Stochastic Analysis of DASH-based Video Service in High-Speed Railway Networks

Zhongbai Jiang, Changqiao Xu, Jianfeng Guan, Yang Liu, and Gabriel-Miro Muntean

***Abstract*—The latest increasing popularity of High-Speed Railways (HSR) has stimulated growing demands for wireless Internet services in HSR networks, especially for video streaming. However, due to the high variability and unpredictability of wireless communications in HSR networks, it is still difficult for the existing solutions to provide high-quality video streaming services to HSR passengers. This paper addresses this crucial problem first by reporting on field experiments performed to investigate the characteristics of HSR networks. Then the paper formulates an intractable optimization problem for Dynamic Adaptive Streaming over HTTP (DASH)-enabling service in HSR networks considering various factors, including packet loss, energy consumption, video service quality, etc. By leveraging Lyapunov optimization approaches, the formulated optimization problem is transformed into a queue stability problem which is of high scalability and generality. Moreover, in order to overcome the intractability of the initial optimization problem, the queue stability problem is further decomposed into three subproblems which can be easily solved individually. Finally, a novel Joint Stochastic DASH Optimization (JSDO) mechanism consisting of three algorithms for the derived subproblems is proposed. Rigorous theoretical analyses and realistic dataset-based simulations demonstrate the effectiveness of the proposed JSDO mechanism.**

***Index Terms*—Video Streaming, DASH, Stochastic Analysis, Service Optimization, HSR Networks.**

I. Introduction

THE High-Speed Railway (HSR) system is an advanced intelligent transportation system with many advantages which include high velocity, flexible scheduling, great punctuality and mass transportation of people. Benefiting from these aspects, HSR is becoming increasingly popular and is attracting huge number of passengers. For instance, it is reported that the Chinese HSR has carried more than five billion passengers from its launch until 2016 [1]. With the popularity increase of HSR, the demands for high-quality video streaming services over HSR data network will also increase rapidly, considering video traffic will account for more than 82% of all Internet traffic by 2021 compared with 73% in 2016 [2]. However, our field experiments on data transfer on a real HSR line have shown that high packet loss is a serious problem. At first, the average value of Packet Loss Rate (PLR) reaches 75.9%. Moreover, the fluctuation of PLR is significant and aperiodic which seriously reduces the predictability of PLR. Such high and unpredictable PLR levels greatly impair the quality of video streaming services. Therefore, it can be concluded that it is highly challenging to provide high-quality video streaming services in HSR networks.

In order to help improve the quality of video streaming services, the Dynamic Adaptive Streaming over HTTP (DASH) technique was proposed and has attracted very much attention in recent years. When employing DASH, videos are divided into segments and each segment is encoded into multiple bitrate versions. Through dynamic adaptive selection of bitrate according to current communication quality levels, DASH enables provision of improved video streaming. Building on this advantage, many researches proposed solutions and implementations based on DASH [3]–[12] and all demonstrate that the video service quality is improved. However, among the limitations of current research, two major aspects attract attention in relation to adaptive video streaming service deliveries in HSR networks.

1) Currently, the commonly used bitrate selection algorithms in DASH rely primarily on the available transmission rate. However, based on our field experiments, we find out that serious packet loss problem is a salient feature of HSR networks, which is seldom considered in previous researches. Actually, the high PLR levels in HSR networks are a major factor in the serious reduction of the effective transmission rate. Additionally, the unpredictability of PLR also makes difficult to perform optimal control of bitrate selection. Therefore, the performance of most existing bitrate adaptation algorithms will be seriously affected in HSR networks.

2) Most current research efforts focus on optimizing adaptive video streaming services by considering a few factors only, such as network capacity and service quality. At the same time, due to a lack of scalable analysis and optimization approaches, existing solutions are difficult to be extended to incorporate more factors without affecting negatively their performance. Therefore, adapting the bitrate selection according to many factors in an integrated manner and providing a scalable optimization approach for future research are still challenges in HSR networks and have not been explicitly targeted.

Motivated by the aforementioned aspects, this paper focuses on the DASH-based video service optimization problem in HSR networks. A scalable and general service optimization solution is provided and a new Joint Stochastic DASH Optimization (JSDO) mechanism is introduced. The main contributions of this paper as follows.

This work is supported in part by the National Natural Science Foundation of China (NSFC) under Grant 61871048, Grant 61522103, Grant 61872253 and Grant 61602038, in part by EU Horizon 2020 Grant 688503.

Zhongbai Jiang, Changqiao Xu, Jianfeng Guan, and Yang Liu are with State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing, 100876, China. E-mail: {zbjiang, cqxu, jfguan, liu.yang}@bupt.edu.cn.

Gabriel-Miro Muntean is with the Performance Engineering Laboratory, School Electronic Engineering, Dublin City University, Dublin, Ireland. E-mail: gabriel.muntean@dcu.ie.

Corresponding author: Changqiao Xu.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2018.2881095, IEEE Transactions on Multimedia

1) In order to investigate the characteristics of HSR networks, we first develop a PLR measurement system to examine the PLR characteristics of HSR networks. Field experiments are conducted along a real HSR line and many PLR datasets are collected. The cause of packet loss is also analyzed.

2) A general system model for DASH-based video service in HSR networks is introduced which consists of a transmission capacity model, a DASH service model, and an energy model. Based on this system model, a DASH service optimization problem is formulated, considering various factors which include packet loss, energy consumption, video service quality and service delivery.

3) By utilizing the Lyapunov optimization approaches, the formulated optimization problem is transformed into a queue stability problem which is of high scalability and generality. This problem is further decomposed into three independent subproblems which can be handled easily. Through incorporating a PLR prohibiting method, the JSDO mechanism is introduced to separately and efficiently solve the three subproblems.

4) The performance of JSDO is validated through rigorous theoretical analysis and extensive simulations using the realistic PLR datasets collected from field experiments. The theoretical analysis demonstrates how the quality of video streaming service is close to the optimal value. The simulation results illustrate how JSDO is able to improve the transmission energy efficiency and video streaming service quality in comparison with other existing mechanisms.

The rest of this paper is organized as follows. Section II briefly discusses related work. Section III introduces the field experiments on a real HSR line. Section IV describes the proposed system model of HSR networks. Section V formulates, transforms and decomposes the video streaming optimization problem. Section VI proposes the JSDO mechanism to solve the formulated optimization problem and section VII shows and discusses the JSDO evaluation results. Section VIII discusses the suitability of transport protocols of video streaming in HSR networks. Section IX concludes this paper.

II. Related Work

Recently, increasing attention has been given to providing high-quality Internet services in HSR networks. In [13], track-side access points are proposed to be deployed along railway lines in order to assist Base Stations (BS) in Internet service delivery. In [14]–[16], the multi-service delivery problem is investigated with considerations for heterogeneous Quality of Service (QoS) requirements. In [17], a dual decomposition method-based resource allocation scheme designed for HSR networks, respecting specific service quality constraints, is described. In [18], admission control and resource allocation are jointly considered and distributed algorithms are proposed to maximize system utility and maintain system stability. In [19], with consideration of resource reservation and preemption, an admission control scheme is proposed to reduce service dropping probability. In [20], a cross-layer optimization framework in HSR networks and a cooperative resource allocation scheme with global optimality are proposed.

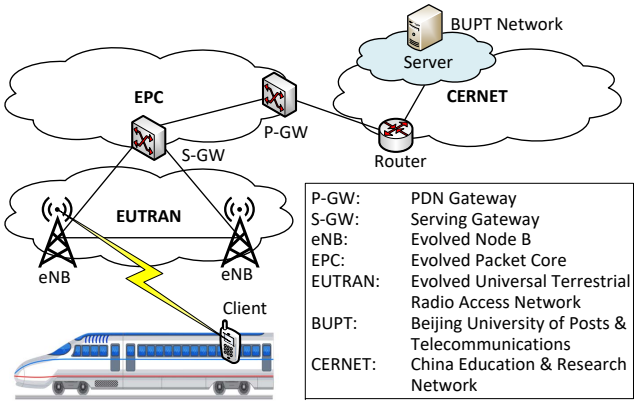


Fig. 1. Measurement System & Environment

Adaptive video streaming also attracts very much interest as it provides a flexible control of video quality according to communication conditions. In [3] and [4], cooperative and collaborative methods are researched and implemented to address the resource competing problem between multiple clients. In [5], a new bitrate adaption algorithm is proposed based on estimated throughput using adaptive forgetting factor. The authors of [6] propose a virtual caching scheme for adaptive video streaming based on network function virtualization to save computing and storage resources. In [7], DASH is implemented and a physiology-aware adaption technique is proposed to transmit critical clinical data reliably. The authors of [8]–[10] focus on improving the quality of adaptive video streaming in terms of content management, resource allocation, and video encoding, respectively. Based on SDNDASH architecture, an enhanced video streaming architecture called SDNHAS is established in [11] which achieves better performance in terms of adaption decision, communication overhead and resource allocation. In [12], an adaptive bitrate selection algorithm is proposed for DASH. By developing a prototype system and running field experiments, the advantage of the proposed algorithm is demonstrated.

Apart from the aforementioned works, there is very much research effort put on improving the video streaming service in wireless mobile networks. The authors of [21] consider the problem of video streaming in content-centric mobile networks and propose a mobility-adaptive video delivery mechanism. [22] describes a cost-efficient multimedia content transmission mechanism proposed for information-centric vehicular networks. In this mechanism, content mobility and tradeoff between supply and demand are analyzed in an original manner. In [23], the concept of social-aware rate delivery is proposed and a social-aware-rate content sharing approach is designed for device-to-device communications. In [24] and [25], video content placement and caching problems are investigated over wireless networks focusing on cost reduction and service quality, respectively.

III. Real-life HSR Experiment and Dataset

This section introduces the experimental testbed and field experiments performed and presents the collected PLR datasets. Additionally, the causes of packet loss in HSR networks are also analyzed.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2018.2881095, IEEE Transactions on Multimedia

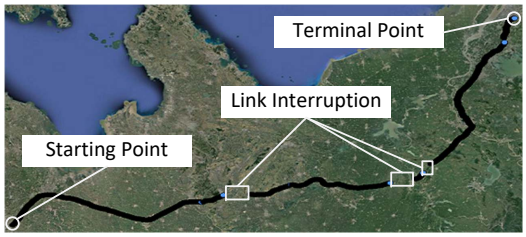


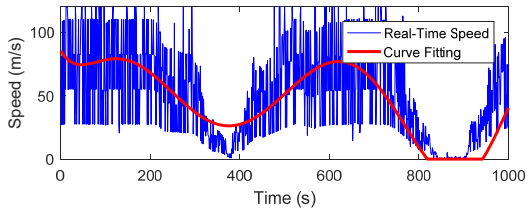
Fig. 2. HSR Line

In order to investigate PLR in HSR networks, we develop a PLR measurement system which is illustrated in Fig. 1. The system is composed of a server and a client. The application at the server is developed using JAVA with jdk1.8.0 and is deployed at campus network of Beijing University of Posts and Telecommunications within China Education and Research Network (CERNET). CERNET only serves for Chinese universities and scientific organizations. The client is an Android application which runs on a Samsung N-9002 device with Android 5.0 Lollipop as operating system. To establish the link, client subscribes the network access services of China Unicom. China Unicom deploys large-scale 4G (FDD-LTE) and 3G (WCDMA) networks in China. The client will automatically connect to 3G network when 4G network is not available. During the measurements, the client sends a single packet to the server every 30 milliseconds using Transmission Control Protocol (TCP). As soon as the server receives the packet, it will reply with a response packet also using TCP. The response timeout of client is set as 3 seconds. If the client receives the response packet within 3 seconds, the packet transmission is viewed as successful. Otherwise, packet loss occurs. At the client, PLR is calculated every 3 seconds and SQLite is used to store the PLR data. Apart from PLR, location information is also obtained from the network operator (i.e. eNode-B) and recorded.

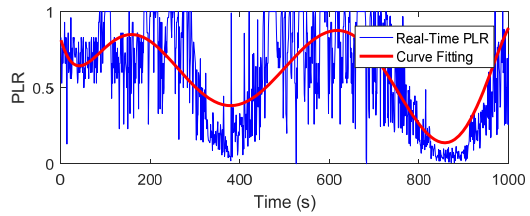
The field experiments are performed along a real HSR line which is shown in Fig. 2. The railway line is around 1300 kilometers long of which about 910 kilometers are viaducts. The blank spaces marked in this HSR line illustration are due to the loss of location information which is caused by communication link interruption.

Fig. 3 shows the speed of High Speed Trains (HST) and PLR in HSR networks. The speed of HST is derived from the location information collected every 3s. Because there exist errors in our collected location information, it seems that the speed changes frequently. Actually, HST is always driving smoothly. It can be observed that, during 800s to 950s, the HST is stopped at a railway station and its speed is reduced to 0m/s. PLR decreases accordingly. After 950s, HST starts to drive again and PLR also increases. Therefore, it can be concluded that PLR is in direct proportion to HST speed.

Besides the experiments performed along HSR line, we also conduct similar experiments in static state and freeway scenario using the same measurement system. Experiment results in the three scenarios are shown in Fig. 4 and Fig. 5. Fig. 4 provides a holistic view about the PLR status. In static state, PLR is always less than 20% and remains below 10% for nearly 85% of test time. In freeway scenario, the



(a) Speed of HST



(b) PLR of HSR Networks

Fig. 3. Speed and PLR in HSR Scenario

maximum value of PLR increases to 80%. Meanwhile, PLR maintains at a low level, i.e. 0%–20%, for about half of test time. Thus, the overall communication quality in freeway scenario is acceptable. In HSR scenario, PLR is larger than 90% for more than half of test time and remains below 30% for only 15% of test time. Fig. 5 shows parts of PLR datasets in the three scenarios. It can be observed that PLR changes in a small range in static state and it is more stable than the PLR measured in other scenarios. In freeway scenario, PLR vibrates between 0%–65%, which is a bit worse than the PLR in static state. As for HSR scenario, PLR changes frequently and it increases to about 90% from low levels for numerous times. Therefore, it can be concluded that PLR status is the worst in HSR scenario.

From Fig. 3–Fig. 5, we can conclude that there exists a tradeoff between HST speed and PLR. If PLR with low level is required, HST should be slowed down. Otherwise, PLR will be increased. Meanwhile, in DASH, high PLR will result in frequent packet retransmission since HTTP/TCP is employed. It further indicates that the energy consumed by successful data transmissions and data transmission delay will be greatly increased because of additional retransmissions. Then the quality of DASH-based video services will be also impaired because of longer transmission delay. Therefore, it can be deduced that the increased speed of HST will degrade the performance of transmission energy consumption and DASH-based video service.

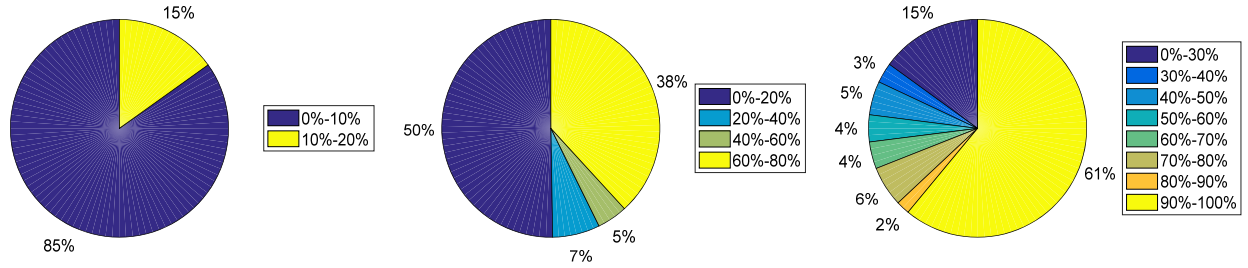
It is already known that the wireless links between mobile users and HST are the bottlenecks in HSR networks [26] and determine packet loss. Therefore, the causes of packet loss are analyzed in more details next, with focus on the wireless links.

1) The frame[1] transmitted over the wireless links are prone to corruption and loss. The high velocity of HST causes serious Doppler frequency shift. The bit error rate increases with the growth of the Doppler frequency shift. This further results in the increase probability of frame corruption and loss.

2) Handover is performed frequently in HSR networks. It is reported that handover frequency is greatly increased and

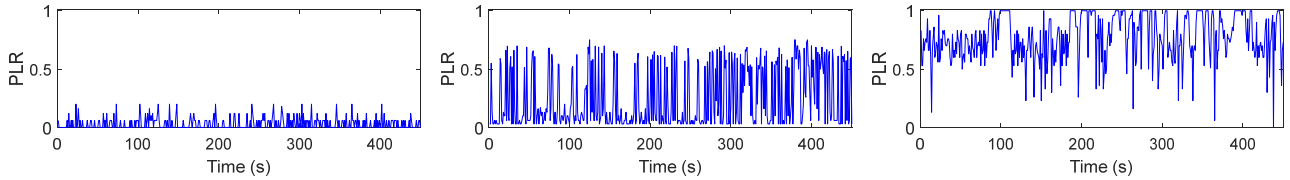
[1]Data packets from network layer are encapsulated into frames at link layer.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2018.2881095, IEEE Transactions on Multimedia



(a) Static state: velocity is 0km/h (b) Freeway: velocity is 100km/h (c) HSR: velocity is 300km/h

Fig. 4. Holistic View of PLR Status in Different Scenarios



(a) Static state: velocity is 0km/h (b) Freeway: velocity is 100km/h (c) HSR: velocity is 300km/h

Fig. 5. PLR Dataset in Different Scenarios

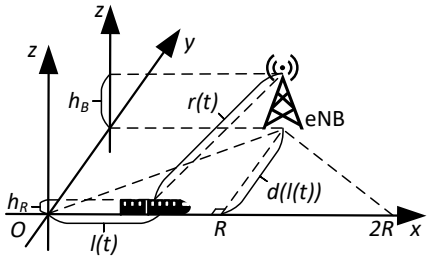


Fig. 6. Abstract Model of HSR Networks

handover failure rate is about 21% [27]. This is unprecedented in any other mobile networks and the problem of packet loss is therefore aggravated in HSR networks.

3) The metal cabin of HST causes serious penetration loss. This greatly impairs the wireless access of mobile users. However, the penetration loss can be alleviated through a two-hop architecture [28]. This enables mobile users access to the wireless network through access points installed on train cabins.

Based on the above analysis, this paper focuses on the frame loss problem at data link layer (both corrupted and lost frames are considered as lost). The frequent handover and penetration loss problems will be researched in the future. Moreover, the collected PLR dataset will be used as an approximation for real time frame loss probability in Delivery Capacity Model in Section IV-B.

IV. System Model

This section describes the proposed general system model which includes the transmission capacity model, delivery capacity model, DASH model and energy model. Table I lists the main notations used in this paper.

A. Transmission Capacity Model

Since the wireless links between eNB and mobile users in HST are the bottleneck of HSR networks, we focus on the transmission capacity of the wireless links. The abstract model of the wireless communication environment is first extracted as shown in Fig. 6. In this model, a LTE network architecture is employed as network infrastructure because it is one of the most commonly used wireless communication networks nowadays. For ease of analysis, it is assumed that eNB are allocated along the railway line with an equal distance $2R$ between them. Therefore, the length of the railway line covered by each eNB is $2R$. The heights of an eNB and mobile user are denoted as h_B and h_R, respectively. Assume HST starts driving from point O with a speed $v(t)$ along the railway line. At time t, the distance that HST has traveled is $l(t) = \int_0^t v(t)\,dt$. The vertical distances between the railway

TABLE I
Parameter Setting

Notation	Explanation
$r(\cdot)$	Distance between eNB and mobile user
$pl(\cdot)$	Path loss of wireless link
$C(\cdot)$	Total network capacity
Δt	Time slot length
$c_i^{\kappa}(\cdot)$	Number of arrived frames for mobile user i
L	Size of data frames
$\mathcal{N}$	Set of mobile user $\mathcal{N} = \{1, \ldots, n_m\}$
$m_i(\cdot)$	Number of duplicated for mobile user i
$r_i(\cdot)$	Bitrate of video streaming for mobile user i
P_e	Power of eNB
W	System bandwidth
$C^f(t)$	Transmission capacity
$D_i(t)$	Delivery capacity
$n(\cdot)$	Number of different frames transmitted in a time slot
$p_i(\cdot)$	Frame loss probability of mobile user i
$e_i(\cdot)$	Energy Efficiency Rate for mobile user i
M	Consumed energy of eNB in a time slot
E_e	Energy cost for transmission link establishment
$\Phi(\cdot)$	Quality of video streaming
α	Positive constant to balance the video quality and video quality variation
$K(\cdot)$	Stability of Video streaming
$\mathcal{L}(\cdot)$	Lyapunov function
$G_i(\cdot), H_i(\cdot)$ $I_i(\cdot), R_i(\cdot)$	Lengths of virtual queues

line and eNBs are determined by the eNB that HST are connected, which is further determined by the location of HST. Then it can be denoted as $d\left(l\left(t\right)\right)$. Denote $r\left(t\right)$ as the distance between eNB and HST and it is calculated as equation (1).

$$r\left(t\right)=\sqrt{\left[R-\bmod\left(l\left(t\right),2R\right)\right]^{2}+d^{2}\left(l\left(t\right)\right)+\left(h_{B}-h_{R}\right)^{2}} \quad (1)$$

where $\bmod\left(a,b\right)$ denotes the remainder of a/b.

In HSR networks, the path loss of the wireless link between eNB and mobile user is given in equation (2) according to [14].

$$pl\left(r\left(t\right)\right)=\begin{cases}44.2+21.5\log\left(r\left(t\right)\right)+L, r\left(t\right)<l_{bp}\\ 44.2+40\log\left(r\left(t\right)/l_{bp}\right)+L_{bp}+L, r\left(t\right)\geq l_{bp}\end{cases} \quad (2)$$

where L_{bp}, l_{bp} and L are constants, $L=20\log\left(f/5\times10^{9}\right)$, $L_{bp}=21.5\log\left(l_{bp}\right)$ and $l_{bp}=4h_{B}h_{R}f/\omega$. f is the carrier frequency and ω is the speed of light. If P_e is the power of eNB and N_0 is the noise power, we have the Signal-to-Noise Ratio (SNR) of the wireless link shown in equation (3).

$$SNR=P_{e}-pl\left(r\left(t\right)\right)-N_{0} \quad (3)$$

We assume HSR networks operate in slotted time and all time slots are of the same length Δt. Δt is a constant. The total network capacity of the wireless link during time slot t is calculated as in equation (4).

$$C\left(t\right)=\int_{t}^{t+\Delta t}W\log_{2}\left(1+SNR\right)dt \quad (4)$$

where W is system bandwidth. Therefore, the number of data frames that can be transmitted over wireless link in time slot t, i.e. transmission capacity, is calculated as in equation (5).

$$C^{f}\left(t\right)=\frac{C\left(t\right)}{L} \quad (5)$$

where L is the size of data frame.

It is noteworthy that equation (5) gives the upper bound of transmission capacity for the wireless link between client and eNB. Since the data transmission suffers on the lossy wireless links between mobile users and eNB, a more realistic expression which incorporates the collected PLR datasets and data transmission process is given in next subsection based on this upper bound.

The maximum value of $C^{f}\left(t\right)$ is further denoted as C_{max}^{f} and is obtained when $\bmod\left(l\left(t\right),2R\right)=R$. The transmission capacity for each user $i\in\mathcal{N}$ at time slot t is denoted as $C_{i}^{f}\left(t\right)$. Because mobile users are able to acquire different transmission capacities through different subscription plans and wireless access technologies, we denote the rate of the transmission capacity allocated to mobile user i as $q_{i}\left(t\right)$ and $\sum_{i=1}^{n_{m}}q_{i}\left(t\right)=1$. Therefore, the allocated transmission capacity allocated to mobile user i is $C_{i}^{f}\left(t\right)=q_{i}\left(t\right)C^{f}\left(t\right)$.

B. Delivery Capacity Model

We define the delivery capacity $D_{i}\left(t\right)$ as the amount of data associated with the successful transmitted frames for user $i\in\mathcal{N}$. Considering the frame loss over the wireless links between eNB and mobile users, it is immediately clear that $D_{i}\left(t\right)\leq C_{i}^{f}\left(t\right)$. At link layer, the Selective Repeat-Automatic Repeat reQuest (SR-ARQ) is always employed to ensure the data reliability of wireless transmission [29]. In SR-ARQ, each data frame should be acknowledged by an ACKnowledgment (ACK) frame. Only an acknowledged data frame is viewed as transmitted successfully; otherwise, it will be resent. However, high frame loss rate severely degrades the performance of SR-ARQ [30]. Therefore, the Loss-Aware Adaptive Scalable Transmission (LAAST) mechanism [31] that we proposed previously is employed. LAAST basically utilizes dynamic frame duplication to reduce the negative effect of frame loss. In LAAST, the number of duplicated frames for user i is denoted as $m_{i}\left(t\right)$, which is determined by the real-time frame loss probability $p_{i}\left(t\right)$. As analyzed in previous section, the frame loss over wireless link is the primary cause of packet loss in HSR networks. Thus, in delivery capacity model, $p_{i}\left(t\right)$ is less than and approximates PLR. In this way, the collected PLR dataset can be incorporated to reflect the overall negative influences of lossy wireless links. Then the number of different frames transmitted in a time slot is calculated using floor function as in equation (6).

$$n\left(t\right)=\left\lfloor C_{i}^{f}\left(t\right)/m_{i}\left(t\right)\right\rfloor \quad (6)$$

where $\lfloor\rfloor$ denotes taking the largest integer less than the value. In this way, the frame loss probability at time slot t is reduced to $\left(p_{i}\left(t\right)\right)^{m_{i}(t)}$ for mobile user i. Since the ACK frame is also suffering from the frame loss, it is also duplicated in LAAST. However, since the frame loss probability of ACK frame is always smaller than that of data frame with payload, the frame loss probability is denoted as $\phi p_{i}\left(t\right)$, where ϕ is a real number in the range of $[0,1]$ and denotes the ratio between the frame loss probabilities of ACK frame and data frame. Therefore, the delivery capacity for user i is denoted as $D_{i}\left(t\right)$ and is calculated as in equation (7).

$$D_{i}\left(t\right)=\underbrace{n\left(t\right)\cdot\left(1-\left(p_{i}\left(t\right)\right)^{m_{i}(t)}\right)}_{\text{data frame}}\cdot\underbrace{\left(1-\left(\phi p_{i}\left(t\right)\right)^{m_{i}(t)}\right)}_{\text{ACK frame}} \quad (7)$$

As equation (7) is derived in our previous work [31], we omit the details for brevity.

C. DASH Model

Assume DASH [32] is adopted by the video streaming service. In DASH, a video is divided into consecutive segments. We denote the set of segments as $\mathcal{S}=\{1,2,\ldots,s\}$. Each segment is of several seconds (1–10 seconds) and encoded into different bitrates. The length of segment $\kappa\in\mathcal{S}$ is $t_{l}\left(\kappa\right)$. We denote the set of mobile users in HST who are using video streaming services as $\mathcal{N}=\{1,2,\ldots,n_{m}\}$. The selected bitrate of segment κ for mobile user $i\in\mathcal{N}$ at time slot t is denoted as $r_{i}^{\kappa}\left(t\right)$. Then the number of frames generated by segment κ at time slot t is calculated as in equation (8).

$$c_{i}^{\kappa}\left(t\right)=\frac{r_{i}^{\kappa}\left(t\right)\Delta t}{L} \quad (8)$$

We further denote the maximum and minimum values of $c_{i}^{\kappa}\left(t\right)$ as c_{max} and c_{min}, respectively.

D. Energy Model

The energy model is established considering both eNB and user device. Related to eNB, since it is typically connected to permanent power, we mainly focus on its energy efficiency. Regarding user device, we concentrate on the constraints of energy consumption because of its limited battery capacity.

1) Energy Efficiency Rate for eNB: To optimize energy efficiency of eNB, we define the Energy Efficiency Rate (EER) as the ratio between the energy consumption of eNB and delivery capacity $D_i(t)$. For ease of analysis, we assume eNB consumes a fixed amount of energy in each time slot and advanced power allocation approaches are beyond the scope of this paper. However, it is worth noting that power allocation approaches can be easily incorporated into our model.

The definition of EER is shown as equation (9).

$$e_i(t) = \frac{M}{D_i(t) + \sigma} \tag{9}$$

where M denotes the consumed energy in a time slot and σ is a positive constant number which is used to prevent the denominator be equal to zero.

From equation (9), it is immediately clear that $e_i(t)$ denotes the amount of energy consumed by successfully transmitting a data frame. It can also be deduced that the small value of $e_i(t)$ means large number of frames are transmitted and indicates higher utilization ratio of energy.

2) Energy Consumption for User Device: The energy consumption of user device's Network Interface Card (NIC) is composed of two parts [33]. The one is the energy cost for transmission link establishment, which is a constant and denoted as E_e. The other one is data transfer energy consumption. We denote the energy consumed by transmitting one frame over wireless link as E_t. Then the data transfer energy consumed during previous t time slots is $E_t \sum_{\tau=1}^{t} c_i^{\kappa}(\tau)$. The total energy consumption of user device's NIC is calculated as (10).

$$E_i(t) = E_e + E_t \sum_{\tau=1}^{t} c_i^{\kappa}(\tau) \tag{10}$$

V. Problem Formulation, Transformation & Decomposition

In this section, the problem of video streaming service delivery in HSR networks is first formulated based on the established system model. Considering the intractability of this problem, it is further transformed into a queue stability problem and decomposed into three subproblems which can be solved separately and easily.

A. Problem Formulation

According to [35], the perceived quality of video streaming service is mainly influenced by three aspects, including average video quality, video quality variation and rebuffering. Even though the exact expressions of the three aspects are provided, they are not suitable for our optimization framework. Therefore, some minor revisions are made without affecting the basic ideas.

Average video quality is evaluated through a function of average bitrate. According to [36], a logarithmic function which has the diminishing returns property is employed and is shown in equation (11).

$$\Phi\left(\overline{c_i^{\kappa}(t)}\right) = \zeta \ln\left(\overline{c_i^{\kappa}(t)}\right) + \beta \tag{11}$$

where ζ and β are constants. The notation of $\overline{c_i^{\kappa}(t)}$ denotes the time average, that is $\overline{c_i^{\kappa}(t)} = \lim_{t\to\infty} \frac{1}{t} \sum_{\tau=1}^{t} c_i^{\kappa}(\tau)$.

Video quality variation is evaluated by the squared difference between the current and time averaged bitrate at time slot t, and it is measured as in equation (12). The essential idea behind equation (12) comes from the concept of variance which measures how far a set of numbers diverge from their average value.

$$K\left(\overline{c_i^{\kappa}(t)}, t\right) = -\left[c_i^{\kappa}(t) - \overline{c_i^{\kappa}(t)}\right]^2 \tag{12}$$

It is worth noting that frequent change of video bitrate will impair the quality of video streaming service. Therefore, $K\left(\overline{c_i^{\kappa}(t)}, t\right)$ is set negative.

Rebuffering always happens when network transmission capacity is not able to support the amount of requested video contents. To avoid the rebuffering happening, video bitrate $c_i^{\kappa}(t)$ should be less than network delivery capacity $D_i(t)$. Then we have the constraint (13) to ensure that videos can be played smoothly.

$$\overline{c_i^{\kappa}(t)} \le \overline{D_i(t)} \tag{13}$$

Besides the quality of video streaming service, energy consumption is also an important consideration. As presented in Section IV-D, energy-related constraints are established in terms of eNB and mobile user devices.

EER constraint of eNB is formulated as (14).

$$\overline{e_i(t)} \le e_{av} \tag{14}$$

where e_{av} denotes the desired average EER.

As shown in equation (10), energy consumption of user devices is composed of connection establishment energy E_e and data transfer energy $E_t \sum_{\tau=1}^{t} c_i^{\kappa}(\tau)$. Minimizing the connection establishment energy is related to analyzing and designing wireless communication protocols. It is out of the scope of this paper. We mainly focus on minimizing data transfer energy consumption, i.e. $E_t \sum_{\tau=1}^{t} c_i^{\kappa}(\tau)$. Then the energy constraint of user device is shown as in inequality (15).

$$E_t \overline{c_i^{\kappa}(t)} \le c_i^e \tag{15}$$

where c_i^e is average energy consumption.

Given formulae (11)-(15), we formulate the optimization problem for the video streaming service as in equation (16).

$$\begin{aligned} \max \quad & \Phi\left(\overline{c_i^{\kappa}(t)}\right) + \alpha K\left(\overline{c_i^{\kappa}(t)}, t\right), \ \forall t \\ \text{s.t.} \quad & \overline{c_i^{\kappa}(t)} \leq \overline{D_i(t)}, i \in \mathcal{N} \\ & \overline{e_i(t)} \leq e_{av}, i \in \mathcal{N} \\ & E_t\overline{c_i^{\kappa}(t)} \leq c_i^e, i \in \mathcal{N} \end{aligned} \tag{16}$$

where α is a positive constant which is used to balance the video quality and video quality variation.

The problem from equation (16) is mathematically intractable, because it includes various factors such as frame loss, energy consumption, service delivery, video streaming bitrate, video quality and video quality variation. Moreover, there also exist intricate and implicit relationships between these factors. Therefore, to address this problem, it is necessary to transform this problem by utilizing the Lyapunov optimization approach [36]. The main insight behind this approach is to transform the constraints of optimization problem into virtual queues. Then these constraints can be satisfied by maintaining the stability of these virtual queues. Moreover, the relationships between the variables in problem (16) will become mathematically tractable by virtue of these virtual queues. In this context, the difficulty and complexity of problem (16) can be reduced.

B. Problem Transformation

We first transform the optimization objective of problem (16). Since it involves the function of time average $\overline{c_i^{\kappa}(t)}$ and is not linear, an auxiliary variable $a_i(t)$ $(0 \leq a_i(t) \leq c_{max})$ is introduced according to Lyapunov optimization approaches. $a_i(t)$ should satisfy the constraint from equation (17).

$$\overline{a_i(t)} \leq \overline{c_i^{\kappa}(t)} \tag{17}$$

By leveraging the auxiliary variable $a_i(t)$, the optimization objective of problem (16) is transformed as in equation (18).

$$\begin{aligned} & \overline{\sum_{i=1}^{n_m} \Phi(a_i(t)) + \alpha \sum_{i=1}^{n_m} K(a_i(t), t)} \\ & = \sum_{i=1}^{n_m} \left[\zeta \ln(a_i(t)) + \beta\right] - \alpha \sum_{i=1}^{n_m} \left[c_i^{\kappa}(t) - a_i(t)\right]^2 \end{aligned} \tag{18}$$

Based on the transformation of the constraints of problem (16) and constraint from equation (17), the following virtual queues are derived which are shown in equations (19)–(22).

$$G_i(t+1) = \max\left[G_i(t) - e_{av}, 0\right] + e_i(t) \tag{19}$$

$$H_i(t+1) = \max\left[H_i(t) - D_i(t), 0\right] + c_i^{\kappa}(t) \tag{20}$$

$$I_i(t+1) = \max\left[I_i(t) - c_i^e, 0\right] + E_t c_i^{\kappa}(t) \tag{21}$$

$$R_i(t+1) = \max\left[R_i(t) - c_i^{\kappa}(t), 0\right] + a_i(t) \tag{22}$$

where $G_i(0)$, $H_i(0)$, $I_i(0)$ and $R_i(0)$ are 0.

Lemma 1. *Taking the queue from equation (19) and constraint $\overline{e_i(t)} \leq e_{av}$ as an example, $\overline{e_i(t)} \leq e_{av}$ is satisfied only when the queue is mean rate stable, which is defined as $\lim_{t\to\infty} \frac{\mathbb{E}\{G_i(t)\}}{t} = 0$, where $\mathbb{E}\{\cdot\}$ denotes the expectation.*

The proof of **Lemma 1** is shown as Appendix A

Given **Lemma 1**, the optimization problem (16) can be transformed into a problem of queue stability which is shown in equation (23).

$$\begin{aligned} \max \quad & \overline{\sum_{i=1}^{n_m} \Phi(a_i(t)) + \alpha \sum_{i=1}^{n_m} K(a_i(t), t)} \\ \text{s.t.} \quad & \text{Queues (19)–(22) are mean rate stable, } i \in \mathcal{N} \end{aligned} \tag{23}$$

It is noteworthy that with the help of virtual queues, the intricate relationships between these considered factors are described in the form of queue dynamics. It makes these relationships much clearer and more tractable. It also provides great convenience to formulate the optimization mechanism which will be shown in next sections. In this way, we can add more factors in initial optimization problem (16) to make it more general and accurate without the concern of unsolvability. Therefore, the analyzing and optimizing process in this paper is of high scalability and generality and can be easily extended to incorporate more factors.

C. Problem Decomposition

By establishing the following row matrices:

$$\begin{aligned} \mathbf{R}(t) &= [R_1(t), R_2(t), \ldots, R_{n_m}(t)] \\ \mathbf{G}(t) &= [G_1(t), G_2(t), \ldots, G_{n_m}(t)] \\ \mathbf{H}(t) &= [H_1(t), H_2(t), \ldots, H_{n_m}(t)] \\ \mathbf{I}(t) &= [I_1(t), I_2(t), \ldots, I_{n_m}(t)] \end{aligned}$$

a row matrix $\boldsymbol{\eta}(t) = [\mathbf{R}(t), \mathbf{G}(t), \mathbf{H}(t), \mathbf{I}(t)]$ is obtained. Then the Lyapunov function of $\boldsymbol{\eta}(t)$ can be given as in equation (24).

$$\begin{aligned} & \mathcal{L}(\boldsymbol{\eta}(t)) = \tfrac{1}{2}\boldsymbol{\eta}(t)\boldsymbol{\eta}^T(t) \\ & = \tfrac{1}{2}\sum_{i=1}^{n_m}\left[R_i^2(t) + G_i^2(t) + H_i^2(t) + I_i^2(t)\right] \end{aligned} \tag{24}$$

where the superscript $(\cdot)^T$ denotes the transposition of matrix. $\mathcal{L}(\boldsymbol{\eta}(t))$ indicates the overall status of the above queue lengths. A large value of $\mathcal{L}(\boldsymbol{\eta}(t))$ indicates at least one queue is congested. On the contrary, a small value of $\mathcal{L}(\boldsymbol{\eta}(t))$ implies all queue lengths are short.

Then the Lyapunov drift is formulated as in equation (25).

$$\Delta(t) = \mathbb{E}\left\{\mathcal{L}(\boldsymbol{\eta}(t+1)) - \mathcal{L}(\boldsymbol{\eta}(t)) \,|\boldsymbol{\eta}(t)\right\} \tag{25}$$

$\Delta(t)$ represents the variation of Lyapunov function value in one time slot. If $\Delta(t)$ becomes smaller, the above queues will be stabler. Therefore, aiming at improving the queue stability and maximizing quality of video streaming service, the drift-minus-penalty is defined as in equation (26).

$$\Delta(t) - V\mathbb{E}\left\{\sum_{i=1}^{n_m}\Phi(a_i(t)) + \alpha\sum_{i=1}^{n_m}K(a_i(t), t)\right\} \tag{26}$$

where V is a positive real number used to represent the significance of video streaming service quality. By further considering the inequality $(\max[a-b,0]+c)^2 \leq a^2 + b^2 + c^2 - 2a(b-c)$, the upper bound of drift-minus-penalty is shown as in inequality (27).

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2018.2881095, IEEE Transactions on Multimedia

$$\Delta(t) - V\mathbb{E}\left\{\sum_{i=1}^{n_m}\Phi(a_i(t)) + \alpha\sum_{i=1}^{n_m}K(a_i(t),t)\,|\boldsymbol{\eta}(t)\right\} \leq \mathcal{B}$$

$$-\sum_{i=1}^{n_m}\mathbb{E}\left\{H_i(t)\,D_i(t) - G_i(t)\,e_i(t) + G_i(t)\,e_{av}\,|\boldsymbol{\eta}(t)\right\} \quad (27a)$$

$$-\sum_{i=1}^{n_m}\mathbb{E}\left\{[R_i(t)\,E_t - H_i(t) - I_i(t)]\,c_i^{\kappa}(t) + I_i(t)\,c_i^{e}\,|\boldsymbol{\eta}(t)\right\} \quad (27b)$$

$$-\sum_{i=1}^{n_m}\mathbb{E}\left\{V[\Phi(a_i(t)) + \alpha K(a_i(t),t)] - R_i(t)\,a_i(t)\,|\boldsymbol{\eta}(t)\right\} \quad (27c)$$

where $\mathcal{B} = \frac{1}{2}n_m\left(4c_{max}^2 + (c_i^e)^2 + e_{av}^2 + \left(\frac{M}{\sigma}\right)^2 + \left(C_{max}^f\right)^2\right)$.

Now by minimizing the upper bound of drift-minus-penalty, the problem from equation (23) can be decomposed into the following three subproblems.

1) Delivery Control Problem:

$$\max \sum_{i=1}^{n_m}[H_i(t)\,D_i(t) - G_i(t)\,e_i(t) + G_i(t)\,e_{av}] \quad (28)$$

This optimization problem is derived from equation (27a). In problem (28), given equations (7)–(9), we can observe that $e_i(t)$ is derived from $D_i(t)$ and $D_i(t)$ is further related with frame duplication number $m_i(t)$. Therefore, by solving this problem, the value of $m_i(t)$ can be determined and controlled for each time slot.

2) Bitrate Selection Problem:

$$\max \sum_{i=1}^{n_m}\{[R_i(t)\,E_t - H_i(t) - I_i(t)]\,c_i^{\kappa}(t) + I_i(t)\,c_i^{e}\} \quad (29)$$

The problem from equation (29) is basically formulated based on equation (27b). Because both the two terms in equation (27b) relate with $c_i^{\kappa}(t)$, this problem provides insight on deciding the value of $c_i^{\kappa}(t)$, which further determines the bitrate of adaptive video streaming.

3) Auxiliary Computation Problem:

$$\max \sum_{i=1}^{n_m}[V\Phi(a_i(t)) + \alpha VK(a_i(t),t) - R_i(t)\,a_i(t)] \quad (30)$$

This maximization problem is derived from equation (27c). It involves the video quality, video quality variation, queue $R_i(t)$ and auxiliary variable $a_i(t)$. The purpose of this problem is to select an appropriate value of $a_i(t)$ so as to optimize the quality of video streaming service and maintain the stability of queues.

Now the problem (16) is transformed and decomposed into the above three mathematically tractable subproblems which provide insight on optimizing the video streaming service with numerous consideration factors in an integrated manner in HSR networks. By solving the three subproblems, the aforementioned JSDO mechanism can be built efficiently.

VI. JSDO Mechanism

First this section introduces the proposed JSDO mechanism, which has components corresponding to the solutions of the three subproblems described in the previous section. Then the detailed algorithms employed in these modules are presented.

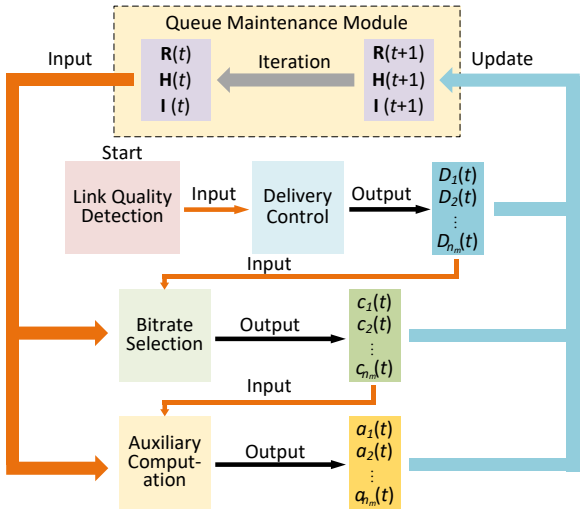


Fig. 7. JSDO Control Loop

A. JSDO Overview

Fig. 7 illustrates the control loop of the JSDO mechanism. It is composed of a link quality detection module, delivery control module, bitrate selection module, auxiliary computation module and queue maintenance module. The link quality detection module detects the real-time frame loss probability $p_i(t)$ and transmission capacity $C_i^f(t)$ of users. The detected information is sent to the delivery control module to solve the problem described in equation (28) and determine the frame duplication number $m_i(t)$ so as to control the delivery capacity $D_i(t)$. Based on $D_i(t)$, bitrate selection module is responsible for addressing the problem from equation (29) and choosing the appropriate bitrate for adaptive video streaming. The auxiliary computation module is used to derive the auxiliary variables to optimize the quality of video streaming service and maintain the stability of queues considering the output of bitrate selection module, which corresponds to the problem from equation (30). After all these modules finishing computation, results are collected by queue maintenance module to update the queue lengths. Then the updated queue lengths are further given to other modules to start another new computation. By taking the queue lengths as common inputs and influencing the queue lengths jointly, JSDO mechanism optimizes the adaptive video streaming and maintains the queue stability in an integrated and stochastic manner.

B. Delivery Control Module

The delivery control module takes the detection results of link quality detection module (i.e. $p_i(t)$ and $C_i^f(t)$) as inputs to perform delivery scheduling. The frame loss probability $p_i(t)$ can be measured using the method proposed in [37]. It can also be estimated by PLR using the measurement method described in Section III. Such estimation is reasonable since the wireless link between HST and eNB is the bottleneck of HSR network [26]. Most of packet losses are caused by frame losses on the wireless link.

Algorithm 1: Delivery Control Algorithm

Input: Real-time $C_i^f(t)$, real-time $p_i(t)$
Output: The delivery capacity $D_i(t)$

At the beginning of each time slot t;
for i = 1 *to* n_m **do**
  derive $m_i(t)$ through solving problem (33);
  duplicate the frames for $m_i(t)$ times and start transmission;
  derive the value of $D_i(t)$ using equation (7);
  return $D_i(t)$;
end

For the delivery control algorithm, considering equation (9), the optimization objective of problem from equation (28) can be written as in equation (31):

$$F(D_i(t)) = H_i(t) D_i(t) - G_i(t) \frac{M}{D_i(t)+\sigma} + G_i(t) e_{av} \tag{31}$$

The first derivative of $F(D_i(t))$ with respect to $D_i(t)$ is calculated as in equation (32).

$$F'(D_i(t)) = H_i(t) + G_i(t) \frac{M}{(D_i(t)+\sigma)^2} \tag{32}$$

Since $F'(D_i(t)) \geq 0$, $F(D_i(t))$ is monotonic increasing. According to the monotonicity of the composite function, the monotonicity of $F(D_i(t))$ is the same with $D_i(t)$. Therefore, in order to maximize the objective function of problem (28), we only need to find the maximum value of $D_i(t)$. Considering equation (7), the $\lfloor\rfloor$ operator is taken out to conduct an approximate analysis. Then we arrive at a similar optimization problem with that in [31], which is shown in equation (33).

$$\begin{aligned} \max \quad & D_i(t) = \frac{C_i^f(t)}{m_i(t)} \times \left(1 - (p_i(t))^{m_i(t)}\right) \\ & \times \left(1 - (\phi p_i(t))^{m_i(t)}\right) \\ \text{s.t.} \quad & m_i(t) \text{ is an integer number} \end{aligned} \tag{33}$$

This problem can be addressed using the method proposed in our previous work [31] and corresponding $m_i(t)$ and its upper bound can be obtained. In general, $m_i(t)$ is in direct proportion to $p_i(t)$. It implies when $p_i(t)$ increases, duplicating data frames for $m_i(t)$ times is able to increase delivery capacity $D_i(t)$. The *Delivery Control Algorithm* is shown in **Algorithm 1**. Note that the complexity of **Algorithm 1** is $\mathcal{O}(n_m)$. The proof of algorithm complexity is given as in Appendix B.

C. Bitrate Selection Module

Since the problem described in equation (29) is used to determine the value of $c_i^\kappa(t)$ for each user i which is related with biterate of video streaming, the solution is shown in equation (34).

$$c_i^\kappa(t) = \begin{cases} c_{max}, & R_i(t) E_t > H_i(t) + I_i(t) \\ c_{min}, & else \end{cases} \tag{34}$$

Algorithm 2: Bitrate Selection Algorithm

Input: Queue lengths of $R_i(t)$, $H_i(t)$ and $I_i(t)$ and delivery capacity $D_i(t)$
Output: The bitrate of DASH video $c_i^\kappa(t)$

At the beginning of each time slot t;
for i = 1 *to* n_m **do**
  if *Segement* κ *is not downloaded completely* **then**
    if $D_i(t) < c_i^\kappa(t-1)$ **then**
      return $[1-\theta_i(t)]\, c_i^\kappa(t-1)$;
    else
      return $c_i^\kappa(t-1)$;
    end
  end
  $dif = R_i(t) E_t - H_i(t) - I_i(t)$;
  if $dif > 0$ **then**
    $c_i^\kappa(t) = c_{max}$;
  else
    $c_i^\kappa(t) = c_{min}$;
  end
  if $D_i(t) < c_i^\kappa(t)$ **then**
    return $[1-\theta_i(t)]\, c_i^\kappa(t)$;
  else
    return $c_i^\kappa(t)$;
  end
end

Algorithm 3: Auxiliary Computation Algorithm

Input: Queue length of $R_i(t)$, bitrate $c_i^\kappa(t)$
Output: The value of auxiliary variable $a_i(t)$

At the beginning of each time slot t;
for i = 1 *to* n_m **do**
  solve $U'(a_i(t)) = 0$, and derive $a_i^*(t)$;
  let $a_i(t) = a_i^*(t)$;
  return $a_i(t)$;
end

In the above solution, when the values of $H_i(t)$ and $I_i(t)$ are small, it means the leveraged delivery capacity and energy consumption are relatively small. Meanwhile, large value of $R_i(t)$ indicates the bitrates selected previously are also small. Thus, when $R_i(t) > H_i(t) + I_i(t)$, it is necessary to increase the value of $c_i^\kappa(t)$.

Meanwhile, it is noteworthy that the above solution is effective when network delivery capacity of mobile user $i \in \mathcal{N}$ is able to support minimum video quality. But it is not always the case, especially when the number of mobile users increases greatly. Two methods can be used to address this problem. The one is to employ advanced wireless communication technology such as millimeter-wave technology [39] to provide higher communication bandwidth. This method can be easily incorporated into our system model by updating equations (2)–(4). The other one is to incorporate video admission control [40] into this module so as to further reduce the amount of requested contents when delivery capacity is insufficient. We define Content Drop Rate (CDR) as the percentage of the amount

of dropped video contents and denote CDR as $\theta_i(t)$. Then the amount of admitted video contents is $[1-\theta_i(t)]\,c_i^{\kappa}(t)$. However, determining the expression of $\theta_i(t)$ is out of the scope of this paper which involves a lot of considerations including video encoding, Quality of Experience and network bandwidth, etc. In this paper, we mainly focus on providing a general optimization framework which is scalable for admission control in future research. The *Bitrate Selection Algorithm* is shown in **Algorithm 2**. Similar with **Algorithm 1**, the complexity of **Algorithm 2** is $\mathcal{O}(n_m)$. Proof of algorithm complexity is the same as in Appendix B.

D. Auxiliary Computation Module

The optimization objective of the problem from equation (30) can be written as the function from equation (35).

$$\begin{aligned} U(a_i(t)) = {} & V\zeta \ln(a_i(t)) + \beta \\ & - V\alpha(c_i^{\kappa}(t) - a_i(t))^2 - R_i(t)\,a_i(t) \end{aligned} \tag{35}$$

The first and second derivatives of $U(a_i(t))$ with respect to $a_i(t)$ are calculated as in equations (36) and (37).

$$U'(a_i(t)) = V\zeta\frac{1}{a_i(t)} + 2V\alpha\,(c_i^{\kappa}(t) - a_i(t)) - R_i(t) \tag{36}$$

$$U''(a_i(t)) = -V\zeta\frac{1}{a_i^2(t)} - 2V\alpha \tag{37}$$

Since $U''(a_i(t)) < 0$ and the value range of $a_i(t)$ is $(0,+\infty)$, $U(a_i(t))$ is a concave function and $U'(a_i(t))$ is monotonic decreasing. Moreover, because $\lim_{a_i(t)\to 0^+} U'(a_i(t)) = +\infty$ and $\lim_{a_i(t)\to+\infty} U'(a_i(t)) = -\infty$, we can always find a unique root for equation $U'(a_i(t)) = 0$ on the interval $(0,+\infty)$. It implies that the maximum value point $a_i^*(t)$ of function $U(a_i(t))$ can be obtained by solving equation $U'(a_i(t)) = 0$ and the problem (30) is solved. The insight behind this problem is to tune the auxiliary variable $a_i(t)$ according to video quality, video quality variation and queue length $R_i(t)$. Then $a_i(t)$ will further influence the result of *Bitrate Selection Algorithm* in next time slot. In this way, the quality of video streaming service can be optimized. The *Auxiliary Computation Algorithm*, solving the problem from equation (30), is shown in **Algorithm 3**. The complexity of **Algorithm 3** is $\mathcal{O}(n_m)$. Proof of algorithm complexity is the same as in Appendix B.

E. JSDO Implementation Discussions

1) Implementation Methods

In general, the JSDO mechanism designed in previous sections can be implemented by two methods: centralized or distributed. The centralized implementation method requires a powerful computing platform such as a cloud computing center to guarantee the operating efficiency of the JSDO mechanism. Numerous queues can be maintained at the computing platform. Centralized implementation method also provides great convenience for module synchronization. Because all the modules are deployed at the same platform, the operating time lines of these modules can be easily maintained identical. Meanwhile, the start time and finish time for each computing iteration of delivery control module, bitrate selection module and auxiliary computation module can be regulated by queue maintenance module. Specifically, the finish time can be set as the time when queue maintenance module requests the computing results. The start time can be set as the time when it issues the updated queue lengths information.

To achieve distributed implementation, the three algorithms of JSDO should be divided into two categories to remove the dependency from the centralized computing platform. The first category is performed at eNB and includes the *Delivery Control Algorithm*. The second category is performed at mobile user devices in a fully distributed manner and includes the *Auxiliary Computation Algorithm*, *Bitrate Selection Algorithm* and *Queue Maintenance Module*. Additionally, the selection of energy constraint c_i^e should also be performed by mobile users. Meanwhile, the user devices should maintain and update three queue lengths including $R_i(t)$, $H_i(t)$ and $I_i(t)$. eNB should send its delivery control decisions $D_i(t)$ to each mobile user device to help them perform their algorithms. It is noteworthy that the queue length of $G_i(t)$ is not necessary to be maintained and updated. This is because as analyzed in section VI-B, the delivery control decisions can be made without the information $G_i(t)$. Therefore, the computation load can be further reduced at eNB.

In the above distributed implementation description, it can be noted that neither eNB, nor mobile user devices are necessary to undertake heavy computation loads. This approach saves the potential cost of renting or establishing a computation platform (i.e. cloud). But it has difficulty in module synchronization. Because all the modules are deployed at different devices, these modules will operate at different time lines and algorithms will be performed asynchronously. In this context, an additional time synchronization module is necessary to ensure these modules to operate at an identical time line [38]. However, time synchronization is not within the scope of this paper. We will further research the impact of time synchronization on our proposed mechanism in future research.

2) Signaling Overhead Analysis

Centralized implementation method performs the three algorithms and maintains the queue lengths at a computing platform. The computing platform needs to obtain the information about capacity $C_i^f(t)$ and frame loss probability $p_i(t)$ from eNBs. Meanwhile, it also needs to issue the frame duplication number $m_i(t)$ to eNBs and the decisions about $c_i^{\kappa}(t)$ to mobile users. Therefore, four kinds of signaling should be transmitted.

In distributed implementation method, powerful computing platform is no longer necessary because all the algorithms are performed at eNBs and user devices. Therefore, only one kind of signaling exchange is required. That is the information about $D_i(t)$ from eNB to user devices.

From the above analyses, we can conclude that the signaling overhead of centralized implementation method is four times as large as distributed one. Thus, distributed implementation method is signaling-efficient. However, it is noteworthy that user devices are always constrained by limited battery capacity.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2018.2881095, IEEE Transactions on Multimedia

Employing additional computing function at user devices will increase their energy consumption. A potential future research is hybrid implementation method which combines centralized and distributed implementation methods. Specifically, if the energy of user devices is enough, JSDO can be operated distributedly. Otherwise, computing platform will be used.

3) Large-scale Deployment

With the increase of deployment scale, module synchronization problem will become much more serious in distributed implementation method. This is because huge amount of user devices and eNBs are involved. Synchronizing the operating time lines of them is of great difficulty. However, computation power problem is natively eliminated in distributed implementation method because the computation loads are shared by user devices and eNBs. On the contrary, the computation power will become an essential problem and signaling consumption will also be increased greatly in centralized implementation method for large-scale deployment. But module synchronization can be achieved relatively easily.

F. Optimality Analysis

In this section, we analyze the optimality of the proposed JSDO mechanism in terms of video streaming service quality (i.e. video quality and video quality variation) and queue lengths. This is shown in **Theorem 1**.

Theorem 1. *For any parameter $V > 0$, we have the following performance guarantees when all queues are mean rate stable:*

1) The video streaming service quality achieved by JSDO differs from the optimal service quality by $\mathcal{O}\left(\frac{1}{V}\right)$, which is shown as in inequality (38).

$$\begin{aligned}&\mathbb{E}\left\{\overline{\sum_{i=1}^{n_m}\Phi^*\left(a_i\left(t\right)\right)+\alpha\sum_{i=1}^{n_m}K^*\left(a_i\left(t\right),t\right)}\right.\\&\left.-\overline{\sum_{i=1}^{n_m}\Phi\left(a_i\left(t\right)\right)+\alpha\sum_{i=1}^{n_m}K\left(a_i\left(t\right),t\right)}\right\}\leq\frac{\mathcal{B}+\varepsilon}{V}\end{aligned}\tag{38}$$

where $\mathbb{E}\left\{\overline{\sum_{i=1}^{n_m}\Phi^*\left(a_i\left(t\right)\right)+\alpha\sum_{i=1}^{n_m}K^*\left(a_i\left(t\right),t\right)}\right\}$ denotes the optimal video streaming service quality and ε is a positive constant.

2) The queue lengths satisfy inequality (39).

$$\begin{aligned}&\mathbb{E}\left\{\overline{\sum_{i=1}^{n_m}R_i\left(t\right)}+\overline{\sum_{i=1}^{n_m}G_i\left(t\right)}+\overline{\sum_{i=1}^{n_m}H_i\left(t\right)}+\overline{\sum_{i=1}^{n_m}I_i\left(t\right)}\right\}\\&\leq\frac{\mathcal{B}}{\pi}+\frac{V}{\pi}\mathbb{E}\left\{\overline{\sum_{i=1}^{n_m}\Phi^*\left(a_i\left(t\right)\right)+\alpha\sum_{i=1}^{n_m}K^*\left(a_i\left(t\right),t\right)}-\Lambda^s\right\}\end{aligned}\tag{39}$$

where Λ^s denotes the achieved video streaming service quality under Slater Condition which is stated in equation (45).

The proof of **Theorem 1** is given in Appendix C.

Theorem 1 proves that the time-averaged video streaming service quality achieved by the JSDO mechanism diverges from the optimal video streaming service quality at most with $\mathcal{O}\left(\frac{1}{V}\right)$. At the same time, the time-averaged queue lengths are bounded by $\mathcal{O}\left(V\right)$. It also shows that there exists a tradeoff between the system stability and video streaming service quality. When parameter V becomes large, the system puts more emphasis on the service quality maximization according to equation (26) and when $V \to +\infty$, the time-averaged service quality will approach the optimal service quality. However, the queue lengths will also become extremely long according to inequality (39). This implies the delay of data transmission becomes unacceptable.

TABLE II
PARAMETER SETTING

Parameter	Value	Parameter	Value
R	1500m	Δt	1s
v	100m/s	σ	10
SNR	115dB	M	135J
$d\left(l\left(t\right)\right)$	300m	c_{min}	256kbps
W	10M	c_{max}	1024kbps
h_B	50m	L	12000bit
h_R	2.5m	n_m	10

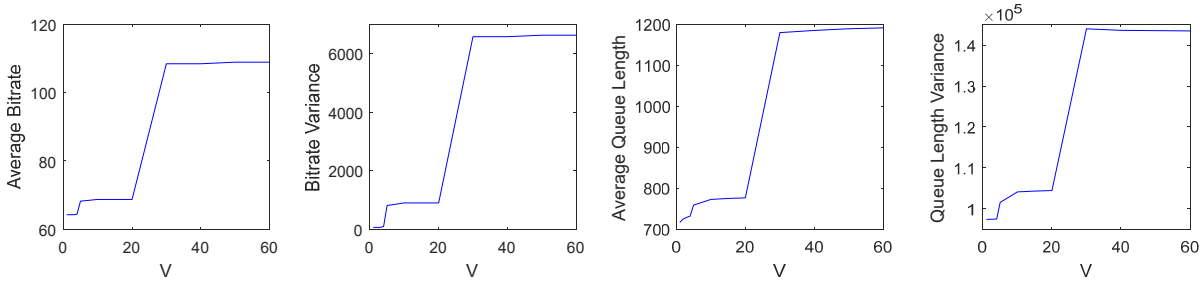
VII. PERFORMANCE EVALUATION

The three algorithms enable the JSDO mechanism solve the optimization problem from equation (16) efficiently. In this section, we evaluate the JSDO performance using numerical simulations and the realistic dataset collected from our real-life field experiments described in Section III. The numerical simulations are performed on MATLAB 2016a. During the simulations, the frame loss probability $p_i\left(t\right)$ is approximated by the collected PLR datasets since most of packet losses are caused by frame losses on the wireless links in HSR networks. We set time slot length Δt as 1 second. For video parameters, we first set the segment duration as 1 second and maximum and minimum bitrate levels are set as 256kbps and 1024kbps. Different segment durations are also considered in our simulations and its impact on JSDO performance will be analyzed. Meanwhile, since we need to examine time averaged performance of JSDO, we assume the total duration of video is long enough and mobile users are always using video streaming services. It is worth noting that the constants in equation (11) (i.e. β and ζ) take different values according to different kinds of video streaming services, user devices and environments. As formulating the exact function for the average video quality in HSR network environments is beyond the scope of this paper, we set these parameters to 1. For the same reason, we set the parameter ϕ as 0.5 and E_t as 1. The other parameter settings are shown in Table II.

JSDO is compared with Smart Schedule Algorithm (SSA) [12] and BOLA [41]. As for SSA, it only considers network bandwidth. Therefore, we further incorporate packet loss into SSA and formulate the Smart Schedule Algorithm-Packet Loss (SSA-PL), which is also used for comparison. Similarly, packet loss is also incorporated into the BOLA.

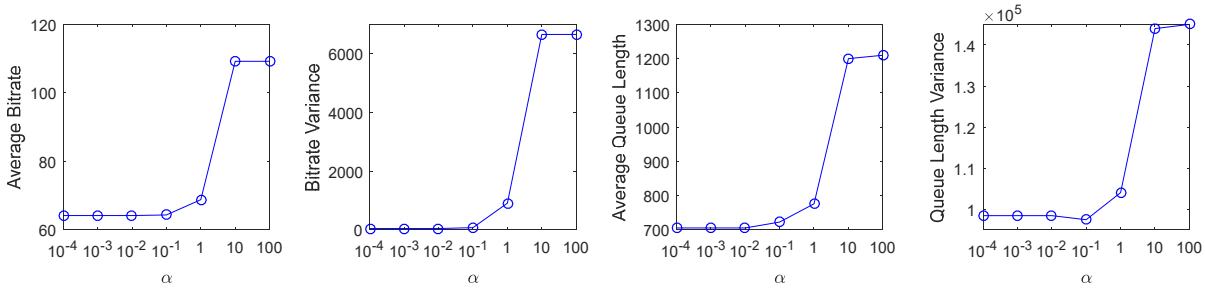
Fig. 8 shows the effect of parameter V variation on the performance of JSDO which is evaluated through average bitrate, bitrate variance, average queue length and queue length variance. The average value and variance of queue length are measured by the summation of averages and variances of queue lengths $R_i\left(t\right)$, $G_i\left(t\right)$, $H_i\left(t\right)$ and $I_i\left(t\right)$. For the simulation, e_{av} is set to 0.2. It can be observed that with the growth in parameter V, average bitrate, bitrate variance, average queue length and queue length variance have a similar growing trend. When $1 \leq V \leq 6$, they increase moderately. They keep stable for $7 \leq V \leq 20$ and increase sharply

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2018.2881095, IEEE Transactions on Multimedia



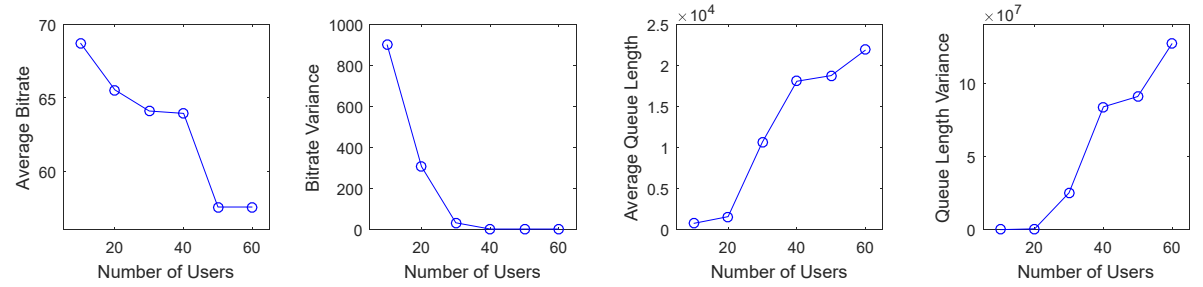
(a) Average Bitrate (b) Bitrate Variance (c) Average Queue Length (d) Queue Length Variance

Fig. 8. Effect of V



(a) Average Bitrate (b) Bitrate Variance (c) Average Queue Length (d) Queue Length Variance

Fig. 9. Effect of α



(a) Average Bitrate (b) Bitrate Variance (c) Average Queue Length (d) Queue Length Variance

Fig. 10. Effect of Number of Users

for $20 \leq V \leq 25$. Then they become stable again when $V \geq 25$. These phenomena indicate that the increase of V can effectively improve the average bitrate. However, improved average bitrate also brings many data packets into HSR networks. It makes queue lengths become much longer and variate frequently. Meanwhile, because video bitrate is determined according to queue lengths, unstable queue lengths will influence the stability of video bitrate. Therefore, as shown in Fig. 8b, the bitrate variance also increases. The above analyses confirm formula (26). As V increases, the second term of formula (26) which is related to video bitrate will become more influential for the value of formula (26). A moderate reduction of video bitrate will reduce the value of formula (26) greatly. Otherwise, video bitrate should be reduced a lot so as to minimize the value of formula (26) and the stability of queues is therefore improved.

Fig. 9 shows the effect of parameter α variation on average bitrate, bitrate variance, average queue length and queue length variance in JSDO. For the simulation, e_{av} is set to 0.2. It can be observed that average bitrate can be improved with the increase of α. Meanwhile, bitrate variance is also increased. Considering Fig. 9c and Fig. 9d, we can find out the increased video bitrates make the queues unstable. Moreover, it is noteworthy that when $\alpha = 1$, average bitrate can be improved without greatly increasing bitrate variance. Therefore, we set α as 1 in our simulations.

Fig. 10 shows the effect of different user numbers on the performance of JSDO. CDR is set as 0.1. With the increase of user number, the average bitrate is reduced. This is because the network transmission capacity allocated to each user is reduced. When user number is larger than 50, average bitrate is not reduced anymore and bitrate variance is reduced. This is because network transmission capacity is not able to support the lowest video quality and bitrate cannot be reduced anymore. At the same time, it can also be observed from Fig. 10c and Fig. 10d that more and more data frames are backlogged and it leads to the increased average value and variance of queue lengths.

Fig. 11 shows the effect of e_{av} on average EER and average delivery capacity over time in JSDO. During the simulation, parameter V was set to 20. It can be observed that both the average delivery capacity and average EER have similar

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2018.2881095, IEEE Transactions on Multimedia

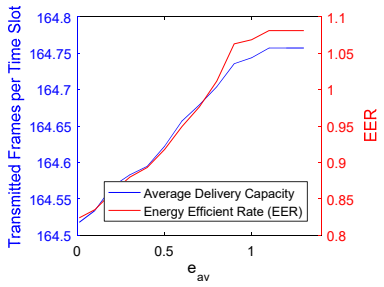


Fig. 11. Effect of e_{av}

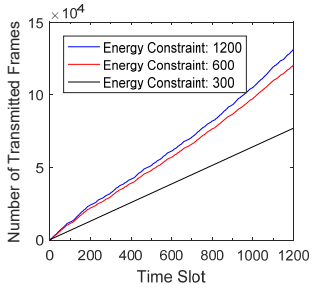


Fig. 12. Effect of Energy Constraint

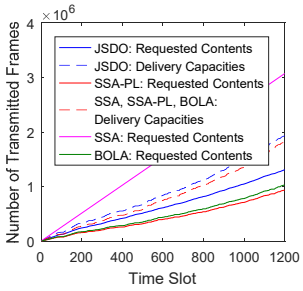


Fig. 13. Requested Video Contents & Delivery Capacity

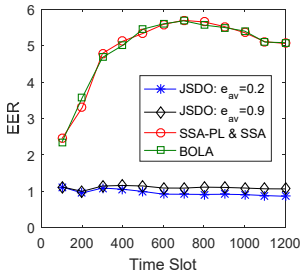


Fig. 14. Comparison of EER

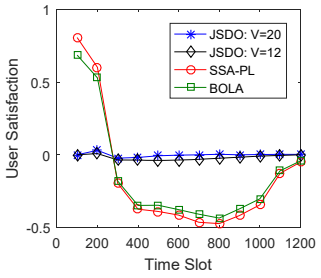


Fig. 15. Comparison of User Satisfaction

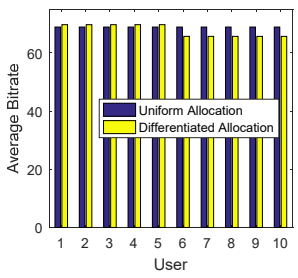


Fig. 16. Fairness of JSDO

growth trend. At first, both of them increase with the growth of e_{av}. This is because eNB can transmit more video content in more time slots with high EERs to mobile users after the increase of e_{av}. It also demonstrates the tradeoff between the average EER and average delivery capacity. That is when increasing the amount of delivered video content, the energy utilization ratio decreases. Then after e_{av} reaches 1.1, both of them do not have an obvious increase anymore. There are two reasons for this phenomenon. First reason is the fact that most EERs of residual time slots are too high to be used for data transmission. Therefore, only a small number of time slots can be added for video content transmission. The other reason is related to the fact that the newly added time slots can not provide large enough delivery capacity. This leads to a small increase in the delivery capacity. Additionally, it is worth noting that when e_{av} increases from 0.01 to 1.3, the average delivery capacity and average EER increase 0.14% and 31.26%, respectively. Therefore, considering the tradeoff between EER and delivery capacity, it would be better to set e_{av} to a small value for the purpose of energy conservation.

Fig. 12 shows the effect of energy constraint on the received frames of video streaming service. It can be observed that with the decrease of energy constraint, the number of received data frames decreases. This indicates the bitrate of the video streaming service also decreases and consequently the video quality is reduced.

Fig. 13 shows the comparison between the amount of requested video streaming and delivery capacity for JSDO, SSA, SSA-PL and BOLA, respectively. During the simulation, e_{av} and V are set to 0.2 and 20, respectively. At first, it can be observed that the total delivery capacity of JSDO is larger than those of SSA, SSA-PL and BOLA. This is because a frame duplication mechanism is employed in JSDO and therefore the frame loss probability of JSDO is lower than those of SSA, SSA-PL and BOLA. In this way, the delivery capacity increases in JSDO. Additionally, for JSDO, SSA-PL and BOLA, it can also be observed that the total amount of requested video contents is always lower than the total delivery capacity. This indicates that the playback buffers of the three mechanisms are never exhausted and rebuffering is avoided. However, in SSA, the total amount of requested video content is always larger than the total delivery capacity. This is because the packet loss problem is not taken into consideration in SSA, and SSA can not provide appropriate bitrate selection. Therefore, serious rebuffering will occur and the quality of video streaming will be reduced.

Fig. 14 shows the comparison of EER between the JSDO, SSA, SSA-PL and BOLA. During the simulation, V is set to 20 and the energy constraint is relaxed. It can be observed that the EER of JSDO is always lower than that of SSA-PL, SSA and BOLA. This is because the delivery control module performs effective delivery scheduling. Through the delivery control algorithm, JSDO is able to avoid transmitting frames when EER is high. Therefore, the EER of JSDO is reduced and energy is conserved. Additionally, for JSDO, the EER when $e_{av} = 0.9$ is higher than the EER for $e_{av} = 0.2$. However, the difference between them becomes greater in time. This indicates that the JSDO mechanism is becoming more effective with increase in the number of iterations.

Fig. 15 illustrates comparative user satisfaction for JSDO, SSA-PL and BOLA in terms of both video quality and video quality variation. It is worth noting that video quality and video quality variation are first normalized before evaluation. In this comparison, the SSA is not taken into consideration since its total requested content is always larger than its delivery capacity and the loss severely affects the quality. Fig. 15 also shows how the user satisfaction when JSDO is employed is greater than that of SSA-PL and BOLA. In particular, in JSDO, before the 500th time slot, the user satisfaction for the case when $V = 20$ is similar with the user satisfaction

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2018.2881095, IEEE Transactions on Multimedia

when $V = 12$. After the 500th time slot, the user satisfaction for $V = 12$ is lower than that for $V = 20$. The figure also illustrates how JSDO is becoming more effective with the increase in the number of iterations.

The fairness of JSDO is evaluated in Fig. 16. For uniform allocation, all users equally share the transmission capacity. In differentiated allocation, 75% of network transmission capacity is allocated to mobile users 1–5 equally and 25% to mobile users 6–10. It can be observed that all mobile users acquire similar video quality in uniform allocation. In differentiated allocation, mobile users 1–5 obtain better service qualities than mobile users 6–10, because different amounts of network transmission capacity are allocated to the two parts of users. It can be concluded that JSDO natively supports quality fairness among mobile users if transmission capacity is allocated fairly.

The impacts of different segment durations and time slot lengths on the performance of JSDO are shown as in Fig. 17. With the increase of segment duration, average bitrate and bitrate variance are increased. Recalling Fig. 8 and Fig. 9, such phenomenon indicates that the queue lengths are increased and network stability is impaired. This is because when segment duration is increased, the size of each segment is also increased. Then a segment is not able to be delivered completely within one time slot. Therefore, video bitrate cannot be selected timely according to the variation of transmission capacity. In this context, we recommend to set the segment duration as small as possible in JSDO. However, when segment duration is small and buffer is large, JSDO will experience long startup delay. To address this problem, a practical approach is to increase the length of time slot. In this way, the appropriate bitrates of videos with long segment durations can be determined and long startup delay can be avoided. It is also noteworthy that increasing the time slot length in JSDO means we need to calculate the network capacity for longer duration. Thus, advanced wireless link quality prediction is also necessary.

The impact of mobility on the performance of JSDO is shown in Fig. 18. It can be observed that with the increase in speed, the average bitrate is reduced and bitrate variance is increased. When the HST speed is relatively low, PLR and transmission capacity are good and a high bitrate can be selected without increasing queue lengths greatly. This indicates a tradeoff between the HST speed and quality of video streaming service. Specifically, if better service quality is required, HST should be slowed down. Otherwise, service quality is lower at high speeds.

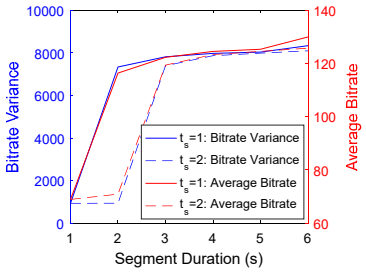


Fig. 17. Impact of Different Segment Durations

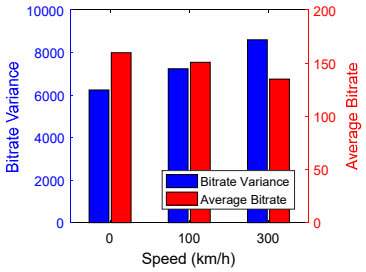


Fig. 18. Impact of Mobility

VIII. Discussions of Transport Protocols

In this section, we focus on analyzing communication protocols for video streaming service in HSR networks. Because paper length is limited and most protocols are designed based on User Datagram Protocol (UDP) and TCP, we mainly analyze the strength and weakness of these two transport protocols.

UDP is always used for network services which put more attention on timeliness over reliability. Considering video services are often delay sensitive and do not require high reliability, a lot of protocols are designed to provide real-time data transmission for video services, such as Quick UDP Internet Connection, Real-time Transport Protocol and Real Time Streaming Protocol. However, currently used video compression techniques depend on the interdependence of a series of frames such as H.264. Meanwhile, UDP is prone to packet loss. In H.264, if I-frames are lost, it is greatly difficult for video players to reconstruct the group of pictures. Considering packet loss is a serious problem in HSR networks, UDP is not suitable for video services in HSR networks.

TCP is designed to ensure service reliability. It initially is not leveraged by video services widely. Afterwards, because HTTP web server is becoming popular. Moreover, TCP traffic can traverse firewalls easily compared with UDP traffic and it works well when Network Address Translations is deployed. Transmitting video contents over HTTP/TCP becomes a dominant approach and DASH technique is widely used. The visual fidelity of videos is therefore guaranteed. However, TCP cannot distinguish packet loss and network congestion which will result in inappropriate adjustment of congestion window. This problem will become much more serious in high packet loss environment, such as HSR networks. Meanwhile, frequent retransmissions and long transmission delays will also be incurred because of high PLR in HSR networks.

Based on the above analysis, one possible approach to ameliorate the negative effect of high PLR in HSR networks is to provide reliable transmission for important video frames such as I-frame. Other frames with less significance in video reconstruction can be transmitted without acknowledgment. Moreover, advanced congestion detection technology should also be researched to avoid inappropriate adjustment of congestion window.

IX. Conclusion

This paper first presents results of field experiments performed to examine the characteristics of HSR networks in

terms of PLR and discusses the packet loss causes, including Doppler shift, frequent handover, and penetration loss. Then a DASH-enabling service optimization problem is formulated for HSR networks and is transformed into a queue stability problem which is highly tractable, scalable and general. By solving this problem, a novel JSDO mechanism is proposed to improve the quality of DASH-based video delivery services with consideration of multiple factors. JSDO includes five modules and three new algorithms for delivery control, bitrate selection and auxiliary data computation. Using these modules and algorithms, JSDO detects the wireless link state and intelligently adjusts multiple factors in an integrated manner. The simulation results employing the field experimental results demonstrate how JSDO achieves better performance in terms of video quality, video quality variation, and energy consumption in comparison with other state-of-the-art solutions.

APPENDIX A
PROOF OF LEMMA 1

Proof. For user $i \in \mathcal{N}$, according to the queue from equation (19), we can derive the inequality $G_i(t+1) - G_i(t) \geq e_i(t) - e_{av}$. Then by summing this inequality over $t \in \{0, \ldots, T\}$, we have the inequality from equation (40):

$$\frac{G_i(T)}{T} - \frac{G_i(0)}{T} + \frac{1}{T}\sum_{t=0}^{T} e_{av} \geq \frac{1}{T}\sum_{t=0}^{T} e_i(t) \tag{40}$$

Considering $G_i(0) = 0$, we take the expectation and take the limit of T to infinity of inequality (40) and then we have the inequality from equation (41).

$$\lim_{T\to\infty} \frac{\mathbb{E}\{G_i(T)\}}{T} + \lim_{T\to\infty} \frac{1}{T}\sum_{t=0}^{T} e_{av} \geq \lim_{T\to\infty} \frac{1}{T}\sum_{t=0}^{T} e_i(t) \tag{41}$$

If the queue $G_i(t)$ is mean rate stable, we have $\lim_{T\to\infty} \frac{\mathbb{E}\{G_i(T)\}}{T} = 0$. Considering inequality (41), the constraint $\overline{e_i(t)} \leq e_{av}$ is satisfied. Thus, **Lemma 1** is proved. □

APPENDIX B
PROOF OF ALGORITHM COMPLEXITY

Proof. In each time slot, the complexity of this algorithm is determined by the number of mobile users. As the number of mobile users is n_m, the algorithm complexity is $\mathcal{O}(n_m)$. □

APPENDIX C
PROOF OF THEOREM 1

Proof. Given equation (26), we have the inequality (42).

$$\begin{aligned}
&\Delta(t) - V\mathbb{E}\left\{\sum_{i=1}^{n_m} \Phi(a_i(t)) + \alpha\sum_{i=1}^{n_m} K(a_i(t), t)\,|\boldsymbol{\eta}(t)\right\} \\
&\leq \mathcal{B} - \sum_{i=1}^{n_m} \mathbb{E}\{R_i(t)[c_i^{\kappa}(t) - a_i(t)]\,|\boldsymbol{\eta}(t)\} \\
&\quad - \sum_{i=1}^{n_m} \mathbb{E}\{G_i(t)[e_{av} - e_i(t)]\,|\boldsymbol{\eta}(t)\} \\
&\quad - \sum_{i=1}^{n_m} \mathbb{E}\{H_i(t)[D_i(t) - c_i^{\kappa}(t)]\,|\boldsymbol{\eta}(t)\} \\
&\quad - \sum_{i=1}^{n_m} \mathbb{E}\{I_i(t)[c_i^{e} - c_i^{\kappa}(t)]\,|\boldsymbol{\eta}(t)\} \\
&\quad - V\mathbb{E}\left\{\sum_{i=1}^{n_m} \Phi(a_i(t)) + \alpha\sum_{i=1}^{n_m} K(a_i(t), t)\,|\boldsymbol{\eta}(t)\right\}
\end{aligned} \tag{42}$$

We further assume there exists a feasible control mechanism δ which achieves the following inequalities:

$$\begin{aligned}
\mathbb{E}\left\{c_i^{\kappa\delta}(t) - a_i^{\delta}(t)\right\} &\leq \varepsilon \\
\mathbb{E}\left\{e_{av} - e_i^{\delta}(t)\right\} &\leq \varepsilon \\
\mathbb{E}\left\{D_i^{\delta}(t) - c_i^{\kappa\delta}(t)\right\} &\leq \varepsilon \\
\mathbb{E}\left\{c_i^{e} - c_i^{\kappa\delta}(t)\right\} &\leq \varepsilon \\
\mathbb{E}\left\{\Phi^{\delta}(a_i(t)) + \alpha K^{\delta}(a_i(t), t)\right\} &\geq \\
\mathbb{E}\{\Phi^*(a_i(t)) + \alpha K^*(a_i(t), t)\} - \varepsilon&
\end{aligned}$$

for any $\varepsilon > 0$. By plugging the above inequalities into the right-hand-side of equation (42), we have:

$$\begin{aligned}
&\Delta(t) - V\mathbb{E}\left\{\sum_{i=1}^{n_m} \Phi(a_i(t)) + \alpha\sum_{i=1}^{n_m} K(a_i(t), t)\,|\boldsymbol{\eta}(t)\right\} \\
&\leq \mathcal{B} - V\mathbb{E}\left\{\sum_{i=1}^{n_m} \Phi^*(a_i(t)) + \alpha\sum_{i=1}^{n_m} K^*(a_i(t), t) - \varepsilon\,|\boldsymbol{\eta}(t)\right\}
\end{aligned} \tag{43}$$

Given $\boldsymbol{\eta}(0) = 0$ and all queues are rate stable, by summing inequality (43) over $t \in \{0, \ldots, T\}$, we have:

$$\begin{aligned}
&\mathbb{E}\left\{\overline{\sum_{i=1}^{n_m} \Phi^*(a_i(t)) + \alpha\sum_{i=1}^{n_m} K^*(a_i(t), t)} - \right. \\
&\left.\overline{\sum_{i=1}^{n_m} \Phi(a_i(t)) + \alpha\sum_{i=1}^{n_m} K(a_i(t), t)}\right\} \leq \tfrac{B+\varepsilon}{V}
\end{aligned} \tag{44}$$

Thus, the first part of **Theorem 1** is proved.

To ensure the stability of queues from equations (19)–(22), we establish the Slater Condition [36], shown as follows:

$$\begin{aligned}
\mathbb{E}\{c_i^{\kappa s}(t)\} &\leq \mathbb{E}\{a_i^{s}(t)\} - \pi \\
\mathbb{E}\{e_{av}\} &\leq \mathbb{E}\{e_i^{s}(t)\} - \pi \\
\mathbb{E}\{D_i^{s}(t)\} &\leq \mathbb{E}\{c_i^{\kappa s}(t)\} - \pi \\
\mathbb{E}\{c_i^{e}\} &\leq \mathbb{E}\{c_i^{\kappa s}(t)\} - \pi \\
\sum_{i=1}^{n_m} \Phi^s(a_i(t)) + \alpha\sum_{i=1}^{n_m} K^s(a_i(t), t) &= \Lambda^s
\end{aligned} \tag{45}$$

where $\pi > 0$ and Λ^s denotes the achieved video streaming service quality under Slater Condition. The Slater Condition basically ensures the input of a queue is always smaller than its output. By plugging the above formulas into the right-hand-side of (42), we have the inequality (46).

$$\begin{aligned}
&\Delta(t) - V\mathbb{E}\left\{\sum_{i=1}^{n_m} \Phi(a_i(t)) + \alpha\sum_{i=1}^{n_m} K(a_i(t), t)\,|\boldsymbol{\eta}(t)\right\} \\
&\leq B - \pi\sum_{i=1}^{n_m} \mathbb{E}\{R_i(t) + G_i(t) + H_i(t) + I_i(t)\,|\boldsymbol{\eta}(t)\} - V\Lambda^s
\end{aligned} \tag{46}$$

Given $\boldsymbol{\eta}(0) = 0$ and all queues are rate stable, by summing inequality (46) over $t \in \{0, \ldots, T\}$, we have inequality (47).

$$\begin{aligned}
&\mathbb{E}\left\{\overline{\sum_{i=1}^{n_m} R_i(t)} + \overline{\sum_{i=1}^{n_m} G_i(t)} + \overline{\sum_{i=1}^{n_m} H_i(t)} + \overline{\sum_{i=1}^{n_m} I_i(t)}\right\} \\
&\leq \tfrac{B}{\pi} + \tfrac{V}{\pi}\mathbb{E}\left\{\overline{\sum_{i=1}^{n_m} \Phi^*(a_i(t)) + \alpha\sum_{i=1}^{n_m} K^*(a_i(t), t)}\right\}
\end{aligned} \tag{47}$$

Finally, the second part of **Theorem 1** is also proved. □

REFERENCES

[1] ChinaDaily, "China's Bullet Trains Make Five Billion Trips," News Report, Aug. 2016.

[2] Cisco, "Cisco Visual Networking Index: Forecast and Methodology, 20162021," Technical Report, Jun. 2017.

[3] H. Yuan, X. Wei, F. Yang, J. Xiao and S. Kwong, "Cooperative Bargaining Game-Based Multiuser Bandwidth Allocation for Dynamic Adaptive Streaming Over HTTP," in IEEE Transactions on Multimedia, vol. 20, no. 1, pp. 183-197, Jan. 2018.

[4] K. T. Bagci, K. E. Sahin and A. M. Tekalp, "Compete or Collaborate: Architectures for Collaborative DASH Video Over Future Networks," in IEEE Transactions on Multimedia, vol. 19, no. 10, pp. 2152-2165, Oct. 2017.

[5] M. Aguayo, L. Bellido, C. M. Lentisco and E. Pastor, "DASH Adaptation Algorithm based on Adaptive Forgetting Factor estimation," in IEEE Transactions on Multimedia, vol. PP, no. 99, pp. 1-1.

[6] G. Gao, Y. Wen and J. Cai, "vCache: Supporting Cost-Efficient Adaptive Bitrate Streaming," in IEEE MultiMedia, vol. 24, no. 3, pp. 19-27, 2017.

[7] M. Hosseini, Y. Jiang, R. R. Berlin, L. Sha and H. Song, "Toward Physiology-Aware DASH: Bandwidth-Compliant Prioritized Clinical Multimedia Communication in Ambulances," in IEEE Transactions on Multimedia, vol. 19, no. 10, pp. 2307-2321, Oct. 2017.

[8] G. Gao, Y. Wen, W. Zhang and H. Hu, "Cost-efficient and QoS-aware content management in media cloud: Implementation and evaluation," 2015 IEEE International Conference on Communications, pp. 6880-6886, 2015.

[9] H. B. Chang, I. Rubin, S. Colonnese, F. Cuomo, O. Hadar, "Joint Adaptive Rate and Scheduling for Unicasting Video Streams in Cellular Wireless Networks," IEEE Transactions on Vehicular Technology , vol.PP, no.99, pp.1-1, Mar. 2017.

[10] G. Gao, Y. Wen and H. Hu, "QDLCoding: QoS-differentiated low-cost video encoding scheme for online video service," IEEE Conference on Computer Communications, 2017, pp. 1-9.

[11] A. Bentaleb, A. C. Begen, R. Zimmermann and S. Harous, "SDNHAS: An SDN-Enabled Architecture to Optimize QoE in HTTP Adaptive Streaming," in IEEE Transactions on Multimedia, vol. 19, no. 10, pp. 2136-2151, Oct. 2017.

[12] K. Kanai et al., "Proactive Content Caching for Mobile Video Utilizing Transportation Systems and Evaluation Through Field Experiments," IEEE Journal on Selected Areas in Communications, vol. 34, no. 8, pp. 2102-2114, Aug. 2016.

[13] Y. Hu, Z. Chang, H. Li, T. Ristaniemi, Z. Han, "Service Provisioning and User Association for Heterogeneous Wireless Railway Networks," IEEE Transactions on Communications, vol. 65, no. 7, pp. 3066-3078, Jul. 2017.

[14] S. Xu, G. Zhu, C. Shen, B. Ai, "A QoS-Aware Scheduling Algorithm for High-Speed Railway Communication System," 2014 IEEE International Conference on Communications, pp. 2855-2860, Jun. 2014.

[15] Y. Lei et al., "Delay-Aware Dynamic Resource Allocation in High-Speed Railway Networks," 2016 IEEE 83rd Vehicular Technology Conference, pp. 1-5, May 2016.

[16] T. Li, K. Xiong, P. Fan, K. B. Letaief, "Service-Oriented Power Allocation for High-Speed Railway Wireless Communications," IEEE Access, vol. 5, pp. 8343-8356, May 2017.

[17] H. Ghazzai, T. Bouchoucha, A. Alsharoa, E. Yaacoub, M. S. Alouini, T. Y. Al-Naffouri, "Transmit Power Minimization and Base Station Planning for High-Speed Trains With Multiple Moving Relays in OFDMA Systems," IEEE Transactions on Vehicular Technology, vol. 66, no. 1, pp. 175-187, Jan. 2017.

[18] S. Xu, G. Zhu, C. Shen, Y. Lei, Z. Zhong, "Analysis and Optimization of Resource Control in High-Speed Railway Wireless Networks," Mathematical Problems in Engineering, vol. 2014, Apr. 2014.

[19] Q. Xu, H. Ji, X. Li and H. Zhang, "Admission Control Scheme for Service Dropping Performance Improvement in High-Speed Railway Communication Systems," IEEE Transactions on Vehicular Technology, vol. 65, no. 7, pp. 5251-5263, Jul. 2016.

[20] N. Sun, Y. Zhao, L. Sun, Q. Wu, "Distributed and Dynamic Resource Management for Wireless Service Delivery to High-Speed Trains," IEEE Access, vol. 5, pp. 620-632, 2017.

[21] C. Xu, P. Zhang, S. Jia, M. Wang, G.-M. Muntean, "Video Streaming in Content-Centric Mobile Networks: Challenges and Solutions," IEEE Wireless Communications, vol. 24, no. 5, pp. 157-165, Oct. 2017.

[22] C. Xu, W. Quan, A. V. Vasilakos, H. Zhang, G.-M. Muntean, "Information-Centric Cost-Efficient Optimization for Multimedia Content Delivery in Mobile Vehicular Networks," vol. 99, pp. 93-106, Computer Communications, Aug. 2016.

[23] D. Wu, L. Zhou and Y. Cai, "Social-Aware Rate Based Content Sharing Mode Selection for D2D Content Sharing Scenarios," in IEEE Transactions on Multimedia, vol. 19, no. 11, pp. 2571-2582, Nov. 2017.

[24] Y. Jin, Y. Wen and K. Guan, "Toward Cost-Efficient Content Placement in Media Cloud: Modeling and Analysis," in IEEE Transactions on Multimedia, vol. 18, no. 5, pp. 807-819, May 2016.

[25] W. Zhang, Y. Wen, Z. Chen and A. Khisti, "QoE-Driven Cache Management for HTTP Adaptive Bit Rate Streaming Over Wireless Networks," IEEE Transactions on Multimedia, vol. 15, no. 6, pp. 1431-1445, Oct. 2013.

[26] Y. Zhou, B. Ai, "Handover Scheme and Algorithms of High-Speed Mobile Environment: A Survey," Comput. Commun., vol. 47, pp. 1-5, Jul. 2014.

[27] W. Ali, J. Wang, H. Zhu and J. Wang, "Distributed antenna system based frequency switch scheme evaluation for high-speed railways," 2017 IEEE International Conference on Communications, Paris, 2017, pp. 1-6.

[28] J. Wang, H. Zhu, N. Gomes, "Distributed Antenna System for Mobile Communications in High Speed Trains," IEEE Journal on Selected Areas in Communications, vol. 30, no. 4, pp, 675-683, May 2012.

[29] F. Chiti et al., "Evaluation of the Resequencing Delay for Selective Repeat ARQ in TDD-Based Wireless Communication Systems," IEEE Transactions on Vehicular Technology, vol. 63, no. 5, pp. 2450-2455, Jun. 2014.

[30] Y. Hayashi, J. Suzuiki, M. Kan, "Delay-Bounded Transport using Rateless Codes for I/O Bus over Wireless Ethernet," 2016 IEEE International Conference on Communications, Jul. 2016.

[31] Z. Jiang, C. Xu, J. Guan et al., "Loss-Aware Adaptive Scalable Transmission in Wireless High-Speed Railway Networks," 2017 IEEE International Conference on Communications, May 2017.

[32] I. Sodagar, "The MPEG-DASH Standard for Multimedia Streaming Over the Internet," IEEE MultiMedia, vol. 18, no. 4, pp. 62-67, Apr. 2011.

[33] M. Hosseini, J. Peters, S. Shirmohammadi, "Energy-budget-compliant Adaptive 3D Texture Streaming in Mobile Games" Proceedings of the 4th ACM Multimedia Systems Conference, Mar. 2013.

[34] J. Qiao, Y. He and X. S. Shen, "Proactive Caching for Mobile Video Streaming in Millimeter Wave 5G Networks," IEEE Transactions on Wireless Communications, vol. 15, no. 10, pp. 7187-7198, Oct. 2016.

[35] X. Yin, A. Jindal, V. Sekar, and B. Sinopoli, "A Control-Theoretic Approach for Dynamic Adaptive Video Streaming over HTTP," 2015 ACM Conference on Special Interest Group on Data Communication, pp. 325-338. Aug. 2015.

[36] M. J. Neely, *Stochastic Network Optimization with Application to Communication and Queueing Systems*, Morgan & Claypool, 2010.

[37] C. Xu, Z. Li, J. Li, H. Zhang, G. M. Muntean, "Cross-Layer Fairness-Driven Concurrent Multipath Video Delivery Over Heterogeneous Wireless Networks," IEEE Transactions on Circuits and Systems for Video Technology, vol. 25, no. 7, pp. 1175-1189, Jul. 2015.

[38] Y. J. Chang, H. Jung, S. Cho and M. A. Weitnauer, "Network time synchronization for large multi-hop sensor networks using the cooperative analog-and-digital (CANDI) protocol," 2014 IEEE Wireless Communications and Networking Conference (WCNC), Istanbul, 2014, pp. 1950-1955.

[39] H. Song, X. Fang and Y. Fang, "Millimeter-Wave Network Architectures for Future High-Speed Railway Communications: Challenges and Solutions," in IEEE Wireless Communications, vol. 23, no. 6, pp. 114-122, December 2016.

[40] Q. M. Qadir, A. A. Kist and Z. Zhang, "A Novel Traffic Rate Measurement Algorithm for Quality of Experience-Aware Video Admission Control," in IEEE Transactions on Multimedia, vol. 17, no. 5, pp. 711-722, May 2015.

[41] K. Spiteri, R. Urgaonkar and R. K. Sitaraman, "BOLA: Near-optimal bitrate adaptation for online videos," IEEE INFOCOM 2016 - The 35th Annual IEEE International Conference on Computer Communications, San Francisco, CA, 2016, pp. 1-9.