNN model based on different parameters. (a) Avg channel SNR = 13 dB. (b) Avg channel SNR = 10 dB. (c) Training Error Rate vs Epochs. (d) Test Error Rate vs Window Size.

One can see that the proposed RNN model achieves a higher estimation precision than other models; especially, when the condition of poor path status occurs, the RNN model is able to accurately forecast the bad statuses based on the previous data. To see how the system parameters influence the learning performance, we change the learning rate to 0.001, 0.01, and 0.1. The relationship curves between the training error rates and the training epochs under different learning rates are provided in Fig. 12(c). We see that when the learning rate is quite small (0.001), the RNN model fails to minimize the loss function under the limited training epochs owing to a small gradient descend step. However, when the learning rate becomes large (0.1), the RNN model fails to learn the data features, as it always skips the optimal point on a large descend step. Moreover, we validate the test error rate of the neural network by increasing the window size K. As shown in Fig. 12(d), the test error rate shows a decreasing trend as the window size increases. When the size is larger than 100, the error rate becomes stable. Since the window size is sufficient for acquiring enough correlated information among the observed data, the model reaches the bottleneck of the error rate. However, increasing the window size would increase the computational complexity and memory consumption owing to the feature of exponential computing complexity in most deep learning systems.

IV. CONCLUSION

In this paper, we presented a novel framework for the ROI-based video transmission in heterogeneous wireless networks with multihomed terminals. The framework was able to guarantee acceptable qualities for the ROI components in a frame, especially, under limited bandwidths. Based on the mathematical discussions on the video R-D model and the wireless network model , we proposed a novel scheduling approach in which both the delay and distortion factors were taken into consideration. The scheduling approach ensured that the MTUs belonging to the ROI group were delivered over paths with low delays and distortions, first. Further, the water-filling idea adopted in our scheduling approach maximized the network's resource utilization rate and minimized the total end-to-end delay. We also proposed a deep-learning approach for channel-status estimation, which improved the estimation precision. This enhanced the performance of the proposed scheduling approach as well. In future works, we will consider employing rate-control methods in the codec, which suit the current network conditions. Besides, more work will be considered on the FEC coder , in which the trade-off between error bits and redundant bits is a meaningful issue.

REFERENCES

[1] C. Eklund, R. B. Marks, K. L. Stanwood, and S. Wang, "IEEE standard 802.16: A technical overview of the WirelessMAN/sup TM/air interface for broadband wireless access," *IEEE Commun. Mag.*, vol. 40, no. 6, pp. 98–107, Jun. 2002.

[2] C. Zhang, S. L. Ariyavisitakul, and M. Tao, "LTE-advanced and 4G wireless communications [guest editorial]," *IEEE Commun. Mag.*, vol. 50, no. 2, pp. 102–103, Feb. 2012.

[3] "Cisco visual networking index: Forecast and methodology, 2016–2021," Cisco Cooperation, San Jose, CA, USA, White Paper, Feb. 2011.

[4] S. Han, H. Joo, D. Lee, and H. Song, "An end-to-end virtual path construction system for stable live video streaming over heterogeneous wireless networks," *IEEE J. Sel. Area Commun.*, vol. 29, no. 5, pp. 1032–1041, May 2011.

[5] J. Yoon, H. Zhang, S. Banerjee, and S. Rangarajan, "MuVi: A multicast video delivery scheme for 4G cellular networks," in *Proc. 18th Annu. Int. Conf. Mobile Comput. Netw. (Mobicom)*, New York, NY, USA, 2012, pp. 209–220. [Online]. Available: http://doi.acm.org/10.1145/2348543.2348571

[6] J. Wu, X. Qiao, Y. Xia, C. Yuen, and J. Chen, "A low-latency scheduling approach for high-definition video streaming in a heterogeneous wireless network with multihomed clients," *Multimedia Syst.*, vol. 21, no. 4, pp. 411–425, Jul. 2015. [Online]. Available: http://dx.doi.org/10.1007/s00530-014-0388-7

[7] L. Tong and K. R. Rao, "Region of interest based H. 263 compatible codec and its rate control for low bit rate video conferencing," in *Proc. Int. Symp. Intell. Signal Process. Commun. Syst. (ISPACS)*, Dec. 2005, pp. 249–252.

[8] L. Yang, L. Zhang, S. Ma, and D. Zhao, "A ROI quality adjustable rate control scheme for low bitrate video coding," in *Proc. Picture Coding Symp.*, May 2009, pp. 1–4.

[9] H. Meuel, M. Munderloh, and J. Ostermann, "Low bit rate ROI based video coding for HDTV aerial surveillance video sequences," in *Proc. CVPR WORKSHOPS*, Jun. 2011, pp. 13–20.

[10] M. Meddeb, M. Cagnazzo, and B. Pesquet-Popescu, "ROI-based rate control using tiles for an HEVC encoded video stream over a lossy network," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2015, pp. 1389–1393.

[11] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand, "Overview of the high efficiency video coding (HEVC) standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1649–1668, Dec. 2012.

[12] D. Mukherjee *et al.*, "The latest open-source video codec VP9—An overview and preliminary results," in *Proc. Picture Coding Symp. (PCS)*, Dec. 2013, pp. 390–393.

[13] L. Lu, J. V. Mccanny, and S. Sezer, "Reconfigurable system-on-a-chip motion estimation architecture for multi-standard video coding," *IET Comput. Digit. Techn.*, vol. 4, no. 5, pp. 349–364, Sep. 2010.

[14] G. Pastuszak, "Architecture design of the H.264/AVC encoder based on rate-distortion optimization," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 25, no. 11, pp. 1844–1856, Nov. 2015.

[15] J. L. Nunez-Yanez, A. Nabina, E. Hung, and G. Vafiadis, "Cogeneration of fast motion estimation processors and algorithms for advanced video coding," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 20, no. 3, pp. 437–448, Mar. 2012.

[16] W. Song and W. Zhuang, "Performance analysis of probabilistic multipath transmission of video streaming traffic over multi-radio wireless devices," *IEEE Trans. Wireless Commun.*, vol. 11, no. 4, pp. 1554–1564, Apr. 2012.

[17] 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.

[18] JV Groups. *H.264/AVC Reference Software JM 18.6*. Accessed: Aug. 2014. [Online]. Available: http://iphome.hhi.de/suehring/tml/

[19] K. Chebrolu and R. R. Rao, "Bandwidth aggregation for real-time applications in heterogeneous wireless networks," *IEEE Trans. Mobile Comput.*, vol. 5, no. 4, pp. 388–403, Apr. 2006.

[20] D. Jurca and P. Frossard, "Video packet selection and scheduling for multipath streaming," *IEEE Trans. Multimedia*, vol. 9, no. 3, pp. 629–641, Apr. 2007. [Online]. Available: http://dx.doi.org/10.1109/TMM.2006.888017

[21] X. Zhu, P. Agrawal, J. P. Singh, T. Alpcan, and B. Girod, "Distributed rate allocation policies for multihomed video streaming over heterogeneous access networks," *IEEE Trans. Multimedia*, vol. 11, no. 4, pp. 752–764, Jun. 2009.

[22] T. Goff and D. S. Phatak, "Unified transport layer support for data striping and host mobility," *IEEE J. Sel. Areas Commun.*, vol. 22, no. 4, pp. 737–746, May 2004.

[23] A. El Gamal, J. Mammen, B. Prabhakar, and D. Shah, "Optimal throughput-delay scaling in wireless networks—Part I: The fluid model," *IEEE Trans. Inf. Theory*, vol. 52, no. 6, pp. 2568–2592, Jun. 2006.

[24] A. El Gamal, J. Mammen, B. Prabhakar, and D. Shah, "Optimal throughputŰdelay scaling in wireless networks—Part II: Constant-size packets," *IEEE Trans. Inf. Theory*, vol. 52, no. 11, pp. 5111–5116, Nov. 2006.

[25] J. Abouei, A. Bayesteh, and A. K. Khandani, "Delay-throughput analysis in decentralized single-hop wireless networks," in *Proc. IEEE Int. Symp. Inf. Theory*, Jun. 2007, pp. 1401–1405.

[26] C. Gong and X. Wang, "Adaptive transmission for delay-constrained wireless video," *IEEE Trans. Wireless Commun.*, vol. 13, no. 1, pp. 49–61, Jan. 2014.

[27] J. Lee and S. Bahk, "On the MDP-based cost minimization for video-on-demand services in a heterogeneous wireless network with multihomed terminals," *IEEE Trans. Mobile Comput.*, vol. 12, no. 9, pp. 1737–1749, Sep. 2013.

[28] J. Wang, R. V. Prasad, and I. G. M. M. Niemegeers, "Solving the incertitude of vertical handovers in heterogeneous mobile wireless network using MDP," in *Proc. IEEE Int. Conf. Commun.*, May 2008, pp. 2187–2192.

[29] H. Song and C.-C. J. Kuo, "A region-based H.263+ codec and its rate control for low VBR video," *IEEE Trans. Multimedia*, vol. 6, no. 3, pp. 489–500, Jun. 2004.

[30] H. Li, Z. Wang, H. Cui, and K. Tang, "An improved ROI-based rate control algorithm for H. 264/AVC," in *Proc. 8th Int. Conf. Signal Process.*, vol. 2. Nov. 2006, pp. 1–5.

[31] B. Lee and M. Kim, "Modeling rates and distortions based on a mixture of Laplacian distributions for inter-predicted residues in quadtree coding of HEVC," *IEEE Signal Process. Lett.*, vol. 18, no. 10, pp. 571–574, Oct. 2011.

[32] B. Lee, M. Kim, and T. Q. Nguyen, "A frame-level rate control scheme based on texture and nontexture rate models for high efficiency video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 24, no. 3, pp. 465–479, Mar. 2014.

[33] Z. Zhang, T. Jing, J. Han, Y. Xu, and F. Zhang, "A new rate control scheme for video coding based on region of interest," *IEEE Access*, vol. 5, pp. 13677–13688, 2017, doi: 10.1109/ACCESS.2017.2676125.

[34] X. Zhou, C. Yang, H. Zhao, and W. Yu, "Low-rank modeling and its applications in image analysis," *ACM Comput. Surveys*, vol. 47, no. 2, pp. 36:1–36:33, Dec. 2014. [Online]. Available: http://doi.acm.org/10.1145/2674559, doi: 10.1145/2674559.

[35] E. Candés, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis: Recovering low-rank matrices from sparse errors," in *Proc. IEEE Sensor Array Multichannel Signal Process. Workshop (SAM)*, Oct. 2010, pp. 201–204.

[36] C. Guyon, T. Bouwmans, and E.-H. Zahzah, "Foreground detection based on low-rank and block-sparse matrix decomposition," in *Proc. 19th IEEE Int. Conf. Image Process.*, Orlando, FL, USA, Oct. 2012, pp. 1225–1228.

[37] Y. Peng, A. Ganesh, J. Wright, W. Xu, and Y. Ma, "RASL: Robust alignment by sparse and low-rank decomposition for linearly correlated images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 11, pp. 2233–2246, Nov. 2012.

[38] Z. Zivkovic, "Improved adaptive Gaussian mixture model for background subtraction," in *Proc. 17th Int. Conf. Pattern Recognit. (ICPR)*, vol. 2. Aug. 2004, pp. 28–31.

[39] C. Stauffer and W. E. L. Grimson, "Adaptive background mixture models for real-time tracking," in *Proc. IEEE CVPR*, vol. 2. Jun. 1999, p. 252.

[40] P. Chiranjeevi and S. Sengupta, "Detection of moving objects using multichannel kernel fuzzy correlogram based background subtraction," *IEEE Trans. Cybern.*, vol. 44, no. 6, pp. 870–881, Jun. 2014.

[41] J. Yao and J.-M. Odobez, "Multi-layer background subtraction based on color and texture," in *Proc. IEEE Conf. CVPR*, Jun. 2007, pp. 1–8.

[42] H.-M. Hang and J.-J. Chen, "Source model for transform video coder and its application. I. Fundamental theory," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 7, no. 2, pp. 287–298, Apr. 1997.

[43] J. Ribas-Corbera and S. Lei, "Rate control in DCT video coding for low-delay communications," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 9, no. 1, pp. 172–185, Feb. 1999.

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

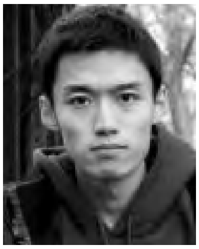
[45] E. N. Gilbert, "Capacity of a burst-noise channel," *Bell Syst. Tech. J.*, vol. 39, no. 5, pp. 1253–1265, 1960.

[46] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016. [Online]. Available: http://www.deeplearningbook.org

[47] JV Groups. *OMNeT++ Offical*. [Online]. Available: https://omnetpp.org/

[48] H. Adiseshu, G. Parulkar, and G. Varghese, "A reliable and scalable striping protocol," in *Proc. ACM SIGCOMM*, 1996, pp. 131–141.

[49] T. Wiegand, H. Schwarz, A. Joch, F. Kossentini, and G. J. Sullivan, "Rate-constrained coder control and comparison of video coding standards," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 688–703, Jul. 2003.



ZHEWEI ZHANG received the B.S. degree from Beijing Jiaotong University, Beijing China, where he is currently pursuing the Ph.D. degree of electronic engineering and information system. His current research interests include video compression, image process, machine learning, and pattern recognition and analysis.

TAO JING (M'14) received the M.S. degree from the Changchun Institute of Optics in 1994 and the Ph.D. degree in fine mechanics and physics from the Chinese Academy of Sciences in 1999. He is currently a Professor with the School of Electronic and Information Engineering, Beijing Jiaotong University, China. His research interests include capacity analysis, spectrum prediction and resource management in cognitive radio networks, RFID in intelligent transporting system, smart phone application, and multimedia.

JINGNING HAN (M'11) received the B.S. degree in electrical engineering from Tsinghua University, Beijing, China, in 2007, and the M.S. and Ph.D. degrees in electrical and computer engineering from the University of California at Santa Barbara, Santa Barbara, CA, USA, in 2008 and 2012, respectively. He is currently with the WebM Codec Team, Google Inc., Mountain View, CA, USA, where he is involved in video compression, processing, and related technologies. His research interests include video coding and computer architecture. He was a recipient of the Outstanding Teaching Assistant Awards from the Department of Electrical and Computer Engineering, University of California at Santa Barbara, in 2010 and 2011, the Dissertation Fellowship in 2012, and the Best Student Paper Award at the IEEE International Conference on Multimedia and Expo in 2012.

YAOWU XU (M'10) received the B.S. degree in electrical engineering from Tsinghua University, Beijing, China, in 2007, and the M.S. and Ph.D. degrees in electrical and computer engineering from the University of Rochester at New York, New York, USA, in 2008 and 2012, respectively. He is currently with the WebM Codec Team, Google Inc., Mountain View, CA, USA, where he is involved in video compression, processing, and related technologies. He is an expert in video compression and processing, digital image processing, embedded audio/video architecture, mobile audio/video architecture, hardware-software integration, and real-time multimedia embedded systems.

XUEJING LI received the B.E. degree in communication and information system from Beijing Jiaotong University, Beijing, China, in 2014, where he is currently pursuing the Ph.D. degree with the Next Generation Network Center. His research interests are cognitive radio networks, energy harvesting, and mobile social networks.

MEILIN GAO received the B.S. degree in information engineering from Shijiazhuang Tiedao University, Shijiazhuang, China, in 2013. She is currently pursuing the Ph.D. degree with the State Key Laboratory of Rail Traffic Control and Safety, Beijing Jiaotong University, Beijing, China. Her current research interests include wireless resource allocation and network architecture for high-mobility broadband wireless communications.