Streaming high-definition real-time video to mobile devices with partially reliable transfer

Wu, Jiyang; Tan, Rui; Wang, Ming

2019

Wu, J., Tan, R., & Wang, M. (2019). Streaming High-Definition Real-Time Video to Mobile Devices with Partially Reliable Transfer. *IEEE Transactions on Mobile Computing*, 18(2), 458-472. doi:10.1109/TMC.2018.2836914

<https://hdl.handle.net/10356/90125>

<https://doi.org/10.1109/TMC.2018.2836914>

© 2018 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works. The published version is available at: <https://doi.org/10.1109/TMC.2018.2836914>.

Downloaded on 26 Apr 2023 12:49:31 SGT

Streaming High-Definition Real-Time Video to Mobile Devices With Partially Reliable Transfer

Jiyan Wu, *Member, IEEE*, Rui Tan, *Senior Member, IEEE*, and Ming Wang

Abstract—Delivering High-Definition (HD) real-time video to mobile devices is challenged with stringent constraints in delay and reliability. In the presence of network dynamics (*e.g.*, channel errors and bandwidth fluctuations), the existing error-control mechanisms [*e.g.*, ARQ (Automatic Repeat reQuest) and FEC (Forward Error Correction)] frequently induce deadline violations and quality degradations. To strike an effective balance between delay and reliability in real-time video transmission, this research presents an application-layer solution dubbed PERES (Partial rELiability based Real-timE Streaming) to perform partially reliable transfer. First, we develop an analytical framework to model the delay-constrained partial reliability for ACK (acknowledgement) and NAK (negative acknowledgement) based real-time video streaming. Second, we propose scheduling algorithms for video-aware reliability adaptation and network-adaptive buffer control. *PERES is able to maximize the transmission reliability of high-priority video frames within stringent delay constraint.* We implement the proposed transmission scheme in embedded video monitoring systems and evaluate the efficacy over different wireless network environments. Evaluation results demonstrate PERES achieves appreciable improvements over the reference schemes in perceptual video quality, delay performance and bandwidth efficiency.

Index Terms—Mobile video streaming; stringent delay constraint; real-time video application; partial reliability; buffer control; wireless networks.

1 INTRODUCTION

Driven by the technological advancements in hand-held devices (*e.g.*, smartphone and iPad) and video compression (*e.g.*, H.264/AVC [1], H.264/SVC [2] and H.265 [3]), mobile video streaming has already dominated Internet multimedia applications. According to the latest Cisco visual networking index [4], video streaming accounts for 55% of mobile Internet traffic in 2015 and this portion will exceed 75% by 2020. In particular, High-Definition (HD) video services have prevailed over standard-quality streaming in 2012 and are becoming more prevalent as mobile connection speed increases.

An important tendency with the tremendous growth of HD video streaming is the proliferation of real-time applications, *e.g.*, Skype, Google+, Web Real-Time Communication [5], video gaming, cloud video recording [6], *etc.* However, streaming HD real-time videos to mobile devices poses crucial challenges in unreliable wireless environments. The critical issues include the following aspects.

- **Delay constraint.** Real-time video transmission has stringent delay constraint of less than 200 ms to prevent video stalls and glitches.
- **Reliability requirement.** To achieve excellent perceptual video quality, the transmission reliability needs to be above 99% [7]. However, the packet loss rate in wireless environments is up to 30% [8].

- **Bandwidth consumption.** Streaming HD real-time video is bandwidth-demanding, *e.g.*, a good-quality Skype video call consumes bandwidth above 900 Kbps [9]. Furthermore, higher bandwidth consumption also substantially increases the energy consumption of battery-limited mobile devices [10][45].
- **Network dynamics.** Wireless networks are characterized by time-varying channel status and limited radio resources. In particular, the bandwidth fluctuations and channel errors frequently occur to induce noticeable video quality degradations.

A plethora of error-control transmission schemes (*e.g.*, Automatic Repeat reQuest, Forward Error Correction, Hybrid ARQ) have been proposed to optimize end-to-end real-time video transmission over wireless networks. First, the ARQ (*i.e.*, retransmission) schemes [11–13] enable 100% reliability but also suffer from long retransmission delay. Second, the FEC solutions [14–18] provide a certain level of reliability but also induce higher bandwidth consumption (*e.g.*, 20 – 50% redundancy) and larger coding latency. Third, the HARQ (Hybrid ARQ) methods [30][46] are conventionally plagued with the deficiencies of both FEC and ARQ schemes. Thus, the real-time and reliability performance of the existing error-control solutions are still inadequate to achieve excellent real-time video quality.

To address these critical problems, this paper presents an application-layer transmission scheme dubbed PERES (Partial rELiability based Real-timE Streaming). To provide flexible error resilience against network packet losses, PERES adopts a hybrid NAK (negative acknowledgement) and ACK (acknowledgement) mechanism for video data protection. The maximum retransmissions

• Jiyan Wu and Rui Tan are with the School of Computer Science and Engineering, Nanyang Technological University, Singapore 639798. Email: wujiyan@ieee.org, tanrui@ntu.edu.sg.
 • Ming Wang is with the State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, P. R. China. Email: wangming_bupt@bupt.edu.cn.

and timeout values are dynamically adjusted for video frames with different priorities. PERES is able to effectively mitigate unnecessary and overdue retransmissions to optimize delay performance and bandwidth efficiency. The packet buffers in communication terminals are automatically controlled to guarantee video streaming fluency. Distinct from the existing ARQ/retransmission schemes, PERES strikes an effective balance between real-time and reliability performance with proactive retransmission and buffer control. *Therefore, the proposed solution addresses the shortcomings of FEC and ARQ schemes to maximize real-time video streaming quality.*

The contributions of this paper are summarized as follows.

- (C1) Develop a mathematical model to analyze the real-time and reliability performance for the NAK & ACK based real-time video transmission over wireless networks (Section 3). An integer programming problem is formulated to maximize video quality under bandwidth and delay constraints.
- (C2) Propose scheduling algorithms for reliability adaptation and buffer control (Section 4).
 - **Reliability adaptation:** dynamically controls the timeout events and maximum retransmissions for different video frames within imposed deadline and available bandwidth.
 - **Buffer control:** adaptively regulates the buffer behavior and streaming fluency with network-adaptive and delay-aware control algorithm.
- (C3) Implement the proposed PERES in an embedded video monitoring system and evaluate the efficacy in different wireless network environments (Section 5). Experimental results demonstrate PERES achieves appreciable performance improvements over the FEC [19], ARQ and TCP (Transmission Control Protocol) [20] in perceptual video quality, end-to-end delay and bandwidth efficiency.

The remainder of this paper is structured as follows. Section 2 reviews relevant existing work. The system model and problem formulation are presented in Section 3. Section 4 describes the scheduling algorithms of the proposed PERES scheme. The system implementation and evaluation results are provided in Section 5. Section 6 concludes this research and discusses the future work.

2 RELATED WORK

The existing studies close to this work can be divided into two categories: 1) HD and real-time video communication; 2) error-resilient video transmission schemes (e.g., ARQ, FEC and HARQ). The term “partially reliable transfer” is also used in the stream control transmission protocol [21] at the transport layer. In this paper, partially reliable transfer represents a completely different solution using ACK & NAK mechanism at the application layer.

2.1 HD and Real-Time Video Communication

Mukerjee *et al.* [22] introduce a centralized controller named VDN to optimize quality, cost and scalability of Internet-scale live video delivery. Baik *et al.* [23] propose a video-aware MAC that incorporates perceptual error tolerance and NAK-based adaptive window for real-time video streaming. A delay-stringent coded transmission scheme named ASCOT is proposed in reference [24] to address the problem of large-size video frames in HD video streaming. In literature [18], the authors propose a priority-aware FEC coding scheme to mitigate the sporadic and burst packet losses of HD mobile video delivery with TCP. Wu *et al.* [25] propose a Content-Aware Concurrent Multipath Transfer scheme that incorporates video-aware chunk scheduling in Stream Control Transmission Protocol (SCTP) to perform unequal data transmission. A video frame scheduling approach based on Weibull distribution and graph theory is presented in [26] to optimize the delay performance of multipath HD video transmission.

In summary, the problem of streaming real-time video with hybrid NAK & ACK mechanism has not been investigated in the existing studies. Since NAK and ACK have supplementary characteristics in different transmission scenarios, it is important to study the problem for partially reliable video transfer.

2.2 Error-Resilient Schemes

The authors in literature [27] generally review and discuss the recent studies on FEC coding for video communication. Ahmad *et al.* [28] propose a rateless coding scheme that keeps on sending the encoded symbols until receiving an acknowledgement or passing the deadline. Tournoux *et al.* [29] propose an on-the-fly erasure coding scheme called Tetrys that considers the feedback information during the FEC coding process. Huang *et al.* [30] propose a hybrid FEC/ARQ protocol built on a packet streaming code. Sgardoni *et al.* [14] propose an APP/PHY cross-layer design featured by Raptor code awareness to optimize high-quality live video unicast over mobile broadband networks. In [15], a cross-layer FEC scheme using Raptor and rate compatible punctuated convolutional (RCPC) codes is proposed for video transmission over wireless channels. Xiao *et al.* [16] propose a sub-GoP level FEC coding scheme to reduce end-to-end delay in real-time video applications. A transport protocol is developed by the authors in reference [17] to stream video over multiple paths using systematic Raptor codes. A FEC coding scheme named CLOSET is developed in reference [31] to leverage the delay-friendliness of TCP in real-time video communication. In [43], a randomized expanding FEC coding scheme is proposed to append parity packets for current and previous frames for enhanced data protection. Lee *et al.* [45] propose an LT encoding pattern aware packetization algorithm that maps encoding symbols to different video frames in order to minimize packet loss effects.

In conclusion, the above error-control schemes (ARQ, FEC, HARQ) are not specifically designed to balance the tradeoff between delay and reliability should network dynamics (e.g., bandwidth fluctuations and packet losses) occur.

To the best of our knowledge, this is the first research to: 1) develops an analytical framework to model the delay and reliability performance for NAK & ACK based real-time video streaming; 2) implements an application-layer partially reliable transmission scheme with hybrid NAK & ACK in real-time embedded systems.

3 SYSTEM MODEL

3.1 System Overview

The system overview of the proposed transmission framework is presented in Figure 1. We consider the problem of end-to-end real-time video transmission to mobile devices in wireless networks (e.g., LTE, Wi-Fi, 3G, etc.). The objective of the proposed transmission scheme is to balance the delay and reliability performance of real-time streaming to achieve optimal video quality. The proposed PERES is developed for streaming real-time videos encoded with *H.264 video codec*. In particular, the video frame information (e.g., the I, P frame types) is taken into account in the reliability adaptation (Algorithm 1) and buffer control (Algorithm 2) schemes. Furthermore, the codec efficiency also has significant effect on the video transmission performance. This is because we consider real-time applications with stringent delay constraint. To achieve low-latency video compression, we use the hardware encoder in the processing chip as described in Section 5.

In particular, the basic system settings of the proposed PERES framework are listed as follows.

Setting 1 (UDP socket). *In this real-time video communication system, the User Datagram Protocol (UDP) socket is adopted for video data and control information (e.g., NAK, ACK, command packets) transfer.*

Setting 2 (128-KByte buffer size). *The sender and receiver buffer sizes are both 128 KBytes (packets) to balance the resilience against network dynamics and timeliness of received videos. This buffer size is also a widely adopted setting in real-time multimedia streaming systems.*

Setting 3 (Hybrid NAK & ACK mechanism). *The error-resilience scheme adopted in the video communication system is a hybrid ACK (acknowledgment) and NAK (negative acknowledgment) retransmission mechanism. These terminologies are defined as follows.*

Definition 1 (ACK-based retransmission). *An ACK packet informs the sender of the successful delivery of the packet. If the ACK of a packet is not received within the timeout (RTO), this packet will be retransmitted from the sender to the receiver.*

Definition 2 (NAK-based retransmission). *A NAK packet notifies the sender of the detected lost packet and proactively requires the retransmission.*

The working procedures and decision blocks in both end hosts are summarized in the following.

Sender: In each decision epoch, the video frames are pulled from the video encoder to the sender buffer. The solution procedure to deal with the cases of sender buffer full/available is presented in Section 4.3. Each video frame is split into several packets with size S . If the number of buffered packets is less than the maximum buffer size, these packets are pulled to the sender buffer and delivered to the destination simultaneously. To respect the network friendliness, the TCP-friendly rate control (TFRC) algorithm [32] is employed to dynamically regulate transmission rate. However, the proposed PERES is different from TCP-friendly Rate Control (TFRC) in the following aspects: 1) NAK and ACK mechanisms are adopted in PERES for data protection while TFRC does not include error-control schemes; 2) PERES takes into account the video frame priority in reliability adaptation (Algorithm 1) while TFRC is a video-agnostic rate control scheme. The retransmission timeout and maximum retransmission are specified in the algorithm of Section 4.2.

Receiver: At the receiver side, the arrived packets are stored and reordered in the receiver buffer. Once a packet has successfully arrived at the receiver, an ACK packet with the sequence number will be sent back to the sender side. In order to facilitate the window sliding and mitigate the delayed/lost ACK problem, a single ACK packet notifies the sender buffer to change the lower bound as the same sequence number. Furthermore, the NAK packets also carry the lower bound of receiver buffer for piggybacking ACK information. If the received packet is not the lower bound of the receiver buffer, the NAK packets are sent back to require the transmissions of these absent packets. The details of sender & receiver buffer control algorithm are described in Section 4.2.

To formulate the optimization problem of video quality maximization, we first present an analytical framework to study the NAK and ACK based real-time video streaming over wireless networks. This analytical model is suitable and specific for HD video content because: 1) HD video streaming is characterized by high bit rate and throughput demand (e.g., above 4 Mbps). NAK and ACK mechanisms are able to provide bandwidth-efficient and timely data protection, i.e., without incurring fixed-level data redundancy or substantial packet retransmissions; 2) A major issue of HD video streaming is the frame-level delay [33]. This model provides detailed analysis on the delay performance at video frame level (Section 3.2). The proposed algorithm in Section 4.1 also adjusts transmission reliability at frame level to improve delay profile. Therefore, this model and algorithm are suitable for streaming HD real-time videos featured by long frame-level latency. The mathematical notations and acronyms used in this paper are summarized in Table 1.

Fig. 1. Overall design of the proposed partial reliability based real-time streaming (PERES) framework.

TABLE 1
Basic notations.

Symbol	Definition
$\mathbb{P}, \mathbb{E}$	the probability, expectation value.
π_B	the packet loss rate.
G/B	the Good/Bad state of the communication network.
ξ_B/ξ_G	the state transition probability from G/B to B/G .
P_f	the data loss ratio of f -th frame.
R_f, E_f	the reliability value, expired loss rate of f -th frame.
l_f	the distortion impact value of f -th frame.
S_f	the delivery status of the f -th video frame.
d_f	the end-to-end delay of the f -th video frame.
$N_{\max}$	the maximum retransmission number.
T	the delay constraint for each video GoP.
n_f	the number of video packets for the f -th frame.
$K_{\langle i,j \rangle}(\theta)$	the transition probability from state i to j in time θ .
S	the application-layer video packet size.
F, M	the GoP size, video frame rate.
Acronym	Definition
RTT	round trip time.
RTO	retransmission timeout.
ACK	acknowledgement.
NAK	negative acknowledgement.
MSS	maximum segment size.

3.2 ACK and NAK based Real-Time Streaming

This paper considers the problem of ACK & NAK based real-time transmission at video GoP level. A GoP includes F frames and each is numbered with an index f ($1 \leq f \leq F$). Let the symbol M denote the video frame rate and the duration for each GoP is F/M . The B (bidirectional) video frames induce large coding delay because of the prediction from previous and subsequent frames. Therefore, the bidirectional mode is disabled to ensure low-delay video coding [24], *i.e.*, the encoded video stream consists exclusively of I (Intra) and P (Predicted) frames. Suppose the f -th video frame is divided into n_f UDP packets and $n_f = \lceil L_f/S \rceil$, in which $\lceil x \rceil$ denotes the smallest integer larger than x . A video frame can be successfully delivered if all the UDP packets are correctly received within the playout duration for each GoP (the receiver buffer is automatically moved forward for every F/M seconds). In the real-time video streaming system of this paper, the frame-level error-

concealment method for H.264 videos is adopted to mitigate consecutive frame losses. In order to facilitate the analysis of video quality, we present the derivations based on the frame-level distortion model developed in reference [33][34]. Specifically, the distortion of a single video frame is dependent on the frame-level data loss ratio (P_f) and the distortion impact from parent frames (denoted with symbol Θ_f) in the same GoP (*i.e.*, drifting distortion/error propagation), *i.e.*,

$$D_f = P_f + \sum_{f' \in \Theta_f} (P_{f'} \cdot l_{f'}). \quad (1)$$

The distortion impact value $l_{f'}$ is configured for video frames according to different types, *i.e.*, 5 for I frames and 2 for P frames [35]. The frame-level data loss ratio (P_f) in the above equation is a combined probability of transmission reliability (R_f) and expired loss rate (E_f), *i.e.*,

$$P_f = 1 - R_f + R_f \cdot E_f. \quad (2)$$

This frame-level data loss ratio P_f includes both the transmission impairments caused by communication network and expired packets received after deadline. Such loss ratio estimation method has also been verified and used in recent studies (*e.g.*, [24][33]) on video communication. Therefore, P_f is able to characterize the data losses of video frame with high accuracy. To further analyze this frame-level data loss ratio P_f , we provide derivations on the reliability R_f and delay performance d_f as follows.

Reliability Performance

The reliability of the f -th video frame is inversely proportional to the network packet loss ratio, *i.e.*,

$$R_f = 1 - \frac{\mathbb{E}[L(n_f + N)]}{n_f}, \quad (3)$$

in which L ($0 \leq L \leq n_f$) denotes the number of lost packets and N ($N \in [0, \infty)$) represents the number of retransmissions. L is the sum of transmission packets in the bad (B) network state, *i.e.*, $\sum_{i=1}^{n_f} 1_{\{c_i=B\}}$. Suppose the interval between the consecutive packet transmission is θ (*e.g.*, the execution time of each loop in the firmware program of embedded video systems). The packet loss pattern is analyzed based on Gilbert loss model [36]

and continuous-time Markov chain. Let the symbol $\mathbb{P}(L)$ represent the probability of lost L packets. The expected value of packet loss number ($\mathbb{E}(L)$) can be estimated as follows.

$$\begin{aligned} \mathbb{E}(L) &= \sum_{L=0}^{n_f+N} [L \cdot \mathbb{P}(L)], \\ &= \sum_{L=0}^{n_f} \left\{ L \cdot \left\{ \pi_{c_1} \cdot \prod_{i=1}^{n_f+N-1} [\mathbf{K}_{\langle c_i, c_{i+1} \rangle}(\theta)] \right\} \right\}_{\sum_{i=1}^{n_f+N} 1_{\{c_i=B\}}=L} \end{aligned} \quad (4)$$

In the above equation, $\mathbf{F}_{\langle c_i, c_{i+1} \rangle}(\theta)$ denotes the state transition probability from the i -th to $(i+1)$ -th packet in time θ . Suppose π_B and π_G denote the stationary probabilities of the Bad (B) and Good (G) states. The packet loss rate π_B is periodically measured based on the number of sent and received packets. This loss rate only indicates the stationary probability of packets encountering network losses, and does not include the packets received after deadline. The relationship between π_B and the average loss burst length $1/\xi_B$ can be expressed as $\pi_B = \frac{\xi_B}{\xi_B + \xi_G}$. According to the property of continuous-time Markov chain, we have the following formulas for state transitions from the good (G) state.

$$\begin{aligned} \mathbf{K}_{\langle G, G \rangle}(\theta) &= \pi_G + \pi_B \cdot \exp[-(\xi_B + \xi_G) \cdot \theta], \\ \mathbf{K}_{\langle G, B \rangle}(\theta) &= \pi_B - \pi_B \cdot \exp[-(\xi_B + \xi_G) \cdot \theta]. \end{aligned}$$

On the other side, the transition probabilities from the bad (B) state are expressed as follows.

$$\begin{aligned} \mathbf{K}_{\langle B, G \rangle}(\theta) &= \pi_G - \pi_G \cdot \exp[-(\xi_B + \xi_G) \cdot \theta], \\ \mathbf{K}_{\langle B, B \rangle}(\theta) &= \pi_B + \pi_G \cdot \exp[-(\xi_B + \xi_G) \cdot \theta]. \end{aligned}$$

From the above transition probability equations, we can observe $\mathbf{K}_{\langle c_i, c_{i+1} \rangle}(\theta) < 1$ for any $\theta > 0$. In particular, we have the following proposition on the relationship between reliability level and number of packet retransmissions.

Proposition 1. *For the same network stationary probabilities (π_B, π_G) and state transitions ($\mathbf{K}_{\langle c_i, c_{i+1} \rangle}(\theta)$), a larger number of packet retransmissions leads to higher reliability level, i.e.,*

$$N_1 > N_2 \Rightarrow \mathbf{R}_f(N_1) > \mathbf{R}_f(N_2).$$

Proof: Recall that the state transition probability $\mathbf{K}_{\langle c_i, c_{i+1} \rangle}(\theta)$ is smaller than 1 for any $\theta > 0$. We can obtain $\prod_{i=1}^{n_f+N_1-1} [\mathbf{K}_{\langle c_i, c_{i+1} \rangle}(\theta)] < \prod_{i=1}^{n_f+N_2-1} [\mathbf{K}_{\langle c_i, c_{i+1} \rangle}(\theta)]$ for the case of $N_1 < N_2$. According to Equation (4), the following derivation presents the relationship between number of packet retransmissions (N) and expected

value of lost packets ($\mathbb{E}(L)$)

$$\begin{aligned} N_1 > N_2 \\ &\Rightarrow \pi_{c_1} \cdot \prod_{i=1}^{n_f+N_1-1} [\mathbf{K}_{\langle c_i, c_{i+1} \rangle}(\theta)] < \pi_{c_1} \cdot \prod_{i=1}^{n_f+N_2-1} [\mathbf{K}_{\langle c_i, c_{i+1} \rangle}(\theta)], \\ &\Rightarrow \mathbb{P}[L(N_1)] < \mathbb{P}[L(N_2)] \Rightarrow L \cdot \mathbb{P}[L(N_1)] < L \cdot \mathbb{P}[L(N_2)], \\ &\Rightarrow \sum_{L=0}^{n_f} L \cdot \mathbb{P}[L(N_1)] < \sum_{L=0}^{n_f} L \cdot \mathbb{P}[L(N_2)], \\ &\Rightarrow \mathbb{E}[L(N_1)] < \mathbb{E}[L(N_2)]. \end{aligned}$$

From the above derivations, a larger of retransmissions leads to lower expected number of packet losses. Since the reliability value is $\mathbf{R}_f = 1 - \mathbb{E}[L(n_f + N)]/n_f$, we can obtain $N_1 > N_2 \Rightarrow \mathbf{R}_f(N_1) > \mathbf{R}_f(N_2)$. The proof for Proposition 1 is completed. $\square$

For the completely reliable transfer schemes (e.g., ARQ and TCP), the reliability value approaches 100% since the number of retransmissions is unlimited, i.e., $\lim_{N \rightarrow \infty} \mathbf{R}_f(N) = 100\%$.

Delay Performance

As introduced in Section 3.2, the f -th video frame in a GoP is divided into n_f UDP packets with size S . The end-to-end delay of a single video frame is estimated with the last packet latency, which includes the transmission and retransmission delays. Let the notation d denote the one-way (end-to-end) transmission delay of a video packet in the sender buffer. After receiving a video packet, d is updated based on the time stamp in the header and this information is sent back in the feedback packets. To facilitate the analysis, we estimate the NAK retransmission delay with $d + RTT_i$ (the actual delay for information feedback should be smaller than RTT_i). The retransmission delay is determined by the smaller value between timeout RTO_i and $d + RTT_i$, i.e., $\min\{RTO_i, d + RTT_i\}$. Specifically, $2d + RTT_i$ indicates the retransmission is triggered by NAK events and $d + RTO_i$ refers to the timeout retransmission delay. Suppose a packet encounters losses and is successfully delivered after N_i retransmissions, the end-to-end delay can be expressed as (assume there is no large variation during this packet transmission)

$$d_i = d + N_i \cdot \{d + \min\{RTT_i + d, RTO_i\}\}.$$

The end-to-end video frame delay is determined by the latency of the latest-arrival packet, i.e.,

$$d_f = d + (n_f - 1) \cdot \theta + \sum_i^{n_f} N_i \cdot \{d + \min\{RTT_i + d, RTO_i\}\},$$

where $(n_f - 1) \cdot \theta$ represents the transmission intervals between the n_f packets. The end-to-end video frame delay d_f takes into account the latencies of one-way transmission and possible retransmissions. The expired loss rate of the f -th video frame is estimated according to the empirical equation in [37]. And this equation is derived based on a large collection of experimental results

from real wireless networks. Suppose T represents the delay constraint on end-to-end video transmission (e.g., 150 ms for excellent video quality). The expired loss rate of the f -th video frame can be estimated as follows [37].

$$E_f = \exp \left\{ -\frac{T}{d_f} \right\}, \quad (5)$$

$$= \exp \left\{ -\frac{T}{d + (n_f - 1) \cdot \theta + \sum_i^{n_f} N_i \cdot \{d + \min \{RTT_i + d, RTO_i\}\}} \right\}.$$

Apparently, the end-to-end delay is proportional to the number of packet retransmissions, i.e., larger retransmission number entails higher end-to-end latency. The following proposition gives the relationship between the number of retransmissions and expired packet loss probability.

Proposition 2. For the same delay constraint T and one-way transmission latency d , a larger number of packet retransmissions incurs higher expired packet loss probability, i.e.,

$$N_1 > N_2 \Rightarrow E_f(N_1) > E_f(N_2).$$

Proof: Following the analysis on end-to-end delay in this subsection, the comparison of delay performance is as follows.

$$\begin{aligned} N_1 > N_2 &\Rightarrow d + N_1 \cdot \{d + \min \{RTT_i + d, RTO_i\}\} > \\ &d + N_2 \cdot \{d + \min \{RTT_i + d, RTO_i\}\}, \\ &\Rightarrow d_i(N_1) > d_i(N_2) \Rightarrow d_f(N_1) > d_f(N_2), \\ &\Rightarrow -\frac{T}{d_f(N_1)} > -\frac{T}{d_f(N_2)}. \end{aligned}$$

The exponential function $\exp(x)$ is monotonically increasing for any $x \in (-\infty, +\infty)$. From Equation (5), E_f is directly proportional to the end-to-end frame delay, i.e., $-\frac{T}{d_f(N_1)} > -\frac{T}{d_f(N_2)} \Rightarrow E_f(-\frac{T}{d_f(N_1)}) > E_f(-\frac{T}{d_f(N_2)})$. Thus, we can have $N_1 > N_2 \Rightarrow E_f(N_1) > E_f(N_2)$. $\square$

3.3 Network Measurements

Fig. 2. Network measurements for different RTT values.

For real-time video streaming, ACK and NAK mechanisms generate different effects on the packet arrival time. In the case of short-distance transmission, the NAK mechanism can perform early detection of packet losses. For the long-distance video delivery, ACK mechanism can periodically identify packet drops after RTO if the NAK feedback packets cannot return timely. To profile and analyze the performance of NAK & ACK based real-time video streaming, the network measurements

from an embedded video monitoring system are presented in this subsection. The detailed descriptions of system implementation and experimental environments are introduced in Section 5. Figure 2a shows the bandwidth consumption and retransmission ratio (number of retransmissions/total number of sent packets) with regard to different RTT values. These RTT levels indicate the transmission distances. The plots in Figure 2b depict the frame loss and video PSNR for these RTT values. Obviously, the average PSNR substantially decreases with RTT enlarges. This is mainly caused by the drop events of large-size I frames.

Fig. 3. Illustration of spurious NAKs and lost/delayed ACKs.

The results in Figure 2a show that the actual bandwidth consumption of the 1-Mbps video streaming almost doubles when the average RTT reaches around 350 ms. Such excessive data rate is caused by the superfluous retransmissions triggered by NAK and timeout events. As observed in the debug log of firmware, the same packet in the sender buffer (especially the one buffered at the lower bound) is highly likely to be retransmitted for multiple times. These unnecessary retransmissions are mainly induced by the spurious NAK and delayed/lost ACK. The spurious NAKs are caused by the out-of-order arrival packets as depicted in Figure 3. For instance, the NAK with sequence number 3 will be sent back if the third packet arrives later than the fourth packet. The delayed and lost ACK packets also incur unnecessary timeout retransmissions and increase the end-to-end delay.

In the case of massive packet losses, there may be many duplicate NAK packets sent from the receiver to request the retransmissions of same packet(s). These duplicate-NAKs significantly increase the unnecessary retransmissions, bandwidth consumption and end-to-end delay.

3.4 Motivation for Partial Reliability

This subsection presents the motivation for partial reliability of ACK & NAK based real-time video streaming. In order to provide different reliability levels, the maximum retransmissions are in each scenario deliberately limited to be 10%, 20%, 30%, 40% and 50% of total number of sent packets. The relationships between retransmission ratio, video PSNR and delay are presented in Figure 4a. As the results indicate, the end-to-end delay substantially increases as the retransmission ratio becomes larger.

Fig. 4. Performance of different retransmission ratios.

Paradoxically, a higher retransmissions ratio does not always guarantee better video quality in terms of PSNR. The evaluations demonstrate the retransmission ratio of 30% is able to achieve the highest PSNR values.

Figure 4b provides the explanation for the video quality differences. As the retransmission ratio increases, the reliability level R_f is effectively improved, *i.e.*, lower network packet loss ratio. However, the increased retransmission delay also incurs more expired packet arrivals as shown in the yellow plot. The frame-level loss ratio reaches the minimal value at the retransmission level of 30%.

Therefore, the motivation of the proposed partially reliable transfer scheme is to: 1) provide an appropriate level of reliability (by adapting the maximum number of retransmissions) to achieve optimal real-time streaming quality; 2) minimize the frame-level data loss ratio of high-priority video frames (*i.e.*, I frames) to mitigate severe quality degradations.

3.5 Problem Formulation

The above analysis shows that the number of retransmissions N_i and timeout RTO_i are critical to balance the delay and reliability tradeoff. According to the affine model in reference [33], the total distortion for a video GoP can be expressed as the sum of all single-frame distortions, *i.e.*,

$$\sum_{f=1}^F D_f = \sum_{f=1}^F \left\{ P_f + \sum_{f' \in \Theta_f} (P_{f'} \cdot I_{f'}) \right\}. \quad (6)$$

Furthermore, the less-important frames may be dropped if the number of buffered packets reaches the maximum buffer size (*e.g.*, network outage, link failure, or packet loss occurs). This frame-level delivery status in the decision process of PERES is denoted by the symbol S_f , where $S_f = 0$ means *dropped*, and $S_f = 1$ means *sent*.

The optimization problem to minimize the sum of total distortion can be stated as follows. Given the estimated packet loss rate π_B , round trip time RTT , one-way packet delay d and transmission deadline T , the objective of the proposed transmission framework is to achieve the minimum video distortion by adapting the per-packet retransmission timeout RTO_i , maximum number of retransmissions N_i , and delivery status S_f for each

frame. Formally,

$$OP1: \left\{ \{RTO_i, N_i\}_{1 \leq i \leq n_f}, S_f \right\}_{f=1}^F = \arg \min \left(\sum_{f=1}^F D_f \right).$$

$$\text{subject to: } \begin{cases} \sum_{f=1}^F d_f + (F-1) \cdot \theta \leq F/M, & (7a) \\ \frac{\sum_{f=1}^F (n_f + N_f) \cdot S}{\sum_{f=1}^F d_f + (F-1) \cdot \theta} \leq R_{TCP}. & (7b) \end{cases}$$

where:

$$D_f = P_f + \sum_{f' \in \Theta_f} (P_{f'} \cdot I_{f'}),$$

$$P_f = 1 - R_f + R_f \cdot E_f, R_f = 1 - \frac{\mathbb{E}[L(n_f + N)]}{n_f},$$

$$d_f = d + (n_f - 1) \cdot \theta + \sum_i^{n_f} N_i \cdot \{d + \min\{RTT_i + d, RTO_i\}\},$$

$$E_f = \exp \left\{ -\frac{T}{d + (n_f - 1) \cdot \theta + \sum_i^{n_f} N_i \cdot \{d + \min\{RTT_i + d, RTO_i\}\}} \right\},$$

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

The transmission deadline for each GoP is stated in (7a). This delay constraint is equal to the playout duration to prevent playback buffer starvation. For instance, F/M is 500 ms for a GoP of 15 frames at the encoding frame rate 30 fps. Condition (7b) specifies the upper limit of instantaneous transmission rate as ruled in TCP-friendly rate control (TFRC) [32]. The optimization problem (OP1) is NP-hard since it is infeasible to consider all the possible combinations of N_i , RTO_i and S_f . Therefore, we develop heuristic scheduling algorithms for the reliability adaptation and buffer control to achieve sub-optimal performance with polynomial time complexity.

4 SCHEDULING ALGORITHMS

This section presents the scheduling algorithms of the partial reliability based real-time streaming framework. As analyzed in Section 3, the absolute reliability is undesirable for real-time video transmission. To implement partially reliable transfer, the solution procedure in this paper includes the reliability adaptation and buffer control. In particular, the reliability adaptation is to calculate the per-packet retransmission timeout RTO_i and the maximum number of retransmissions N_i . The buffer control dynamically determines the buffer usage for video frames to be transmitted.

4.1 Video-Aware Reliability Adaptation

The objective of the reliability adaptation is to adjust the timeout value and the maximum number of retransmissions to minimize the overall video distortion. First, we

provide analysis on the maximum retransmission number from the perspective of delay and rate constraints. To respect the transmission deadline in (7a), the maximum retransmission number for a video GoP can be estimated as

$$N_{\max}^{\dagger} = \frac{T - F \cdot d - F \cdot (n_f - 1) \cdot \theta}{d + \min\{RTO, RTT + d\}}.$$

The number of packet retransmissions is dynamically allocated to the video frames in a GoP to minimize the total distortion, *i.e.*,

$$\{N_i\}_{1 \leq i \leq F} = \arg \min \left(\sum_{f=1}^F D_f \right).$$

Furthermore, the maximum number of retransmissions per packet is constrained by the TFRC expressed as follows.

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

With regard to the constraint in (7b), the maximum retransmission number is estimated as

$$\frac{(N_{\max} + \sum_{f=1}^F n_f) \cdot S}{\sum_{f=1}^F d_f + (n - 1) \cdot \theta} = R_{\text{TCP}}.$$

Therefore, the value of $N_{\max}$ is estimated as

$$N_{\max}^{\dagger} = \frac{R_{\text{TCP}} \cdot [\sum_{f=1}^F d_f + (n - 1) \cdot \theta] - \sum_{f=1}^F n_f \cdot S}{S}.$$

Finally, the value of $N_{\max}$ is chosen from the delay and rate constraints, *i.e.*,

$$N_{\max} = \min \{N_{\max}^{\dagger}, N_{\max}^{\ddagger}\}.$$

The reliability adaptation is to appropriately allocate the $N_{\max}$ retransmissions for a total of $\sum_{f \in F} n_f$ packets in a video GoP to minimize the total distortion. However, it is computationally infeasible to solve the problem with exhaustive search algorithms since there are $(\sum_{f=1}^F n_f)^{N_{\max}}$ possible combinations of retransmission number allocations. For instance, if there are 30 video packets in a GoP and the maximum retransmission number is 20, there will be $30^{20} \approx 3.49 \times 10^{29}$ possible allocations of the maximum number of retransmissions.

We develop a polynomial-time search algorithm with sub-optimal performance. In each loop, the algorithm tries to allocate a retransmission to the F possible video frames and the corresponding total distortion $\sum_{f=1}^F D_f$ is estimated with Equation (6). If the number of retransmissions for a specific frame becomes zero, the packets belonged to this frame will be removed from the sender buffer (*i.e.*, the retransmission is not performed for this video frame).

The retransmission timeout adaptation is also a challenging and interesting problem. If the timeout interval is too short, there may be substantial amount of unnecessary retransmissions before receiving the ACKs. On the

other hand, the end-to-end delay becomes large due to the delayed retransmissions. In this paper, the RTO_i for each packet is estimated to minimize the expired loss ratio of a single video frame, *i.e.*,

$$RTO_i = \arg \min_{RTT \leq RTO_i \leq F/M} \left\{ \exp \left\{ -\frac{T}{d_f(RTO_i)} \right\} \right\}.$$

As illustrated in Figure 4a, there is a delay value d to achieve optimal quality in terms of PSNR. Thus, the optimal RTO value entails the minimal frame-level data loss ratio. In our algorithm, the RTO is adapted at the interval of $\Delta RTO = \Delta RTT + \overline{RTT}/8$, in which ΔRTT represents the standard deviation of the latest ten RTT values. Furthermore, the value of RTO does not exceed the delay constraint F/M for each GoP. Algorithm 1 describes the reliability adaptation process. The time complexity of the algorithm is given in Proposition 3.

Algorithm 1: Reliability Adaptation

Input: RTT, π_B, S, F, n_f ;
Output: $\{RTO_i, N_i\}_{1 \leq i \leq n_f}$;

- 1 Maximum retransmission estimation:
- 2 $N_{\max}^{\dagger} = \frac{T - F \cdot d - F \cdot (n_f - 1) \cdot \theta}{d + \min\{RTO, RTT\}};$
- 3 $N_{\max}^{\ddagger} = \frac{R_{\text{TCP}} \cdot [\sum_{f \in F} d_f + (n - 1) \cdot \theta] - \sum_{f=1}^F n_f \cdot S}{S};$
- 4 $N_{\max} = \min \{N_{\max}^{\dagger}, N_{\max}^{\ddagger}\};$
- 5 $\bar{D} = \infty, \{N_f\}_{1 \leq f \leq F} = 0;$
- 6 **for** $i = 1$ **to** $N_{\max}$ **do**
- 7 Per-frame retransmission estimation:
- 8 **for** $f = 1$ **to** F **do**
- 9 $N_f = N_f + 1;$
- 10 $d_f = d + (n_f - 1) \cdot \theta + N_f \cdot \{d + \min \{RTT_i + d, RTO_i\}\};$
- 11 $R_f = 1 - \frac{\mathbb{E}[L(n_f + N)]}{n_f}, E_f = \exp \left\{ -\frac{T}{d_f} \right\};$
- 12 Call procedure timeout estimation;
- 13 $P_f = 1 - R_f + R_f \cdot E_f;$
- 14 $D_f = 1 - P_f + \sum_{f' \in \Theta_f} (1 - P_{f'}) \cdot I_{f'};$
- 15 **if** $\bar{D} > \sum_{f=1}^F (D_f)$ **then**
- 16 $N_f = N_f - 1;$
- 17 **end**
- 18 **else**
- 19 $N_{\max} = N_{\max} - 1, \bar{D} = \sum_{f=1}^F (D_f);$
- 20 **end**
- 21 **end**
- 22 **end**
- 23 Retransmission timeout estimation:
- 24 **for** $i = 1$ **to** n_f **do**
- 25 $RTO_i = RTT, \Delta RTO = \Delta RTT + \overline{RTT}/8;$
- 26 $\bar{P} = \exp \left\{ -\frac{T}{d_f(RTO_i)} \right\};$
- 27 **while** $RTO_i < F/M$ **do**
- 28 $RTO_i = RTO_i + \Delta RTO;$
- 29 $d_f =$
 $(n_f - 1) \cdot \theta + \sum_i n_i \cdot \{d + \min \{RTT_i + d, RTO_i\}\};$
- 30 **if** $\exp \left\{ -\frac{T}{d_f(RTO_i)} \right\} > \bar{P}$ **then**
- 31 **break**;
- 32 **end**
- 33 $\bar{P} = \exp \left\{ -\frac{T}{d_f(RTO_i)} \right\};$
- 34 **end**
- 35 **end**

Proposition 3. *The worst-case time complexity of Algorithm 1 is $O\left(N_{\max} \cdot F \cdot n_f \cdot \frac{F/M - RTT}{\Delta RTO}\right)$, in which $N_{\max}$ denotes the number of maximum retransmissions [Equation (9)], F represents the video GoP size and n_f stands for the number of video packets.*

Proof: The time complexity of the main loop for retransmission allocation is $O(N_{\max})$. In each cycle, the execution time of the per-frame retransmission estimation is F (GoP size), and $n_f \cdot \frac{F/M - RTT}{\Delta RTO}$ for the timeout estimation. Therefore, the total complexity of Algorithm 1 is $O\left(N_{\max} \cdot F \cdot n_f \cdot \frac{F/M - RTT}{\Delta RTO}\right)$. $\square$

4.2 Network-Adaptive Buffer Control

Due to the dynamics in communication networks (*e.g.*, bandwidth fluctuations, packet losses, network outage, *etc.*), the sender and receiver are often plagued with buffer overflow and starvation. These abnormal buffer behaviors induce noticeable video quality degradations (*e.g.*, video stall, freeze, rebuffering, *etc.*). To regulate buffer level and improve video fluency, this research presents a network-adaptive buffer control framework. In particular, the network adaptiveness includes the following aspects: 1) to proactively substitute the sender buffer in the presence of network congestion and bandwidth shrink; 2) to automatically slide the receiver window (*i.e.*, lower and upper bounds of receiver buffer) to cope with packet losses and delay jitters. In what follows, we present the buffer control algorithms at the sender and the receiver.

4.2.1 Sender Buffer Control

The main procedure for the sender buffer control can be summarized as: 1) normal packet delivery and buffer update; 2) proactive buffer substitution for high-priority frames. As shown in Figure 1, the information process unit in the sender side handles the ACK & NAK packets to update the sender buffer status (*e.g.*, slide the window, perform retransmission, *etc.*). In the normal packet sending process, the delivery status S_f of the f -th video frame is 1 (sent) if the buffer usage and rate control constraints are satisfied. Formally,

$$S_f = 1, \text{ if } n_f \leq n_{\max} - n_{\text{buffered}} \text{ \&\& } R < R_{\text{TCP}},$$

where R denotes the estimated transmission rate in each rate control period (*e.g.*, 0.5 seconds). After determining the delivery status, the video frame is either dropped or sent to the receiver. If this frame is to be sent, the per-packet retransmission timeout and maximum retransmissions are estimated with Algorithm 1. However, the sender buffer overflows (*i.e.*, $n_{\text{buffered}} = n_{\max}$) frequently should bandwidth fluctuations, packet losses or network congestions occur. To implement the partially reliable transfer and improve video fluency, we present a sender buffer control solution to proactively substitute occupied sender buffers with high-priority video packets

in Algorithm 2. The handlers for ACK and NAK packets in the sender side are described as follows.

ACK Packets Handler: If the received packet is an ACK and the sequence number (ack_num) is within the limit, the buffer controller automatically changes the upper bound to this value and updates the number of buffered packets. The sender buffers for these acknowledged packets are released to enable the normal packet delivery process.

NAK Packets Handler: If a NAK packet is received, the sender immediately retransmits the packet if this number (nak_num) falls within the sender window and the frame-level retransmission number N_f is larger than zero. If a video packet is retransmitted due to NAK, the timer for this packet is immediately reset and the RTO is re-calculated. Algorithm 2 summarizes the main control process of the sender buffer.

Timeout Retransmission: The timeout retransmission is performed with the per-packet timer and retransmission timeout (*RTO*) value in the system program. Different from the conventional retransmission schemes (*e.g.*, in TCP), the *RTO* value in PERES is configured for proactive retransmission. In the case of long-distance transfer, the delay for receiving NAK packet is intolerable for real-time applications. Thus, an appropriate *RTO* value is desirable to enable retransmission before the NAK detection.

4.2.2 Receiver Buffer Control

In the receiver side, the general process for buffer control can be summarized as follows.

- ACK & NAK feedback according to the received packet sequence number and buffered packets.
- Deadline-constrained sliding window to periodically clear the receiver buffer and provide the arrived video frames to the decoder.

ACK & NAK Feedback: After receiving a video packet, the first step is to identify whether the sequence number is the lower bound of the receiver buffer. If this packet sequence number is the lower bound, an ACK packet will be delivered to notify the sender side. Otherwise, the NAK packet(s) will be sent back to require the retransmission of the absent packet(s). In order to mitigate the problem of delayed/lost ACKs and facilitate the sender buffer release, the feedback NAK packets also carry the latest lower bound of receiver buffer for acknowledgement.

Deadline-Constrained Sliding Window: In the case of bandwidth shrink or network outage, the receiver buffer may be blocked because of dropped and overdue video packets. To improve the smoothness of real-time streaming, we develop a deadline-constrained sliding window algorithm to adaptively slide the receiver window. Specifically, the timer for each GoP is started once the first packet of the I frame is received and stored in the sender buffer. If the timer equals to the GoP duration and there are still remaining packets in the sender

Algorithm 2: Sender Buffer Control

Input: L_f, RTT, π_B ;
Output: $S_f, SEQ_{low}, SEQ_{upr}$;

1 Normal packet delivery process:
 $R_{TCP} = \frac{MSS}{RTT \cdot \sqrt{\frac{2 \times \pi_B}{3}} + 12 \times \sqrt{\frac{3 \times \pi_B}{8}} \cdot \pi_B \cdot [1 + 32 \times (\pi_B)^2]}$;

2 $n_f = \lceil L_f / S \rceil$, $R = \frac{n_{trans} \cdot S}{Interval}$, $n_{trans} = 0$;

3 **if** $n_f \leq n_{max} - n_{buffered}$ && $R < R_{TCP}$ **then**

4 $S_f = 1$, $\{N_i, RTO_i\}_{1 \leq i \leq n_f} = \text{Algorithm 1}$;

5 Send the n_f video packets to the receiver;

6 $SEQ_{upr} + = n_f$, $n_{buffered} + = n_f$;

7 $n_{trans} + = n_f$;

8 **end**

9 **else if** $(n_f > n_{max} - n_{buffered})$ && $fram\ type == \text{Intra}$ **then**

10 $S_f = 1$, $\{N_i, RTO_i\}_{1 \leq i \leq n_f} = \text{Algorithm 1}$;

11 $j = n_f$;

12 **while** $j \geq 0$ **do**

13 Remove the packet in the $(SEQ_{low} + j)$ -th buffer;

14 Send the $n_f - j + 1$ -th packet of the frame f ;

15 Reset the timer and retransmission with $\{N_{n_f - j + 1}, RTO_{n_f - j + 1}\}$;

16 $j - -$;

17 **end**

18 $n_{trans} + = n_f$;

19 **end**

20 ACK packets handler:

21 **while** $between(SEQ_{low}, ack_num, SEQ_{upr})$ **do**

22 Cancel the timer of lower bound;

23 Release the lower bound of sender buffer;

24 $n_{buffered} - -$, $SEQ_{low} + +$;

25 **end**

26 NAK packets handler:

27 **if** $between(SEQ_{low}, nak_num, SEQ_{upr})$ && $N_f > 0$ **then**

28 Retransmit the packet with nak_num ;

29 $N_f - -$, $n_{trans} + = n_f$;

30 **if** $N_f == 0$ **then**

31 Remove all the video packets of the f -th frame;

32 **end**

33 **end**

buffer, these packets will be provided to the decoder with the slice-level error concealment from previously-received frames. The loss recovery time using this error concealment is approximately 1 – 2 GoPs.

Algorithm 3 outlines the pseudo code for the receiver buffer control in PERES.

5 IMPLEMENTATION AND EVALUATION

This section describes the system implementation and evaluation results of the proposed PERES scheme. In particular, the experiments are conducted with an embedded video monitoring system in different wireless network environments. The first subsection presents the implementation details of the real-time video system. Then, we present and discuss the experimental results. The modular validation is presented in the end of this section.

5.1 System Implementation

The proposed PERES scheme is implemented in a embedded video monitoring system as shown in Figure

Algorithm 3: Receiver Buffer Control

Input: seq - received packet sequence number, SEQ_{low}, SEQ_{upr} ;

Output: $SEQ_{low}, SEQ_{upr}, n_{buffered}$;

1 ACK feedback:

2 **while** $between(SEQ_{low}, seq, SEQ_{upr})$ **do**

3 Send ACK packet with sequence SEQ_{low} ;

4 $n_{buffered} = n_{buffered} - 1$;

5 $SEQ_{low} + +$, $SEQ_{upr} + +$;

6 **end**

7 NAK feedback:

8 **if** $between(SEQ_{low}, seq, SEQ_{upr})$ **then**

9 **if** $seq \neq SEQ_{low}$ **then**

10 $i = SEQ_{low}$;

11 **while** $between(SEQ_{low}, i, seq)$ **do**

12 **if** the i -th packet is not received **then**

13 Send NAK packet with sequence i ;

14 **end**

15 $i + +$;

16 **end**

17 **end**

18 **end**

Fig. 5. Image sensor and evaluation board of the embedded video monitoring system for performance evaluation.

5. In particular, the evaluation board supports both CBR (Constant Bit Rate) and VBR (Variable Bit Rate) video encoding. The initialized (default) sensor effects include 1280×720 video size and 30 fps. These effect parameters are immediately updated after obtaining the video coding parameters from the mobile App side (the video coding parameters can be configured in the user interface of the App). We develop the firmware based on the SDK (Software Development Kit) and API (Application Program Interface) of the evaluation board. The binary (.bin) file is generated with the standard GCC and a cross-compiler in 32-bit Linux. The reliability adaptation and buffer control algorithms of PERES are included in the video communication module of the firmware.

5.1.1 Experimental Setup

Video codec: The evaluation board shown in Figure 5 includes H.264 hardware codec that supports video compression up to 720p and 30 fps. In particular, the deblocking filter and macroblock skipping are the special features of the hardware codec. The video encoding bit rates are 1, 1.5, 2, and 3 Mbps. In order to make better comparison of the image quality, the *Big Buck Bunny* and

Sintel sequences in 720p format are used as the video test sequences (*i.e.*, the captured videos for the image sensor). The IDR period is configured as 1 to guarantee the I frames of different GoPs are completely independent of each other, *i.e.*, there is no decoding reference among these different I frames. Table 2 summarizes the video coding parameters configured in the experiments.

TABLE 2
Video encoding parameters

Value	Parameter
1, 1.5, 2, 3 Mbps	Encoding bit rate
8, 10, 15 frames	GoP size
15, 20, 30 fps	Frame rate
IPPPPP...P	GoP structure
1280 × 720	Resolution
40, 20	max, min quantization parameter
1	IDR period

Network environment: The network topology for performance evaluation is presented in Figure 6. The devices and smartphones are placed in the locations of Singapore, Hong Kong and Beijing. In addition to the P2P transfer mode, we also place a TURN (Traversal Using Relays around NAT) [38] server in Shanghai for data relay. With this relay control protocol, it is possible to enable data communications between end hosts behind NAT (Network Address Translator). P2P and TURN represent the two most commonly-used transfer modes in video communication. In most cases, P2P transfer mode generally implies shorter physical distance and path propagation delay. Thus, the rebuffering ratio and transmission delay in P2P mode are often lower than the results measured in TURN mode. The many-to-one and one-to-many communication modes are not considered in the current paper. This is because these communication scenarios are more complex than the one-to-one (unicast) scenario considered in this paper. For instance, we need to consider the different network conditions of receivers and multimedia synchronization in the one-to-many scenarios. In our future work, we will consider improving the modules of PERES to accommodate many-to-one and one-to-many communication modes.

The devices access the Internet with Wi-Fi connections while the mobiles may use either Wi-Fi or LTE. As shown in the network topology, the contention smartphones and devices are placed nearby to introduce wireless interference and losses. For instance, the contention mobiles are receiving FTP (File Transfer Protocol), VOIP (Voice over IP), CBR (Constant Bit-Rate) and VBR (Variable Bit-Rate) data services during the video transmission.

5.1.2 Reference Schemes

- TCP. TCP New Reno [20] has the ability to recover from multiple losses within the same loss window, and thus avoids frequent retransmission timeout

events. This TCP version has been widely adopted in current Internet for data communication.

- FEC. The LDPC (Low-Density Parity Check) 2 – 8 code from OpenFEC [19] is used as a reference. In the FEC coding, the redundancy is fixed at 25% ($k = 4, n = 5$) and the symbol size is 512 Bytes.
- ARQ. This ARQ scheme refers to the reliable transfer based on NAK and ACK mechanism without including the reliability adaptation and buffer control algorithms.

The above reference schemes are selected and implemented because of the following reasons.

- **Representativeness:** TCP, FEC and HARQ represent three different types of error-control schemes used for video communication. In particular, TCP adopts packet retransmission for data protection. FEC uses redundant data packets to combat against network packet losses. And HARQ leverages both packet retransmission and error correction code for loss recovery.
- **Implementation:** The reference schemes (TCP, FEC, HARQ) are commonly used in real multimedia communication systems. And there are many open-source programs to implement these reference schemes. These reference schemes are also widely used in research as baseline approaches due to their easy implementation.

Thus, the comparative evaluation results against TCP, FEC, and HARQ will be representative and provide understanding regarding the performance improvements achieved by PERES over many real systems.

5.1.3 Performance Metrics

- PSNR. PSNR (Peak Signal-to-Noise Ratio) is the standard objective metric to measure video quality. This parameter is expressed as a function of the mean squared error between the original and the reconstructed video frames. In particular, the EvalVid toolset [39] is used for the PSNR measurement.
- PVQM. According to reference [40], the real-time streaming quality is also measured with the perceptual video quality metrics (PVQM) of MOS (Mean Opinion Score), SSIM (Structural Similarity), VSNR (Visual Signal-to-Noise Ratio) and VIF (Visual Information Fidelity).
- Delay. The end-to-end delay of a video frame is counted from the generation to the decoding based on the time stamps of sent/received packets. The end-to-end packet delay values are measured in the experiments to reflect the delay performance of the competing schemes. The deadline violation ratio of video frames is also presented to reveal the video streaming fluency level.
- Bandwidth consumption. The bandwidth consumption is measured at UDP/TCP packet level for the video application on smartphones. This traffic estimation is based on the TrafficStats class in Android SDK to calculate the received data of a single thread.

Fig. 6. Network topology for performance evaluation.

5.2 Experimental Results

Video Quality

Fig. 7. Average PSNR values in different video encoding rates and data transfer modes.

The average PSNR values in different encoding bit rates are presented in Figure 7a. As the video streaming rate increases, larger performance gaps can be observed between the proposed PERES and reference solutions. Constrained by network bandwidth and transmission deadline, higher video streaming rate indicates lower transfer reliability for all the video frames. In such cases, it is important to provide differentiated protection level for high-priority video frames to improve real-time streaming quality. In the case of lower bit-rate video streaming (1 and 1.5 Mbps), the NAK & ACK based ARQ scheme outperforms the FEC coding scheme. However, FEC achieves higher video quality than ARQ at higher streaming rates of 2 and 3 Mbps. This is because ARQ guarantees video streaming quality with a large amount of retransmissions (*i.e.*, significantly higher bandwidth consumption as illustrated in Figure 16a). In the case of high video streaming rates, the excessive retransmissions are intolerable in the capacity-limited communication networks.

Figure 7b shows the video PSNR measured in LAN, P2P and TURN transfer modes at the encoding rate of 2 Mbps. The PSNR values in the LAN transfer mode generally represent the lossless video quality since there is seldom packet loss or bandwidth fluctuations in the same local network experiments. The P2P and TURN transfer modes reflect minor and moderate impairments. As shown in the bar-charts, the received video quality is obviously lower after experiencing longer-distance transmissions in TURN server mode. Furthermore, the cloud

server status (*e.g.*, CPU load, input/output bandwidth, memory usage, *etc.*) also affects the streaming quality (especially in peak hours during weekdays).

Fig. 8. Average PSNR values for VBR video streaming and different access networks.

VBR video is featured by frequent bit-rate variations within a range (*e.g.*, 0.2 – 1 Mbps), and has already become an important multimedia application over the Internet [41]. In order to evaluate the performance for VBR videos, we conduct the experiments with the streaming rates of [1, 1.2], [1.5, 1.7] and [2.5, 2.8] Mbps and the evaluation results are presented in Figure 8a. PERES dynamically adjusts the reliability level according to different frame types and time-varying network status. Hence, the proposed scheme exhibits superiority over the competing solutions in coping with the traffic variations of VBR video streaming. FEC outperforms the ARQ at the video streaming rate of [2.5, 2.8] Mbps. Figure 8b compares the video PSNR measured in different access networks. LTE exhibits higher stability than Wi-Fi and a better-quality video streaming is received.

The PSNR per-frame values for the *Big Buck Bunny* and *Sintel* sequences are presented in Figure 9 to have a microscopic view of the evaluation results. The noticeable PSNR drops for the reference schemes are possibly caused by the data loss of important frames. A drop event of I frame incurs the consecutive decoding failures of subsequent P frames. The loss recovery time from the error propagation is often larger than two GoPs (*i.e.*, 1 second).

In order to compare the image quality observed on the mobile App, Figure 10 presents the typical video frames measured from the *Big Buck Bunny* and *Sintel* test sequences. In these experiments, the source video to the evaluation board is from the video sequence instead of from the image sensor. The received image quality of the

Fig. 10. Comparison of received image quality measured from the *Big Buck Bunny* and *Sintel* video sequences.

Fig. 9. Instantaneous PSNR values for video frames indexed from 500 to 1500 in both video test sequences.

proposed PERES is obviously higher than the reference solutions. For instance, the rabbit in the image of PERES can be clearly observed while there are visual artifacts in other images.

Fig. 11. Number and percentage of lost video frames in different experimental scenarios.

The number and ratio of dropped video frames measured from 1000 GoPs (GoP size = 10, IPP.P) at 20 fps are presented in Figure 11. The drop ratio and number refer to the video frames violating the transmission deadline. In particular, the I frame loss of PERES is apparently lower than those of the competing transfer schemes. In

unreliable communication networks, the I frames are more likely to experience network and expired losses because of large frame size. Therefore, these high-priority frames are protected with different reliability levels and buffer control solutions in our proposed algorithms.

TABLE 3
Comparison of video quality in different metrics.

Method	PERES	ARQ	FEC	TCP
MOS	4.61	3.83	3.71	3.56
SSIM	0.972	0.893	0.861	0.839
VSNR	0.957	0.886	0.858	0.821
VIF	0.983	0.891	0.852	0.827

Since the video PSNR has limitations in correlating with perceptual video quality [40], we also present the subjective video quality results including MOS (Mean Opinion Score), SSIM (Structural Similarity), VIF (Visual Information Fidelity) and VSNR (Visual Signal-to-Noise Ratio) in Table 3. These are commonly-used QoE (Quality-of-Experience) metrics to characterize user-perceived quality based on human visual system features. For instance, the SSIM [42] is a subjective quality assessment metric method on the degradation of structural information. This similarity measurement includes the comparison of luminance, contrast and structure. It can be observed that the proposed PERES exhibits superior performance advantages over the reference schemes in improving user-perceived quality. In particular, the average MOS value of PERES (4.61) is up to the excellent quality level (> 4.5) while the reference schemes are at the good level (> 3.6).

The instantaneous values of SSIM and VIF are presented in Figure 12 for comparison of per-frame subjective quality values. The stability of the proposed PERES scheme indicates the higher perceptual quality perceived by end users.

Delay Profile

The average end-to-end frame delay profile with regard to different video frame rates in P2P mode is presented

Fig. 12. Instantaneous SSIM and VIF values for video frames indexed from 500 to 1500.

Fig. 13. Average delay values in different encoding frame rates and I-frame delay.

in Figure 13a. The latency values of all the schemes become smaller as the frame rate increases. This delay reduction is due to the smaller length (in Bytes) of video frames. PERES outperforms the reference schemes in reducing the end-to-end frame latency because: 1) we provide partial reliability to flexibly configure the total number of retransmissions; 2) the retransmission timeout (RTO) is flexibly configured to perform effective retransmissions within the deadline. Due to the different reliability mechanisms/levels, there is significant difference in the average delay of I frames in TURN transfer mode as shown in Figure 13b. The FEC scheme achieves the lowest I frame delay since no retransmission is performed for this method. In HD video streaming, the end-to-end delay of I frames is a critical problem for user-perceived quality due to the large frame length (in Bytes). The delayed arrival of an I frame induces noticeable quality degradations on the subsequent P frames.

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

The cumulative distribution functions (CDF) of end-

to-end delay are plotted in Figure 14 to depict the microscopic delay performance. PERES delivers more than 80% of the packets within 45 ms. For the retransmission-based reference schemes (ARQ and TCP), the end-to-end packet delays are often larger than 70 ms due to the frequent retransmissions.

Fig. 15. Number of total and effective retransmissions in P2P and TURN transfer modes.

To further investigate the delay profile differences among the evaluated schemes, the numbers of total and effective retransmissions (in P2P and TURN transfer modes) are illustrated in Figure 15 for comparison. Specifically, the effective retransmission indicates the retransmitted packets are successfully received within the deadline T . The evaluation results include PERES, ARQ and TCP since the FEC scheme does not perform packet retransmission. PERES achieves effective retransmissions above 70% with obviously lower total numbers. The successful retransmission probability is achieved with the retransmission limitation and timeout configuration strategies in the proposed algorithms. As observed in the experiments, more than 90% of the consecutive retransmissions caused by spurious NAKs with the same sequence are superfluous and delay the normal packet sending process.

Bandwidth and Energy Consumption

Fig. 16. Average and instantaneous values of bandwidth consumption for 1-Mbps video streaming.

Figure 16a presents the bandwidth consumption with regard to different transfer modes for the 1-Mbps video streaming. Expectedly, the FEC schemes consumes a fixed bandwidth level at 125% of the video encoding rate. The bandwidth of PERES, ARQ and TCP are significantly different due to the diverse numbers of packet

retransmissions. In particular, ARQ consumes the highest bandwidth because of the frequent retransmissions triggered by timeout and NAK events. The substantial bandwidth reduction of PERES is expected since we deliberately alleviate unnecessary retransmissions with video-aware reliability adaptation. Since data communication constitutes the main portion of energy consumption in video streaming applications, the bandwidth improvements also indicate lower power consumption of mobile devices for the proposed PERES. The bandwidth conservation of PERES is mainly attributed to the mitigation of excessive packet retransmissions in the case of network losses.

Fig. 17. Average energy consumption of mobile device in different transfer modes.

The energy consumption of mobile device with regard to different transfer modes are presented in Figure 17. These results are measured with the Monsoon power monitor [47] in 200 seconds. PERES achieves the lowest energy in all the transfer modes and the results' pattern is similar to that in Figure 16a. This is because data communication with radio interfaces is energy-consuming and the value is proportional to the bandwidth consumption (*i.e.*, received data amount).

5.3 Modular Validation

Fig. 18. Percentage of frame-level allocated maximum and actual-performed retransmissions.

In Figure 18a, the packet retransmission allocation and actual retransmissions for different video frame indexes are presented to validate the efficacy of Algorithm 1. The number of allocated packet retransmission decreases as the index becomes larger (from 1 to 15). In the mathematical model of PERES, the total number of retransmissions $N_{\max}$ is limited by the network bandwidth and delay constraint. Therefore, more packet retransmissions are allocated to the video frames with higher priority. The percentage of ACK and NAK based retransmission in different transfer modes are profiled in Figure 18b. As the

transmission distance becomes longer, there is significant higher possibility of 'proactive' timeout retransmission in the PERES to reduce retransmission delay. The percentage indicates the importance of both ACK and NAK mechanisms in the proposed PERES scheme.

Fig. 19. Rebuffering ratio and buffer usage in different transfer modes.

The bar-charts in Figure 19a present the average rebuffering ratios ($\frac{\text{buffering time}}{\text{buffering and play time}}$ of the video player) and this metric is computed as the ratio of the time the video player spends buffering to the sum of buffering and play time. The evaluation results demonstrate the efficacy of the proposed buffer control algorithms in preventing playback buffer starvation. The performance of PERES and FEC are close to each other in both P2P and TURN transfer modes. For the reference schemes (*e.g.*, ARQ and TCP), the receiver side is frequently plagued with video stall events in TURN transfer mode. The buffer usage for I frames is presented in Figure 19b. PERES provides higher buffer utilization level than the reference ARQ scheme, especially in the TURN transfer mode. Such utilization differences are the consequences of proactive sender buffer substitution Algorithm 2.

Fig. 20. Execution time of the proposed Algorithm 1.

The execution time of the proposed Algorithm 1 is presented in Figure 20. As the results show, the time cost for each GoP is generally less than 7 ms and this latency is negligible compared to the end-to-end transmission delay. This execution time overhead is low for real-time video content delivery. Another observation from the results is that the execution time increases with bit rate becomes higher since more packets need to be processed.

6 CONCLUSION AND DISCUSSION

The technological evolutions of mobile Internet prompt the surprising proliferation of HD real-time video applications on smartphones. Due to the network dynamics

and stringent deadline, it still remains problematic to effectively stream HD real-time videos to mobile devices. The existing transmission schemes (e.g., ARQ and FEC) suffer from the performance deficiencies either in delay or reliability. This paper proposes a partially reliable transfer scheme dubbed PERES to balance the critical tradeoff between delay and reliability. Through system modeling and theoretical analysis, we develop scheduling algorithms for reliability adaptation and buffer control. The proposed PERES is implemented in an embedded video monitoring system and evaluated under different wireless environments. Evaluation results demonstrate the performance advantages over the reference solutions.

As future work, we will consider the following important directions:

- To include the energy consumption model to improve the power efficiency of real-time video streaming on mobile devices.
- To optimize the perceptual quality by taking advantage of video QoE (Quality of Experience) metrics.
- To cope with the dynamic of participating nodes in P2P transfer mode.

ACKNOWLEDGEMENT

This work is supported in part by an NTU Start-up Grant. The authors would like to express their sincere gratitude to the anonymous reviewers and Associate Editor for the valuable comments.

REFERENCES

- [1] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H. 264/AVC video coding standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 560-576, 2003.
- [2] H. Schwarz, D. Marpe, and T. Wiegand, "Overview of the scalable video coding extension of the H. 264/AVC standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 9, pp. 1103-1120, 2007.
- [3] 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, 2012.
- [4] Cisco company, "Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2015-2020," Feb. 2016.
- [5] Google WebRTC, <https://webrtc.org/>
- [6] J. He, D. Wu, X. Xie, M. Chen, et al., "Efficient Upstream Bandwidth Multiplexing for Cloud Video Recording Services," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 26, no. 12, pp. 1893-1906, 2016.
- [7] Y. Li, Y. Zhang, L. Qiu, and S. Lam, "Smarttunnel: Achieving reliability in the Internet," in *Proc. of IEEE INFOCOM*, 2007.
- [8] 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, 2012.
- [9] X. Zhang, Y. Xu, H. Hu, Y. Liu, Z. Guo, and Y. Wang, "Profiling skype video calls: Rate control and video quality," in *Proc. of IEEE INFOCOM*, 2012.
- [10] W. Hu, G. Cao, "Energy-Aware Video Streaming on Smartphones," in *Proc. of IEEE INFOCOM*, 2015.
- [11] M. Asefi, J. W. Mark, and X. S. Shen, "A mobility-aware and quality-driven retransmission limit adaptation scheme for video streaming over VANETs," *IEEE Trans. Wireless Commun.*, vol. 11, no. 5, pp. 1817-1827, 2012.
- [12] H. Bobarshad, M. V. D. Schaar, A. H. Aghvami, R. S. Dilmaghani, and M. R. Shikh-Bahaei, "Analytical modeling for delay-sensitive video over WLAN," *IEEE Trans. Multimedia*, vol. 14, no. 2, pp. 401-414, 2012.
- [13] M. V. der Schaar, D. S. Turaga, "Cross-layer packetization and retransmission strategies for delaysensitive wireless multimedia transmission," *IEEE Trans. Multimedia*, vol. 9, no. 1, pp. 185-197, 2007.
- [14] V. Sgardonis and A. R. Nix, "Raptor code-aware link adaptation for spectrally efficient unicast video streaming over mobile broadband networks," *IEEE Trans. Mobile Comput.*, vol. 14, no. 2, pp. 401-415, 2015.
- [15] Y. Wu, S. Kumar, F. Hu, et al., "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, 2014.
- [16] J. Xiao, T. Tillo, C. Lin, and Y. Zhao, "Dynamic Sub-GOP Forward Error Correction Code for Real-Time Video Applications," *IEEE Trans. Multimedia*, pp. 1298-1308, vol. 14, no. 4, 2012.
- [17] O. C. Kwon, Y. Go, Y. Park, and H. Song, "MPMTP: Multipath multimedia transport protocol using systematic Raptor codes over wireless networks," *IEEE Trans. Mobile Comput.*, vol. 14, no. 9, pp. 1903-1916, 2015.
- [18] J. Wu, B. Cheng, M. Wang, J. Chen, "Priority-Aware FEC Coding for High-Definition Mobile Video Delivery Using TCP," to appear in *IEEE Trans. Mobile Comput.*, DOI: 10.1109/TMC.2016.2584049.
- [19] OpenFEC, <http://openfec.org/>
- [20] S. Floyd, A. Gurtov, and T. Henderson, "The NewReno modification to TCP's fast recovery algorithm," *IETF RFC 6582*.
- [21] R. Stewart, M. Ramalho, Q. Xie, M. Tuexen, and P. Conrad, "Stream control transmission protocol (SCTP) partial reliability extension," *IETF RFC 3758*.
- [22] M. K. Mukerjee, D. Naylor, et al. "Practical, real-time centralized control for cdn-based live video delivery," in *Proc. of ACM SIGCOMM*, 2015.
- [23] E. Baik, A. Pande, P. Mohapatra, "Efficient MAC for Real-Time Video Streaming over Wireless LAN," *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 11, no. 4, article 50, 2015.
- [24] J. Wu, C. Yuen, N.-M. Cheung, 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, 2016.
- [25] J. Wu, C. Yuen, M. Wang, J. Chen, "Content-Aware Concurrent Multipath Transfer for High-Definition Video Streaming over Heterogeneous Wireless Networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 27, no. 3, pp. 710-723, 2016.
- [26] J. Wu, B. Cheng, Y. Shang, J. Huang, J. Chen, "A Novel Scheduling Approach to Concurrent Multipath Transmission of High Definition Video in Overlay Networks," *J. Netw. Comput. Appl.*, vol. 44, pp. 17-29, 2014.
- [27] Y. Huo, C. Hellig, Wiegand T, et al., "A Tutorial and Review on Inter-Layer FEC Coded Layered Video Streaming," *IEEE Commun. Surveys Tur.*, vol. 17, no. 2, pp. 1166 - 1207, 2015.
- [28] S. Ahmad, R. Hamzaoui, 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, 2010.
- [29] P. U. Tournoux, E. Lochin, et al., "On-the-fly erasure coding for real-time video applications," *IEEE Trans. Multimedia*, vol. 13, no. 4, pp. 797-812, 2011.
- [30] Y. Huang, S. Mehrotra, and J. Li, "A hybrid FEC-ARQ protocol for low-delay lossless sequential data streaming," in *Proc. of IEEE ICME*, 2009.
- [31] J. Wu, C. Yuen, 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, 2015.
- [32] S. Floyd, 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, 1999.
- [33] N. Freris, C. Hsu, J. Singh, and X. Zhu, "Distortion-Aware Scalable Video Streaming to Multinetwork Clients," *IEEE/ACM Trans. Netw.*, vol. 21, no. 2, pp. 469-481, 2013.
- [34] N. Freris, C. Hsu, et al., "Resource allocation for multihomed scalable video streaming to multiple clients," in *Proc. of IEEE ISM*, 2010.
- [35] D. Jurca and P. Frossard, "Video packet selection and scheduling for multipath streaming," *IEEE Trans. Multimedia*, vol. 9, no. 3, pp. 629-641, 2007.
- [36] E. Gilbert, "Capacity of a burst-noise channel," *Bell Syst. Tech. J.*, vol. 39, no. 9, pp. 1253-1265, 1960.
- [37] L. Zhou, X. Wang, W. Tu, et al., "Distributed scheduling scheme for video streaming over multi-channel multi-radio multi-hop wireless networks," *IEEE J. Sel. Area Commun.*, vol. 28, no. 3, pp. 409-419, 2010.
- [38] R. Mahy, P. Matthews, and J. Rosenberg, "Traversal Using Relays Around NAT (TURN): Relay extensions to session traversal utilities for NAT (S-TURN)," *IETF RFC 5766*.
- [39] The EvalVid toolset, <http://www2.tkn.tu-berlin.de/research/evalvid/fw.html>
- [40] W. Lin, C. J. Kuo, "Perceptual visual quality metrics: A survey," *J. Vis. Commun. Image R.*, vol. 22, no. 4, pp. 297-312, 2011.
- [41] F. M. Tabrizi, J. Peters, and M. Hefeeda, "Dynamic control of receiver buffers in mobile video streaming systems," *IEEE Trans. Mobile Comput.*, vol. 12, no. 5, pp. 995-1008, 2013.
- [42] Z. Wang, A. C. Bovik, H. R. Sheikh, E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600-612, 2004.
- [43] J. Xiao, T. Tillo, 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, 2013.
- [44] D. Lee, and H. Song, "A robust luby transform encoding pattern-aware symbol packetization algorithm for video streaming over wireless network," *IEEE Trans. Multimedia*, vol. 13, no. 4, pp. 788-796, 2011.
- [45] M. A. Hoque, et al., "Using crowd-sourced viewing statistics to save energy in wireless video streaming," in *Proc. of ACM MobiCom*, 2013.
- [46] C. Zhu, Y. Huo, B. Zhang, R. Zhang, M. El-Hajjar, and L. Hanzo, "Adaptive-Truncated-HARQ-Aided Layered Video Streaming Relying on Interlayer FEC Coding," *IEEE Trans. Veh. Technol.*, vol. 65, no. 3, pp. 1506-1521, 2016.
- [47] Monsoon monitor, <https://www.monsoon.com>.

Jiyan Wu received the Ph.D. degree in computer science and technology from the Beijing University of Posts and Telecommunications, Beijing, China, in June 2014, the Master's degree from China University of Mining and Technology (Beijing) in June 2011, and the Bachelor's degree from North China Institute of Science and Technology in June 2008.

He was a Software Developer (C++) with Sinosoft Technologies Company Ltd., Beijing, from Oct. 2010 to Mar. 2014, a Post-Doctoral Research Fellow with the Singapore University of Technology and Design, Singapore, from Mar. 2014 to Jan. 2016, and a Senior Software Engineer with OmniVision Technologies Singapore from Jan. 2016 to Sep. 2017. Currently, he is working on applying machine/deep learning algorithms in industrial vision inspection system and real-time sensory data analysis.

His research interests include video communication, video coding, forward error correction, heterogeneous wireless networks, concurrent multipath transfer and energy optimization. His programming experiences/interests are in the areas of client-server video communication, embedded multimedia systems, peer-to-peer data transfer, and back-end server programs.

Rui Tan is an Assistant Professor at School of Computer Science and Engineering, Nanyang Technological University, Singapore. Previously, he was a Research Scientist (2012-2015) and a Senior Research Scientist (2015) at Advanced Digital Sciences Center, a Singapore-based research center of University of Illinois at Urbana-Champaign (UIUC), a Principle Research Affiliate (2012-2015) at Coordinated Science Lab of UIUC, and a postdoctoral Research Associate (2010-2012) at Michigan State University. He

received the Ph.D. (2010) degree in computer science from City University of Hong Kong, the B.S. (2004) and M.S. (2007) degrees from Shanghai Jiao Tong University. His research interests include cyber-physical systems, sensor networks, and pervasive computing systems. He received the Best Paper Awards from IPSN'17, CPSR-SG'17, Best Paper Runner-Ups from IEEE PerCom'13 and IPSN'14.

Ming Wang is currently pursuing the PhD degree in computer science and technology in the State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications (BUPT). He received the Master's degree from University of Science and Technology Beijing (USTB) in Jan. 2011, and the Bachelor's degree from the North China Institute of Science and Technology in Jun. 2008. He was a Software Engineer with the China Academy of Space Technology from 2011 to 2014 and

a visiting PhD candidate in Singapore University of Technology and Design from Mar. 2015 to Oct. 2016. His research interests include multimedia networking and image processing.