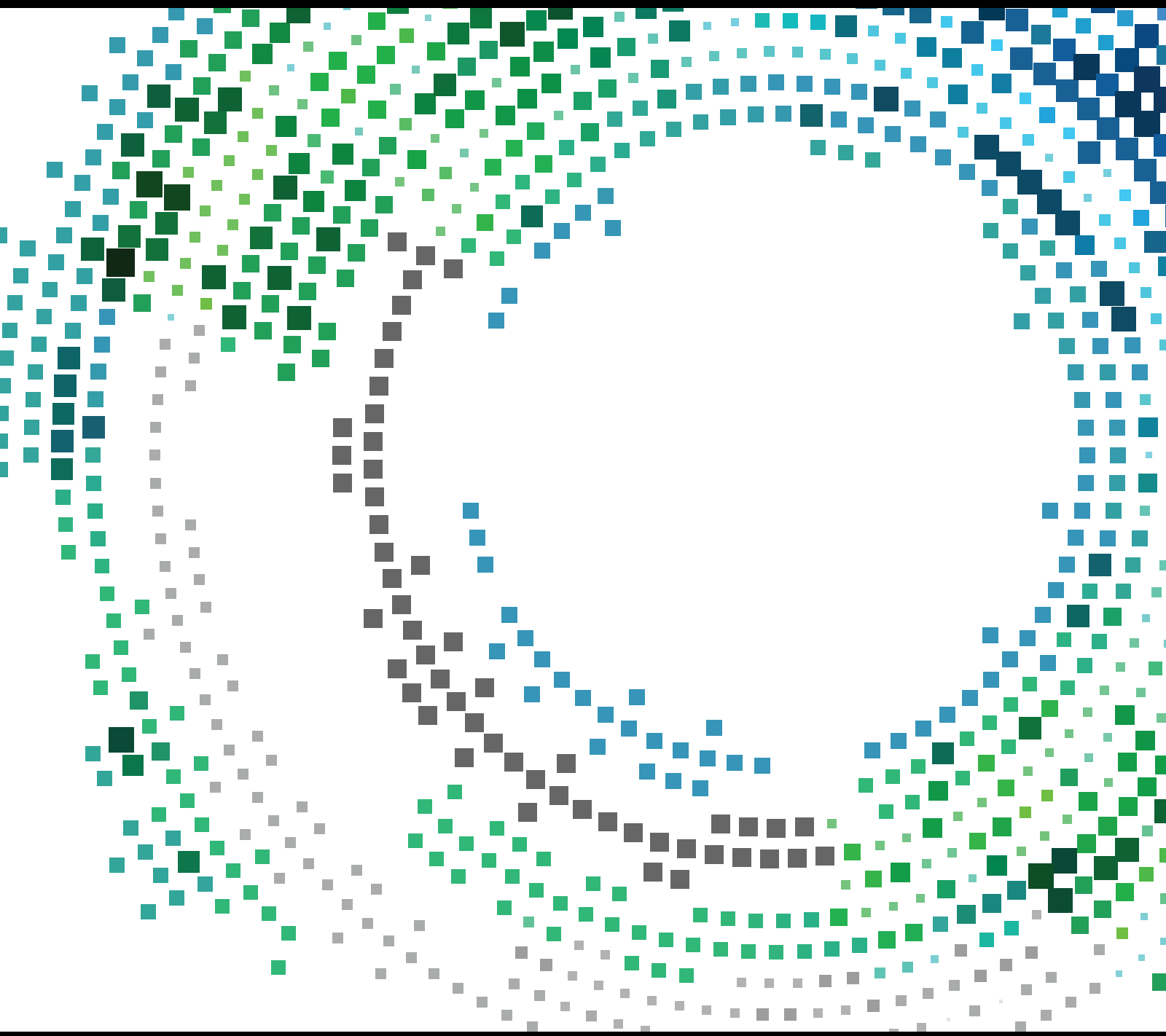
Mobile Information Systems

Advances in Mobile Video Networking

Guest Editors: Qi Wang, Guojun Wang, Christos Grecos, and Ansgar Gerlicher



Advances in Mobile Video Networking

Mobile Information Systems

Advances in Mobile Video Networking

Guest Editors: Qi Wang, Guojun Wang, Christos Grecos, and Ansgar Gerlicher

Contents

Hindawi Publishing Corporation
Mobile Information Systems
Volume 2016, Article ID 9706378, 2 pages
http://dx.doi.org/10.1155/2016/9706378

Editorial

Advances in Mobile Video Networking

Qi Wang,[1] Guojun Wang,[2] Christos Grecos,[3] and Ansgar Gerlicher[4]

[1]*University of the West of Scotland, Paisley PA1 2BE, UK*
[2]*Guangzhou University, Guangzhou 510006, China*
[3]*Central Washington University, 400 E. University Way, Ellensburg, WA 98926, USA*
[4]*Stuttgart Media University, Hochschule der Medien Nobelstraße 10, 70569 Stuttgart, Germany*

Correspondence should be addressed to Qi Wang; qi.wang@uws.ac.uk

Received 21 July 2016; Accepted 21 July 2016

Video applications have been increasingly dominating worldwide mobile network traffic in the last years. The growing popularity of smart phones and other mobile devices, on-demand and surveillance video services, multimedia social networking, the latest advances in video coding and transmission, and the significant increase in mobile network capacity have all contributed to this global phenomenon. Both challenges and opportunities have emerged in multiple research disciplines involving video signal processing, 4G such as LTE (Long Term Evolution) and even 5G mobile networks, mobile cloud computing, Big Visual Data, mobile user experience, and so on. Research in these leading-edge areas has been gaining accelerated global momentum in both developing and developed economies.

Therefore, it is high time to call for contributions to further stimulate the continuing efforts to resolve the intrinsic conflicts between resource-demanding video applications and resource-constrained mobile networks, optimise video networking performance in mobile networks and achieve improved Quality of Experience (QoE) for end users, and leverage cloud computing and other cutting-edge technologies and standards in supporting adaptive, secure, energy-efficient, and reliable video delivery in various mobile systems. This special issue attracted numerous submissions, out of which five papers have been accepted to be published. These accepted papers are expected to highlight some of the important aspects outlined above. It is the pleasure for the guest editorial team to introduce these papers as follows.

The first paper entitled "Energy-Efficient Transmission Strategy by Using Optimal Stopping Approach for Mobile Networks" by Y. Peng et al. contributes to greener video delivery in mobile networks. An optimised distributed opportunistic scheduling algorithm is proposed to maximise the usage of good quality slots of the time-varying wireless transmission channels to improve video data delivery ratio and thus energy efficiency, whilst taking into account the delay constraint for video applications.

The second paper "Network-Aware Reference Frame Control for Error-Resilient H.264/AVC Video Streaming Service" by H.-S. Gang et al. focuses on enhancing the reliability of video streaming in error-prone networks such as mobile/wireless networks whilst making a trade-off between error resilience and coding efficiency. In particular, error propagation across video frames caused by video packet loss or corruption is effectively mitigated, through controlling the number of reference video frames based on the awareness of network channel status.

In the third paper, "A Framework for QoE-Aware 3D Video Streaming Optimisation over Wireless Networks" by I. Politis et al., the challenging topic of 3D video streaming in LTE networks with good perceived quality is addressed. To model the QoE of end users, machine learning techniques are utilised and parameters from multiple layers are monitored and collected. In addition, a media-aware proxy is introduced to the proposed system, integrated with the QoE controller, to achieve QoE-aware, optimised 3D video transmission.

The fourth paper, "A Low Energy Consumption Storage Method for Cloud Video Surveillance Data Based on SLA Classification" by Y. Xiong et al., also deals with the energy efficiency in a video system although the main concern is on the storage side other than the transmission side as in the first paper. The targeted application is Cloud Video Surveillance,

and the design objective is to reduce the power consumption without compromising the Service Level Agreement (SLA) for users to access the stored surveillance video data in the cloud on demand in a timely fashion. Based on classifying the access time periods and thus virtual machines and storage nodes, the proposed system reduces overall power consumption in the data centre by placing selected nodes into an energy-saving mode.

In the final paper "A Novel Adaptation Method for HTTP Streaming of VBR Videos over Mobile Networks," H. T. Le et al. investigate adaptive on-demand transmission of variable bitrate (VBR) videos through HTTP streaming. Their adaptation scheme is designed to be tolerant to large variations of video bitrate, featured with a buffer-based algorithm that considers smoothed throughput estimate, representative bitrate, and instant bitrate.

Acknowledgments

The guest editorial team would like to thank all the authors for submitting their manuscripts to this special issue and all the invited reviewers for their time and constructive feedback. Moreover, the team appreciate the staff of this journal for their continuous support and patience in creating this special issue.

Qi Wang
Guojun Wang
Christos Grecos
Ansgar Gerlicher

Hindawi Publishing Corporation
Mobile Information Systems
Volume 2016, Article ID 2920850, 11 pages
http://dx.doi.org/10.1155/2016/2920850

Research Article

A Novel Adaptation Method for HTTP Streaming of VBR Videos over Mobile Networks

Hung T. Le,[1] Hai N. Nguyen,[2] Nam Pham Ngoc,[2] Anh T. Pham,[1] and Truong Cong Thang[1]

[1]*University of Aizu, Tsuruga, Ikki-machi, Aizu-Wakamatsu 965-8580, Japan*
[2]*Hanoi University of Science and Technology, 1 Dai Co Viet, Hanoi 10000, Vietnam*

Correspondence should be addressed to Truong Cong Thang; thang@u-aizu.ac.jp

Received 16 November 2015; Accepted 12 June 2016

Academic Editor: Qi Wang

Recently, HTTP streaming has become very popular for delivering videos over the Internet. For adaptivity, a provider should generate multiple versions of a video as well as the related metadata. Various adaptation methods have been proposed to support a streaming client in coping with strong bandwidth variations. However, most of existing methods target at constant bitrate (CBR) videos only. In this paper, we present a new method for quality adaptation in on-demand streaming of variable bitrate (VBR) videos. To cope with strong variations of VBR bitrate, we use a local average bitrate as the representative bitrate of a version. A buffer-based algorithm is then proposed to conservatively adapt video quality. Through experiments in the mobile streaming context, we show that our method can provide quality stability as well as buffer stability even under very strong variations of bandwidth and video bitrates.

1. Introduction

Recently, video streaming has rapidly gained popularity over the mobile Internet. It is predicted that global video traffic will reach 80 percent of total consumer Internet traffic in 2019 [1]. Besides, HTTP protocol has become a cost-effective solution thanks to the abundance of Web platform and broadband connections [2, 3]. Furthermore, for interoperability of HTTP streaming in the industry, ISO/IEC MPEG has developed "Dynamic Adaptive Streaming over HTTP" (DASH) [4] as the first standard for video streaming over HTTP.

Due to the heterogeneity of communication networks nowadays, adaptivity is a principal requirement for any streaming clients. In DASH, multiple versions of an original video as well as related metadata (e.g., describing bitrates and resolutions) are generated and stored at servers [4, 5]. Based on the information of metadata as well as the terminal and networks, a client can adaptively decide which/when data parts should be downloaded. Currently, the question of video adaptation for HTTP streaming is still an open issue [6].

Various adaptation methods for HTTP streaming have been proposed over the past few years [7–17]. These methods can be roughly classified into two groups, throughput-based and buffer-based; each has its own strengths and weaknesses [18]. Throughput-based methods decide the version based on the estimated throughput only, while buffer-based methods mainly use buffer characteristics as references for making decisions. Throughput-based methods are usually able to react quickly to throughput variations; however, the streaming quality may be unstable [7]. Meanwhile, buffer-based methods try to maintain a smooth video stream, but may cause sudden changes in video quality when the buffer level drastically drops [8–11]. It should be noted that as the data size of each segment varies according to the requested version, the buffer size and buffer level should be measured in seconds of media.

So far, existing adaptation methods have mostly focused on CBR (constant bitrate) videos. The research on HTTP streaming for VBR (variable bitrate) videos is still limited. The problem with VBR videos is that even though the throughput is stable, strong fluctuations of video bitrate may result in buffer underflows [12]. Our previous work in [12] is the first study on HTTP streaming that supports VBR videos by estimating both the instant bitrate and the instant throughput. In the context of managed IPTV networks, where

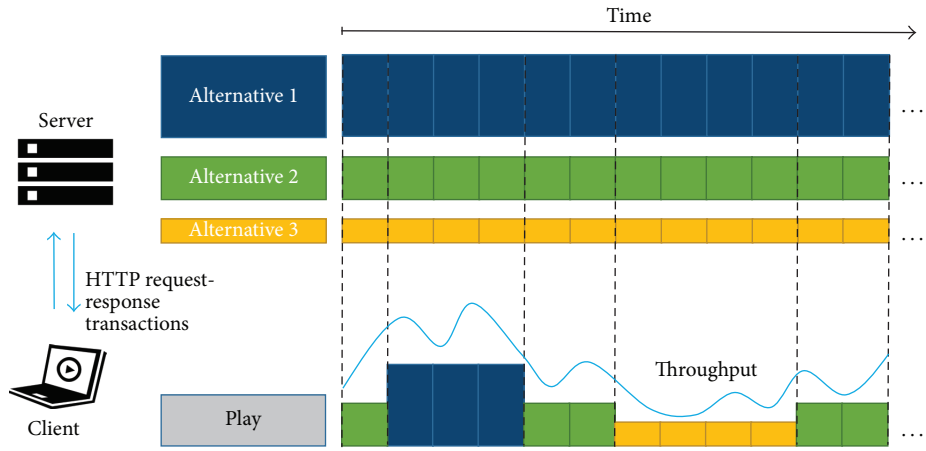


Figure 1: Media delivery hierarchy in HTTP streaming.

the bandwidth is allocated in advance, the delay-quality tradeoff of a VBR video is optimally achieved by replacing some high-bitrate segments with low-bitrate segments [13]. In [9], a buffer-based adaptation method is proposed for VBR video streaming by using a partial-linear buffer prediction model along with a strategy to select versions in different buffer ranges. However, this method does not consider bitrate values, and so it cannot avoid sudden changes of quality when video bitrate and throughput are drastically varying.

In this paper, we present a novel adaptation method which can effectively support VBR videos. By extending our preliminary work in [14], the proposed method can cope with variations of throughput as well as video bitrate. Our method especially takes into account the moving average of bandwidth and video bitrate to provide stable streaming quality without sudden changes. The experimental results in the mobile streaming context show that our approach can provide consistent VBR video streaming with smooth video quality and stable buffer level. To the best of our knowledge, this is the first method that can provide smooth version transitions for VBR video streaming over HTTP. It should be noted that our method does not require the exact values of video segment bitrates to make decisions.

The organization of this paper is as follows. Section 2 presents an overview of HTTP streaming and the related work. The principles of our method as well as the algorithm description are presented in detail in Section 3. Section 4 provides our experimental results and discussions. Finally, conclusion and direction for future work are given in Section 5.

2. Related Work

The general architecture of an HTTP streaming system consists of servers, delivery networks, and clients [3, 4]. In MPEG DASH terminology, to support adaptivity, a video is encoded in multiple versions (also called alternatives or representations), each of which is further divided into short segments. Video segments together with metadata are hosted at a server and will be requested by the client. In most cases, for each request from the client, the server will send one segment. Therefore, a video will be delivered by a sequence of HTTP request-response transactions. The version (low or high) of a requested segment is decided based on the metadata and status of terminals/networks. Figure 1 depicts an illustration of media delivery in DASH. More information about the HTTP streaming structure as well as DASH concepts could be found in [2, 4].

In general, an adaptation method needs to answer two key questions: (1) should the current version be maintained? and (2) if not, which version should be switched to? As already mentioned, existing methods can be divided into a throughput-based group and a buffer-based group. Throughput-based methods are different in the ways they estimate or use the throughput. In terms of throughput estimation, the simplest way is to use the measured throughput right after having fully received a segment (called instant throughput) as the throughput estimate of the next segment. Another approach is to use a smoothed throughput measure [3, 10] to avoid short-term fluctuations, which are a drawback of using instant throughput. However, this may cause late reaction of the client to large throughput drops. In [3] we propose a throughput estimation method that has the advantages of both instant throughput and smoothed throughput. Furthermore, other studies also present ways to obtain the estimated throughput based on sampled throughput values and RTT [12], probing or stored data (lookup table) [15]. Once the client obtains the estimated throughput, the version can be decided in many ways. A simple solution is using a safety margin to compute an appropriate version for the next segment [3]. In [7], the version is controlled by a TCP-like mechanism where a measure proportional to the instant throughput is used as the key input.

Buffer-based methods, which mainly use buffer characteristics to decide the video versions, may take into account a throughput estimate as well. A popular strategy of these methods is dividing the buffer into multiple ranges with buffer thresholds $\beta_1, \beta_2, \beta_3, \beta_{\max}$ ($0 < \beta_1 < \beta_2 < \beta_3 \leq \beta_{\max}$) [8, 10, 11]. When the buffer level stays in different ranges,

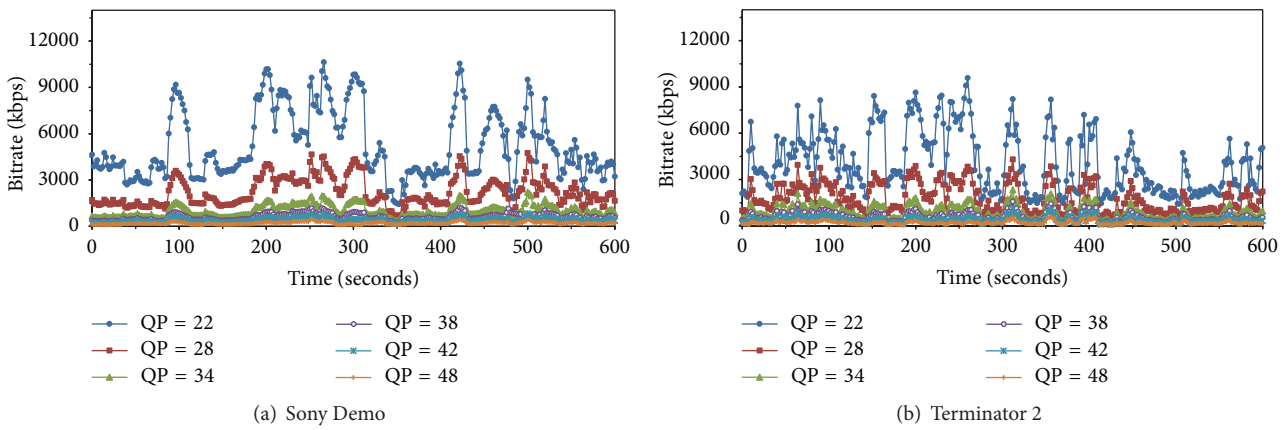


Figure 2: Bitrates of the versions of two test videos [19].

different actions are applied. For instance, methods of [8, 10] try to maintain the current version in a specific buffer range (e.g., $\beta_2 \sim \beta_3$ for the method of [8] and $\beta_2 \sim \beta_{\max}$ for the method of [10]) and dynamically switch up/down the quality in other buffer ranges. Meanwhile, the method of [11] chooses video versions following the variations of instant throughput (also with the use of upscaling and downscaling factors) based on different buffer ranges. In [16] we introduce a trellis-based method that represents all possible changes of the versions and corresponding buffer levels in the near future. Thus, this approach can make good decisions on the bitrates of some future segments. In [17], the buffer level deviation and instant throughput are employed as inputs of a proportional-integral controller for adaptation.

Yet, most of the existing methods have been developed only for the context of CBR videos, where the bitrate of a version is constant. Compared to CBR videos, videos encoded in VBR mode have important advantages in terms of quality and network resource usage [20]. However, the variations of video bitrate over time, together with throughput fluctuations, result in a big challenge for HTTP adaptive streaming [12]. Two examples of how bitrates of different versions vary, especially in some scene changes, are illustrated in Figure 2. Detailed information of these videos will be described in Section 4.

Our previous work in [12] is the first study on VBR video streaming over HTTP. Besides throughput estimation, this method also considers the estimation of the instant video bitrate, which can be divided into (1) intrastream estimation and (2) interstream estimation. The former estimates the bitrates of segments within a version, while the latter estimates the bitrates of segments across different versions. This method can provide a very stable buffer and, moreover, can support a CBR-like streaming service from VBR videos. In [9], a buffer-based adaptation method for VBR videos is proposed, where the buffer is divided into multiple ranges. In order not to take into account varying video bitrates, this method uses a partial-linear trend prediction of the buffer level for choosing versions in different buffer ranges. If no significant change of buffer level is estimated, the client will maintain the current version for the next segment. However, this method still causes sudden version changes if the actual buffer level declines drastically.

In this paper, we propose a novel adaptation method that can effectively support VBR videos in on-demand streaming. The distinguishing features of our method include the following:

(i) To cope with strong variations of video bitrates, we propose using a local average bitrate as the (moving) representative bitrate of a version. Since this representative bitrate is stable in short term, it helps the client clearly differentiate between the available versions and make a good version selection at each time instance.

(ii) Because the client does not have enough information about the segment bitrates of all versions, we provide an algorithm to obtain an estimated representative bitrate of each version.

(iii) Regarding the first key question, quality stability is effectively maintained by (1) using a smoothed throughput estimate when the buffer is not in danger, (2) using the representative bitrate, and (3) being conservative in quality switching.

(iv) Regarding the second key question, smooth transitions of quality are supported by avoiding jumping simply to the lowest version in panic case and by early switching down when there is a throughput-bitrate mismatch.

3. The Proposed Adaptation Method

In this section, we present a new buffer-based quality adaptation method for VBR video streaming. Some notations along with their definitions used in the paper are provided in Notations.

3.1. Handling Throughput and Bitrate Fluctuations. Generally, based on the measured throughput and the video bitrate, the client should choose an appropriate version for the next segment. Suppose that after receiving the current (or last)

segment i of version I_i, the client measures the bitrate B_{i,I_i} and throughput T_i of this segment. Now the client will decide the version I_{i+1} for the next segment $i+1$. Our proposed method will also leverage the client buffer to cope with the fluctuations of both the throughput and the video bitrate. Depending on the current buffer level β_{cur}, the client will decide whether the version should be increased, decreased, or maintained.

For the version selection of the next segment, it is necessary to estimate the throughput based on the throughput history of received segments. To avoid the effects of short-term fluctuations of instant throughput, we use a smoothed throughput measure T_i^s [3, 10] as the throughput estimate T_{i+1}^{est} for the next segment $i+1$:

$$T_{i+1}^{\text{est}} = T_i^s = \begin{cases} (1-\delta) \times T_{i-1}^s + \delta \times T_i & \text{if } i > 0; \\ T_i & \text{if } i = 0, \end{cases} \quad (1)$$

where δ is a weighting value, which is set to 0.1 in this paper.

As video bitrate is highly fluctuating, we propose using a representative bitrate for each version, which can differentiate the available versions and can also be appropriate for maintaining quality stablity. Denote $B_{i,k}^{\text{rep}}$ the representative bitrate for version k at segment index i. In our method, $B_{i,k}^{\text{rep}}$ is calculated as the average bitrate of N recent segments of version k.

The problem is that the client only knows the bitrates of the received segments, which may belong to different versions. So, after receiving each segment i, we will estimate the segment bitrates of other versions with the same index i. This is enabled by the bitrate estimation method proposed in our previous study [12], where the segment bitrates of other versions are estimated from the bitrate of the received segment using the interstream bitrate prediction. Specifically, the estimated bitrate $B_{i,k}^e$ of version k can be calculated from the (actual) bitrate $B_{i,n}^o$ of the received segment i with the selected version n as follows:

$$B_{i,k}^e = \theta \times B_{i,n}^o \times 2^{(\text{QP}_n - \text{QP}_k)/6}, \quad (2)$$

where QP_k and QP_n are the quantization parameter (QP) values of the versions and $\theta = 1.05$ is an empirical factor used as the compensation for the approximation error of the model [12]. In our notation (see Notations), bitrate $B_{i,k}$ can be either the actual bitrate $B_{i,k}^o$ or the estimated one $B_{i,k}^e$. Once the bitrates of all segments are obtained, the representative bitrate of each version at segment index i is calculated as the average of the bitrates of segment i and $N-1$ previous segments in that version. The algorithm to calculate the representative bitrates is provided in Algorithm 1. In this algorithm, the current representative bitrate is actually computed using the previous representative bitrate. Also, it should be noted that, when $i < N$, N will be set to i. In Section 4, we will investigate how different values of N affect the performance of our method.

3.2. Adaptation Algorithm. Our algorithm will address the two key questions above, so as to avoid rebuffering and to reduce the number and the degree of (version) switches. Based on the current buffer level of the client, we define four possible cases in a streaming session, which are uptrend, stable, downtrend, and panic cases. In these four cases, the client will be likely to switch up, maintain, switch down, or aggressively decrease the version. For this purpose, our method divides the buffer into three ranges with thresholds $\beta_{\min}$ and β_i^{th} ($\beta_{\min} < \beta_i^{\text{th}} < \beta_{\max}$). Here $\beta_{\max}$ is the buffer size and also the target buffer level of the adaptation method.

When the current buffer level exceeds $\beta_{\max}$, the uptrend case is activated (for the reason why the buffer level could be higher than $\beta_{\max}$, please refer to our previous work [18]). However, it is not good if the client frequently goes back and forth between the uptrend case and downtrend case. To avoid this fluctuation, our method switches up to the next higher version ($I_i + 1$) only if the representative bitrate B_{i,I_i+1}^{rep} of that version is smaller than the throughput estimate of the next segment (i.e., $B_{i,I_i+1}^{\text{rep}} < T_{i+1}^{\text{est}}$); otherwise, the client will maintain the current version.

The stable case is determined by the condition $\beta_i^{\text{th}} \le \beta_{\text{cur}} < \beta_{\max}$. In this case, the buffer level is judged as in a very safe condition, so no change in quality is needed. The client just maintains the current version to avoid unnecessary switches.

The downtrend case is activated when $\beta_{\min} \le \beta_{\text{cur}} < \beta_i^{\text{th}}$. In this case, the client needs to carefully decide the requested version in order to avoid buffer underflows as well as sudden quality changes when the throughput and/or the video bitrate change drastically. In general, the version should be decreased; however, it is unnecessary to switch down always when the buffer level is in this range. In this process, we define the target bitrate for the next segment $i+1$ as the highest representative bitrate, which is lower than the throughput estimate: $B_{i+1}^{\text{tar}} = \max\{B_{i,k}^{\text{rep}} \mid B_{i,k}^{\text{rep}} < T_{i+1}^{\text{est}}\}$. If the instant bitrate and the representative bitrate do not exceed the target bitrate ($B_{i,I_i} \le B_{i+1}^{\text{tar}}$ and $B_{i,I_i}^{\text{rep}} \le B_{i+1}^{\text{tar}}$), the current version will be maintained. Otherwise, the client will switch down to the next lower version.

Input: N, $B_{i-1,k}^{\text{rep}}|_{1\le k\le V}$, $B_{i-j,k}|_{0\le j<N,1\le k\le V}$
Output: $B_{i,k}^{\text{rep}}|_{1\le k\le V}$
for $k \leftarrow 1, 2, \ldots, V$ **do**
  // Check if bitrate is the original bitrate
  if $k = I_i$ **then**
    $B_{i,k} \leftarrow B_{i,k}^o$;
  // Otherwise the bitrate is the estimated bitrate
  else
    Estimate $B_{i,k}^e$ by (2);
    $B_{i,k} \leftarrow B_{i,k}^e$;
  end
  // Compute $B_{i,k}^{\text{rep}}$
  $B_{i,k}^{\text{rep}} \leftarrow B_{i-1,k}^{\text{rep}} + (B_{i,k} - B_{i-N,k})/N$;
end

Algorithm 1: Representative bitrate computation (after receiving the last segment i and for all video versions).

Input: T_i, β_{cur}, I_i, B_{i,I_i}, B_{i,I_i}^{rep}, T_{i+1}^{est}
Output: I_{i+1}
// *Uptrend case*
(1) **if** $\beta_{\text{cur}} > \beta_{\max}$ **then**
  if $B_{i,I_i}^{\text{rep}} < T_{i+1}^{\text{est}}$ **then**
    $I_{i+1} \leftarrow I_i + 1$;
  else
    $I_{i+1} \leftarrow I_i$;
  end
// *Stable case*
(2) **else if** $\beta \in [\beta_i^{\text{th}}, \beta_{\max})$ **then**
  $I_{i+1} \leftarrow I_i$;
// *Downtrend case*
(3) **else if** $\beta \in [\beta_{\min}, \beta_i^{\text{th}})$ **then**
  $B_{i+1}^{\text{tar}} \leftarrow \max\{B_{i,k}^{\text{rep}} \mid B_{i,k}^{\text{rep}} < T_{i+1}^{\text{est}}\}$;
  if $B_{i,I_i} \le B_{i+1}^{\text{tar}}$ *and* $B_{i,I_i}^{\text{rep}} \le B_{i+1}^{\text{tar}}$ **then**
    $I_{i+1} \leftarrow I_i$;
  else
    $I_{i+1} \leftarrow I_i - 1$;
  end
// *Panic case*
(4) **else**
  Select I_{i+1} based on (4);
end

Algorithm 2: Adaptation algorithm for VBR video streaming.

We can see that sometimes the instant throughput might be much smaller than the instant bitrate. Even though the buffer is not in danger yet, the downtrend case should be activated earlier to avoid having to quickly reduce the version in the future. This is enabled by increasing the value of β_i^{th}. In our method, β_i^{th} is controlled using a logistic function as follows:

$$\beta_i^{\text{th}} = \beta_{\max} - \frac{1}{1+e^{\sigma}} \times (\beta_{\max} - \beta_{\min}), \quad \text{where } \sigma = 1 - \frac{T_i}{B_{i,I_i}}. \tag{3}$$

In this way, the larger the mismatch between T_i and B_{i,I_i} becomes, the higher the value of β_i^{th} will be, while still satisfying the condition $\beta_{\min} \le \beta_i^{\text{th}} < \beta_{\max}$.

The final case, called the panic case, is when the buffer level is in danger; that is, $\beta_{\text{cur}} < \beta_{\min}$. To ensure that the buffer will not be empty, the client will aggressively switch down the version. Yet, instead of jumping directly to the lowest version, the client employs the method of our previous work [12] in this case. Specifically, the client will choose a version of which the instant bitrate is the highest but still lower than the instant throughput:

$$I_{i+1} = \arg\max_{1 \le k \le V} \{B_{i,k} \mid B_{i,k} < T_i\}. \tag{4}$$

The algorithm of our method is summarized by pseudocode as in Algorithm 2.

Generally, it can be seen that our method tries to provide a smooth video stream by two techniques. First, it is somewhat conservative in increasing the quality because that is allowed only when the buffer is full. In addition, the selected version is constrained by the long-term values of throughput and bitrate. Second, the downtrend case is also conservative by avoiding continuously switching down the quality. Meanwhile, in the panic case, we take into account the instant bitrate and instant throughput. Thus, the client can switch down gradually when possible; and there is no need to switch to the lowest version if the instant bitrate of a higher version still meets the throughput constraint.

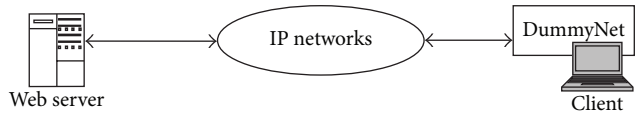


Figure 3: Test-bed organization for experiments.

4. Experimental Results and Discussions

In this section, we will evaluate our proposed method and two reference methods in the context of on-demand VBR streaming over mobile networks, focusing on the behaviors of version switching and buffer level after the initial buffering stage. Two bandwidth traces, a simple one and a complex one, are employed in our experiments.

4.1. Experiment Setup. Our test-bed organization used for the experiments is similar to that of [18], which consists of an HTTP Web server, a streaming client, and an IP network as in Figure 3. The IP network includes a router and wired connections connecting the server and the client. The server is an Apache HTTP server of version 2.2.21 running on Ubuntu 12.04. Our test-bed uses DummyNet tool [21] installed at the client side to emulate network characteristics. The packet loss rate is set to 0%, assuming that the bandwidth trace used in the experiments already takes into account the fluctuations caused by packet loss. RTT value of DummyNet is set to 40 ms. The client is implemented in Java and runs on a Windows 7 notebook with Core i5 2.6 GHz CPU and 4 GB RAM.

The test videos are "Sony Demo" and "Terminator 2" [19] with a frame rate of 30fps and a resolution of 1280 × 720. The duration of each video is 600 seconds. We encode 6 VBR versions by the high profile of H.264/AVC [22], corresponding to 6 different values of QP, namely, 22, 28, 34, 38, 42, and 48. Each version is divided into small video segments of 2 seconds. The version index, QP, and the average bitrate of each version are listed in Table 1. The bitrate traces of the video versions are shown in Figure 2.

As we focus on on-demand streaming, the buffer size of the client is set to 50 s (i.e., 25 segment durations), which is similar to previous work (e.g., [9, 14]). For comparison, two reference methods which are the instant throughput-instant bitrate based method [12] (called ITB) and the buffer-based method with trend prediction [9] (called TBB) are employed. The TBB method is implemented with buffer thresholds $(B_{\min}, B_{\text{low}}, B_{\text{high}}, B_{\max}) = (10\text{ s}, 20\text{ s}, 40\text{ s}, 50\text{ s})$, which are the

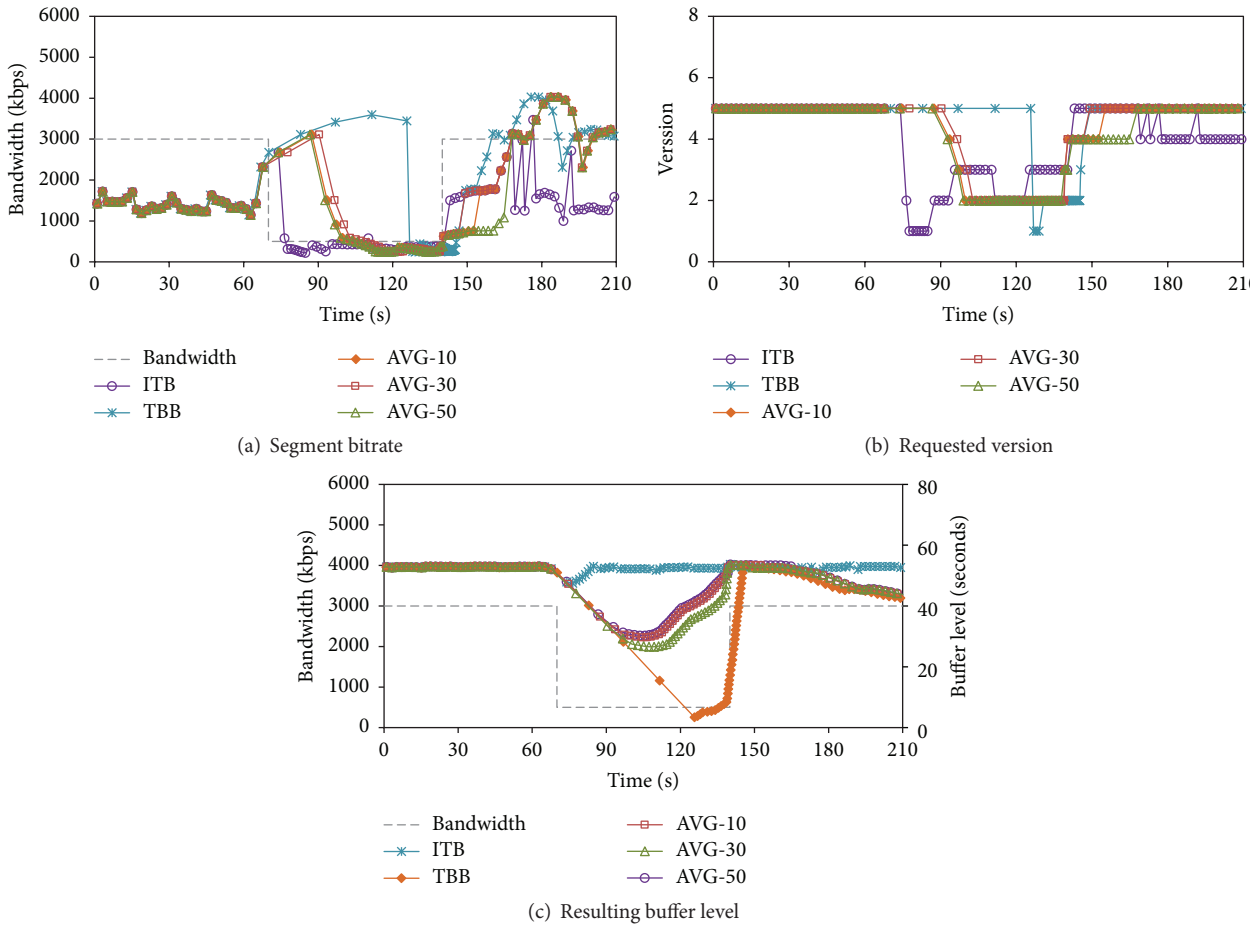


Figure 4: Adaptation results of the three methods in simple bandwidth scenario.

Table 1: Version information of the two test videos.

Index	QP	Average bitrate (kbps)	
		Sony Demo	Terminator 2
1	48	203.77	201.55
2	42	390.75	377.97
3	38	602.96	567.02
4	34	991.32	882.29
5	28	2194.05	1798.93
6	22	5180.58	4127.86

same as the settings in [9]. In our method, $\beta_{\min}$ and $\beta_{\max}$ are 10 s and 50 s, respectively. Also, we investigate whether the number of segments N used for representative bitrates affects our method's performance. The values of N considered are 10, 30, and 50, and these options of our method are referred to as AVG-10, AVG-30, and AVG-50, respectively.

4.2. Simple Bandwidth Scenario. First, we investigate the performance of the methods in a simple bandwidth scenario, when the available bandwidth drops suddenly. The bandwidth has a rectangular shape with two bandwidth levels, 2500 kbps and 500 kbps, as shown in Figure 4(a). This case is important in evaluating adaptation methods because we need to know how they perform when the bandwidth drops drastically. In this case, the Sony Demo video is used.

Figure 4 shows the comparisons of requested versions, bitrates, and buffer levels of the three methods. It is clear that the TBB method tries to maintain the high quality (version 5) for too long even when the available bandwidth drops and stays at the low level for a long interval, resulting in the worst buffer level curve (Figure 4(c)) as well as a drastic drop of quality (around $t = 125$ s, from version 5 to version 1 in Figure 4(b)). As for the ITB method, because it requests versions following the variations of instant throughput and instant video bitrate, the quality is aggressively changed over time while the buffer level variations are very small.

As for our method, all the three options have similar behaviors in terms of bitrate, version switch, and buffer level. Our method reduces the video quality gradually with no switches larger than 1 while the buffer level is higher than 25 seconds. Also, the minimum version provided by our method is 2, while the other two methods sometimes jump to version 1.

4.3. Complex Bandwidth Scenario. In this part, we evaluate the adaptation methods with a complex bandwidth trace (Figure 5), which was obtained from a mobile network [11].

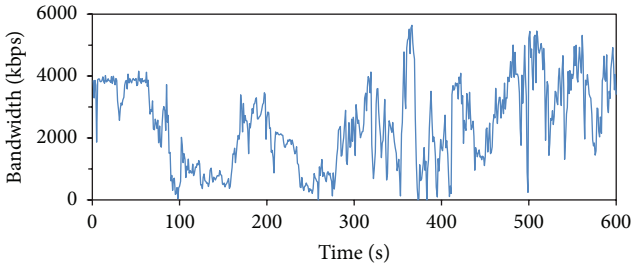


Figure 5: The bandwidth trace used in the complex bandwidth scenario.

Both test videos, "Sony Demo" and "Terminator 2," are employed in this scenario.

Figure 6 shows the experimental results for the "Sony Demo" video. As seen in Figure 6(a), it is very difficult to use the curves of segment bitrates to compare the adaptation methods. From Figure 6(b), it is obvious that the TBB method's behavior is similar to that in the simple bandwidth case. Usually, this method tries to maintain a high version as long as the buffer level permits. This behavior results in sudden drops of quality (e.g., from version 6 to version 3 at 100 s and from version 5 to version 1 at 240 s). Also, the buffer level curve of this method is the worst (Figure 6(c)), as in the previous scenario.

As for the ITB method, its bitrate curve closely follows the throughput curve, resulting in a highly fluctuating version curve with many switches, including switches of large degrees. Anyway, this method has the most stable buffer level curve as seen in Figure 6(c).

Meanwhile, our method provides both quality stability and buffer stability. The version curves of our method (Figure 6(b)) have no sudden switches. Moreover, when the throughput decreases, the selected version of our method is always higher than or equal to those of the other methods. The buffer level curve of our method is not as stable as that of the ITB method; however, it is always higher than that of the TBB method. Compared to the TBB method, our method is more conservative in switching-up and less conservative in switching-down.

Some statistics of the adaptation results are provided in Table 2. The statistics are related to bitrate, requested version, version switch, and buffer level. In terms of bitrate, the ITB method has the lowest average bitrate. Its quality is also very fluctuating with many switches and large switch degree/deviation. The TBB method has the highest average bitrate since this method tends to select and maintain the best possible quality. However, its average version is interestingly lower than that of our AVG-10 option (4.19 vs. 4.30). Even, the ITB method has very low average bitrate, but its average version is higher than that of the TBB method. In fact, the average version values of all methods are very similar (about 4.2 or 4.3). Among the three options, the AVG-10 option is the most aggressive as it always tries to request higher versions; however, this causes a little more switches than other options. These points will be explained and discussed further in the next part.

Table 2: Statistics of the reference methods and the proposed method with three different options for "Sony Demo." Here, STD is the standard deviation.

Statistics	ITB	TBB	AVG-10	AVG-30	AVG-50
Average bitrate (kbps)	1555.2	1951.5	1777.7	1632.0	1637.1
Average version	4.29	4.19	4.30	4.18	4.16
Maximum version	6	6	6	6	6
Minimum version	1	1	2	2	2
Number of switches	94	18	17	15	15
Maximum switch degree	4	4	1	1	1
STD of switch degrees	0.56	0.37	0.23	0.22	0.22
Minimum buffer level (s)	42	5.2	15.1	14.3	14.4
STD of buffer levels (s)	1.59	15.31	11.49	11.41	10.43

Table 3: Statistics of the reference methods and the proposed method with three different options for "Terminator 2." Here, STD is the standard deviation.

Statistics	ITB	TBB	AVG-10	AVG-30	AVG-50
Average bitrate (kbps)	1544.3	1926.2	1962.9	1779.4	1691.3
Average version	4.51	4.34	4.61	4.55	4.46
Maximum version	6	6	6	6	6
Minimum version	1	1	2	2	2
Number of switches	128	17	18	12	12
Maximum switch degree	3	3	1	1	1
STD of switch degrees	0.63	0.34	0.24	0.20	0.20
Minimum buffer level (s)	42.5	1.5	18.3	21.8	22.7
STD of buffer levels (s)	1.64	16.44	10.43	9.22	8.98

Moreover, our method provides the smallest values in terms of the number of switches and the degree of switches. In addition, the minimum version of our method is higher than the other methods. The minimum value and standard deviation (STD) of buffer level of our method are also much better than those of the TBB method. For more information about the buffer, the cumulative distribution functions (CDFs) of the buffer levels are provided in Figure 7(a). Again, it is evident that the buffer of our method is much more stable than that of the TBB method.

Figure 8 shows the experimental results for Terminator 2 video, where similar behaviors of the methods can be found. Our method again provides a smoother version curve and a more stable buffer level curve than the TBB method. The aggressiveness of AVG-10 can also be seen in Figure 8(b). For example, during 330 s~370 s, the AVG-10 option shortly increases the quality to version 6 and then reduces the quality to version 5 and version 4. Meanwhile, the other two options still request version 5 during the same interval.

The statistics of the adaptation results for Terminator 2 video is provided in Table 3. With this video, the average version values of our method are all higher than that of the TBB method. The average bitrate of the AVG-10 option especially is higher than that of the TBB method. This is because version bitrates of Terminator 2 are smaller than those of Sony Demo, so the AVG-10 option can easily reach the highest possible quality. In addition, the other parameters

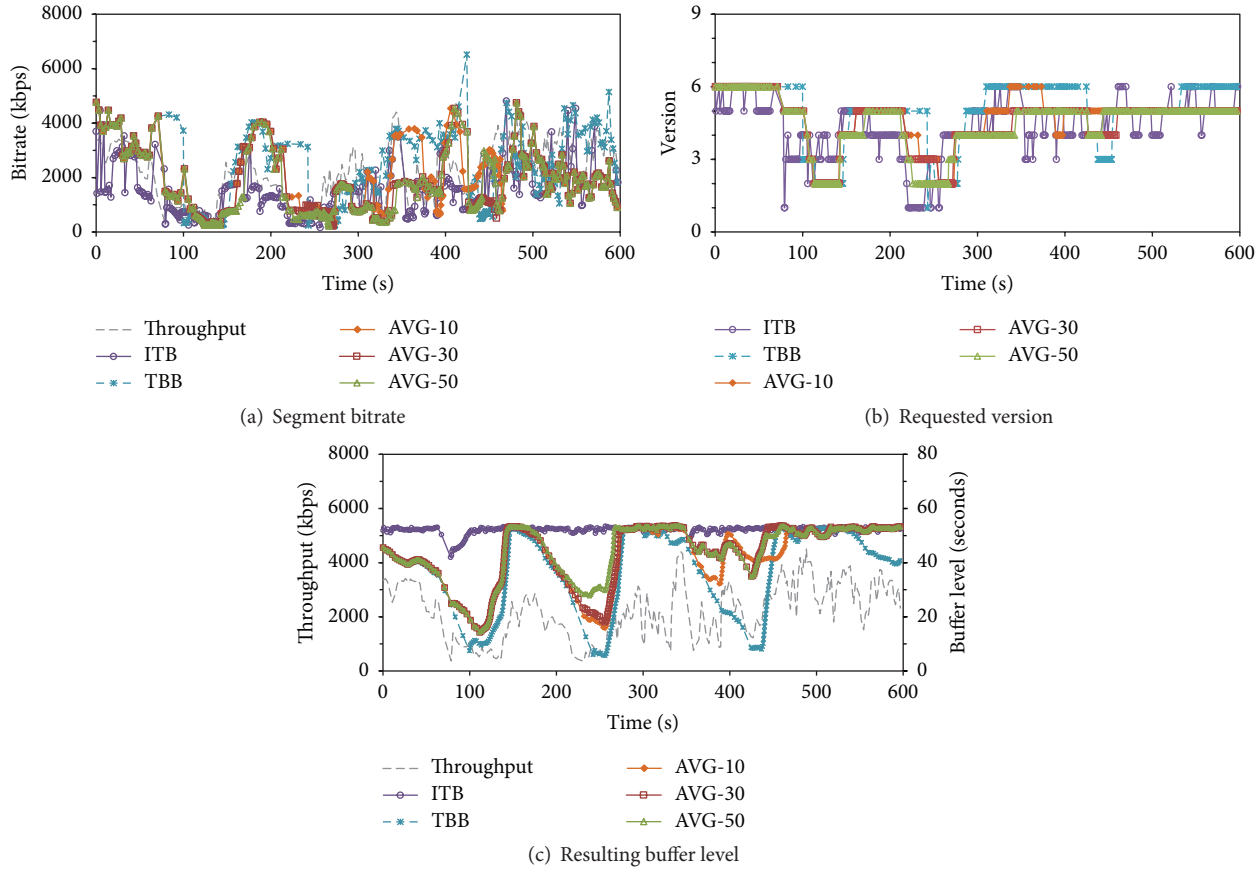


FIGURE 6: Adaptation results of the three methods in the complex bandwidth scenario with "Sony Demo" video.

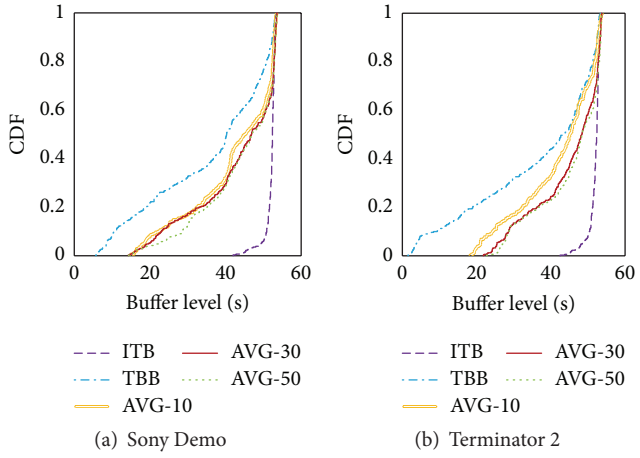


FIGURE 7: Cumulative distribution functions (CDFs) of buffer level in the complex bandwidth scenario.

also show that our method provides a smoother quality and a more stable buffer than the TBB method.

Besides, the CDFs of buffer levels (Figure 7(b)) confirm the stability of buffer level of our method. Figure 7(b) also shows that the buffer levels of the AVG-30 and AVG-50 options are a little better than that of the AVG-10 option. This is because the AVG-10 option is more aggressive than the other options. Especially compared to the case of Sony Demo video (Figure 7(a)), the buffer of the TBB method is worse while the buffer of our method is better. This is actually due to the characteristics of the Terminator 2 video as discussed in the next part.

4.4. Discussions. From the above experimental results, we can see some interesting points. First, in some cases the average bitrate of the TBB method is higher than that of our proposed method, while its average version is lower than that of our method. This can be explained by the fact that sometimes the instant bitrate is very high. So, having some more segments of high bitrates, especially those belonging to the top version, will significantly increase the average bitrate. In case of Terminator 2 video, which has lower version bitrates than Sony Demo video, the average bitrate of TBB method is not better than that of our method.

Meanwhile, in low-throughput periods, the selected versions of the TBB method are lower than those of our method, resulting in a lower value of average version. So, as the bitrate of VBR video is highly fluctuating, the parameter of average bitrate should not be the main indicator to evaluate the performance of VBR video streaming (as in [9]); rather, the parameter of average version should be considered. This is different from CBR video streaming, where the average bitrate is always of high interest.

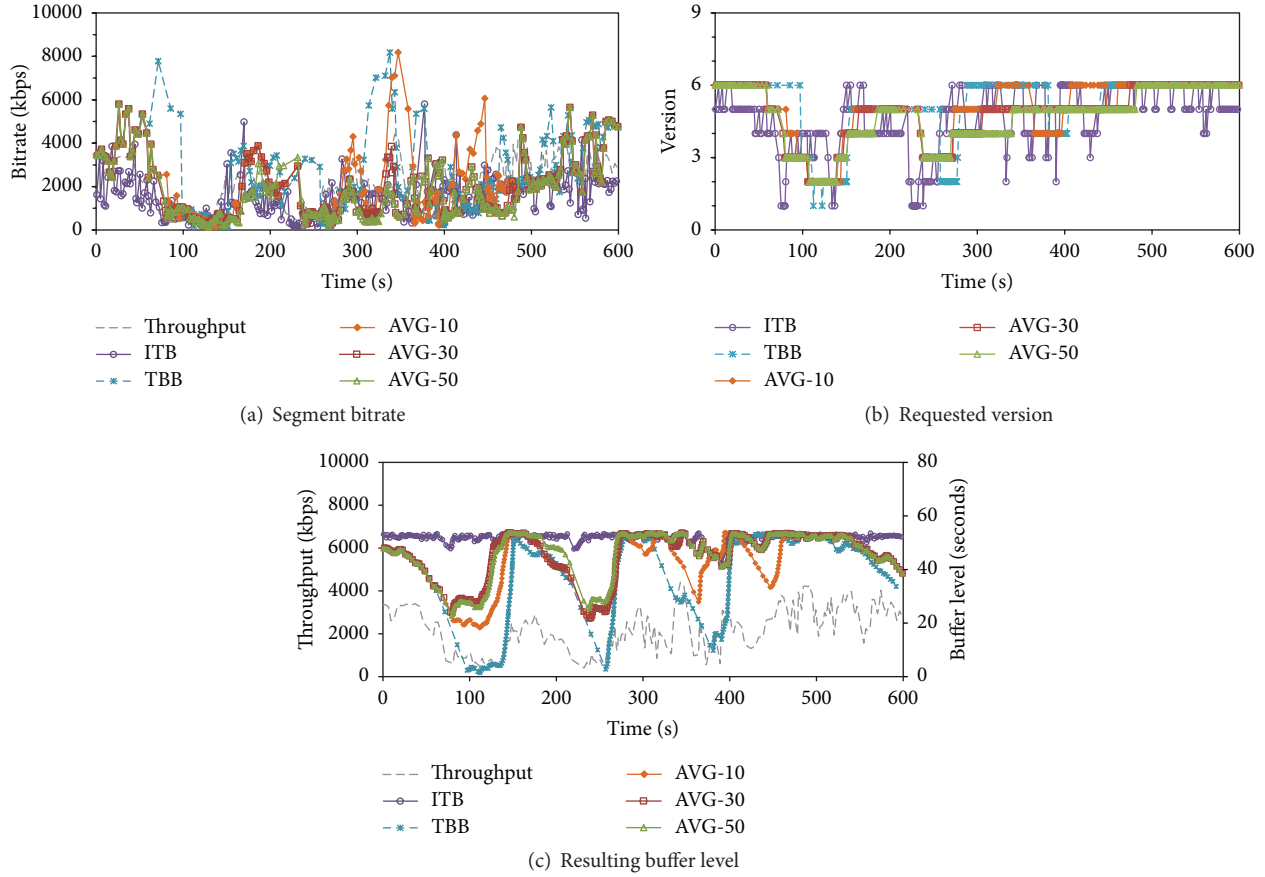


FIGURE 8: Adaptation results of the three methods in the complex bandwidth scenario with "Terminator 2" video.

The results of our adaptation method show that a normal buffer size (i.e., similar to that in CBR video streaming) is already enough to cope with strong bitrate variations of VBR videos. Also, a representative bitrate for each version is important for VBR videos. Even the ITB method benefits a lot from learning the instant bitrate. Though having a highly fluctuating version curve, this method can be used for live streaming thanks to its very stable buffer. On the other hand, the TBB method does not consider bitrate values and thus poorly handles the strong variations of VBR video. For example, although Terminator 2 video has lower version bitrates than Sony Demo video, the buffer stability of the TBB method in the case of Terminator 2 video is worse than in the case of Sony Demo video. This is just because the version bitrates of Terminator 2 video are extremely varying (with repeated and quick changes from a high value to a low value), whereas the performance of our method is not degraded in the case of Terminator 2 video.

The three options of our method are similar in terms of average version values. Yet, among the three options, the AVG-10 option seems to be more aggressive in switching up and maintaining a high quality. This is because the representative bitrate of this option is more "local" and can quickly detect low-bitrate segment intervals, where the client can switch to a higher version when the throughput permits. Such property can increase the average version; however, that also causes more switches as seen in the statistics of the AVG-10 option. The above statistics show that AVG-30 and AVG-50 options have nearly the same performance with good quality stability and good buffer stability.

In general, it is shown that our method is more effective than the reference methods in providing stable streaming quality, with smooth version transitions even when the available bandwidth dramatically drops. If a user prefers streaming with the highest possible quality, more "local" representative bitrates should be used. On the other hand, if the user prefers a stable streaming session with fewer version switches, more "global" representative bitrates should be used.

It should be noted that, actually, the quality of a CBR video is "variable" and the quality of a VBR video is "constant." Meanwhile, as shown in [12], it is totally feasible to provide a quasi-CBR streaming service from VBR video data sets. The above results also show that, with the simple method of representative bitrate estimation, we do not need to know exactly the bitrates of all VBR video segments even though the instant bitrate is extremely varying. This means that VBR videos and the client-based adaptation method proposed in this paper can be easily deployed in DASH-compliant

streaming systems without the need to modify the current standard specification.

5. Conclusion

In this paper, we have presented an adaptation method for VBR videos and evaluated the method in the mobile streaming context. To cope with strong variations of video bitrate, we employed a local average bitrate as the representative bitrate of a version. A buffer-based algorithm was then proposed to conservatively adapt video quality by taking into account a smoothed throughput estimate, the representative bitrate, and even the instant bitrate. The experimental results showed that our method can provide smooth video quality and stable buffer level, even under very strong variations of bandwidth and video bitrates. For future work, we will focus on live streaming scenarios such as surveillances with VBR video sources.

Notations

T_i: The throughput of segment i
T_{i+1}^{est}: The throughput estimate of segment $i + 1$
β_{cur}: The current buffer level
β_{min}: The minimum buffer threshold
β_i^{th}: The flexible buffer threshold
β_{max}: The buffer size and also the target buffer level
$B_{i,k}$: The bitrate of the segment i in version k. It could be the actual value $B_{i,k}^o$ or the estimated value $B_{i,k}^e$
$B_{i,k}^{\text{rep}}$: The representative bitrate of version k at segment i
V: The number of available video versions
I_i: The index of the version which is chosen for segment i (version of higher quality has a higher index value).

Competing Interests

The authors declare that they have no competing interests.

References

[1] Cisco, "Cisco visual networking index: forecast and methodology, 2014—2019," White Paper, 2015.

[2] A. C. Begen, T. Akgul, and M. Baugher, "Watching video over the web: part 1: streaming protocols," *IEEE Internet Computing*, vol. 15, no. 2, pp. 54–63, 2011.

[3] T. C. Thang, Q.-D. Ho, J. W. Kang, and A. T. Pham, "Adaptive streaming of audiovisual content using MPEG DASH," *IEEE Transactions on Consumer Electronics*, vol. 58, no. 1, pp. 78–85, 2012.

[4] T. Stockhammer, "Dynamic adaptive streaming over HTTP: standards and design principles," in *Proceedings of the 2nd Annual ACM Multimedia Systems Conference (MMSys '11)*, pp. 133–143, San Jose, Calif, USA, February 2011.

[5] T. C. Thang, J. Y. Lee, J. W. Kang, S. J. Bae, S. Jung, and S. T. Park, "Signaling metadata for adaptive http streaming," in *ISO/IEC JTC1/SC29/WG11m17771*, 2010.

[6] A. C. Begen, T. Akgul, and M. Baugher, "Watching video over the web: part 2: applications, standardization, and open issues," *IEEE Internet Computing*, vol. 15, no. 3, pp. 59–63, 2011.

[7] C. Liu, I. Bouazizi, and M. Gabbouj, "Rate adaptation for adaptive HTTP streaming," in *Proceedings of the 2nd Annual ACM Multimedia Systems Conference (MMSys '11)*, pp. 169–174, San Jose, Calif, USA, February 2011.

[8] K. Miller, E. Quacchio, G. Gennari, and A. Wolisz, "Adaptation algorithm for adaptive streaming over HTTP," in *Proceedings of the 19th International Packet Video Workshop (PV '12)*, pp. 173–178, IEEE, Munich, Germany, May 2012.

[9] Y. Zhou, Y. Duan, J. Sun, and Z. Guo, "Towards simple and smooth rate adaption for VBR video in DASH," in *Proceedings of the IEEE Visual Communications and Image Processing Conference (VCIP '14)*, pp. 9–12, December 2014.

[10] S. Akhshabi, S. Narayanaswamy, A. C. Begen, and C. Dovrolis, "An experimental evaluation of rate-adaptive video players over HTTP," *Signal Processing: Image Communication*, vol. 27, no. 4, pp. 271–287, 2012.

[11] C. Müller, S. Lederer, and C. Timmerer, "An evaluation of dynamic adaptive streaming over HTTP in vehicular environments," in *Proceedings of the 4th Workshop on Mobile Video (MoVid '12)*, pp. 37–42, Chapel Hill, NC, USA, February 2012.

[12] T. C. Thang, H. T. Le, H. X. Nguyen, A. T. Pham, J. W. Kang, and Y. M. Ro, "Adaptive video streaming over HTTP with dynamic resource estimation," *Journal of Communications and Networks*, vol. 15, no. 6, pp. 635–644, 2013.

[13] D. V. Nguyen, D. M. Nguyen, H. T. Tran, N. P. Ngoc, A. T. Pham, and T. C. Thang, "Quality-delay tradeoff optimization in multi-bitrate adaptive streaming," in *Proceedings of the IEEE International Conference on Consumer Electronics (ICCE '15)*, pp. 66–67, IEEE, January 2015.

[14] H. T. Le, N. P. Ngoc, T. A. Vu, A. T. Pham, and T. C. Thang, "Smooth-bitrate adaptation method for HTTP streaming in vehicular environments," in *Proceedings of the IEEE Vehicular Networking Conference (VNC '14)*, pp. 187–188, December 2014.

[15] K. Evensen, A. Petlund, H. Riiser et al., "Mobile video streaming using location-based network prediction and transparent handover," in *Proceedings of the 21st International Workshop on Network and Operating Systems Support for Digital Audio And Video (NOSSDAV '11)*, pp. 21–26, ACM, British Columbia, Canada, June 2011.

[16] H. T. Le, D. V. Nguyen, N. P. Ngoc, A. T. Pham, and T. C. Thang, "Buffer-based bitrate adaptation for adaptive HTTP streaming," in *Proceedings of the International Conference on Advanced Technologies for Communications (ATC '13)*, pp. 33–38, Ho Chi Minh City, Vietnam, October 2013.

[17] G. Tian and Y. Liu, "Towards agile and smooth video adaptation in dynamic HTTP streaming," in *Proceedings of the 8th ACM International Conference on Emerging Networking Experiments and Technologies (CoNEXT '12)*, pp. 109–120, Nice, France, December 2012.

[18] T. C. Thang, H. T. Le, A. T. Pham, and Y. M. Ro, "An evaluation of bitrate adaptation methods for HTTP live streaming," *IEEE Journal on Selected Areas in Communications*, vol. 32, no. 4, pp. 693–705, 2014.

[19] "H264/AVC video trace library," http://trace.eas.asu.edu/h264/.

[20] T. V. Lakshman, A. Ortega, and A. R. Reibman, "VBR video: tradeoffs and potentials," *Proceedings of the IEEE*, vol. 86, no. 5, pp. 952–972, 1998.

[21] L. Rizzo, "Dummynet: a simple approach to the evaluation of network protocols," *ACM SIGCOMM Computer Communication Review*, vol. 27, pp. 31–41, 1997.

[22] T. Wiegand, G. J. Sullivan, G. Bjøntegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 560–576, 2003.

Hindawi Publishing Corporation
Mobile Information Systems
Volume 2016, Article ID 8981251, 16 pages
http://dx.doi.org/10.1155/2016/8981251

Research Article

Energy-Efficient Transmission Strategy by Using Optimal Stopping Approach for Mobile Networks

Ying Peng,[1] Gaocai Wang,[2] and Nao Wang[2]

[1]*College of Electrical Engineering, Guangxi University, Nanning 530004, China*
[2]*School of Computer and Electronic Information, Guangxi University, Nanning 530004, China*

Correspondence should be addressed to Gaocai Wang; wanggcgx@163.com

Received 18 September 2015; Revised 15 January 2016; Accepted 16 February 2016

Academic Editor: Qi Wang

In mobile networks, transmission energy consumption dominates the major part of network energy consumption. To reduce energy consumption for data transmission is an important topic for constructing green mobile networks. According to Shannon formula, when the transmission power is constant, the better the channel quality is, the greater the transmission rate is. Then, more data will be delivered in a given period. And energy consumption per bit data transmitted will be reduced. Because channel quality varies with time randomly, it is a good opportunity for decreasing energy consumption to deliver data in the best channel quality. However, data has delay demand. The sending terminal cannot wait for the best channel quality unlimitedly. Actually, sending terminal has to select an optimal time to deliver data before data exceeds delay. For this, this paper obtains the optimal transmission rate threshold at each detection slot time by using optimal stopping approach. Then, sending terminal determines whether current time is the optimal time through comparing current transmission rate with the corresponding rate threshold, thus realizing energy-efficient transmission strategy, so as to decrease average energy consumption per bit data transmitted.

1. Introduction

Widespread deployment of mobile networks, for example, mobile ad hoc networks and mobile social networks, and rapid development of data services have brought about exponential growth of wireless mobile terminals (MTs) and dramatic increase of energy consumption in mobile networks. However, the batteries of MTs offer very limited energy and lack the capacity for sustainable supply, especially in the case of inadequate network infrastructure or on the move. As energy consumption greatly affects mobile users, it is highly necessary to make efficient use of resources of mobile networks and reduce energy consumption of MTs for improving mobile users' satisfaction. It is an important and urgent subject to be solved for the construction of green mobile computing [1, 2].

In mobile networks environment, combined influence of multipath propagation, user on the move and channel fading, and so forth will bring about rapid fluctuations in capacity and quality of wireless channel with the change of time. If wireless networks always dynamically allocate resources to the channel in the best instantaneous state or MTs always choose the time of the best channel quality to transmit data, the utilization ratio of wireless network resources will be greatly increased and the performance of network is improved. MT delivers data efficiently using the feature of channel quality varying with time. This strategy is named as opportunistic scheduling [3].

There are two types of opportunistic scheduling, which include centralized opportunistic scheduling and distributed opportunistic scheduling. The former assumes the existence of a central scheduler, which can detect current status of all channels in the network and schedule operation and processing in a centralized manner. The latter, without knowing channel status of other devices, accesses and competes for channel in random mode and certain probability. After obtaining the channel through competition, it will either deliver data immediately if in good quality channel or give up and compete again if in bad quality channel, thus realizing full utilization of network resources in good quality channel.

The distributed opportunistic scheduling technique can take full advantage of device diversity of multiple users

and the differences of channel in different time. In order to increase efficiency of the whole network, distributed opportunistic scheduling permits users to make decision after their temporary surveillance on channel. There is no such need of a complex control center and real time information of all channels' quality. In addition, this technique can decrease network energy consumption [4–8] and enhances network performance [9–12], for example, delivery ratio and throughput.

If the transmission power of sending terminal (ST) is given in wireless link, the greater the transmission rate is, the more data the ST can transmit within the same time period. Consequently, average energy consumption per unit data transmitted (AECPUDT) is smaller. However, transmission rate changes with the wireless channel quality fluctuations. If ST selects the time when channel is in good state to send data, the higher the transmission rate is, the lower the AECPUDT will be. Therefore, in order to minimize AECPUDT on the link, ST needs to survey channel condition timely and then chooses good channel quality time to transmit in accordance with present amount of data accumulated. ST chooses better channel quality for data delivery using the nature of channel quality changing with time, which is called distributed opportunistic scheduling [4–12]. In order to obtain an optimal energy efficiency time (i.e., AECPUDT is smallest) to deliver data, ST continuously surveys channel condition in distributed opportunistic scheduling. Therefore, the distributed opportunistic scheduling could be transformed into an optimal stopping rule strategy to be proved. In this strategy, the decision maker (ST) obtains the time of the minimal expected cost (average energy consumption) to stop observation, based on the continuous observation toward random variable (channel quality), and then takes special actions (transmitting data) to achieve the objective of minimal expected cost.

For the energy consumption problem research in [5], authors of that paper assumed that ST always had enough cumulative data and used the maximum transmission power at the maximum delay and then derived every transmission power threshold of each transmission slot time before the delay. Since current cumulative data quantity of ST is not taken into consideration of these thresholds, ST could neither achieve the optimal energy efficiency under different cumulative data quantity nor guarantee the data delivery ratio (i.e., the ratio of the amount of data transmitted to the total amount of data cumulated). In this paper, the authors take the amount of data cumulated in ST as related factor for selecting the optimal transmission time. In order to better derive the optimal stopping rule, it is assumed that the data generation rate of ST is given, which is consistent with reality needs. For example, when ST transmits some online data, the amount of data to be transmitted per unit time is identified. In addition, the rate of ST obtaining data to be forwarded is also assumed to be constant in [6, 7]. Inspired by optimal stopping theory, the proposition of energy-efficient transmission strategy in this paper is based on the following ideas. When delay requirement and data generation rate on wireless link are given, the matter of distributed opportunistic scheduling about ST selecting optimal channel quality is turned into an optimal stopping problem. Then, the optimal threshold of transmission rate at every selection slot time is acquired based on the optimal stopping theory. Finally, the optimal rate time to transmit data is chosen, thus reducing the AECPUDT and increasing data delivery ratio.

The remaining part is arranged as follows. Related research work of other scholars is reviewed in Section 2. Section 3 describes system model and optimization problem. Then, the energy-efficient transmission strategy using optimal stopping approach is proposed in Section 4. Section 5 presents simulation results and analysis. Finally, Section 6 concludes this paper and prospects future work.

2. Related Research Work

At present, opportunistic scheduling based on time-varying wireless channel quality is widely researched and applied to mobile networks, such as mobile ad hoc networks and mobile social networks. These researches are mainly centered on selection of the optimal transmission time, aiming to improve network performance and energy efficiency of the three scenarios, namely, multiple devices and multiple channels, multiple devices and single channel, and single device and single channel. Researchers mainly focus on two optimization objectives, which are to improve energy efficiency [4–8, 13–16] and to increase network throughput [9–12].

(1) For reducing network energy consumption, many scholars utilize method of choosing the optimal time of channel quality to transmit so as to reduce energy consumption of data transmission [4–8, 13]. Others pay attention to energy consumption saving in the entire routing [14–16].

In order to decrease energy consumption for data transmission, the authors in [4, 5] studied how to choose optimal transmission time to improve energy efficiency in the environment of multiple devices and single channel as well as single device and single channel, respectively. The authors constructed an infinite horizon stopping problem based on optimal stopping theory in [4]. When the competing probability of multiple STs is given in homogeneous environment, the optimal threshold of transmission rate can be derived, which effectively optimized energy efficiency of network. They also proposed a heuristic method for heterogeneous scenario. But they did not consider the data transmission delay demand. The authors considered the case of data with the maximum transmission delay demand in [5]. They studied the transmission energy consumption optimization problem where channel quality varies with time. After obtaining the optimal threshold of power at each slot time using optimal stopping method, ST chose the optimal channel quality time to transmit data, thus energy consumption saved and delay guaranteed. According to their assumption, ST had adequate data to transmit all the time, which is an ideal case. In true applications, ST may have much more or much less data to deliver during the transmission period, while ST obtains excellent opportunities to transmit. The authors in [6] studied the cost optimization of that roadside unit transmitting data to passing-by vehicle in vehicle delay tolerant network. By introducing a function of transmission cost and a penalty cost for exceeding delay, they proved that the time when the queueing delay at roadside unit is above certain threshold is an optimal

one for roadside unit to deliver data flow. The simulation results show that the optimal stopping theory can effectively save transmission cost of roadside unit. Furthermore, in [7] they studied this problem and gave a more complete theoretical derivation and experimental proof. We proposed an energy consumption optimization strategy for data transmission in [8] and formed preliminary ideas of utilizing optimal stopping approach to save energy. We present more perfect theoretical derivation and abundant experimental proof in this paper, which includes five points mainly. The first is the full explanation of opportunistic scheduling problem using optimal stopping theory. The second is the detailed related research work and the differences between ours and the work of others in [5]. The third is the correlation between this work and optimal stopping theory. The fourth is the detailed solution and performance analysis of the strategy. The fifth is the relationship of optimal energy efficiency and parameters and analysis of average scheduling period. The problem of choosing good channel with delay constraint in mobile networks is studied by authors of [13]. They utilized stochastic game to obtain the optimal power threshold for successful data transmission in opportunistic scheduling, which reduced the waste of energy caused by channel error and packet collision.

In addition, some others focused on saving energy in the opportunistic network, which is consumed by sending information from source to destination. Due to rapid fluctuation of channel condition, routing information estimated by average channel quality will become out of date, but opportunistic routing can avoid this case. Compared with the traditional way that data is delivered through a predefined end-to-end path, opportunistic routing enables data to be transmitted from source to destination without end-to-end path. For example, [14] proposed a method of cooperative communication for energy efficiency. The authors fully exploited the random change nature of wireless channel and enabled data packet to be transmitted through better path by high energy efficiency relay node. Consequently, energy consumption is reduced. In [15], they introduced the functions of computing end-to-end energy consumption for traditional routing and opportunistic routing. The authors utilized the technique of cross-layer information exchange to reduce energy consumption and designed energy-efficient routing algorithms based on Dijkstra algorithm. The simulation results show that energy-efficient opportunistic routing outperforms the traditional routing. We in [16] proposed the routing strategy of minimizing transmission energy consumption in mobile network, where transmission delay demand of data is considered. The routing strategy decreased network energy consumption effectively.

(2) For increasing network throughput of opportunistic network, researchers propose all kinds of schemes and algorithms to select a good channel for data transmission, so as to obtain more desirable transmission rate and increase data delivery ratio.

In order to improve network throughput in an ad hoc network where many links contend for the same channel by random access, two distributed opportunistic scheduling strategies from the network-centric and user-centric perspective are proposed in [9], respectively. In [10], the authors studied the improvement of network throughput with proportional fairness. In the scenario of multiple devices competing for the same channel, the researchers constructed block fading channel model with different channel detection slot time dependencies in [11]. Taking into account the impact of channel dependencies on transmission scheduling and system performance, they formulated optimal stopping problem of finite horizon and chose the effective decision time, then characterized the system performance by backward induction, and finally solved the problem by recursive algorithm and effectively enhanced throughput of the system. Considering secure and regular links coexisting in a mobile network, [12] designed a QoS-oriented distributed scheduling scheme based on optimal stopping theory to maximize the whole network throughput.

In summary, for the distributed opportunistic scheduling, it is a very important solution to obtain the optimal scheduling time [4–12] using the optimal stopping theory. In these researches including multiple devices competing to select multiple channels, multiple devices competing to select single channel, and single device selecting single channel, researchers constructed different kinds of optimal rule problem. They solved these problems to obtain the optimal transmission rate threshold [4, 8–11], the optimal power threshold [5], and the optimal transmission time [6, 7], respectively, so as to minimize energy consumption [4–8] or maximize network throughout [9–12]. We research the energy-efficient transmission strategy in this paper, when one device selects single channel. Differing from the hypothesis in [5] that ST has adequate data all the time, we assume that data is generated at a certain rate. In addition, the objectives of minimizing expected energy consumption and average energy consumption per unit time are studied in [5]. This paper considers the objective of minimizing AECPUDT, under the constraint condition of finishing transmitting all data accumulated. For this purpose, we construct the minimization problem of AECPUDT with transmission delay demand and the amount of data transmitted constraint. This minimization problem can be turned into a finite horizon optimal stopping rule problem, so as to acquire the optimal transmission rate thresholds. This paper includes the following two main contributions. (1) The transmission energy consumption optimization problem with transmission delay demand and the amount of data transmitted constraint in mobile network is studied using the characteristics of channel quality varying with time. The effect of the given data generation rate and transmission delay on the AECPUDT is analyzed. (2) We construct the finite horizon optimal stopping rule problem about the minimal AECPUDT under amount of data transmitted constraint. Then, the optimal threshold of transmission rate at every slot time is obtained, thus forming the energy-efficient transmission strategy using optimal stopping approach.

3. Theoretical Background and Problem Description

3.1. System Model. In our mobile networks model, we assume that MT accesses channel using Carrier Sense Multiple Access

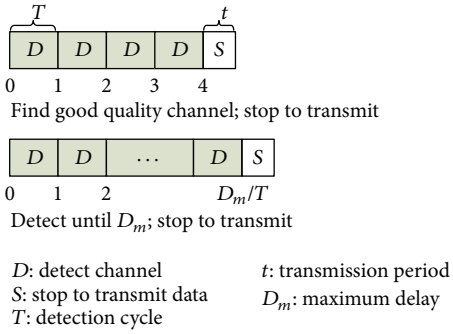


Figure 1: A round of channel detection and data transmission.

with Collision Avoidance (CSMA/CA) protocol. When data needs to be transmitted between two MTs, after establishing the wireless link within transmission range, MT will choose good channel to transmit data immediately. The research purpose of this paper is to minimize the AECPUDT in wireless link under the demand of transmission delay. Meanwhile, the delivery ratio of data transmission must be guaranteed.

In mobile networks, we assume that time is divided into certain slot periods T (s). The channel gain g submits some probability distribution (such as Rayleigh model fading) and remains unchanged in period T [4, 5, 9, 11] on the wireless link construed by ST and a receiving terminal (RT). The data generation rate is c (bit/s). ST delivers the data to the RT within the transmission delay D_m using transmission power P (W). To obtain real time information of channel quality, RT transmits a short signal to ST every period T [5]. Then, ST estimates channel quality according to the signal's power. This consumes energy E_D (J) for each detection. Each duration of detection signal is extremely short and far less than T. After the ST discovers channel in good condition, it will send data during period t. $P \cdot t$ (J) is the energy consumption for data transmission and far bigger than E_D. Because the channel gain g keeps constant in period T, $t \le T$ is satisfied. This entire process from the starting of channel detection to the end of data transmission is called a round of channel detection and data transmission. Its process is given in Figure 1. In a round of detection, the total duration for detection is $n \cdot T$ (s), and the total energy consumption for detection is $n \cdot E_D$ (J). Here, n is the number of detection instances, which is counted from the end of previous round of data transmission. For the first round, the initial counting of n is 0. If transmission rate is R (bit/s), ST can transmit $R \cdot t$ (bit) data in a round. If there is $Rt < c(nT + t)$, ST does not transmit all cumulated data and the remaining data is $c(nT + t) - Rt$ bits. If $Rt \ge c(nT + t)$, ST actually transmits $c(nT + t)$ bits' data. Furthermore, when $Rt > c(nT + t)$ is satisfied, ST wastes power due to being idle in part of duration t. Obviously, ST can increase the amount of data transmitted in given transmission duration by selecting the time of greater transmission rate, thus decreasing the AECPUDT and improving energy efficiency. Shannon formula is presented in the following expression:

$$R = W \log_2 \left(1 + \frac{g \cdot P}{N_0 \cdot W}\right). \quad (1)$$

According to expression (1), transmission rate R in the channel is associated with transmission power P, channel gain g, noise power spectral density N_0, and bandwidth W. When the values of W, P, and N_0 are certain, the value of R is in positive proportion to the value of g. Therefore, to select the time of greater transmission rate, ST must capture the time of larger channel gain value or better channel quality. In order to acquire the optimal channel quality time to transmit data, ST needs to know channel quality timely. The optimal time to transmit is obtained using optimal stopping approach in this paper.

3.2. Optimal Stopping Theory. In order to maximize the expected payoff or minimize the expected cost, the decision maker observes the random variables sequentially in the optimal stopping theory and selects a proper time to take a given action [17]. The stopping rule is defined by two objects:

(1) A sequence of random variables $X_1, X_2, \ldots$, whose joint distribution is assumed to be known.

(2) A sequence of real-valued rewards or cost functions $y_0, y_1(x_1), y_2(x_1, x_2), \ldots, y_\infty(x_1, x_2, \ldots)$.

The associated stopping rule may be described as follows. After observing $X_1 = x_1, X_2 = x_2, \ldots, X_n = x_n$ ($n = 1, 2, \ldots$), the decision maker may stop observation and accept the known reward or cost function $y_n(x_1, \ldots, x_n)$ or continue observing X_{n+1}. If decision maker chooses not to take any observation, he accepts the constant value y_0. If he has not stopped observing, he will receive $y_\infty(x_1, x_2, \ldots)$. This rule enables that the decision maker chooses the optimal time N ($0 \le N \le \infty$) to stop observation, so that the expected reward $E[Y_N]$ is maximal or the expected cost $E[Y_N]$ is minimal. Among them, $Y_N = y_N(x_1, \ldots, x_N)$ is a random cost or reward stopping at N. $E[\cdot]$ expresses the mathematical expected value. If $n \to \infty$, this problem is an infinite horizon optimal stopping problem, which could be calculated through optimality equation. But, in real applications, n is not beyond some value N_m. Then, the problem becomes a finite horizon one and is a special case of infinite horizon. It could be computed using backward induction, which is calculated from the maximum value N_m to the minimum value 0, reversely. Currently, there are a lot of such problems of selecting the optimal time to take actions to maximize the expected reward or minimize the expected cost in the field of communication. Hence, the optimal stopping theory about how to select optimal time according to random variables sequentially observed is an effective tool for solving the mentioned problems.

3.3. Problem Description. In order to minimize AECPUDT and guarantee the delivery ratio, ST selects the optimal time to transmit data under the premise of transmission delay D_m according to the results of continuous observation on channel quality. Hence, the minimization problem in this paper is a finite horizon optimal stopping problem.

In the environment of mobile networks, both the mobility of MT and environmental interference will have some effect on channel quality. The channel condition is uncertain. When the transmission power of ST is constant, the better

Decision maker	⟷	Sending terminal
Observe	⟷	Detect
Random variable	⟷	Channel quality
Cost function	⟷	Average energy consumption
Take action	⟷	Transmit data

Figure 2: Optimal stopping problem elements in data transmission.

the channel quality is (i.e., the faster the transmission rate is), the more the amount of data transmitted in the same duration is. Hence, the AECPUDT is less. The result of the poor quality channel is contrary. At the same time, data to be delivered is generated in given rate and accumulates into ST. If the transmission interval is short, the amount of data accumulated is always less than the one that can be transmitted in transmission duration. Thus, ST wastes transmission power, which causes more AECPUDT. If the transmission interval is too long to transmit all cumulative data in transmission duration, part of data will be discarded due to exceeding the transmission delay.

Actually, MT must decide whether to send data at once or wait for next better moment to transmit data according to current channel quality and the amount of cumulative data. Obviously, due to the stochastic characteristics of link channel quality, ST must detect the channel quality, and then it selects the optimal time to stop detection and transmit data under the accumulative data quantity constraint and the delay demand, so as to minimize the AECPUDT. Therefore, we obtain the relevance between the optimal stopping theory and our optimal stopping problem about energy-efficient transmission strategy, which is shown in Figure 2.

4. Energy-Efficient Transmission Strategy

4.1. Construction of Average Energy Consumption Minimization Problem. We define variable sequence $X_n = \{\Delta T_n, R_n\}$ in the nth channel detection. Here, ΔT_n ($\Delta T_n = T \cdot n$) represents the duration of observation and R_n means the transmission rate at n. ST must detect channel at least once. N is assumed to be time index (time for short) when ST stops detection. The time at which ST must stop detection and deliver data is M, which is $\max\{n : \Delta T_n \le D_m\}$. So $1 \le n \le N \le M$ is satisfied. In a round of detection, the number of channel detection instances is N. Energy consumption for detection of every time is E_D. After the end of detection, ST transmits data in period t. The value of transmission energy consumption is $P \cdot t$. Hence, the total energy consumption in a round is presented in the following expression:

$$E_N = NE_D + Pt. \tag{2}$$

If ST runs this given stopping rule in Y rounds, there will be stopping time sequences $\{N_1, N_2, \ldots, N_y, \ldots, N_Y\}$, energy consumption sequences $\{E_{N_1}, E_{N_2}, \ldots, E_{N_y}, \ldots, E_{N_Y}\}$, and transmission rate sequences $\{R_{N_1}, R_{N_2}, \ldots, R_{N_y}, \ldots, R_{N_Y}\}$. Here, N_y represents the stopping time in the yth round. The initial value of stopping time in each round is 0. With ST detecting channel once in the yth round, the value of N_i is increased by one. And there is $1 \le N_y \le M$. Hence, the total duration of the yth round is sum of detection duration ΔT_{N_y} ($\Delta T_{N_y} = T \cdot N_y$) and transmission duration t. It is equal to $\Delta T_{N_y} + t$. E_{N_y} ($E_{N_y} = N_y E_D + Pt$) is the total energy consumption of the yth round stopping at N_y. At this moment, $c(\Delta T_{N_y} + t)$ bits' data is accumulated and needed to be delivered. R_{N_y} is the transmission rate of the yth round stopping at N_y. L_{N_y} is the value of data that could not be transmitted this time, which is given in the following expression:

$$\begin{aligned} L_{N_y} &= \left(c\left(\Delta T_{N_y} + t\right) - R_{N_y} t\right)^+ \\ &= \begin{cases} c\left(\Delta T_{N_y} + t\right) - R_{N_y} t, & c\left(\Delta T_{N_y} + t\right) > R_{N_y} t, \\ 0, & c\left(\Delta T_{N_y} + t\right) \le R_{N_y} t. \end{cases} \end{aligned} \tag{3}$$

Hence, energy efficiency ζ of AECPUDT is defined in the following expression:

$$\zeta = \frac{\sum_{y=1}^{Y} E_{N_y}}{\sum_{y=1}^{Y} \left(c\left(\Delta T_{N_y} + t\right) - L_{N_y}\right)}. \tag{4}$$

In accordance with the law of large numbers, expression (4) converges to $E[E_N]/E[c(\Delta T_N + t) - L_N]$, where N is the time that ST stops detecting and $E[\cdot]$ represents the mathematical expected value. Thus, an optimal stopping rule problem is constructed, which selects stopping time N to minimize $E[E_N]/E[c(\Delta T_N + t) - L_N]$. There is $1 \le N \le M$. This rule comes from the transmission rate sequence R_N and the total detection duration sequence ΔT_N, which are detected by ST every period T. Meanwhile, energy consumption sequence E_N, accumulative data sequence $c(\Delta T_N + t)$, and data left undelivered sequence L_N are generated. All these sequences values are measurable to be obtained.

When the transmission duration is fixed as t, the amount of data waiting for transmission is $c(\Delta T_N + t)$. If there is $R_N t < c(\Delta T_N + t)$, part of data is left undelivered in this round. On the other side, ST needs to send all data accumulated within duration t. Therefore, this optimization problem includes the constraint condition φ, which is $R_N t \ge c(\Delta T_N + t)$. Consequently, the set of stopping time is given in the following expression:

$$N^+ = \left\{N : 1 \le N \le M,\ E\left[\Delta T_N\right] \le D_m, \varphi\right\}. \tag{5}$$

So the minimization problem of AECPUDT with the amount of data transmitted constraint is described as follows:

$$\begin{aligned} \min_{N \in N^+} \quad & \frac{E\left[NE_D + Pt\right]}{E\left[c\left(\Delta T_N + t\right) - L_N\right]} \\ \text{s.t.} \quad & E\left[R_N t\right] \ge E\left[c\left(\Delta T_N + t\right)\right]. \end{aligned} \tag{6}$$

4.2. Transformation of Optimal Stopping Problem about Average Energy Consumption Minimization. In this paper, it is necessary to derive optimal stopping rule and optimal energy efficiency (the minimum AECPUDT) of ST. In accordance with expression (4), optimal energy efficiency ζ^* is defined as follows:

$$\zeta^* = \inf_{N\in N^+} \frac{E\left[NE_D + Pt\right]}{E\left[c\left(\Delta T_N + t\right) - L_N\right]}. \tag{7}$$

The above expression is turned into the equation in the following expression:

$$\inf_{N\in N^+} \left(E\left[NE_D + Pt\right] - \zeta^* E\left[c\left(\Delta T_N + t\right) - L_N\right]\right) = 0. \tag{8}$$

The minimization problem of expression (8) is an optimal stopping problem related to ζ to minimize $E[Z_N]$. The following equation holds:

$$Z_N = NE_D + Pt - \zeta\left(c\left(\Delta T_N + t\right) - L_N\right). \tag{9}$$

Suppose, for every value of ζ, a set of optimal time $N(\zeta) \in N^+$ to minimize $E[Z_N]$ exists. Our objective is to acquire optimal stopping time $N^* = N(\zeta^*)$, so as to get the optimal energy efficiency ζ^*. Hence, expression (10) is obtained. Consider

$$N^* = \arg\inf_{N\in N^+} \frac{E\left[NE_D + Pt\right]}{E\left[c\left(\Delta T_N + t\right) - L_N\right]}. \tag{10}$$

Therefore, expression (6) is transformed as follows:

$$\begin{aligned} \min_{N\in N^+} \quad & \left(E\left[NE_D + Pt\right] - \zeta E\left[c\left(\Delta T_N + t\right) - L_N\right]\right) \\ \text{s.t.} \quad & E\left[R_N t\right] - E\left[c\left(\Delta T_N + t\right)\right] \geq 0. \end{aligned} \tag{11}$$

On the basis of the Lagrange duality theory, expression (11) is transformed into the following expression:

$$\begin{aligned} \min_{N\in N^+} E\left[Y_N\right] = \min_{N\in N^+} & \left(E\left[NE_D + Pt\right]\right. \\ & - \zeta E\left[c\left(\Delta T_N + t\right) - L_N\right] \\ & \left. + \lambda\left(E\left[c\left(\Delta T_N + t\right)\right] - E\left[R_N t\right]\right)\right). \end{aligned} \tag{12}$$

Here, $\lambda \geq 0$ is a Lagrange multiplier.

Remark 1. In expression (12), the following expression can be gotten:

$$\begin{aligned} &\lambda = 0, \quad E\left[c\left(\Delta T_N + t\right)\right] \leq E\left[R_N t\right], \\ &\lambda > 0, \quad E\left[c\left(\Delta T_N + t\right)\right] > E\left[R_N t\right]. \end{aligned} \tag{13}$$

Hence, we have the following expression:

$$\lambda\left(E\left[c\left(\Delta T_N + t\right)\right] - E\left[R_N t\right]\right) = \begin{cases} 0, & E\left[c\left(\Delta T_N + t\right)\right] \leq E\left[R_N t\right], \\ \lambda\left(E\left[c\left(\Delta T_N + t\right)\right] - E\left[R_N t\right]\right), & E\left[c\left(\Delta T_N + t\right)\right] > E\left[R_N t\right]. \end{cases} \tag{14}$$

According to expression (3), the following equation exists:

$$\lambda\left(E\left[c\left(\Delta T_N + t\right) - R_N t\right]\right) = \lambda E\left[L_N\right]. \tag{15}$$

Therefore, expression (12) is turned into the following expression:

$$\begin{aligned} \min_{N\in N^+} E\left[Y_N\right] = \min_{N\in N^+} & \left(E\left[NE_D + Pt\right]\right. \\ & \left. - \zeta E\left[c\left(\Delta T_N + t\right) - L_N\right] + \lambda E\left[L_N\right]\right). \end{aligned} \tag{16}$$

Remark 2. For energy efficiency ζ, there is $E_D/(cT) < \zeta < Pt/(ct)$. Assume that the ST stops observing and transmits data at the time N. When the amount of data left undelivered L_N is 0, the value of energy efficiency is minimal. The minimum value is $(NE_D + Pt)/(cNT + ct)$. Therefore, energy efficiency ζ is no smaller than that one. That means $\zeta \geq (NE_D + Pt)/(cNT + ct)$ is satisfied.

As $t \leq T$ holds, $\zeta > (NE_D + Pt)/(cNT + cT)$ is satisfied.

That is, $\zeta > E_D/(cT) + (Pt - E_D)/((N + 1)cT)$.

Because transmission energy consumption Pt is greater than detection energy consumption E_D, there is $\zeta > E_D/(cT)$.

Meanwhile, $Pt/(ct)$ is the ratio of transmission energy consumption Pt to accumulative data ct during duration t. If there is $\zeta > Pt/(ct)$, ST would deliver data without channel detection and get smaller value of energy efficiency. Consequently, there is $\zeta < Pt/(ct)$.

4.3. Solution to Optimal Stopping Problem about Average Energy Consumption Minimization. To solve the above minimization problem of AECPUDT, firstly the existence of optimal stopping rule needs to be proved. Then, description of the optimal stopping strategy is called for. Finally, the solution to the optimal stopping problem is expected. Proposition 3 is presented as follows.

Proposition 3. *There is optimal stopping rule in expression (16).*

Proof. According to [17], if the problem satisfies the two conditions,

(A1) $E[\inf_n Y_n] > -\infty$,

(A2) $\liminf_{n\to\infty} Y_n \geq Y_\infty$ a.s.,

the optimal stopping rule exists. Because ΔT_n is equal to nT, expression (16) is converted into the following equation:

$$Y_n = nE_D + Pt - \zeta c\,(nT + t) + (\lambda + \zeta)\,L_n. \tag{17}$$

Because $L_n \geq 0$ is satisfied, there is $Y_n \geq nE_D + Pt - \zeta c(nT + t)$.

$\zeta < Pt/(ct)$ holds, so there is $Y_n > nE_D - \zeta cnT$.

Meanwhile, there is $\zeta > E_D/(cT)$. Furthermore, $nT \leq D_m$ is satisfied. Thus, $Y_n > -\infty$ holds.

Consequently, condition (A1) is satisfied.

If $n \to \infty$, $L_n \to c(nT + t)$ holds. And there is $nE_D \to \infty$ and $c(nT + t) \to \infty$. Consequently, we have the following expression:

$$\begin{aligned}
&\liminf_{n\to\infty} Y_n \\
&\quad = \liminf_{n\to\infty} \left(nE_D + Pt - \zeta c\,(nT + t) + (\lambda + \zeta)\,L_n\right) \\
&\quad = \liminf_{n\to\infty} \left(nE_D + Pt + \lambda c\,(nT + t)\right) = \infty.
\end{aligned} \tag{18}$$

As $Y_\infty = \infty$, condition (A2) holds. □

ST has to detect channel condition every period T and then decides whether the current moment is the optimal moment to stop detecting and deliver data or not. The decision depends on whether the future expected channel condition is better or not. As described in Section 3.3, this is a finite horizon optimal stopping problem. Thus, the expected value can be derived using backward induction, so as to obtain transmission rate threshold at each step that ST stops detecting and transmits data. Therefore, the optimal stopping rule is that ST verifies whether the current transmission rate reaches or exceeds the corresponding threshold or not. If it reaches or exceeds the threshold, ST stops detection and delivers data. Otherwise, ST continues doing detection. When ST continues detection until the delay boundary, it must transmit data unconditionally. In line with [11], the minimal rate of return $W_n(\lambda, \zeta)$ at n is as follows:

$$\begin{aligned}
W_n(\lambda, \zeta) &= \min\left(Pt + nE_D - \zeta c\,(nT + t)\right. \\
&\quad \left. + (\lambda + \zeta)\,L_n, V_{M-n-1}(\lambda, \zeta)\right), \\
V_{M-n-1}(\lambda, \zeta) &= E\left[W_n(\lambda, \zeta) \mid F_n\right] \\
&\qquad\qquad n = 1, 2, \ldots, M - 1.
\end{aligned} \tag{19}$$

Here, F_n represents that the related sequence values from 1 to n have been obtained through observation. In expression (19), the energy consumption cost E_D of ST detecting channel each time is considered. According to the optimal stopping rule, when the rate of return at n is less than or equal to the expected value $V_{M-n-1}(\lambda, \zeta)$, ST will stop detection and transmit data. Hence, the following inequality is obtained:

$$Pt + nE_D - \zeta c\,(nT + t) + (\lambda + \zeta)\,L_n \leq V_{M-n-1}(\lambda, \zeta). \tag{20}$$

According to the definition of L_n in expression (3), when $c(nT+t) > R_n t$ is satisfied, the above expression can be turned into the following:

$$Pt + nE_D + \lambda c\,(nT + t) - (\lambda + \zeta)\,R_n t \leq V_{M-n-1}(\lambda, \zeta). \tag{21}$$

That is,

$$R_n \geq \frac{Pt + nE_D + \lambda c\,(nT + t) - V_{M-n-1}(\lambda, \zeta)}{(\lambda + \zeta)\,t}. \tag{22}$$

When $c(nT + t) \leq R_n t$ holds, expression (20) can be transformed as follows:

$$Pt + nE_D - \zeta c\,(nT + t) \leq V_{M-n-1}(\lambda, \zeta). \tag{23}$$

At last, because of maximum delay constraint D_m of this system, ST stops detection and transmits data when the total duration of detection in a round is $M \cdot t$. Consequently, the threshold of transmission rate at the time M is 0.

Above all, we obtain the threshold of transmission rate to stop detection and deliver data at n, which is given in the following expression:

$$\begin{aligned}
&R_{\mathrm{th},n}(\lambda, \zeta) \\
&\quad = \begin{cases} \alpha, & \beta > \alpha,\ n = 1, 2, \ldots, M - 1, \\ \beta, & \beta < \alpha, c1 \text{ holds},\ n = 1, 2, \ldots, M - 1, \\ 0, & n = M, \end{cases} \\
&\alpha = \frac{Pt + nE_D + \lambda c\,(nT + t) - V_{M-n-1}(\lambda, \zeta)}{(\lambda + \zeta)\,t}, \\
&\beta = \frac{c\,(nT + t)}{t}, \\
&c1 \triangleq Pt + nE_D - \zeta c\,(nT + t) \leq V_{M-n-1}(\lambda, \zeta).
\end{aligned} \tag{24}$$

Assume that the probability density of transmission rate is $f_R(r)$. The cumulative probability of transmission rate at M is $\int_0^{R_{\max}} f_R(r)\,dr$, which is denoted as $F\widetilde{R}$. The corresponding expected value is $\int_0^{R_{\max}} r f_R(r)\,dr$ that is denoted as $\widetilde{R}$. If transmission rate R_M of a round satisfies $R_M \geq c(MT + t)/t$, L_M is equal to 0.

Define $R_{\mathrm{th}} = c(MT + t)/t$. If there is $R_{\mathrm{th}} < R_{\max}$, the cumulative probability of transmission rate which is less than R_{th} at M is $\int_0^{R_{\mathrm{th}}} f_R(r)\,dr$. It is denoted as $F\widehat{R}_{\mathrm{th}}$. The corresponding expected value is $\int_0^{R_{\mathrm{th}}} r f_R(r)\,dr$, which is denoted as $\widehat{R}_{\mathrm{th}}$. Thus, when ST transmits data at the time M, the expected value of the rate of return $V_0(\lambda, \zeta)$ is as follows:

$$V_0(\lambda, \zeta) = \begin{cases} \left(Pt + ME_D\right) F\widetilde{R} + \lambda c\,(MT + t)\,F\widehat{R}_{\mathrm{th}} - (\lambda + \zeta)\,\widehat{R}_{\mathrm{th}} t - \zeta c\,(MT + t)\left(F\widetilde{R} - F\widehat{R}_{\mathrm{th}}\right), & R_{\mathrm{th}} \leq R_{\max}, \\ \left(Pt + ME_D + \lambda c\,(MT + t)\right) F\widetilde{R} - (\lambda + \zeta)\,\widetilde{R}t, & R_{\mathrm{th}} > R_{\max}. \end{cases} \tag{25}$$

According to backward induction, the following equation is obtained that combines with expression (19):

$$\begin{aligned} V_1(\lambda,\zeta) &= E\left[\min\left(Pt+(M-1)E_D\right.\right. \\ &\quad \left.-\zeta c\left((M-1)T+t\right)+(\lambda+\zeta)L_{M-1}, V_0(\lambda,\zeta)\right)\Big] \\ &= \int_{R_{\mathrm{th},M-1}(\lambda,\zeta)}^{R_{\max}} \left(Pt+(M-1)E_D-\zeta c\left((M-1)T\right.\right. \\ &\quad \left.\left.+t\right)+(\lambda+\zeta)L_{M-1}\right) f_R(r)\,dr \\ &\quad + \int_0^{R_{\mathrm{th},M-1}(\lambda,\zeta)} V_0(\lambda,\zeta)\, f_R(r)\,dr. \end{aligned} \tag{26}$$

Similarly, we obtain $V_2,\ldots,V_{M-1}$. Thus, the expected value of the rate of return $V_{M-n}(\lambda,\zeta)$ at n is as follows:

$$\begin{aligned} V_{M-n}(\lambda,\zeta) &= E\left[\min\left(Pt+nE_D-\zeta c\left(nT+t\right)\right.\right. \\ &\quad \left.+(\lambda+\zeta)L_n, V_{M-n-1}(\lambda,\zeta)\right)\Big] = \int_{R_{\mathrm{th},n}(\lambda,\zeta)}^{R_{\max}} \left(Pt+nE_D\right. \\ &\quad \left.-\zeta c\left(nT+t\right)+(\lambda+\zeta)L_n\right) f_R(r)\,dr \\ &\quad + \int_0^{R_{\mathrm{th},n}(\lambda,\zeta)} V_{M-n-1}(\lambda,\zeta)\, f_R(r)\,dr \\ &\qquad\qquad n=1,2,\ldots,M-1, \end{aligned} \tag{27}$$

where $R_{\mathrm{th},n}(\lambda,\zeta)$ is presented in expression (24). Therefore, the optimal stopping rule for expression (16) is presented as follows:

$$N(\zeta^*) = \min\left\{M \ge n \ge 1 : R_n \ge R_{\mathrm{th},n}(\lambda,\zeta^*)\right\}, \tag{28}$$

where $R_{\mathrm{th},n}(\lambda,\zeta^*)$ is defined in expression (24). Next, it is needed to calculate the value of λ and ζ^*. According to [17], the optimal stopping rule has the following optimal equation:

$$\begin{aligned} V^*(\lambda,\zeta^*) &= E\left[\min\left(Pt+NE_D-\zeta^* c\left(NT+t\right)\right.\right. \\ &\quad \left.\left.+\left(\lambda+\zeta^*\right)L_N, V^*(\lambda,\zeta^*)+E_D\right)\right]. \end{aligned} \tag{29}$$

And the optimal solution to the equation is $V^*(\lambda,\zeta^*) = 0$. Consequently, the optimal equation is converted to the following:

$$\begin{aligned} 0 &= E\left[\min\left(Pt+NE_D-\zeta^* c\left(NT+t\right)\right.\right. \\ &\quad \left.\left.+\left(\lambda+\zeta^*\right)L_N, E_D\right)\right]. \end{aligned} \tag{30}$$

Meanwhile, according to the KKT (Karush-Kuhn-Tucker) conditions, the following expression exists:

$$\lambda c E\left[\left(NT+t\right)\right] - \lambda E\left[R_N t\right] = 0. \tag{31}$$

To solve (31) it is needed to obtain the value of $E[R_N]$ and $E[N]$. Next, $E[R_N]$ and $E[N]$ are analyzed. In fact, transmission rate R detected by ST at each period T has the same distribution. Moreover, given a cumulative distribution function $F_G(g)$ of the random variable channel gain G, it is natural to derive the cumulative distribution function $F_R(r)$ of the random variable transmission rate R. Assume that the random variable transmission rate at N is R_N. The cumulative distribution function of the random variable R_N is as follows:

$$\begin{aligned} F_{R_N}(r) &= \Pr\left[R_{N,n} \le r \mid \text{stop at } n\right] \\ &= \begin{cases} \dfrac{F_R(r) - F_R\left(R_{\mathrm{th},n}(\lambda,\zeta^*)\right)}{1-F_R\left(R_{\mathrm{th},n}(\lambda,\zeta^*)\right)}, & r \ge R_{\mathrm{th},n}(\lambda,\zeta^*), \\ 0, & r < R_{\mathrm{th},n}(\lambda,\zeta^*). \end{cases} \end{aligned} \tag{32}$$

The probability of ST stopping at n is defined as follows:

$$\rho_n = \left(\prod_{i=1}^{n-1} F_R\left(R_{\mathrm{th},i}(\lambda,\zeta^*)\right)\right)\left(1-F_R\left(R_{\mathrm{th},n}(\lambda,\zeta^*)\right)\right). \tag{33}$$

Hence, the expected value of the random variable transmission rate R_N is given as follows:

$$\begin{aligned} E\left[R_N\right] &= \sum_{n=1}^{M} E\left[R_{N,n} \mid \text{stop at } n\right]\cdot\rho_n \\ &= \sum_{n=1}^{M}\left(\int_{R_{\mathrm{th},n}(\lambda,\zeta^*)}^{R_{\max}} \frac{r}{1-F_R\left(R_{\mathrm{th},n}(\lambda,\zeta^*)\right)} dF_R(r)\right)\cdot\rho_n. \end{aligned} \tag{34}$$

The expected value of the random variable stopping time N is presented as follows:

$$E[N] = \sum_{n=1}^{M} n\rho_n. \tag{35}$$

Consequently, the following equations can be reasoned out:

$$\begin{aligned} \zeta^* &= \frac{Pt+E[N]E_D}{\left(c\left(E[N]T+t\right), E\left[R_N\right]T\right)^+}, \\ &\left(c\left(E[N]T+t\right), E\left[R_N\right]T\right)^+ \\ &= \begin{cases} c\left(E[N]T+t\right), & c\left(E[N]T+t\right) \le E\left[R_N\right]T, \\ E\left[R_N\right]T, & c\left(E[N]T+t\right) > E\left[R_N\right]T. \end{cases} \end{aligned} \tag{36}$$

$E[R_N]$ and $E[N]$ are defined in expressions (34) and (35), respectively. Expression (36) is solved to obtain the value of ζ^* and λ. Process of solving ζ^* and λ is described in detail below.

Step 1. Start: let $k = 1$, and initialize λ_k, $\lambda_{\max}$, and λ_Δ.

Step 2. If $\lambda_k \le \lambda_{\max}$, initialize ζ_0, and carry out Step 3, or else go to Step 8.

Step 3. Solve $V_0(\lambda_k,\zeta_0)$ according to expression (25). Let $n = M-1$.

Step 4. If $n \ge 1$, carry out Step 5, or else go to Step 6.

Step 5. Obtain $R_{\mathrm{th},n}(\lambda_k,\zeta_0)$ according to expression (24). Compute $V_{M-n}(\lambda_k,\zeta_0)$ according to expression (27). Let $n = n-1$. Return to Step 4.

Step 6. Obtain $E[R_N]$ and $E[N]$ according to expressions (34) and (35), respectively. Calculate ζ_{new} according to expression (36). If $|\zeta_{\text{new}} - \zeta_0| > \varepsilon$ (the error value predefined), let $\zeta_0 = \zeta_{\text{new}}$ and return to Step 3, or else let $\zeta_k^* = \zeta_{\text{new}}$ and carry out Step 7.

Step 7. Let $k = k + 1$ and $\lambda_k = \lambda_{k-1} * \lambda_\Delta$. Return to Step 2.

Step 8. Select the minimum value from the sequence of ζ_k^* as ζ^*, and the corresponding λ_k is saved as λ.

Given the values of λ_k and ζ_0, ζ_k^* is obtained by Newton iterative method. The iterative method is quadratic convergence to the optimal energy efficiency ζ_k^*, and the result becomes stable after 3 iterations. To get the optimal ζ^*, a series of ζ_k^* are found out through a series of λ_k, and then the minimal value is selected out as the optimal ζ^*. Here, $\lambda_k \geq 0$ is Lagrange multiplier. Its value is determined by the value of ζ_k^* and is smaller than that one. The experiment showed that the number of values of λ_k is no more than 10. According to ζ^* and λ, the optimal transmission rate threshold value $R_{\text{th},n}(\lambda, \zeta^*)$ is obtained and presented in expression (24).

This optimal transmission rate threshold $R_{\text{th},n}(\lambda, \zeta^*)$ gives the transmission rate threshold at which ST would stop at every slot time n. It also shows the optimal threshold of transmission rate that ST could obtain the smallest AECPUDT. $R_{\text{th},n}(\lambda, \zeta^*)$ effectively decides the optimal time of ST transmitting data. ST detects channel every period T. If ST discovers that the transmission rate at current slot time n is no smaller than $R_{\text{th},n}(\lambda, \zeta^*)$, ST would stop detection and send data. On the contrary, ST continues doing detection. When ST detects channel till the delay value $M \cdot t$, it has to deliver data. In accordance with this optimal stopping strategy, ST detects channel and delivers data continuously, thus decreasing the AECPUDT and increasing the average delivery ratio.

5. Simulation Results and Analysis

The simulation results are shown in this section, which are obtained by MATLAB simulation tools. The relationships between these five parameters and the value of optimal energy efficiency ζ^* are firstly given, so as to determine their values. The parameters include data generation rate c, detection energy consumption E_D, detection cycle T, transmission duration t, and maximum transmission delay D_m, respectively. Then, a comparison is made between the results of our proposed strategy and the ones of other strategies under different parameters. The results include the average energy consumption, the average success delivery ratio, and the average scheduling period.

Wireless channel fading is small scale fading, and the model is generally simulated as Rayleigh or Rician distribution. The channel conditions of ST and RT obey the same probability distribution. ST obtains channel condition through signal transmitted by RT periodically, as described in Section 3.1. According to [18], probability density function (PDF) in Rayleigh distribution is as follows:

$$f_G(g) = \frac{g}{\sigma^2} \exp\left(-\frac{g^2}{2\sigma^2}\right), \quad g \geq 0. \tag{37}$$

Table 1: Parameter values in simulation.

Parameter	Description	Value
W	Bandwidth [MHZ]	1
N_0	Noise power spectral density [W/HZ]	10^{-6}
σ^2	Value related to mean variance of channel gain	1
g	Channel gain	0~4
P	Transmission power [mW]	100
A	Main signal amplitude peak value	1

Here, g is the channel gain and σ^2 is value related to mean variance of channel gain g. Combining with expression (1), $R_{\max}$-normalized cumulative distribution function (CDF) of transmission rate r in Rayleigh distribution is presented as follows:

$$F_R(r) = \frac{\exp\left(-\left(2^{r/W} - 1\right)^2 \cdot \left(N_0 W\right)^2 / \left(2\sigma^2 P^2\right)\right)}{\exp\left(-\left(2^{R_{\max}/W} - 1\right)^2 \cdot \left(N_0 W\right)^2 / \left(2\sigma^2 P^2\right)\right)}. \tag{38}$$

In accordance with [18], PDF in Rician distribution is as follows:

$$f_G(g) = \frac{g}{\sigma^2} \exp\left(-\frac{g^2 + A^2}{2\sigma^2}\right) I_0\left(\frac{gA}{\sigma^2}\right), \quad g \geq 0. \tag{39}$$

Here, $I_0(\cdot)$ is a Bessel function of first-class 0-order correction and A is main signal amplitude peak value. Combining with expression (1), $R_{\max}$-normalized CDF of transmission rate r in Rician distribution is obtained as follows:

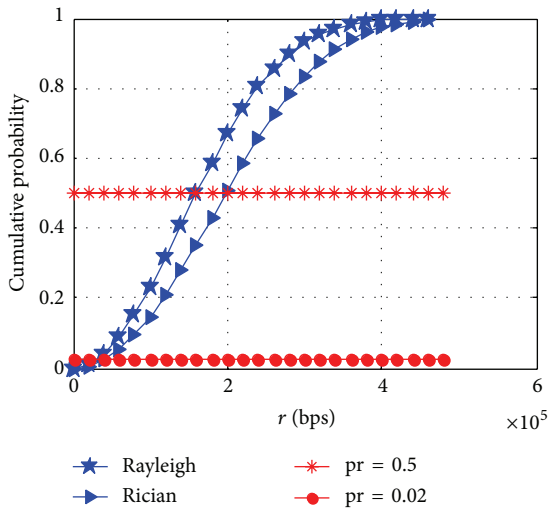
$$F_R(r) = \frac{Q_1\left(A/\sigma, \left(2^{r/W} - 1\right) N_0 W / (\sigma P)\right)}{Q_1\left(A/\sigma, \left(2^{R_{\max}/W} - 1\right) N_0 W / (\sigma P)\right)}, \tag{40}$$

where $Q_1(\cdot)$ is the first-class Marcum Q-function. The parameter values in simulation are shown in Table 1.

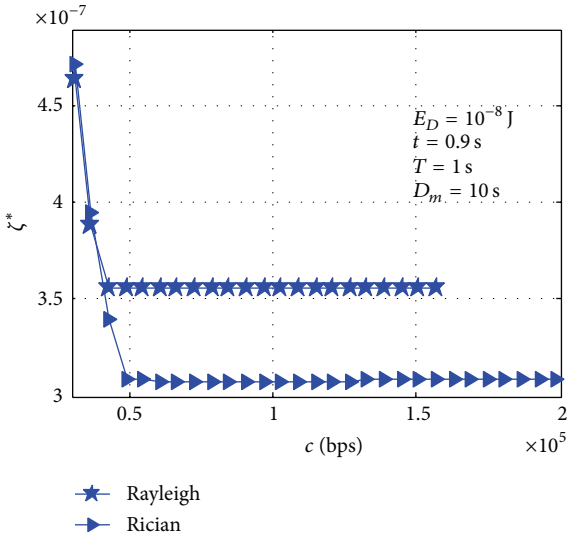
5.1. Effects under Different Parameters on Optimal Energy Efficiency ζ^*. The cumulative distribution curve of transmission rate r is shown in Figure 3(a). As can be seen from the figure, the cumulative probability of r less than 3×10^4 bps is 0.02. That is, the opportunity of link obtaining this rate is smaller. The cumulative probability of r less than 1.6×10^5 bps is 0.5 in Rayleigh distribution and that of r less than 2×10^5 bps is also 0.5 in Rician distribution. This means that the opportunity of link obtaining rate more than that is less than 0.5. But the data loss probability at this moment will be greater than 0.5. Thus, the data generation rate c in Rayleigh distribution is set to be between 3×10^4 and 1.6×10^5 bps. That in Rician distribution is between 3×10^4 and 2×10^5 bps.

As can be seen from expression (36), the optimal energy efficiency ζ^* of the strategy proposed by this paper is closely related to five parameters of ST. The curves of relationships between these parameters and ζ^* are shown in Figures 3(b), 3(c), 3(d), 3(e), 3(f), 3(g), and 3(h).

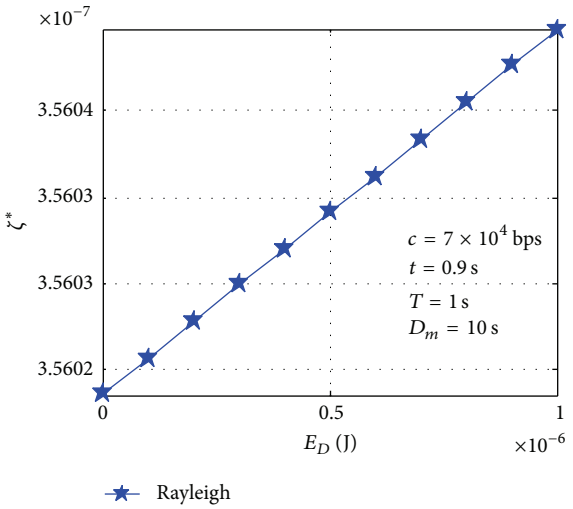
Here, ζ^* is the ratio of the total energy consumption to the amount of data successfully transmitted. The total energy



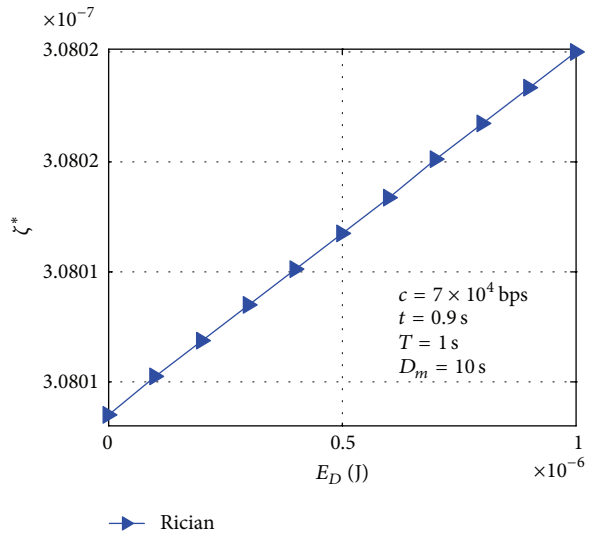
(a) Distribution curve of transmission rate r



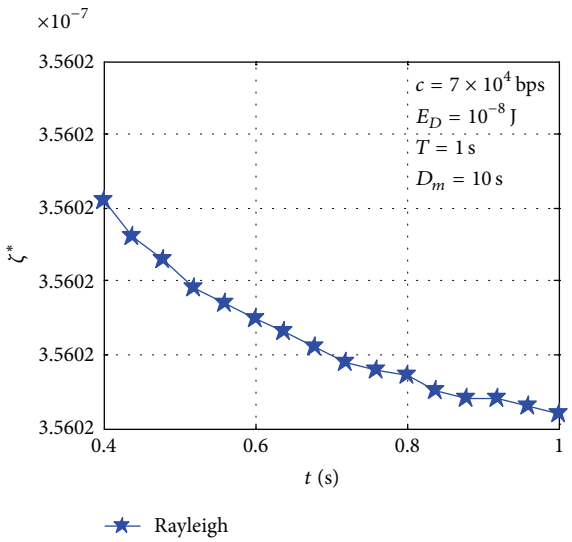
(b) Relationship of c and ζ^*



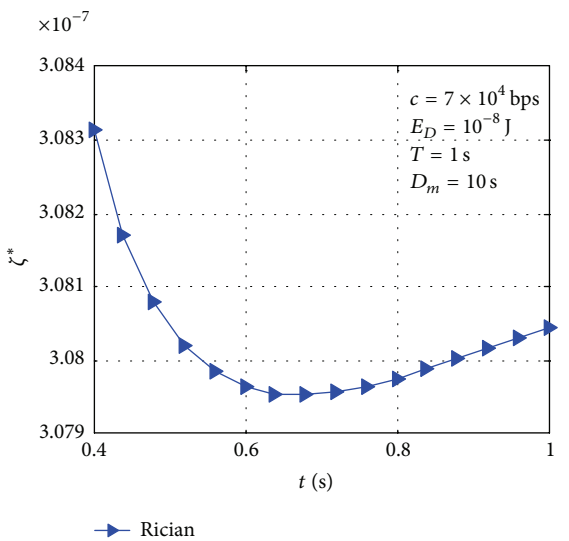
(c) Relationship of E_D and ζ^*



(d) Relationship of E_D and ζ^*

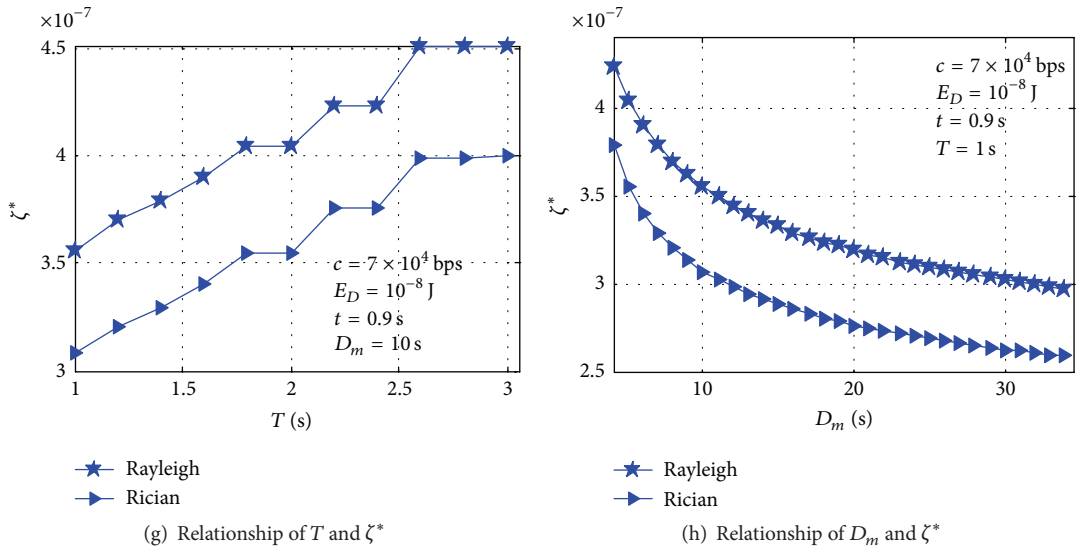


(e) Relationship of t and ζ^*



(f) Relationship of t and ζ^*

Figure 3: Continued.



(g) Relationship of T and ζ^* (h) Relationship of D_m and ζ^*

FIGURE 3: Cumulative distribution curve of transmission rate r and relationships between each parameter and ζ^*.

consumption is energy consumed by ST for detection and transmission in every round. Firstly, the energy consumption for transmission in every round is constant. With c decreasing, the amount of data accumulated is reduced. If the amount of data waiting for transmission is less than the one that can be transmitted within t, ζ^* will increase. Conversely, ζ^* will decrease. Secondly, the total energy consumption of ST includes E_D. Therefore, ζ^* increases with the growth of E_D and decreases with the reduction of E_D. Thirdly, the transmission energy consumption grows with t increasing. If the amount of data accumulated at this moment is less than the one that can be transmitted within t, ζ^* will increase. But the data loss probability reduces. Conversely, ζ^* slightly decreases, but the data loss probability will increase. Fourthly, if T is prolonged, the opportunity that ST finds good channel condition would decrease. So ST has less chance to transmit more data within t, and then ζ^* increases. Fifthly, the amount of data accumulated grows with D_m increasing. So ST can accumulate more data, before it obtains good channel condition. Consequently, ζ^* decreases, but the data loss probability will increase.

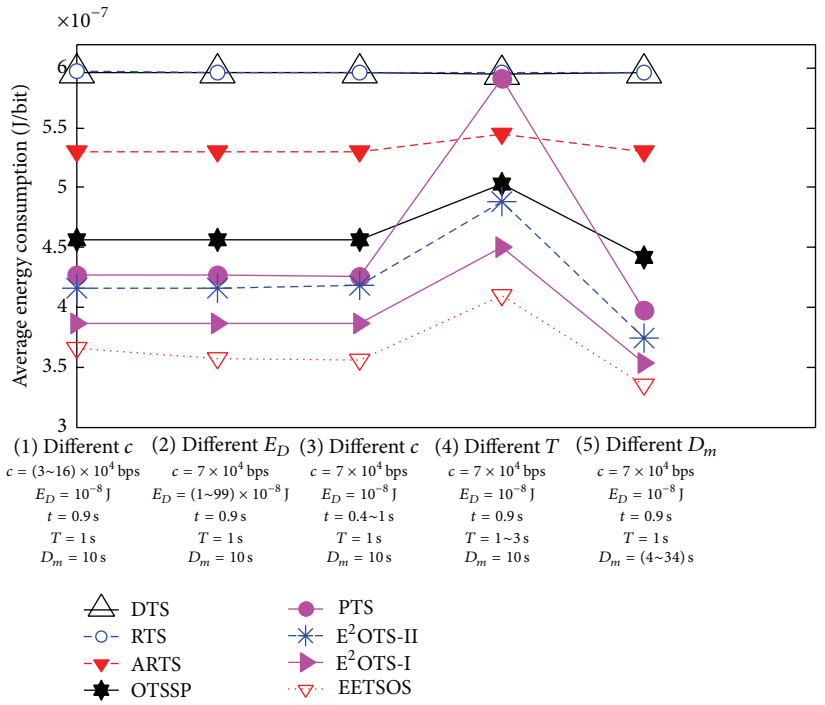
The above analysis shows how the five parameters affect the value of optimal energy efficiency ζ^*. In order to balance the optimal energy efficiency and the data loss probability, these values are taken in simulation, respectively, such that $c = 7 \times 10^4$ bps, $E_D = 10^{-8}$ J, $T = 1$ s, $t = 0.9$ s, and $D_m = 10$ s.

5.2. Performance Comparison and Analysis of Different Strategies. In this section, the comparisons will be made between the Energy-Efficient Transmission Strategy based on Optimal Stopping (EETSOS) theory proposed in this paper and the seven strategies of [5]. The analysis will be made to evaluate the results of average energy consumption, average delivery ratio, and average scheduling period. At first, we describe other strategies used for comparison briefly.

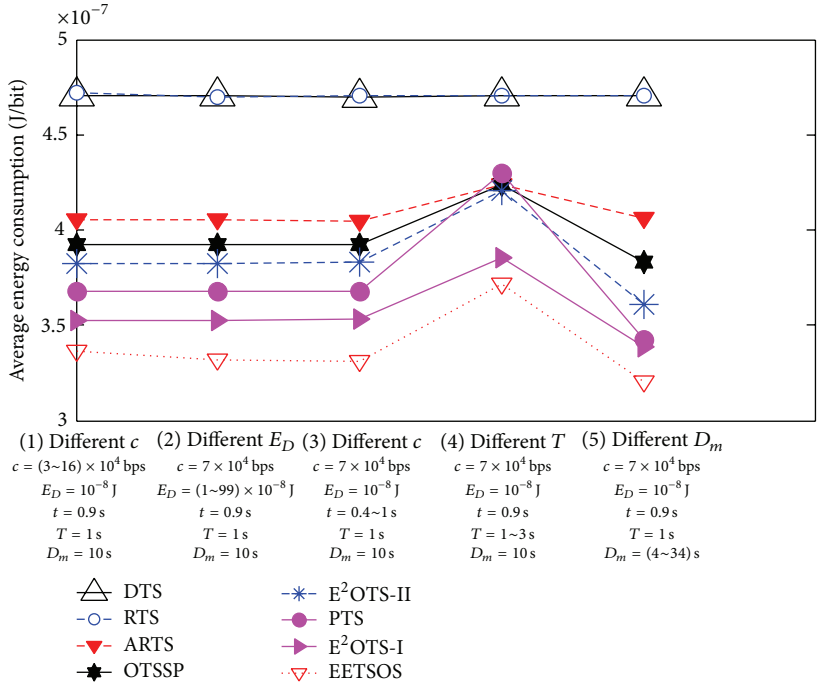
(1) *Deterministic Transmission Strategy (DTS).* ST begins to send data while the time reaches transmission delay D_m. (2) *Random Transmission Strategy (RTS).* From M time of the maximum delay D_m, ST will select randomly one of them to deliver data in $1/M$ probability. (3) *Probabilistic Transmission Strategy (PTS).* ST will deliver data as it forecasts the probability that transmission rate in future being bigger than the present rate would reach or exceed a certain threshold, or else ST goes on doing detection. (4) *Average Rate Transmission Strategy (ARTS).* ST sends data while the present transmission rate is bigger than the average transmission rate in the past. On the contrary, ST will continue doing detection. (5) *Optimal Transmission Strategy Based on Secretary Problem (OTSSP).* ST acquires a maximum transmission rate $R_{c\text{-max}}$ in the period of 37% [5] of the maximum delay D_m using channel detection. In the later period of the remaining 63% of D_m, when ST discovers some rate is bigger than $R_{c\text{-max}}$, it stops transmitting. (6) *Energy-Efficient Opportunistic Transmission Scheduler, E^2OTS [5].* This includes E^2OTS-I and E^2OTS-II, which realize the objective of expected energy consumption minimization and average energy consumption per unit time minimization, respectively. ST will send data as the present power needed is no greater than the optimal threshold of power in corresponding slot time, or else ST will continue doing detection.

5.2.1. Average Energy Consumption. Average energy consumption represents the energy consumption used by transmitting unit data successfully. Its value is the ratio of the total energy consumption to the total amount of data successfully transmitted. The total energy consumption includes not only transmission energy consumption Pt but also detection energy consumption E_D. Comparison results of the average energy consumption of different strategies under different parameters variations of Rayleigh and Rician distribution are shown in Figure 4.

In Figure 4, it shows that EETSOS has the smallest average energy consumption value. It means that EETSOS gains optimal performance of energy consumption optimization.



(a) Average energy consumption in Rayleigh distribution



(b) Average energy consumption in Rician distribution

Figure 4: Comparisons results of average energy consumption under different parameter variations.

EETSOS gets optimum threshold of transmission rate in every slot time using optimal stopping approach. ST will obtain the exact time of the best energy efficiency to deliver data through comparing the corresponding rate threshold of each slot time, thus decreasing average energy consumption. Since DTS and RTS do not consider energy factor for transmission time selection, they have the biggest average energy consumption values. ARTS, PTS, and OTSSP have much smaller values of average energy consumption compared to RTS and DTS, for they take energy consumption as an important factor for the transmission time selection. However, the three strategies do not obtain a transmission

time of the best channel quality. So they consume much more average energy than EETSOS and also more one than E^2OTS. E^2OTS obtains the smaller average energy consumption value. The average energy consumption value of E^2OTS-I is lower than that of E^2OTS-II. The reason lies in that the objective of E^2OTS-II is to minimize the average energy consumption per unit time.

5.2.2. Average Delivery Ratio. Average delivery ratio is the ratio of the amount of data successfully transmitted to the total amount of data to be transmitted of ST. The larger value of ratio represents the less amount of data discarded. Comparison results of the average delivery ratio of different strategies under different parameters variations of Rayleigh and Rician distribution are shown in Figure 5.

We observe in Figure 5 that the average delivery ratio of EETSOS is larger than that of other strategies. It means that EETSOS discards relatively small amount of data. When the minimization problem of AECPUDT (see expression (6)) in EETSOS is constructed, one constraint condition to finish all accumulative data transmission is considered. The optimum thresholds of transmission rate in EETSOS ensure improvement of data delivery ratio. DTS has the least average delivery ratio. Since DTS always transmits at the delay D_m, there will be a lot of data being discarded due to exceeding delay. The average delivery ratios of PTS and RTS are greater than DTS. The two strategies randomly select transmission time and obtain a transmission rate associated with probability distribution. The current rate is compared with the mean value of the past to select transmission time in ARTS, so earlier transmission time to send data is always obtained. Hence, ARTS gets more chances to deliver more data successfully. The average delivery ratio of OTSSP, E^2OTS-I, and E^2OTS-II is small. Because OTSSP takes 37% of the maximum delay to detect channel at least, quite a lot of data will be discarded due to exceeding delay. E^2OTS-I and E^2OTS-II only consider the objective of saving energy and never consider factor of increasing delivery ratio.

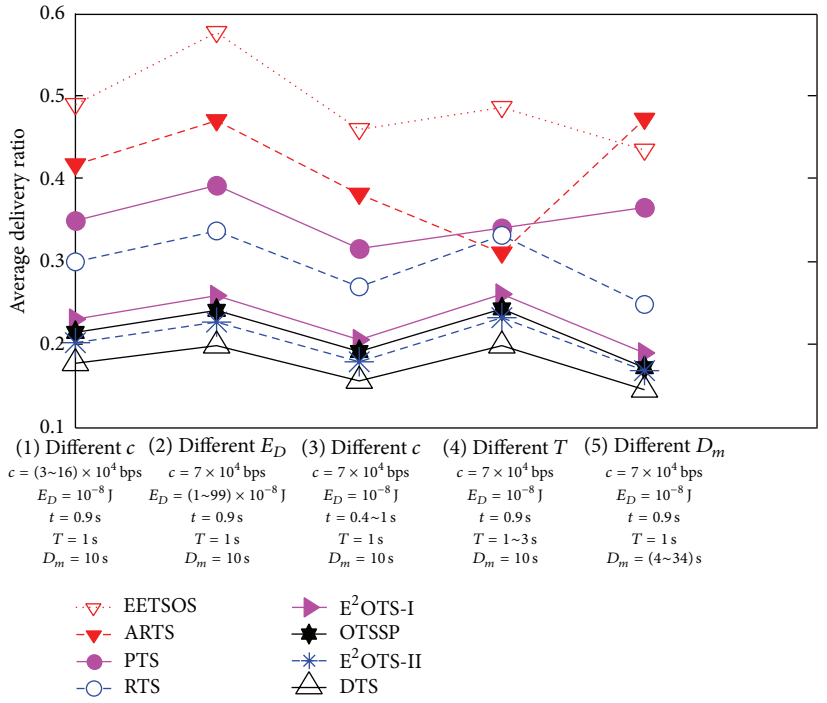
5.2.3. Average Scheduling Period. Average scheduling period is the mean value of detection duration of each round in the given simulation time period. The larger this value is, the longer the average detection duration is and, conversely, the shorter it is. Actually, the average scheduling period is the real average delay of data transmission. Comparison results of the average scheduling period of different strategies under different parameter variations of Rayleigh and Rician distribution are shown in Figure 6.

In Figure 6, it shows that the average scheduling period of EETSOS is smaller than that of RTS, DTS, OTSSP, E^2OTS-I, and E^2OTS-II but is greater than that of RTS, PTS, and ARTS. The average scheduling period of DTS is the longest and equal to the maximum delay. The average scheduling period of RTS is close to the expected value of all detection periods within the maximum delay. The distribution of stopping time of ARTS and PTS is related to the probability distribution of channel. The average scheduling period of OTSSP is longer than 37% of the maximum delay D_m. The average scheduling period of E^2OTS is related to the power threshold and is longer than that of RTS, PTS, and ARTS. Generally, the longer the average scheduling period is, the larger the amount of data accumulated will be. The probability of data discarded due to exceeding delay is increased. Hence, the average delivery ratio of DTS, RTS, and OTSSP is smaller. EETSOS gets the optimal thresholds of transmission rate with the constraint condition about finishing all accumulative data transmission. Therefore, EETSOS not only reduces the average energy consumption but also increases the delivery ratio.

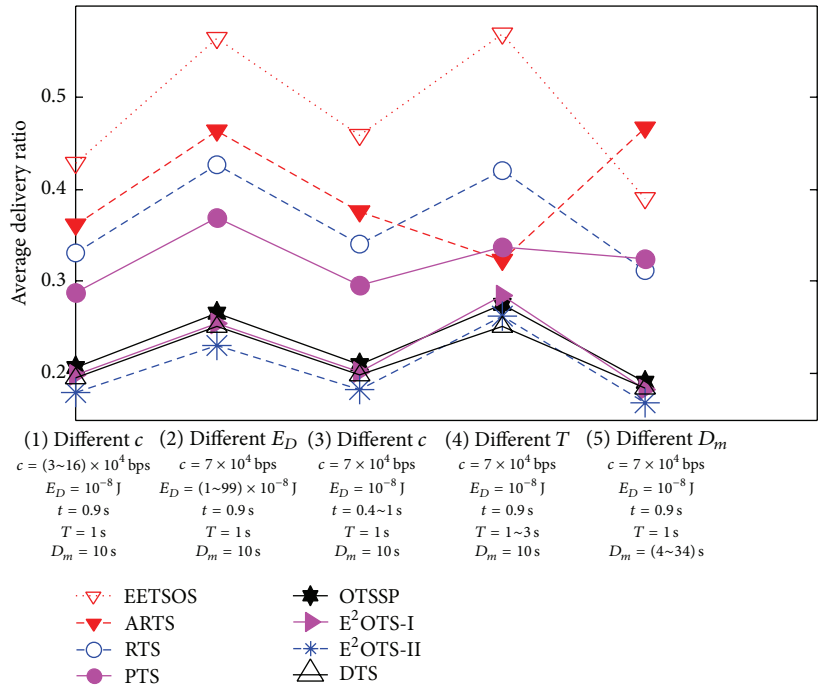
In summary, the energy-efficient transmission strategy proposed using optimal stopping approach achieves the objective of reducing average energy consumption. In addition, the data delivery ratio under the transmission delay constraint is guaranteed. It means that the strategy of this paper improves energy efficiency under the premise of optimizing performance in mobile network.

6. Conclusion

With the rapid advance of network technology, network energy consumption also shows a rapid growth trend. It is one of the important research subjects to construct green mobile computing in the construction and development of mobile networks. In mobile networks, the channel quality in the wireless link varies with time randomly, and it is unstable. If some ST needs to transmit data to another RT in transmission range, ST has to know channel condition timely. In accordance with the information surveyed, ST selects the optimal transmission time to improve energy utilization ratio. Due to the transmission delay demand, it must optimize the data delivery success ratio. Therefore, the minimization problem of AECPUDT with the amount of data transmitted constraint is constructed in this paper. Since wireless channel quality varies with time and distributed opportunistic scheduling can improve network performance and energy utilization of network devices, ST is able to select proper time of the best channel status to send data through distributed opportunistic scheduling. The optimal stopping theory is a useful tool to realize this distributed opportunistic scheduling problem. So this paper proposes transmission strategy using optimal stopping approach, which realizes the objective of decreasing energy consumption through sending data in good channel condition. At first, a minimization problem of AECPUDT is constructed, which considers the transmission delay demand and the amount of data transmitted constraint. This is a finite horizon optimal stopping rule problem. Then the existence of the optimal stopping rule is proved and the solutions and processes of the problem are presented, thus obtaining the optimal threshold of transmission rate on every slot time. ST only compares the current transmission rate obtained by periodical signal with the corresponding optimal threshold and decides which time is the best energy efficiency one to transmit data. Consequently, AECPUDT is reduced, and data delivery ratio is enhanced. The comparison results in simulation indicate that energy-efficient transmission strategy gets the smaller average energy consumption and the larger delivery ratio. The strategy achieves a better optimization

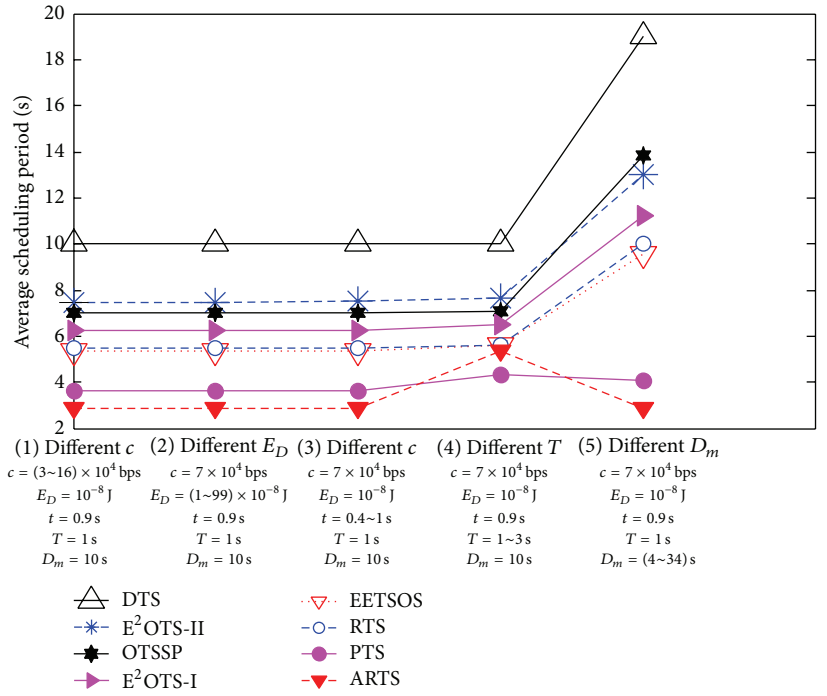


(a) Average delivery ratio in Rayleigh distribution

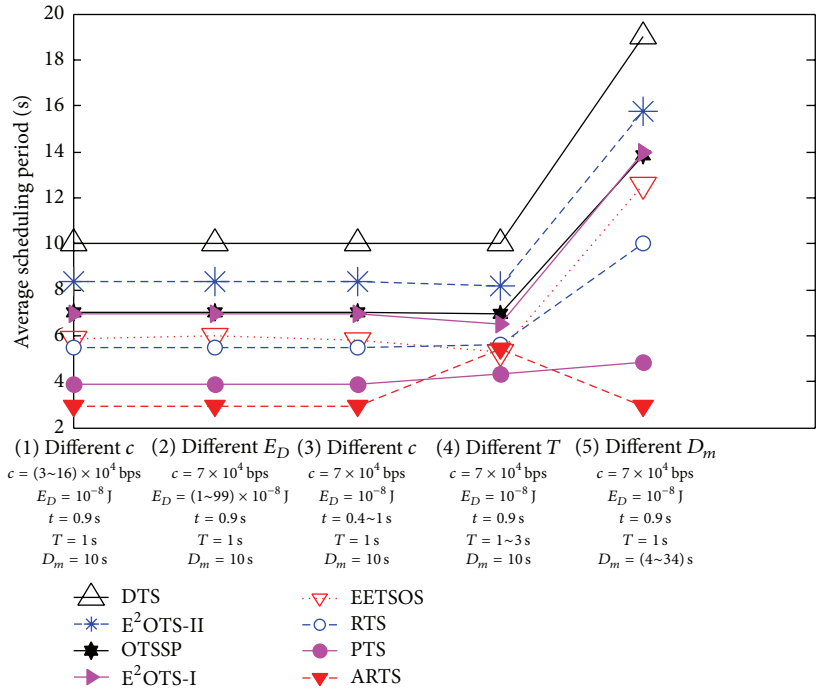


(b) Average delivery ratio in Rician distribution

FIGURE 5: Comparison results of average delivery ratio under different parameter variations.



(a) Average scheduling period in Rayleigh distribution



(b) Average scheduling period in Rician distribution

Figure 6: Comparison results of average scheduling period under different parameter variations.

effect of energy consumption on the basis of guaranteeing network performance.

In addition, how to decrease energy consumption and increase delivery ratio in case of changeable data generation rate is a research topic in the future.

Competing Interests

The authors declare that they have no competing interests.

Acknowledgments

This research is supported in part by the National Natural Science Foundation of China under Grant nos. 61562006 and 61262003 and in part by the Natural Science Foundation of Guangxi Province under Grant no. 2010GXNSFC013013.

References

[1] C. Lin, Y. Tian, and M. Yao, "Green network and green evaluation: mechanism, modeling and evaluation," *Chinese Journal of Computers*, vol. 34, no. 4, pp. 593–612, 2011 (Chinese).

[2] F. Zhang, A. F. Anta, L. Wang, C.-Y. Hou, and Z.-Y. Liu, "Network energy consumption models and energy efficient algorithms," *Chinese Journal of Computers*, vol. 35, no. 3, pp. 603–615, 2012 (Chinese).

[3] A. Asadi and V. Mancuso, "A survey on opportunistic scheduling in wireless communications," *IEEE Communications Surveys and Tutorials*, vol. 15, no. 4, pp. 1671–1688, 2013.

[4] A. Garcia-Saavedra, P. Serrano, and A. Banchs, "Energy-efficient optimization for distributed opportunistic scheduling," *IEEE Communications Letters*, vol. 18, no. 6, pp. 1083–1086, 2014.

[5] M. I. Poulakis, A. D. Panagopoulos, and P. Constantinou, "Channel-aware opportunistic transmission scheduling for energy-efficient wireless links," *IEEE Transactions on Vehicular Technology*, vol. 62, no. 1, pp. 192–204, 2013.

[6] Z. Yan, Z. Zhang, H. Jiang, Z. Shen, and Y. Chang, "Optimal traffic scheduling in vehicular delay tolerant networks," *IEEE Communications Letters*, vol. 16, no. 1, pp. 50–53, 2012.

[7] L. Huang, H. Jiang, Z. Zhang, and Z. Yan, "Optimal traffic scheduling between roadside units in vehicular delay-tolerant networks," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 3, pp. 1079–1094, 2015.

[8] P. Ying, N. Wang, and G. Wang, "An optimization strategy of energy consumption for data transmission based on optimal stopping theory in mobile networks," in *Algorithms and Architectures for Parallel Processing: 15th International Conference, ICA3PP 2015, Zhangjiajie, China, November 18–20, 2015, Proceedings, Part IV*, vol. 9531 of *Lecture Notes in Computer Science*, pp. 281–292, Springer, Berlin, Germany, 2015.

[9] D. Zheng, W.-Y. Ge, and J.-S. Zhang, "Distributed opportunistic scheduling for ad hoc networks with random access: an optimal stopping approach," *IEEE Transactions on Information Theory*, vol. 55, no. 1, pp. 205–222, 2009.

[10] A. Garcia-Saavedra, A. Banchs, P. Serrano, and J. Widmer, "Distributed opportunistic scheduling: a control theoretic approach," in *Proceedings of the IEEE Conference on Computer Communications (INFOCOM '12)*, pp. 540–548, IEEE, Orlando, Fla, USA, March 2012.

[11] H. Chen and J. S. Baras, "Distributed opportunistic scheduling for wireless Ad-Hoc networks with block-fading model," *IEEE Journal on Selected Areas in Communications*, vol. 31, no. 11, pp. 2324–2337, 2013.

[12] W.-G. Mao, S.-S. Wu, and X.-D. Wang, "QoS-oriented distributed opportunistic scheduling for wireless networks with hybrid links," in *Proceedings of the IEEE Global Communications Conference (GLOBECOM '13)*, pp. 4524–4529, Atlanta, Ga, USA, 2013.

[13] C. Van Phan, "A game-theoretic framework for opportunistic transmission in wireless networks," in *Proceedings of the 5th IEEE International Conference on Communications and Electronics (ICCE '14)*, pp. 150–154, Danang, Vietnam, August 2014.

[14] O. Amin and L. Lampe, "Opportunistic energy efficient cooperative communication," *IEEE Wireless Communications Letters*, vol. 1, no. 5, pp. 412–415, 2012.

[15] J. Zuo, C. Dong, H. V. Nguyen, S. X. Ng, L.-L. Yang, and L. Hanzo, "Cross-layer aided energy-efficient opportunistic routing in ad hoc networks," *IEEE Transactions on Communications*, vol. 62, no. 2, pp. 522–535, 2014.

[16] G. Wang, Y. Peng, P. Feng, and N. Wang, "An energy consumption minimization routing scheme based on rate adaptation with QoS guarantee for the mobile environment," *Computer Networks*, vol. 74, pp. 48–57, 2014.

[17] T. S. Ferguson, "Optimal stopping and applications," 2006, http://www.math.ucla.edu/~tom/Stopping/Contents.html.

[18] S. HayKin, S. Tie-Cheng, X. Ping-Ping et al., *Communication System. 4*, Publishing House of Electronics Industry, Beijing, China, 2012 (Chinese).

Hindawi Publishing Corporation
Mobile Information Systems
Volume 2016, Article ID 9076086, 11 pages
http://dx.doi.org/10.1155/2016/9076086

Research Article

Network-Aware Reference Frame Control for Error-Resilient H.264/AVC Video Streaming Service

Hui-Seon Gang, Goo-Rak Kwon, and Jae-Young Pyun

Department of Information and Communication Engineering, Chosun University, 309 Pilmun-daero, Dong-gu, Gwangju 61452, Republic of Korea

Correspondence should be addressed to Jae-Young Pyun; jypyun@chosun.ac.kr

Received 7 August 2015; Revised 21 December 2015; Accepted 15 February 2016

Academic Editor: Qi Wang

To provide high-quality video streaming services in a mobile communication network, a large bandwidth and reliable channel conditions are required. However, mobile communication services still encounter limited bandwidth and varying channel conditions. The streaming video system compresses video with motion estimation and compensation using multiple reference frames. The multiple reference frame structure can reduce the compressed bit rate of video; however, it can also cause significant error propagation when the video in the channel is damaged. Even though the streaming video system includes error-resilience tools to mitigate quality degradation, error propagation is inevitable because all errors can not be refreshed under the multiple reference frame structure. In this paper, a new network-aware error-resilient streaming video system is introduced. The proposed system can mitigate error propagation by controlling the number of reference frames based on channel status. The performance enhancement is demonstrated by comparing the proposed method to the conventional streaming system using static number of reference frames.

1. Introduction

Today, high-quality video content is a basic requirement of multimedia services and is becoming important in mobile communication systems. Because of the low cost of powerful processors and the advancement of mobile communication services, consumers are able to use high-definition multimedia streaming services on their hand-held devices. These multimedia streaming data have been compressed for storage and transmission. Even though many service providers have developed and provided advanced mobile communication services, it remains difficult to reliably transmit high-quality video streams because of the varying channel conditions and limited available bandwidth of wireless channels.

The current streaming video system generally uses motion estimation and compensation procedure at encoder and decoder, respectively, for a high coding efficiency feature. This system considerably reduces the number of bits to encode because it utilizes multiple reference frames to remove temporal redundancy. Because of its high coding efficiency, H.264/AVC and H.265/HEVC are suitable for the streaming system transmitting high-quality video sequences in the environments that have limited channel capacity [1].

However, if the encoded sequences are damaged by channel errors, the damage can be propagated to neighboring macroblocks (MBs) and frames. Even though motion estimation using multiple reference frames can significantly decrease the number of data bits that must be encoded, the compressed sequence can be vulnerable to error propagation. To mitigate the impact of error propagation, the streaming video system includes error-resilience tools. Error-resilience tools preprocess the video data either by reordering each macroblock's coding sequence or by inserting redundant data, such that the damaged blocks can be spreaded out (especially in the case of burst errors). That is, the damaged video is improved by error-resilience tools. These tools can make the encoded video sequence more robust to errors, but coding efficiency will be decreased because of the additional bits [2]. Among the typical error-resilient methods, intrarefresh (IR) algorithm is used often to avoid error propagation in a distorted video sequence over an error-prone network. When IR algorithm is used as an error-resilience method, multiple

reference frame structure used in H.264/AVC motion compensation has recently been found to reduce the received video quality in the presence of transmission errors [3, 4]. This effect occurs because the blocks refreshed by IR coding at the decoder may not be used for further motion compensation of the next frames in multiple reference frame structures; thus, the propagated distortions are not always removed.

In this paper, a new network-aware streaming video system controlling the number of reference frames is proposed for a reliable video transmission system over an error-prone network. The proposed streaming video system uses both the multiple reference frame structure and error-resilience tools for ensuring both error robustness and coding efficiency. To demonstrate the trade-off between error resilience and coding efficiency, various IR and reference frame conditions are used in the performance evaluation.

This paper is organized as follows: Section 2 describes the typical video streaming system. The proposed error-resilient system is explained in Section 3 and the experimental results are presented in Section 4. Finally, Section 5 presents our conclusions.

2. Typical Streaming Video System

2.1. Motion Estimation with Multiple Reference Frames. The key principle of video compression is the elimination of redundancy. Typically, a video encoder compresses a video sequence by removing the temporal, spatial, and statistical redundancies. Specifically, motion estimation and compensation within the compression strategy increase the coding efficiency by removing temporal redundancy between frames. To remove more temporal redundancies, a typical streaming video system based on H.264/AVC and H.265/HEVC uses multiple reference frames for motion compensation and chooses the best reference frame among them based on rate-distortion optimization (RDO). Next, it searches for motion vectors and encodes the residual data in each MB [5]. Motion information regarding blocks is separately encoded and transferred over the network; it is then used for the reconstruction of the original blocks.

However, this compressed motion may be distorted on the unreliable channel. If there are errors caused by packet losses on an unreliable channel, the errors can be propagated into the following frames by the motion compensation procedure even though there are intrarefreshed blocks that are inserted for the error resilience [2]. Furthermore, error propagation becomes more severe as the number of reference frames increases. As a result, motion estimation using multiple reference frames has the advantage of increasing the coding efficiency; however, it also makes the transferred video sequence less robust to errors.

2.2. Error Resilience against Transmission Error. To mitigate quality degradation caused by error propagation, the streaming video system includes error-resilience tools. These tools run the preprocessing in the encoder to make encoded data more robust to errors. In arbitrary slice ordering (ASO), each slice group can be sent in any order and can (optionally) be decoded in order of receipt instead of in the usual scan order; flexible MB ordering (FMO) also reorders the coding sequence of MBs. Even though slices and MBs are consecutively damaged, these errors are scattered and localized in neighboring coding units because these units are reordered at the decoder [1]. Additionally, data partitioning (DP) provides the ability to separate more important and less important syntax elements into different packets of data. Furthermore, redundant slices (RS) are added when the encoder inserts additional picture data such as redundant slice, thus making it more robust to errors [6].

IR coding method is one of the typical error-resilience tools that are widely used in conventional streaming video systems. When the streaming encoder performs motion estimation to code MBs, it decides their coding mode (intra, inter, or skip) using RDO. However, the IR method forces MBs of each frame to be encoded with intracoding mode. The intracoded MBs reduce error propagation and improve the robustness to transmission errors without significantly increasing the RD cost. Additionally, to guarantee that all MBs are eventually refreshed and errors do not propagate indefinitely, random intrarefresh (RIR), which selects MBs randomly in a cyclic mode, can be used. However, the RIR method has some limitations, because RIR randomly decides the locations of intracoded MBs. That is, RIR does not recognize the generated bit-rate difference between moving objects and background area in video sequences [7, 8]. Additionally, the error refresh capabilities can be weakened when multiple reference frames are used in motion compensation, because the damage is propagated into the following frames, even though MBs in the previous frame have been refreshed via the IR method [3, 4].

2.3. Video Streaming over RTP and RTCP. After encoding an input video stream via the video coding layer, the H.264/AVC encoder packetizes the encoded bits into RTP packets for network transmission. RTP is a transport layer protocol that has been developed to carry the encoded video sequence on top of IP and UDP [9]. RTPs companion protocol, RTCP, is used to monitor the transmission status of the media data and provide feedback information including the reception quality [10, 11]. The streaming video server multicasts RTP video packets with sender report (SR) type of RTCP to all clients and the clients reply with receiver reports (RRs) to inform the sender and other receivers about the quality of service. In this way, RTCP RR packets can provide end-to-end feedback information about delay jitter and packet-loss performance [12, 13]. Based on this feedback channel information, the encoder can change its coding strategy to reduce errors and adapt to changing network conditions.

3. Proposed Network-Aware Streaming Video System

3.1. Requirement of Streaming Video System. Typically, the number of reference frames is set to be large for the high coding efficiency that is set up during the initial video

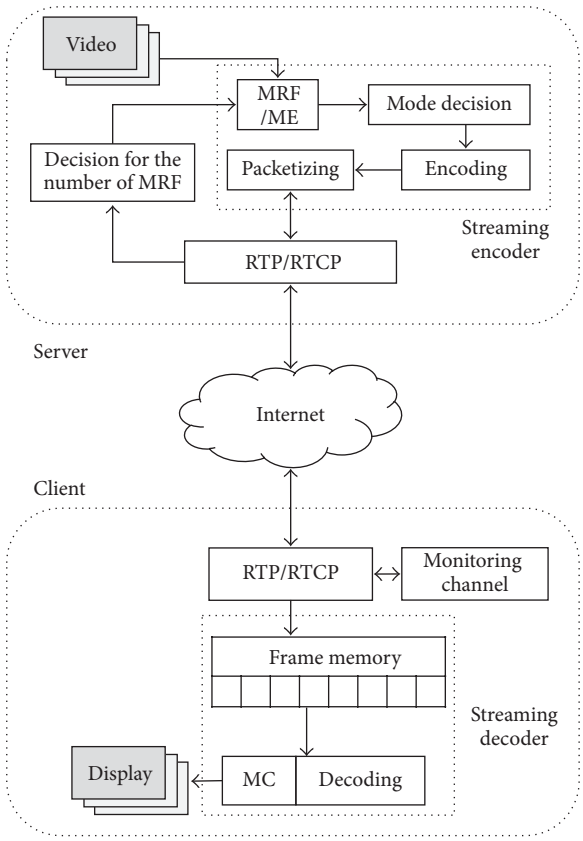


Figure 1: Proposed streaming server and client system.

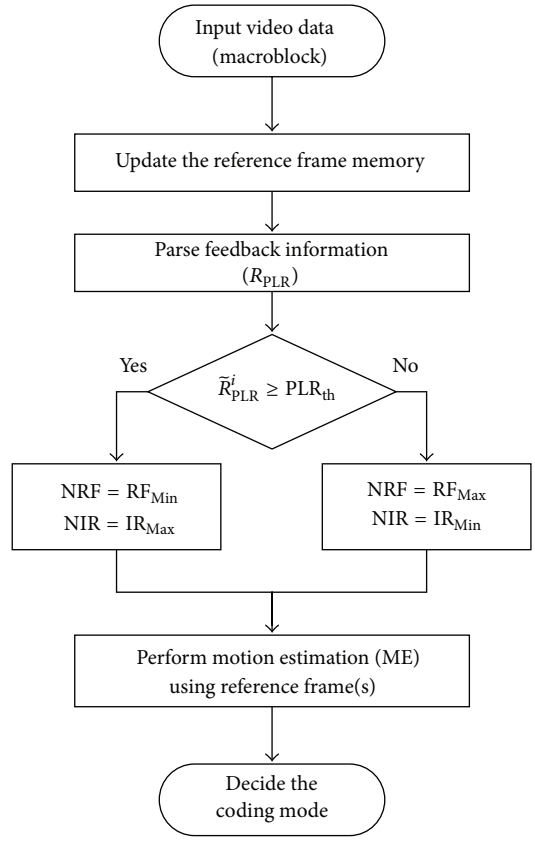


Figure 2: Network-aware decision for the number of reference frames and MBs used in intrarefresh coding.

encoding. This multiple reference frame structure is preferred to single reference frame in the recent video coding methods, even though it requires a large amount of frame memory and computation power for motion estimation and compensation. However, the multiple reference frame coding method combined with typical error-resilience function such as RIR has been found to make the overall video quality worse in an erroneous network [3, 4].

Typical error-resilience methods, that is, FMO [2], ASO, and DP, can not maintain their resilience features under the multiple reference frame structure. Therefore, there should be a strategy combining multiple reference frame structure for high coding efficiency and resilience function for strong error-resilience feature. The strategy is introduced in this paper.

3.2. Proposed Network-Aware Error Resilience. In this section, a network-aware error-resilient streaming video system is proposed. The proposed system monitors RTCP feedback messages (including the channel status) delivered by the client and manages the number of reference frames so as to mitigate error propagation. Additionally, the number of MBs to be intracoded forcefully per frame can be added in this procedure.

Figure 1 shows the proposed streaming server and client system. The streaming server has additional functions to monitor the channel status delivered in the form of RTCP packets and to change the number of reference frames and intracoded MBs being inserted into a frame based on the channel status. After deciding on a suitable number of reference frames and MBs, the streaming server encodes the video and assembles it into RTP packets. The size of a packetized unit is decided by the input parameters of the encoder; then, the packet is delivered to network [14]. At the streaming client, the quality of the decoded video sequence should be observed and returned to the streaming server to control the error resilience. However, it is difficult to measure the quality of the decoded video because there are no undamaged reference frames at the client side. Therefore, the proposed streaming client measures the refined packet loss (R_{PLR}) instead of the decoded video quality. R_{PLR} is smoothed by exponential weighted moving average (EWMA) method that is defined as

$$\widetilde{R}^i_{PLR} = (1 - \alpha)\, \widetilde{R}^{i-1}_{PLR} + \alpha R_{PLR}, \quad (1)$$

where α is a weighting factor that defines the acceleration of averaging and $\widetilde{R}^i_{PLR}$ is the estimated packet-loss ratio (PLR). The delivered $\widetilde{R}_{PLR}$ value in RTCP packet is compared to the predefined PLR threshold PLR_{th} at the streaming server to determine the number of reference frames. The procedure for PLR comparison and controlling the number of reference frames and intracoded MBs is shown in Figure 2.

Table 1: Encoder parameters.

Profile level	Baseline (66)
GOP structure	IPPP...
Bit rate	400 kbps (CBR)
Num. of reference frames	1, 7 (1~7)
Num. of intrarefresh MBs	6, 18
Entropy coding	CAVLC
Packet type	RTP
Reference S/W ver.	JM 17.2 [15]

Table 2: Test video sequences.

Input video	*akiyo*	*carphone*	*football*	*soccer*
Total frames	240 frames			
Frame rate	30 fps			
Resolution	CIF (352 × 288)			
Motion activity	Slow	Slow	Fast	Fast

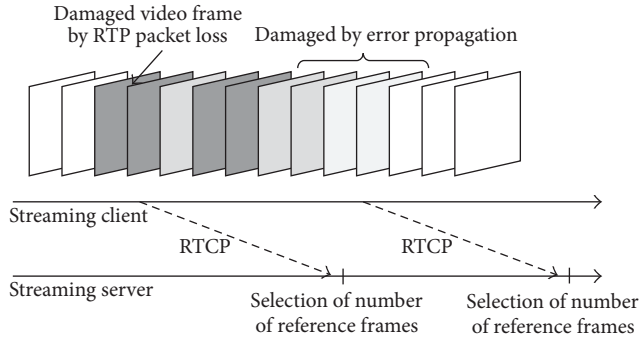


Figure 3: Simulation scenarios for the proposed network-aware reference frame control system.

We present a strategy for achieving high error robustness in the multiple reference frame based streaming video encoding system. The proposed streaming server reduces the number of reference frames (NRF) to RF_{Min} which is the smallest number of reference frames, when $\widetilde{R}_{PLR}$ is greater than PLR_{th}. When there are video frames that are found damaged, the streaming video system makes the number of reference frames small. Then, refreshing feature of the intracoded MBs can be effective [3, 4]. That is, more blocks refreshed by RIR coding will be used for further motion compensation of the next frames in multiple reference frame structures. After the transmission channel becomes stable, that is, $\widetilde{R}_{PLR}$ being less than PLR_{th}, the streaming video encoder increases the number of reference frames up to RF_{Max}, which is the largest number of reference frames. Additionally, the higher number of MBs for intracoded mode, NIR, can be used together with the proposed system to achieve higher error resilience because the intrarefresh will be more effective when the error propagation has been mitigated.

In this way, the proposed system can strike a balance between high coding efficiency (for the stable channel) and error robustness (for the unstable channel).

4. Performance Evaluation

4.1. Experimental Setup. The proposed network-aware reference frame control system is expected to achieve both coding efficiency and error robustness by monitoring the channel status and making suitable adjustments.

For the experiments, reference software of H.264/AVC standard named JM is used. As it is well known, JM includes encoder, decoder, and rtp_loss model. Specifically, the proposed error-resilience method has added to the encoder of JM for the proposed streaming video system. Table 1 shows the encoder parameters used in the experiments. Both the proposed network-aware streaming video system and the conventional streaming system use the same encoding parameters. Additionally, both systems encode the same test sequences shown in Table 2 in baseline profile and packetize the compressed video sequences into RTP packets. To compare the performance of the proposed system under varying conditions, input video sequences depicting different motion activities are used. For example, the *akiyo* and *carphone* sequences have slow motion and a static background. Thus, they are less damaged than the high-activity video sequence when video packets are lost. In contrast, the *football* and *soccer* sequences consist of fast motion.

To simulate the various error-prone wireless networks, both a bursty pattern and a random pattern for packet losses are created as shown in Figure 3. Also, in order to observe the performance of error resilience reducing error propagation, the severe channel error conditions should be considered. That is, 20% packetized frames are lost in the forms of burst error and random error over one second at the simulation time of 2 seconds after starting the streaming video. To decode and analyze the quality of damaged video stream, decoder of JM is used. If the transferred video packets are lost, the frame copy is used for error concealment at the decoder. The $\widetilde{R}_{PLR}$ observed at the streaming client is delivered to the server and is expected to be received after a transmission time of T_{Trans}. T_{Trans} is determined to have a uniform distribution between 1 s and 2 s. Therefore, the high coding efficiency is required before channel error starts, while error-resilience feature is more expected after channel errors. In our experiment, we set the the minimum and maximum numbers of reference frames, RF_{Min} and RF_{Max}, to 1 and 7, respectively. At the same time, 6 and 18 intracoded MBs per CIF picture have been forcefully generated for RIR.

In this experiment, the proposed network-aware reference frame control system using either RF_{Min} or RF_{Max} based on channel condition, named NARF, is compared to two conventional streaming systems, NRF1 and NRF7. Here, NRF1 and NRF7 refer to the conventional system using the static number of reference frames of 1 and 7, respectively.

4.2. Experimental Results and Analysis. For the first error-resilience experiment, we applied a burst error pattern to simulate a worse situation like handover in the mobile network. As shown in Figure 3, the damage to the test sequences is observed for a duration of 1 s at the simulation

Table 3: Average PSNR values observed for test sequences under the burst error conditions.

	akiyo		*carphone*		*football*		*soccer*	
	RIR6	RIR18	RIR6	RIR18	RIR6	RIR18	RIR6	RIR18
				Phase 1 (error-free)				
NRF1	43.97	43.43	37.71	37.44	27.76	27.61	33.06	32.82
NRF7	44.20	43.67	38.28	38.00	27.83	27.74	33.21	32.93
NARF	44.20	43.67	38.28	38.00	27.83	27.74	33.21	32.93
				Phase 2 (after burst error)				
NRF1	31.74	36.11	26.23	25.87	26.74	28.20	28.40	27.00
NRF7	32.23	34.07	25.32	27.22	21.04	24.92	27.33	22.54
NARF	32.88	36.90	26.43	27.83	22.89	26.79	27.83	27.54

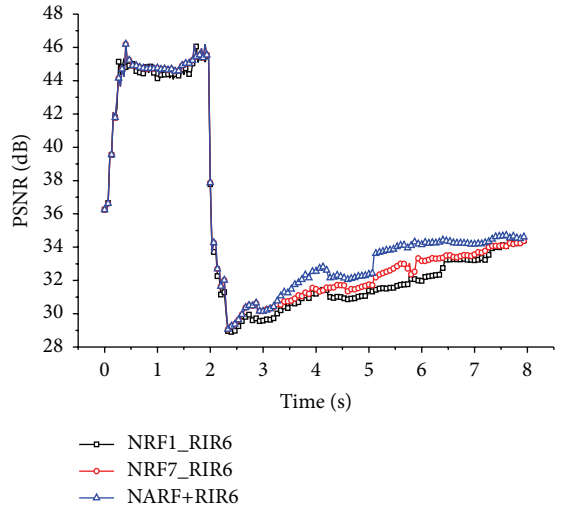


Figure 4: *akiyo* sequence under the burst error condition with RIR6.

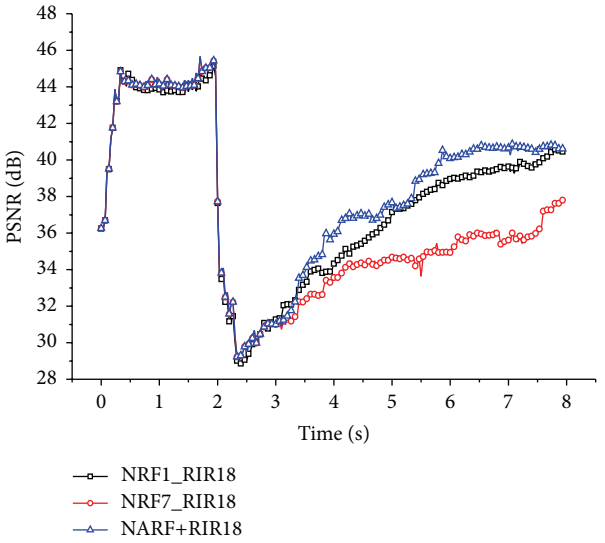


Figure 5: *akiyo* sequence under the burst error condition with RIR18.

time of 2 s. All PSNR results of test sequences are shown in Figures 4–11. Before the packet losses occur, the conventional system of NRF7 shows the more enhanced PSNR result at all test sequences than NRF1 (as shown in Table 3). However, the PSNR result of NRF7 can be worse as the video damage is accumulated and propagated, even though 6 and 18 intrarefresh MBs are inserted. That is, the error-resilience feature of RIR does not work properly under the condition of multiple reference frame structure. However, NRF1 tends to generate the better error recovery feature because its single reference frame structure can reduce the error propagation. Specifically, NRF1 in both *football* and *soccer* sequences shows the stronger error resilience than that in both *akiyo* and *carphone* sequences, because *football* and *soccer* sequences consisting of faster motions have more serious error propagation than others. Also, we could measure another trade-off between NRF1 and NRF7. When small number of previous frames, that is, one or two frames, are damaged, NRF7 partially shows better PSNR performance than NRF1 because NRF7 can perform the motion compensation in the current frame with the blocks in the far-located (undamaged) frames in the multiple reference frame. Therefore, NRF1 does not always show better PSNR than NRF7 even after the packet losses are detected.

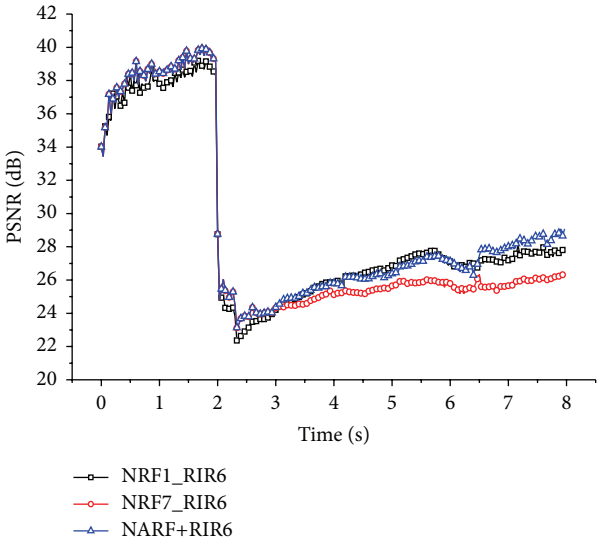


Figure 6: *carphone* sequence under the burst error condition with RIR6.

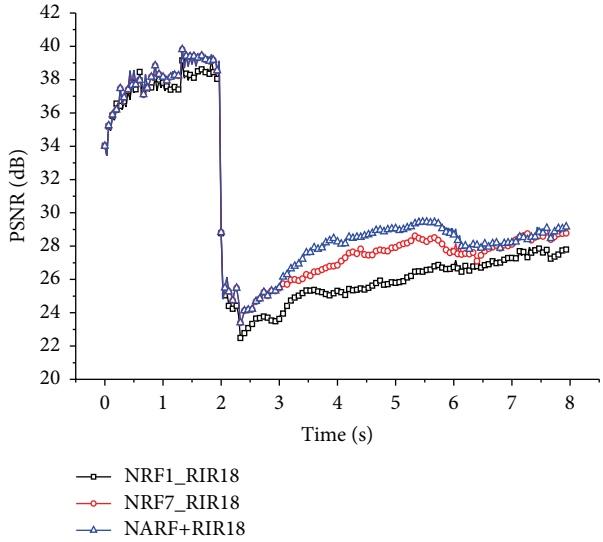


Figure 7: *carphone* sequence under the burst error condition with RIR18.

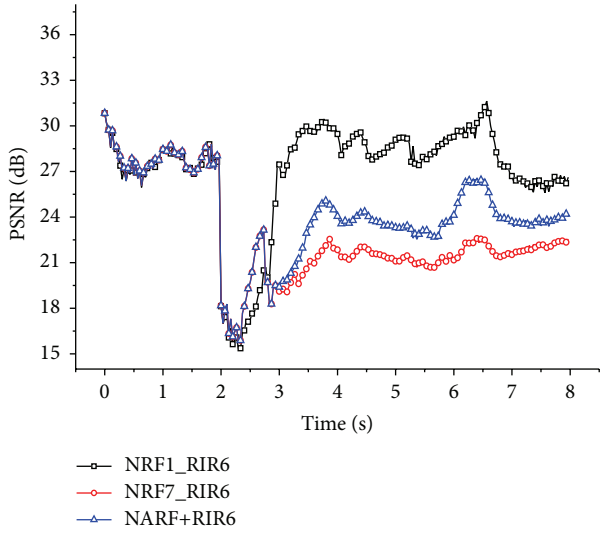


Figure 8: *football* sequence under the burst error condition with RIR6.

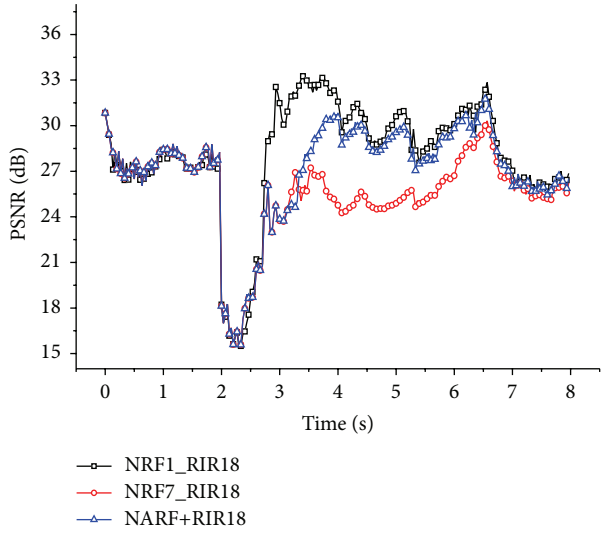


Figure 9: *football* sequence under the burst error condition with RIR18.

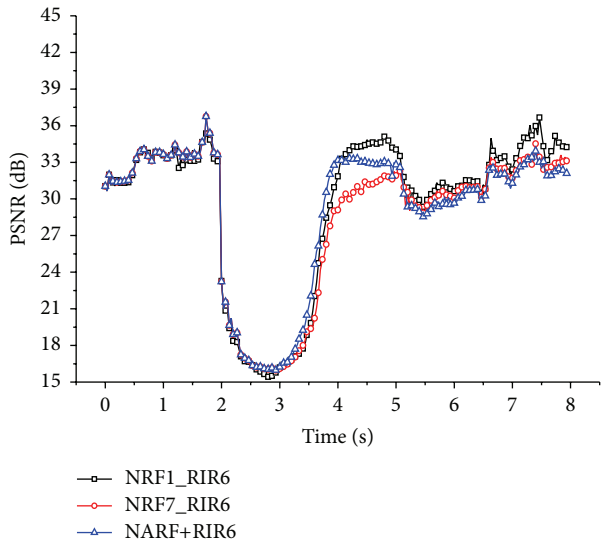


Figure 10: *soccer* sequence under the burst error condition with RIR6.

On the other hand, the proposed NARF system exhibits stronger error robustness than NRF7 for all test sequences because of its channel adaptiveness. The proposed NARF uses RF_{Max} for the higher coding efficiency at the error-free condition, whereas it uses RF_{Min} for the error restoration performance after recognition of channel errors. Indeed, NARF maintains RF_{Max} during the time period when channel errors are notified to the streaming server. This coding balance between RF_{Max} and RF_{Max} makes the proposed NARF perform better than NRF7 all the time and better than NRF1 in some error conditions. It can be effective at both coding efficiency and error robustness. Figure 12 presents representative images of the *soccer* sequence encoded under the various reference frame conditions. Additionally, the average PSNR values observed for the test sequences are shown in Table 3. Here, phase 1 implies the error-free period from 0 s to 2 s at the simulation time, while phase 2 indicates the erroneous period from 2 s to the end of simulation time. During phase 1, NRF7 and NARF using RF_{Max} perform better than NRF1. However, during phase 2, NRF1 and NARF using RF_{Min} perform better than NRF7. That is, the proposed NARF manages its encoding to work like NRF7 at the error-free time and NRF1 at the erroneous time.

For the second error-resilience experiment, we applied the random error pattern to the simulation which is created by random function in the standard C library with rtp_loss model in the JM reference software. Here, the same encoding conditions as in the first experiment (shown in Tables 1 and 2) are used.

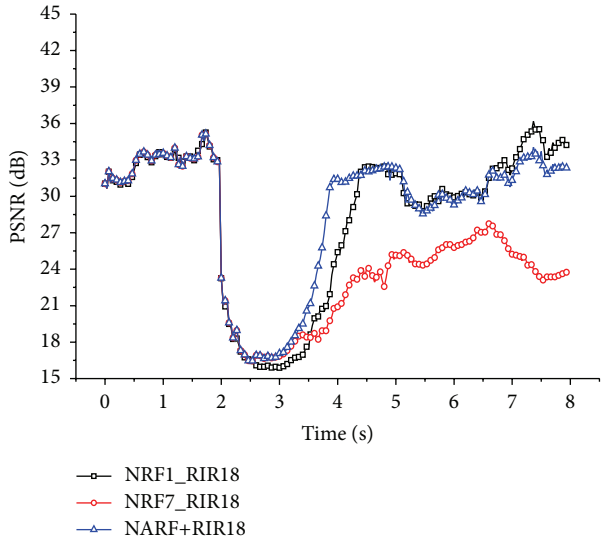


Figure 11: *soccer* sequence under the burst error condition with RIR18.

Figure 12: Visualized *soccer* sequence under the burst error conditions, (a) no damage, (b) NRF1, (c) NRF7, and (d) NARF.

Experimental results of NARF and conventional streaming systems performed under the random errors are shown in Figures 13–20. From the simulation time of 2 s, random number of frames are lost during 1 s. As the same with the first simulation, the NARF system shows better coding efficiency and error resilience than the cases of NRF1 and NRF7 in both cases of RIR6 and RIR18. Figure 21 presents representative images of *football* sequence under various reference frame conditions. The average PSNR values observed for the test sequences are shown in Table 4.

The observation results indicate that the proposed streaming method is effective at achieving more reliable transmission of streaming video because it strikes a balance between coding efficiency and error-resilience features.

Table 4: Average PSNR values observed for test sequences under the random error condition.

	akiyo		*carphone*		*football*		*soccer*	
	RIR6	RIR18	RIR6	RIR18	RIR6	RIR18	RIR6	RIR18
				Phase 1 (error-free)				
NRF1	43.97	43.43	37.71	37.44	27.76	27.61	33.06	32.82
NRF7	44.20	43.67	38.28	38.00	27.83	27.74	33.21	32.93
NARF	44.20	43.67	38.28	38.00	27.83	27.74	33.21	32.93
				Phase 2 (after random error)				
NRF1	38.98	39.49	26.18	26.69	22.92	28.26	24.96	27.11
NRF7	38.44	38.37	26.93	27.13	22.87	26.53	24.59	27.34
NARF	38.51	40.13	27.11	27.94	23.51	27.86	28.56	27.22

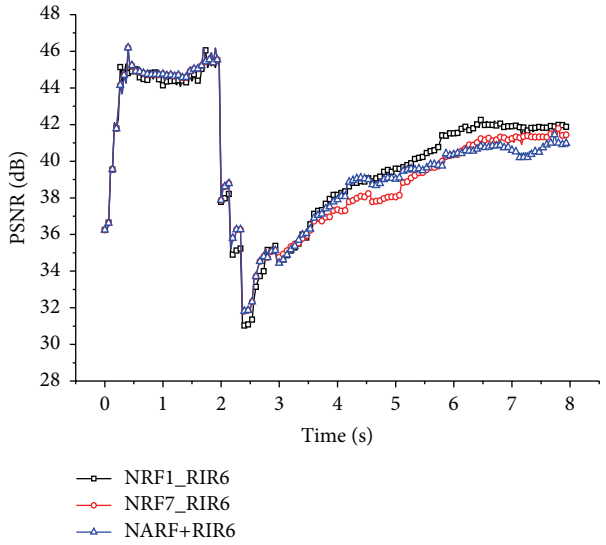


Figure 13: *akiyo* sequence under the random error conditions with RIR6.

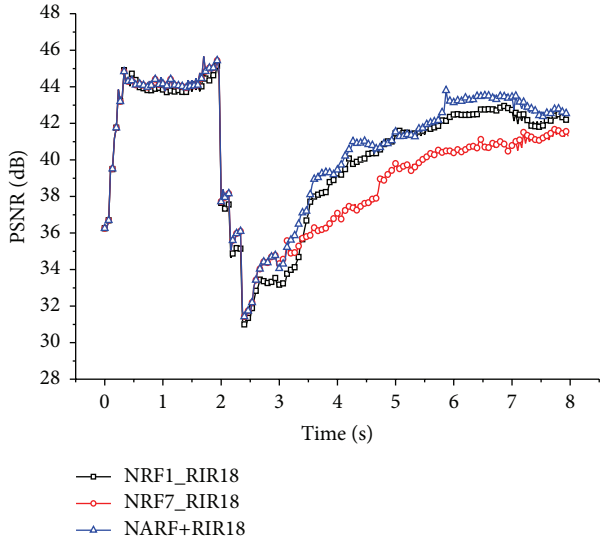


Figure 14: *akiyo* sequence under the random error conditions with RIR18.

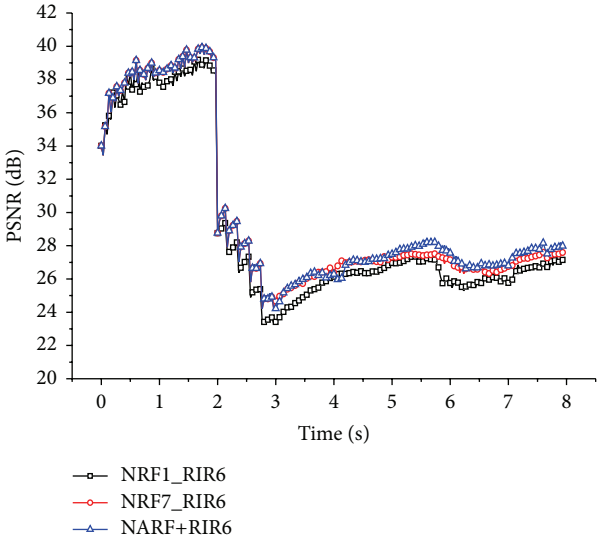


Figure 15: *carphone* sequence under the random error conditions with RIR6.

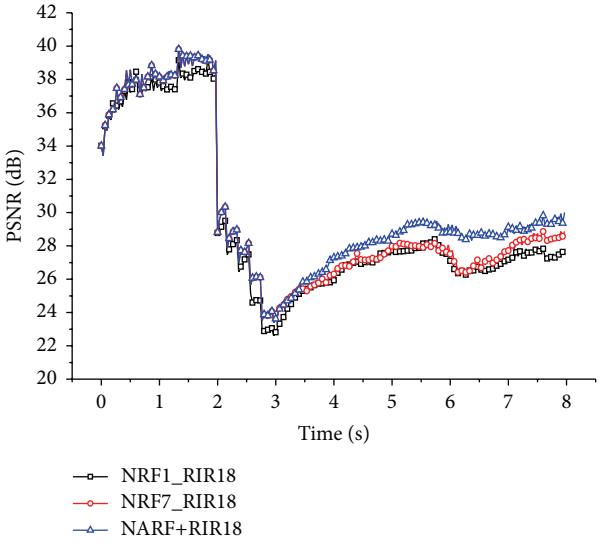


Figure 16: *carphone* sequence under the random error conditions with RIR18.

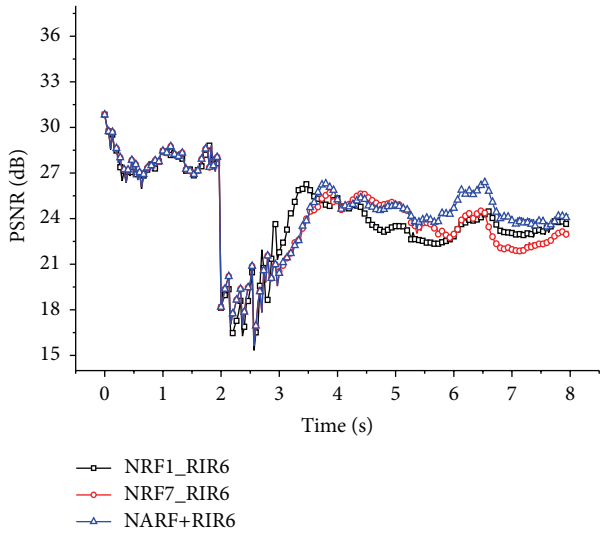


Figure 17: *football* sequence under the random error conditions with RIR6.

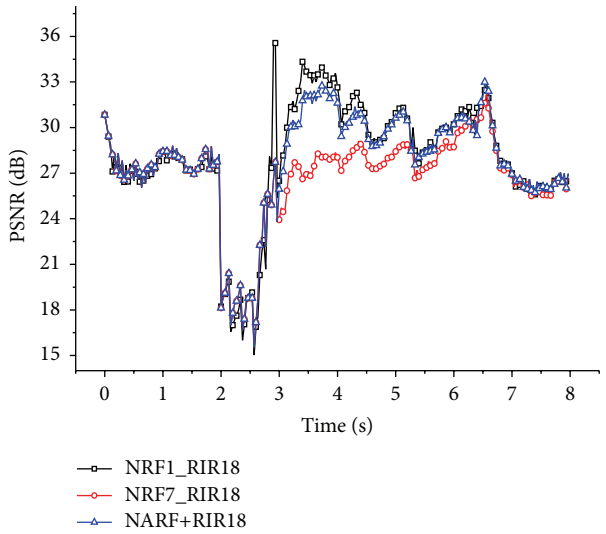


Figure 18: *football* sequence under the random error conditions with RIR18.

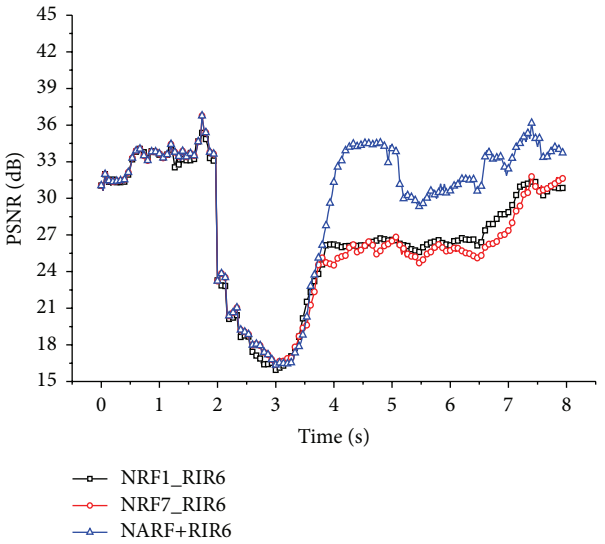


Figure 19: *soccer* sequence under the random error conditions with RIR6.

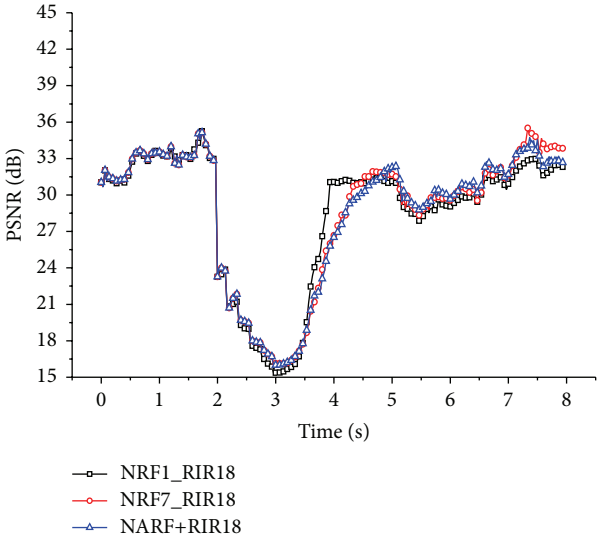
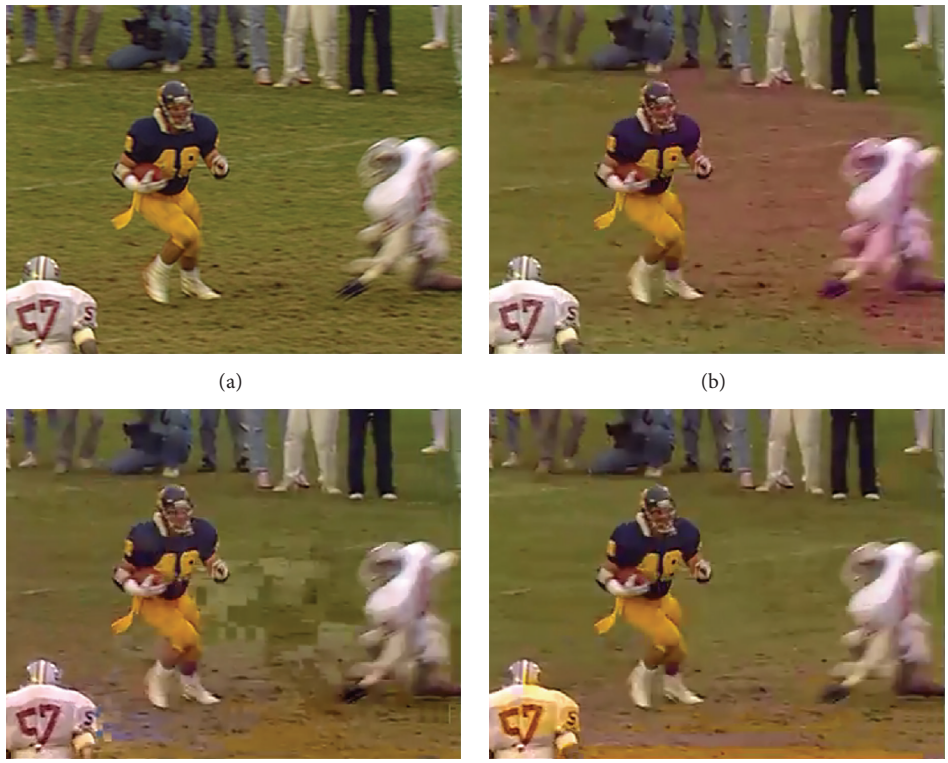


Figure 20: *soccer* sequence under the random error conditions with RIR18.

5. Conclusion

The requirements of reliable real-time streaming video services in the recent mobile and wireless communication environments are enormous. However, these channels remain unreliable while consumer demand for streaming video increases. Therefore, video coding standards typically include both error-resilience tools to cope with the error propagation and multiple reference frame structures to achieve higher coding efficiency. However, error-resilience tools decrease coding efficiency. In addition, the multiple reference frame structure used for higher coding efficiency in motion estimation and compensation interferes with conventional error-resilience tools. In this paper, we propose a network-aware reference frame control system that keeps a balance between coding efficiency and error resilience based on the channel status. Experimental results show that the proposed video streaming system provides better PSNR approximately from 0.2 to 0.5 dB than conventional video streaming system (NRF1) using single reference frame when no video frames are damaged and better PSNR from 0.3 to 3 dB than conventional system (NRF7) using 7 reference frames when the video frames are damaged. That is, the proposed streaming video system maintains the high video quality by controlling the number of reference frames based on the channel status. In addition, the proposed system can adopt the channel-adaptive intrarefresh methods to increase the error robustness.



(a) (b)

(c) (d)

Figure 21: Visualized *football* sequence under the random error conditions, (a) no damage, (b) NRF1, (c) NRF7, and (d) NARF.

Competing Interests

The authors declare that they have no competing interests.

Acknowledgments

This study was supported by research funds from Chosun University, 2015.

References

[1] T. Wiegand, G. J. Sullivan, G. Bjøntegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 560–576, 2003.

[2] T. H. Vu and S. Aramvith, "An error resilience technique based on FMO and error propagation for H.264 video coding in error-prone channels," in *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME '09)*, pp. 205–208, New York, NY, USA, July 2009.

[3] S. Moiron, I. Ali, M. Ghanbari, and M. Fleury, "Limitations of multiple reference frames with cyclic intra-refresh line for H.264/AVC," *Electronic Letters*, vol. 47, no. 2, pp. 103–104, 2011.

[4] S. I. Chowdhury, J.-N. Hwang, P.-H. Wu, G.-R. Kwon, and J.-Y. Pyun, "Error resilient reference selection for H.264/AVC streaming video over erroneous network," in *Proceedings of the IEEE International Conference on Consumer Electronics (ICCE '13)*, pp. 418–419, IEEE, Las Vegas, Nev, USA, January 2013.

[5] P. Nunes, L. D. Soares, and F. Pereira, "Error resilient macroblock rate control for H.264/AVC video coding," in *Proceedings of the IEEE International Conference on Image Processing (ICIP '08)*, pp. 2132–2135, San Diego, Calif, USA, October 2008.

[6] Y. Xu and C. Zhu, "End-to-end rate-distortion optimized description generation for H.264 multiple description video coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 23, no. 9, pp. 1523–1536, 2013.

[7] R. M. Schreier and A. Rothermel, "Motion adaptive intra refresh for the H.264 video coding standard," *IEEE Transactions on Consumer Electronics*, vol. 52, no. 1, pp. 249–253, 2006.

[8] X. Wang, H. Cui, and K. Tang, "Attention-based adaptive intra refresh method for robust video coding," *Tsinghua Science and Technology*, vol. 17, no. 1, pp. 67–72, 2012.

[9] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RTP: a transport protocol for real-time applications," Tech. Rep. RFC 3550, 2003.

[10] M. Sivabalakrishnan and D. Manjula, "Analysis of decision feedback using RTCP for multimedia streaming over 3G," in *Proceedings of the International Conference on Computer and Communication Engineering (ICCCE '08)*, pp. 1023–1026, IEEE, Kuala Lumpur, Malaysia, May 2008.

[11] H. Gharavi, K. Ban, and J. Cambiotis, "RTCP-based frame-synchronized feedback control for IP-video communications over multipath fading channels," in *Proceedings of the IEEE International Conference on Communications*, vol. 3, pp. 1512–1516, June 2004.

[12] N. Baldo, U. Horn, M. Kampmann, and F. Hartung, "RTCP feedback based transmission rate control for 3G wireless multimedia streaming," in *Proceedings of the 15th IEEE International*

Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC '04), vol. 3, pp. 1817–1821, IEEE, September 2004.

[13] L.-L. Wang and J.-D. Lu, "IPTV quality monitoring system based on hierarchical feedback of RTCP," in *Proceedings of the International Conference on Electronics, Communications and Control (ICECC '11)*, pp. 870–873, Ningbo, China, September 2011.

[14] S. Wenger, "H.264/AVC over IP," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 645–656, 2003.

[15] K. Sühring, *H.264/AVC Reference Software—JM 17.2*, 2010, http://iphome.hhi.de/suehring/tml/.

Hindawi Publishing Corporation
Mobile Information Systems
Volume 2016, Article ID 4913216, 18 pages
http://dx.doi.org/10.1155/2016/4913216

Research Article

A Framework for QoE-Aware 3D Video Streaming Optimisation over Wireless Networks

Ilias Politis, Asimakis Lykourgiotis, and Tasos Dagiuklas

CONES Research Group, Hellenic Open University, 26335 Patras, Greece

Correspondence should be addressed to Ilias Politis; ilpolitis@eap.gr

Received 6 October 2015; Revised 5 February 2016; Accepted 29 February 2016

Academic Editor: Ansgar Gerlicher

The delivery of three-dimensional immersive media to individual users remains a highly challenging problem due to the large amount of data involved, diverse network characteristics, and user terminal requirements, as well as user's context. This paper proposes a framework for quality of experience-aware delivering of three-dimensional video across heterogeneous wireless networks. The proposed architecture combines a Media-Aware Proxy (application layer filter), an enhanced version of IEEE 802.21 protocol for monitoring key performance parameters from different entities and multiple layers, and a QoE controller with a machine learning-based decision engine, capable of modelling the perceived video quality. The proposed architecture is fully integrated with the Long Term Evolution Enhanced Packet Core networks. The paper investigates machine learning-based techniques for producing an objective QoE model based on parameters from the physical, the data link, and the network layers. Extensive test-bed experiments and statistical analysis indicate that the proposed framework is capable of modelling accurately the impact of network impairments to the perceptual quality of three-dimensional video user.

1. Introduction

Media-friendly Future Internet needs to cope with an increased demand for streaming applications and three-dimensional (3D) media by modifying and optimising existing protocols for streaming over a hybrid network environment. Accessing the Internet easily, any time, anywhere, and with any device, has changed user's Internet consumption amount and behaviour, especially on media consumption [1]. Traditionally web surfing and file sharing were the dominant applications, whereas today real-time streaming and social media applications (e.g., YouTube, Facebook, etc.) constitute the major portion of the Internet traffic. The change in Internet usage patterns and the increasing needs of novel applications (high performance, availability, security, etc.), together with end users' increasing quality of experience (QoE) expectations, open new challenges for the Future Internet due to increasing popularity of real-time communication, online social networks, and heterogeneity of user devices [2–4].

It is evident that due to recent advances in video acquisition techniques, coding, delivery, and displaying, 3D video has sprouted in the consumer domain through a range of applications and services which provide enhanced visual experience for the viewers. Recently, the research on 3D video coding and transportation intensified, flared by enhancements and updates on coding strategies, such as H.264/SVC (Scalable Video Coding) [5], H.264/MVC (Multiview Video Coding) [6], and High Efficiency Video Coding (HEVC) [7], as well as the wide deployment of LTE (Long Term Evolution) systems for high-speed data delivery to both wireless and mobile users [8].

Based on the different representation formats for stereoscopic and multiview videos, different coding schemes have evolved over time to compress such media efficiently. Pure colour based (e.g., multiview, stereo L-R) or depth based formats (e.g., colour-plus-depth or Layered Video plus Depth) are usually compressed using the legacy block based coding standards, such as MPEG-4 Part 10/H.264 Advance Video Coding (AVC), or its extensions, such as SVC [9] or MVC [10]. AVC has also introduced new profiles to support coding of stereoscopic videos, such as Stereoscopic High Profile [11]. Both AVC derived codecs, namely, SVC and MVC, get their base layer fully compliant with AVC specifications in order to be compliant with legacy deployments. The purpose of SVC is

to provide a universal media bit-stream that can be decoded by multiple decoders of different capacities to produce the reconstructed media at different states. SVC also provides dynamic adaptation to a diversity of networks, terminals, and formats. This is extremely useful for simple adaptation of transmission and efficient storage and is beneficial for transmission services with uncertain resolution, channel conditions, and device types. The MVC standard, on the other hand, in order to impove the overall rate-distortion performance, exploits the inter-view redundancies through the Disparity Compensated Prediction (DCP) [12]. Nonetheless, the multiview bit rate increases to unmanageable levels as the number of source cameras increases. Therefore, it is impractical to encode and transport the entire set of views required to be displayed on most multiview displays, particularly when dealing with heterogeneous wireless access networks. Yet, it remains unclear how all these technologies affect the viewer's perception of 3D video [13, 14]. Apparently, it is still unrealistic to try and estimate the QoE of such application, without resorting to full subjective evaluations.

Any 3D video coding and delivery system will ultimately need to be measured in terms of user's satisfaction and perceived experience. QoE can be defined as the overall acceptability of an application or service strictly from the users point of view. It is a subjective measure of end-to-end service performance from the user perspective and it is an indication of how well the network meets the users needs [15]. Encompassing many different aspects, QoE is riveted on the true feelings of end users when they watch streaming video, listen to digitized music, and browse the Internet through a plethora of methods and devices [16]. There is a real challenge in creating models which will accurately perform learning to model service behaviours by taking into account parameters such as arrival pattern request, service time distributions, I/O system behaviours, and network usage [17]. The aim of such attempts is to estimate QoE for both resource-centric (utilization, availability, reliability, and incentive) and user-centric (response time and fairness) environments over certain predefined Quality of Service (QoS) and QoE thresholds. Evidently, there is still no concrete evidence to prove the correlation between QoS and QoE, particularly for 3D video. Nevertheless, there are several attempts to define the relationship of human perception of video quality to the inherently objective network conditions, and the majority of this work concludes that such a relationship cannot be linear [18, 19]. Specifically, authors in [20] proved the importance of the impact that network delivery speed and latency have on the human satisfaction but also determined that, in a heterogeneous networking environment, neither bandwidth nor latency is sufficient as attributes that indicate the range of issues, which render a service as nonappealing to the end user.

This paper describes a framework for QoE-aware delivering of 3D video across heterogeneous wireless networks. Towards this end, a synergy is proposed between the LTE EPC architecture and IEEE 802.21 protocol that will ensure real-time control and monitoring of key performance indicators and parameters relative to the perceived 3D QoE across different layers and entities. The proposed framework includes a QoE controller module housing a machine learning- (ML-) based decision making engine, which trains data collected from the user and the network planes in order to model, as accurate as possible, the perceptual 3D video quality. It needs to be underlined that the proposed architecture does not require any alteration of the standardised LTE EPC interfaces; hence, it could be smoothly integrated with the current mobile operators' deployments. The paper investigates machine learning-based techniques for producing an objective QoE model of network related impairments. The proposed QoE model is a function of parameters collected not only from the application layer, but also from the underlying layers. Previous efforts on modelling QoE for 3D video have been based on measuring the end-to-end packet loss and determining the perceived QoE, a process that would require feedback from the receiving end, which in the case of wireless networks may never arrive or may arrive too late for the decision engine. This research investigates a new methodology for monitoring network imposed impairments to the perceived video quality that will result in more accurate QoE models. Specifically, a set of QoS parameters is collected from different points in the delivery chain (i.e., bit error rate from physical layer, MAC layer load, and network layer delay/jitter), which account for the overall number of lost packets. Three classification schemes with different characteristics have been studied in order to identify the most appropriate method for modelling 3D video quality due to network impairments.

The rest of the paper is organised as follows. Section 2 briefly presents the concept of LTE EPC and its main entities that are relevant to the purposes of this study. Section 3 provides a detailed description of the proposed media-aware framework and focuses on the synergy between the Media-Aware Proxy, the IEEE 802.21 protocol, and the QoE controller, which enables collecting QoE related key performance indicators across multiple layers and network components, the training of these data, and the modelling of the perceived 3D video QoE. Additionally, in this section the three ML-based classification models are introduced and their main differences are explained. The setup of the experimental testbed is described in Section 4. Moreover, the section defines the physical, MAC, and network layer parameters, whose impact on the perceptual quality of 3D video is under study. The collected Mean Opinion Score (MOS) from the subjective experiments, along with a detailed statistical analysis, is presented in Section 5. Based on the measured 3D MOS, Section 6 compares the performance of the three machine learning methods in terms of precision and accuracy, in order to determine the most suitable classification scheme for implementation as part of the QoE controller. Finally, Section 7 concludes the paper and highlights the future aims of this research.

2. 3GPP EPS Architecture

The technical realization of next generation mobile networks requires that complementary wireless technologies are integrated in order to provide ubiquitous multimedia services "anywhere, any time, and on any device." In the

recent years, 3GPP's Evolved Packet System (EPS) [21] is increasingly deployed by mobile operators in order to satisfy the requirement for uninterrupted, high quality, real-time multimedia services across wireless access networks. EPS consists of the LTE wireless access, which is forming the Evolved UTRAN (E-UTRAN) as the lower part of EPS and an IP connectivity control platform called Evolved Packet Core (EPC) as the upper part of EPS, which enable wireless access networks diversity (i.e., LTE, UMTS, WiMax, WiFi, etc.). EPS is relying on standardized routing and transport mechanisms of the underlying IP network. In parallel, EPS provides capabilities for coexistence with legacy systems and migration to the EPS, while it supports functionalities for access selection based on operators policies, user preferences, and networking conditions. One of the most important features of EPS is its ability to support simultaneous active Packet Data Network (PDN) connections for the same user equipment, whilst maintaining the negotiated QoS across the whole system.

3GPP has proposed the EPC in an effort to support higher data rates, lower latency, packet optimisation, and multiple radio access technologies [22]. As part of EPS, EPC is an all-IP architecture that fulfils those requirements and supports the characteristics of E-UTRAN. The advantages of EPC, as opposed to the legacy GPRS architecture, are a more clear depiction of control and data planes, a more simplified architecture with a single core network, and, finally, the full assumption of the IETF protocols. Therefore, EPC allows for a truly converged packet core for trusted 3GPP networks (i.e., GPRS, UMTS, LTE, etc.), non-3GPP networks (i.e., WiMAX), and untrusted networks (i.e., WLAN). Additionally, EPC maintains seamless mobility and consistent and optimised services provisioning independent of the underlying access network.

The proposed framework for QoE-aware delivery of 3D video content over heterogeneous wireless networks relies on 3GPP EPC. Therefore, a description of the EPC modules related to the study follows. The functionalities of the discussed modules are enhanced as part of the proposed framework in order to support seamless connectivity, transparent real-time adaptation of 3D video streaming, and QoE-aware monitoring and management.

2.1. Key EPC Modules. The relevance to the proposed framework EPC modules [22] is depicted in Figure 1. Specifically, these architectural modules and functions include the following:

(i) Home Subscriber Server (HSS) is the main database of the EPC responsible for storing subscriber's information that may include profile description, several identification information, and information regarding the subscriber's established sessions. The module interfaces with the Mobility Management Entity and the Authentication, Authorization, and Accounting server (implemented as a Diameter server [23]).

(ii) Mobility Management Entity (MME) is the module that performs the mobility management and bearer management and establishment. It is also responsible for performing mobility functions during handovers between 2G or 3G 3GPP access networks. The module has interfaces to the Serving-GW, the eNodeB, and the HSS.

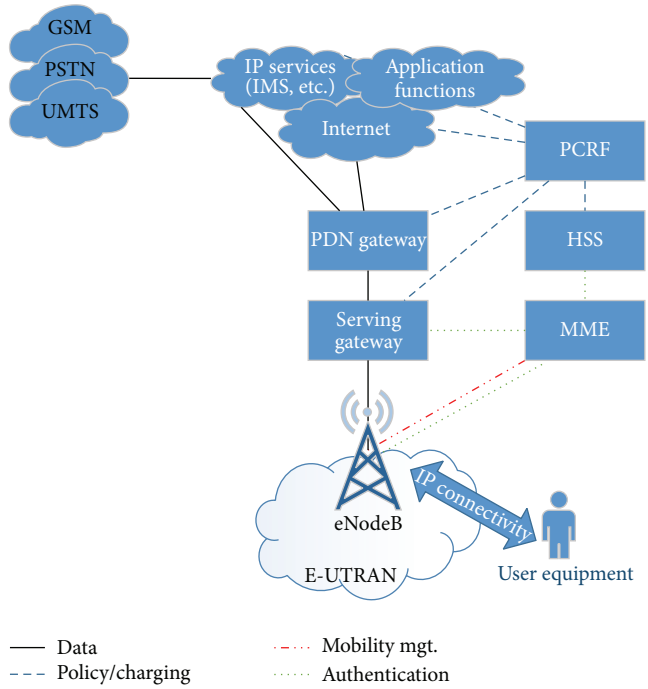


Figure 1: Relevant EPC architectural elements.

(iii) Serving Gateway (Serving-GW) acts as the gateway where the interface towards E-UTRAN terminates. Each piece of user equipment (UE) associated with the EPS is served by a single Serving-GW. In terms of implementation, both the Serving-GW and the PDN-GW may be dwelling in a single physical component.

(iv) Packet Data Network Gateway (PDN-GW) is the gateway which terminates the interface towards the packet data networks (e.g., IMS [24], Internet). In case where the UE is connected with different IP services, several PDN-GW are associated with this UE. It performs UE IP allocation, policy enforcement, packet filtering per user, and packet screening. Moreover, PDN-GW acts as the mobility anchor between 3GPP and non-3GPP access. It has interfaces to the Serving-GW, the Evolved Packet Data Gateway (ePDG), which is responsible for interworking between the EPC and untrusted non-3GPP networks, such as WiFi, interface to the 3GPP AAA server, the trusted non-3GPP access gateway, and the Policy and Charging Rules Function.

(v) Policy and Charging Rules Function (PCRF) performs QoS control, access control, and charging control. The function is policy based and authorizes bearer and session establishment and management. A single PCRF is assigned to all the connections of a subscriber. It interfaces with Policy and

Charging Enforcement Function (PCEF) within the PDN-GW and the Bearer Binding and Event Reporting Functions (BBERFs) within the trusted/untrusted 3GPP/non-3GPP access gateways and the HSS.

(vi) Evolved-UTRAN (E-UTRAN), as explained above, is a simplified radio access network architecture comprised of evolved NodeBs (eNodeBs). In terms of functionality E-UTRAN supports radio resource management (RRM) and selection of MME and is responsible for routing user data to the Serving-GW and for performing scheduling, measurements, and reporting.

The described EPC modules have been implemented as part of the *OpenEPC* testing platform [25], and they have been extended within the context of this work in order to support QoE-aware multimedia delivery, as described in the rest of the paper.

3. Proposed Media-Aware Architecture

The media-aware architecture proposed in this paper aims to provide seamless end-to-end services with ubiquitous use of the heterogeneous technologies. In the centre of this architecture lies a QoE controller, which facilitates a ML-based model of the quality of the perceived video experience. The learning process is based on a collection of key performance indicators from across the network architecture and from different layers. This information is stored in a central database, as described in the rest of this section. Evidently, the proposed framework is required to provide seamless, uninterrupted, and QoE-aware video services across heterogeneous link layer interfaces and user devices. Therefore, the proposed media-aware framework integrates a novel entity named Media-Aware Proxy (MAP) with the IEEE 802.21 Media Independent Handover (MIH) protocol [26]. In order to support seamless handover and IP layer mobility, the proposed framework supports also the Proxy Mobile IP; however, this is beyond the scope of the current research.

This framework is tightly coupled with the LTE EPC functions and modules [22]; however, it neither interrupts the inherent process of EPC nor violates its protocols, as it lies just before the core network. The synergy of MAP and MIH along with a QoE controller, which acts as the QoE-aware decision making function, allows for real-time video stream adaptation (i.e., intelligent quality layer dropping or stream prioritization) based on multiple physical, network, and application layer parameters, collected from both the mobile terminal and the access network side. An overview of the proposed architecture is illustrated in Figure 2.

3.1. Media-Aware Proxy. MAP is defined as a transparent user-space module responsible for low-delay adaptation and filtering of scalable video streams [27]. In conjunction with the QoE controller, which models QoE and informs MAP about the predicted level of the perceived quality, the latter is able to either drop or forward packets that carry specific layers of a stream to the receiving video users. In detail, MAP is a network function based on the Media-Aware Network Element (MANE) standard [28, 29]. Briefly, MANE can be considered as either a middle box or an application layer getaway capable of aggregating or thinning RTP streams by selectively dropping packets that have the less significant impact on the user's video experience. As such, MANE has been proposed as an intermediate system that is capable of receiving and depacketising RTP traffic in order to customise the encapsulated network abstraction layer units, according to client's and access network's requirements. Within the context of the proposed media-aware architecture, MAP's role is twofold. Firstly, it acts as a central point of decision in order to overcome networking limitations imposed by firewalls and Network Address Translation (NAT) protocol that are extensively used in real life networks. Secondly, it receives and parses RTP streams and customises the streaming according to the video client's requirements and network conditions, based on the QoE-aware decision engine of the QoE controller.

In particular, MAP, which is designed to run in Linux kernel level in order to ensure minimum impact on the end-to-end delay, acts as a transparent proxy of the mobile client and it is responsible for parsing the packets that are destined for all mobile users over multiple ports. Each received packet is forwarded in a queue and its header is parsed by an RTP parser process in order to identify the embedded video related information, without changing the header's fields. Moreover, MAP has the responsibility to store this information in media independent information service (MIIS) database. MAP is designed as a MIH-aware entity; hence, it can directly store information required from the QoE model to the MIIS database. Specifically, MAP regularly updates the database with all client IDs (i.e., video service, video view ID, number of layers, incoming data rate, etc.); thus, it enables the QoE controller to train the collected data and predict the perceptual video quality of experience. The timely output of the ML-based QoE model will be utilised by MAP in order to maximise the perceived video QoE by selecting the appropriate stream optimisation method. The abovementioned algorithmic process is shown in Figure 3.

3.2. IEEE 802.21 Overview. While the main purpose of IEEE 802.21 standard is to enhance the handover performance in heterogeneous networks [26], it also supports the important aspect of link adaptation. A resource-intensive multimedia application like video streaming requires link adaptation in order for the network provider to offer the maximum level of video experience. As a result IEEE 802.21 standard can be employed to support cooperative use of information available at both the mobile node and the network infrastructure. This framework introduces a new entity called MIH Function (MIHF), which resides within the protocol stack of the network elements and particularly between the link layer and the upper layers. Any protocol that uses the services provided by the MIHF is called a MIH user.

As shown in Figure 4, MIHF provides three types of services, the media independent event service (MIES), the media independent command service (MICS), and the media independent information service (MIIS). The MIES monitors link layer properties and initiates related events in order to

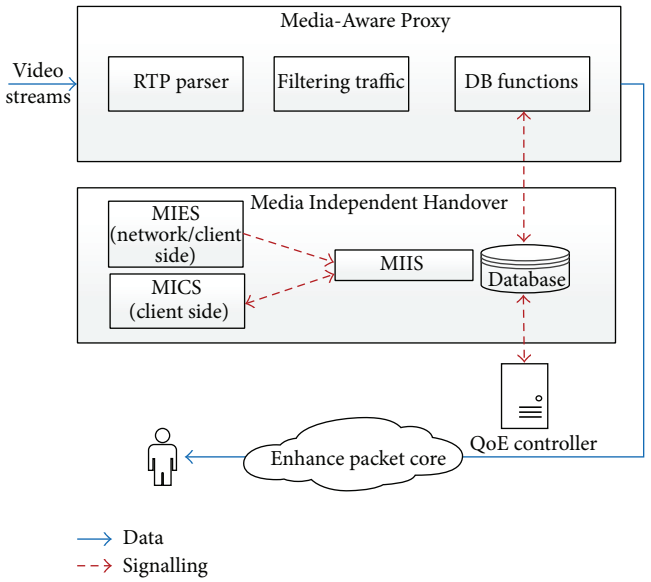


Figure 2: Proposed media-aware framework overview.

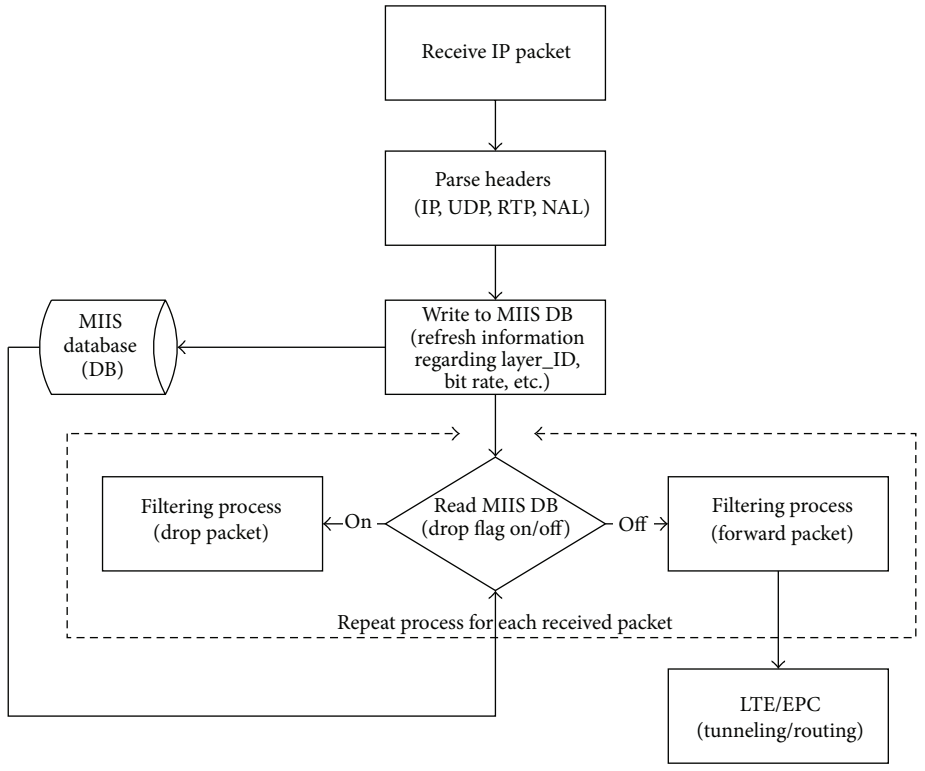


Figure 3: Algorithmic representation of Media-Aware Proxy.

inform both local and remote users. The MICS provides a set of commands for the MIH users to control link properties and interfaces. Finally, the MIIS provides information of static nature for the candidate access networks such as frequency bands, maximum data rate, and link layer address.

On the other hand, there is the 3GGP approach which defines a new component called access network discovery and selection function (ANDSF) [30], which in conjunction with the event reporting function (ERF) aims to facilitate information exchange for the network selection process. However, this approach is not as exhaustive and comprehensive as the MIES and can not support in depth monitoring of the link layer condition. For this reason, in the current work, the integration of MIH in the EPC architecture to support

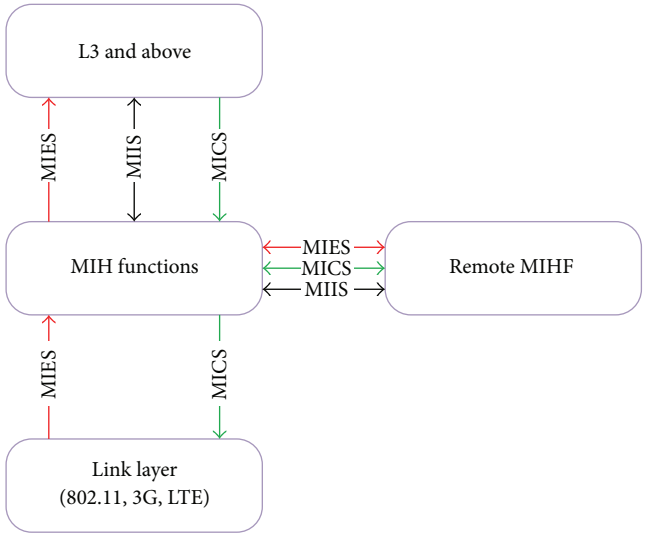


Figure 4: Illustration of overall MIH architecture and services.

link layer monitoring is proposed and the related signalling is presented. Furthermore, the available set of monitored characteristics is extended in order to include the application layer and, hence, to encompass video related attributes.

In order for the MIH framework to be effectively deployed in the EPC architecture, the MIHF must be installed at both network elements and mobile nodes. Additionally, an MIIS server must be introduced that has a database with the static attributes of the available access networks in the current EPC domain. We propose the extension of the MIIS server's functionalities so as to support the MIES and MICS services in order to store the reports of the involved entities and enforce policy rules applied by the QoE controller.

A focal point of the described EPC architecture is the PDN-GW, as it interconnects the various access routers such as the Serving-GW for LTE access and the evolved packet data gateway (ePDG) for non-3GPP access. PGW also provides access to external packet data networks like the Internet through the SGi interface. As a result, it is reasonable to propose that this extended MIH server should be colocated or, at least, should be at the vicinity of the PGW.

3.3. MIH in EPC Architecture. Within the context of this research, MIH is applied as a control mechanism responsible for monitoring and collecting QoE related parameters across different layers and network entities in near real time. Towards this end, MIH primitives including parameters reporting can be utilised. MIH supports different operating modes for parameters reporting. For instance, the `MIH_Link_Get_Parameters` command can be used to obtain the current value of a set of link parameters of a specific link through the `MIH_Link_Parameters_Report` primitive. Additionally, the same primitive can report the status of a set of parameters of a specific link at a predefined regular interval determined by a user configurable timer. Finally, the `MIH_Link_Configure_Thresholds` command can be used to generate a report when a specified parameter crosses a configured threshold. Subsequently, the reporting can be periodical, event-driven, or explicit depending on the desired approach or the available bandwidth for the monitoring functionality.

Table 1: Summary of QoE related KPIs.

Physical layer	Network layer	Application layer
Access technology	Throughput	Type of service
Signal strength	Delay	Spatial index
	Jitter	Temporal index
	Packet loss	Quality layer
		Quantization parameter

The extended MIH server will be the destination of these reports and will store the most recent values of the monitored parameters to the MIIS database that can be shared with the QoE controller. Both static attributes (e.g., the maximum available bandwidth) and dynamic attributes (e.g., the utilised bandwidth) can be stored at the same database. As the MAP is also aware of the characteristics of the video streaming session, the overall set of available monitored parameters will extend from the link layer to the application layer, allowing a more efficient and precise modelling of the perceptual 3D video quality. Figure 5 depicts the basic signalling flow of the periodic reporting scenario. The available links are generating periodic reports of their status and inform their local MIHF through the `Link_Parameters_Report.indication`. Thereafter, the various MIHF report their set of monitoring parameters to MIIS through the `MIH_Link_Parameters_Report.indication` event. Additionally, a similar primitive is generated by the MAP in order to report video related parameters that are available to it.

3.4. QoE Controller

3.4.1. QoE Related Key Performance Indicators. There are three basic categories of QoE parameters related to the physical, network, and application layers that are monitored and collected by MAP and MIH, as shown in Table 1:

(i) Physical layer related information includes the type of the access technology, the received signal strength, and oscillations of the signal strength. Particularly for the WiFi, this information can easily be collected directly from the access point through queries or broadcast messages.

(ii) The network QoS parameters are also an important source of information for the QoE management and include the throughput, the end-to-end IP layer delay, jitter, and packet loss. MAP, which is acting as an IP packet filtering function for the IP based video traffic directed to the wireless users, is able to parse the packet header and retrieve the QoE related KPIs.

(iii) The application parameters monitored involve the type of application (video or audio, although audio is out of the paper's scope), the encoding characteristics, and the video content properties. These are made

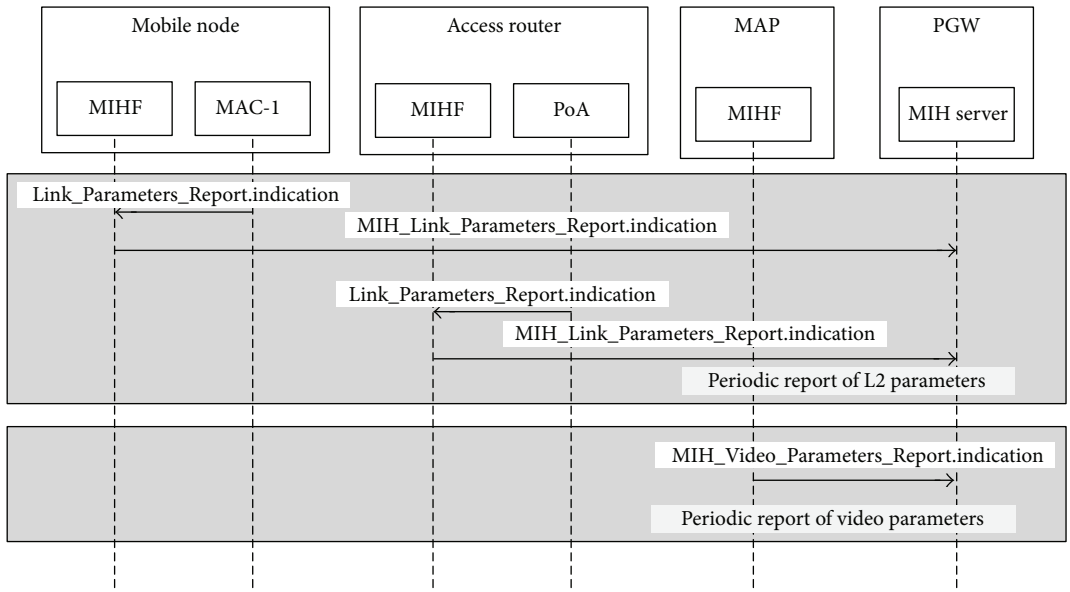


Figure 5: Signalling flow of periodic parameters report.

available from the MAP by parsing the RTP header of the received video traffic.

In addition to the parameters of Table 1, a number of additional measurements need to be considered in order to model and monitor the perceived video QoE. Three main groups of measurements are considered: the radio quality, the control plane performance, and the user plane QoS and QoE measurements [31, 32].

The radio quality related measurements will be restricted in just collecting the *number of active user pieces of equipment*. The number of active user devices indicates how many subscribers, on average, use the resources of the cell over a defined period of time and it can be used for traffic and radio resource planning. Moreover, a set of metrics related with the control plane performance are collected, including the *number of connection drops*, where a suddenly occurring exception, which may include loss of signal on the radio interface, can cause interruption of the connection between the user's equipment and the network. In a scenario that involves real-time video streaming, a loss of an ongoing connection with the content provider will cause extreme deterioration of the user's experience. In this case, it is mandatory in addition to the *connection drop ratio per cell* to also measure the *connection drop ratio per service*.

Furthermore, user plane QoE related performance indicators utilised by the QoE controller include the *packet jitter* defined as the average of the deviations from the mean network latency. This measurement is based on the arrival time stamp of successive received UDP packets at MAP. The *throughput* is defined as the data volume of IP datagrams transmitted within a defined time period (usually this time period is set to 1 second). In order to determine the IP throughput of each client, MAP parses the IP header and collects and stores the source and destination address of the particular packet. The throughput will be measured at a high sampling rate (i.e., every 333 ms) for the duration of the ongoing connection.

3.4.2. ML-Based QoE Model. The supervised ML is a learning process based on instances that produce a generalised hypothesis. In turn, this hypothesis could be applied as a mean to forecast future instances [33]. There are a number of common steps distinct in all ML techniques: (a) data set collection, (b) data prepossessing, (c) features creation, (d) algorithm selection, and (e) learning and evaluation using test data. The accuracy of the model improves through the repetition and the adjustment of any step that needs improvement. Supervised ML algorithms could be categorized as follows [34]:

(i) Logic-based: in this category the most popular algorithms are decision trees, where each node in the tree represents a feature of instances and each branch represents a value that the node can assume. The disadvantage of the algorithms in this category is that they cannot perform efficiently when numerical features are used.

(ii) Perceptron-based: artificial neural networks are included in this category. Neural networks have been applied to a range of different real-world problems and their accuracy is a function of the number of neurons used as well as the processing cost. However, neural networks may become inefficient when fed with irrelevant features.

(iii) Statistical: the most well-known statistical learning algorithms are the bit rate Bayesian network and the k-nearest neighbour. The first has the advantage that it requires short computational time for training, but it is considered partial due to the fact that it assumes that it can distinguish between classes using a

single probability distribution. The second algorithm is based on the principle that neighbouring instances have similar properties. Although very simple to use as it requires only the number of nearest neighbours as input, it is unreliable when applied on data sets with irrelevant features.

(iv) Support Vector Machines (SVM): SVM learning algorithms have been proven to perform better when dealing with multidimension and continuous features as well as when applied to inputs with a nonlinear relationship between them. An example of inputs with nonlinear relationship is the video viewer's QoE and the qualitative metrics gathered from different points of the delivery chain.

Within the context of this study a QoE controller, which is dwelling next to the wireless network provider's core, implements the proposed machine learning-based QoE prediction model. The proposed model could be characterised by fast response time in order to support seamless streaming adaptation, low processing cost, and efficiency. In order to select the best candidate for such a model, three well-known and widely used learning algorithms are investigated and compared in terms of precision and efficiency. The first is bit rate Bayesian classifier [35], the second is a decision tree based on the C4.5 algorithm [36], and the third is a multilayer perceptron network [37].

Naive Bayesian Classifier. A simple method for acquiring knowledge and accurately predicting the class of an instance from a set of training instances, which include the class information, is the Bayesian classifiers. Particularly, the naive Bayesian classifier is a specialized form of Bayesian network that relies on two important simplifying assumptions. Firstly, it assumes that, given the class, the predictive attributes are conditionally independent and, secondly, that the training data set does not contain hidden attributes, which potentially could affect the prediction process.

These assumptions result in a very efficient classification and learning algorithm. In more detail, assuming a random variable C representing the class of an instance, then c is a particular class label. Similarly, X is the vector of random variables that denote the values of the attributes; therefore, x is a particular observed attribute value vector. In order to classify a test case x, the Bayes rule is applied to compute the probability of each class given the vector of observed values for the predictive attributes:

$$p(C = c \mid X = x) = \frac{p(C = c)\, p(X = x \mid C = c)}{p(X = x)}. \quad (1)$$

In (1) the denominator can be calculated given that the event $X = x$ is a conjunction of assigned attribute values (i.e., $X_1 = x_1 \wedge X_2 = x_2 \wedge \cdots \wedge X_k = x_k$) and that these attributes are assumed to be conditionally independent:

$$p(X = x \mid C = c) = \prod_i p(X_i = x_i \mid C = c). \quad (2)$$

Evidently, from (2) it is deduced that each numeric attribute is represented as a continuous probability distribution over the range of the attributes values. Commonly, these probability distributions are assumed to be the normal distribution represented by the mean and standard deviation values. If the attributes are assumed to be continuous, then the probability density function for a normal distribution is

$$p(X = x \mid C = c) = G(x; \mu_c, \sigma_c) \text{ where,}$$
$$G(x; \mu_c, \sigma_c) = \frac{e^{-(x-\mu)^2/2\sigma^2}}{\sqrt{2\pi\sigma}}. \quad (3)$$

Hence, given the class, only the mean of the attribute and its standard deviation are required for the bit rate Bayes classifier to classify the attributes. For the purposes of this study, the naive Bayes classifier will be providing the baseline for the ML-based QoE models in terms of accuracy and precision. This is due to the fact that although it is very fast and robust to outliers and uses evidence from many attributes, it is also characterised by low performance ceiling on large databases and assumes independence of attributes. Moreover, the naive Bayes classifier requires initial knowledge of many probabilities, which results in a significant computational cost.

Logical Decision Tree. By definition a decision tree can be either a leaf node labelled with a class, or a structure containing a test, linked to two or more nodes (or subtrees). Hence, an instance is classified by applying its attribute vector to the tree. The tests are performed into these attributes, reaching one or other leaf, to complete the classification process. One of the most widely used algorithms for constructing decision trees is C4.5 [36]. Although C4.5 is not the most efficient algorithm, it has been selected due to the fact that the produced results would be easily compared with similar research works. There are a number of assumptions commonly applied for C4.5 in order to increase its performance efficiency and classification process:

(i) When all cases belong to the same class, then the tree is a leaf and is labelled with the particular class.

(ii) For every attribute, calculate the potential information provided by a test on the attribute, according to the probability of each case having a particular value for the attribute. Additionally, measure the information gain that results from a test on the attribute, using the probabilities of each case with a particular value for the attribute being of a particular class.

(iii) Depending on the current selection criterion, find the best attribute to create a new branch.

In particular the selection (or splitting) criterion is the normalized information gain. Inherently, C4.5 aims to identify the attribute that possesses the highest information gain and create a splitting decision node. This algorithmic process can be analytically represented with functions (4), assuming that the entropy of the n-dimensional vector of attributes of the sample $H(\vec{y})$ denotes the disorder on the data, while the conditional entropy $H(j \mid \vec{y})$ is derived from iterating over all possible values of $\vec{y}$:

$$H(\vec{y}) = -\sum_{j=1}^{n} \frac{|y_j|}{|\vec{y}|} \log \frac{|y_j|}{|\vec{y}|},$$

$$H(j \mid \vec{y}) = \frac{|y_j|}{|\vec{y}|} \log \frac{|y_j|}{|\vec{y}|},$$
$$\text{Gain}(\vec{y}, j) = H(\vec{y} - H(j \mid \vec{y})). \quad (4)$$

The algorithm, ultimately, needs to perform pruning of the resulting tree in order to minimise the classification error caused by the outliers included in the training data set. However, in the case of classifying video perceptual quality with the use of decision trees, the training set contains specialisations (e.g., a low number of very high or very low MOS measurements), and hence the outlier detection and cleansing of the training set needs to be performed beforehand, as described in Section 5.1.

Multilayer Perceptron. A multilayer perceptron network (also known as multilayer feed-forward network) has an input layer of neurons that is responsible for distributing the values in the vector of predictor variable values to the neurons of the hidden layers. In addition to the predictor variables, there is a constant input of 1.0, called the bias, that is fed to each of the hidden layers. The bias is multiplied by a weight and added to the sum going into the neuron. When in a neuron of the hidden layer, the value from each input neuron is multiplied by a weight and the resulting weighted values are added together producing a weighted sum that in turn is fed to a transfer function. The outputs from the transfer function are distributed to the output layer. Arriving at a neuron in the output layer, the value from each hidden layer neuron is multiplied by a weight and the resulting weighted values are added together producing a weighted sum, which is fed into the transfer function. The output values of the transfer function are the outputs of the network. In principle the transfer function could be any function and could also be different for each node of the neural network. In this study the sigmoidal (s-shaped) function has been used in the form of $f(x) = 1/(1 + \exp(-x))$. This is a commonly used function that also is mathematically convenient as it produces the following derivative property: $d\sigma(x)/dx = \sigma(x)(1 - \sigma(x))$.

The training process of the multilayer perceptron is to determine the set of weight values that will result in a close match between the output from the neural network and the actual target values. The algorithm precision depends on the number of neurons in the hidden layer. If an inadequate number of neurons are used, the network will be unable to model complex data, and the resulting fit will be poor. If too many neurons are used, the training time may become excessively long, and, worse, the network may overfit the data. When overfitting occurs, the network will begin to model random noise in the data. In the context of this study several validation experiments have been performed using different number of neurons in the hidden layer. The best accuracy has been achieved by using five neurons in one hidden layer in the network, as shown in Figure 6. The feed-forward multilayer perceptron minimises the error function but it may take time to converge to a solution (i.e., the minimum value) which may be unpredictable due to the error that is added to the weight matrix in each iteration. Therefore, in order to control the convergence rate, the learning rate parameter was set to 0.3 and 350 iterations were performed until the system converged.

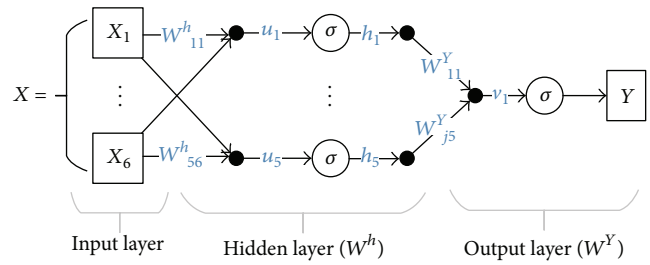


Figure 6: Proposed multilayer perceptron model tested and validated.

4. Experimental Setup

4.1. Capturing and Processing of 3D Video Content. The proposed QoE framework is validated through extensive experiments conducted over the test-bed platform of Figure 8. For the purposes of this study four real-world captured stereo video test sequences ("martial arts," "music concert," "panel discussion," and "report") with different spatial and temporal indexes have been used in left-right 3D format. The 3D video capturing was performed in accordance with the specifications of [38], in order to provide 3D contents which can be simply classified (genres) for 3D video encoding performances evaluation and comparison, to provide multiview contents with relevant 3D features from which one can compute disparity maps and synthesise intermediate views to be rendered on different kinds of display, and to allow advances and exhaustive quality assessment benchmarks. Towards this end, an LG Optimus 3D mobile phone was used as stereoscopic capturing device during the shooting session. The cameras were configured as follows:

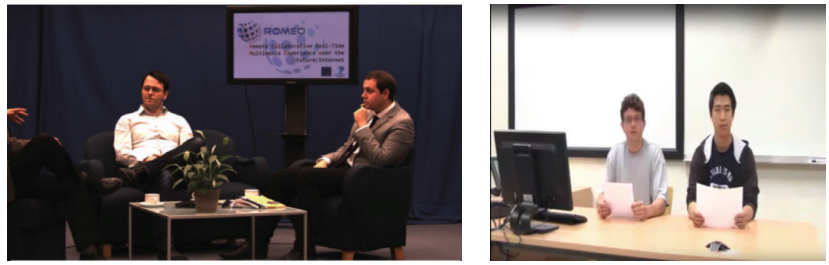
(i) Synchronized stereo camera (f2.8, FullHD, 30 fps, Bayer format, 65 mm spaced).

(ii) Raw formats: bayer format which is then converted to obtain rgb and yuv sequences (+ color correction, noise filtering, mechanical misalignment, and auto-convergence).

(iii) Encoding: FullHD or lower, side-by-side, or top-bottom with subsampling + MPEG4 AVC.

Snapshots of the test video sequences are shown in Figure 7. A detailed description of the capturing scenarios and content characteristics are described in [38]. All sequences are side-by-side with resolutions 640 × 720 pixels and 960 × 1080 pixels per view at 25 frames per second. The H.264/SVC encoding was performed using the encoder provided by *Vanguard Software* tool [39] configured to create two layers (one base layer and one enhancement layer) using MGS quality scalability. The SVC was the favourable choice for the particular experiments, as it allows 3D video content to be delivered over heterogeneous wireless channels with a manageble bit rate. The particular encoder inherently allows

(a) Martial arts (b) Music concert



(c) Panel discussion (d) Report

Figure 7: Test 3D video sequences used for MOS measurements.

the use of a variety of pattent protected error concealment schemes that can compansate for packet losses up to 5%. Each video frame was encapsulated in a single network abstraction layer unit (NALU) with a size of 1400 bytes, which in turn was encapsulated into a single Real-Time Transport Protocol (RTP) packet with a size of 1400 bytes (payload). The generated video packets are delivered through the network using UDP/IP protocol stack. An additional separate channel is responsible for the transmission of the Parameter Sets (PS) to the client through a TCP/IP connection. Moreover, *NetEm* [40] was used for emulating diverse networking conditions, variable delay, and loss in accordance with the described experimental scenarios. Hence, each experiment has been repeated 10 times in order to obtain valid statistical data. Table 2 summarises the encoding and streaming configuration parameters used in all experiments.

Table 2: Encoding and streaming parameters.

Video sequence	*Martial arts*	*Munich*	*Panel discussion*	*Report*
Spatial/temporal index	21/17	25/8	32/15	33/3
Number of frames	400 frames			
Intra period	8 frames			
QP used for base-enhancement layers	42–36 & 36–30			
Frame rate	25 frames per second			
Resolution per view	640 × 720 & 960 × 1080 pixels			
Media transmission unit	1500 bytes			

Wireless Channel Error Model. In order to model the impact of physical impairments on the QoE, the Rayleigh fading channel of the simulated 802.11g is represented by a two-state Markov model. Each state is characterized by a different bit error rate (BER) in the physical layer that results from the state and transitional probabilities of the Markov model, as in Table 3. The wireless channel quality is characterized by the probabilities of an error in the bad and good state, P_B and P_G, and the probabilities P_{GG} and P_{BB} to remain at the good or the bad state, respectively.

4.2. MAC Layer Load Model. The load of the wireless channel will result in time-outs, which will then cause retransmissions in the data link layer due to the contention-based access that is inherent in IEEE 802.11 protocol. Eventually, since the paper investigates real-time video streaming, such a load in the MAC layer will result in losses that in the application layer will be presented as artefacts and distortion of the perceptual video quality. In order to emulate and control the load, UDP traffic is generated and transmitted to both uplink and downlink channels. The constant-sized UDP packets are generated according to the Poisson distribution three times with mean values of 2 Mbps, 3 Mbps, and 4 Mbps, in each direction, respectively. Evidently, such a setup will eventually double the overall background traffic (i.e., load) over the WiFi access channel.

4.3. IP Delay Variation Model. Moreover, for the purpose of this research and in order to model the delay variations in the IP layer, a constant plus gamma distribution is applied [41], similar to the one depicted in Figure 9. More precisely, the delay variations are modelled as a gamma distribution with a constant scale factor of $\theta = 30$ ms and three different

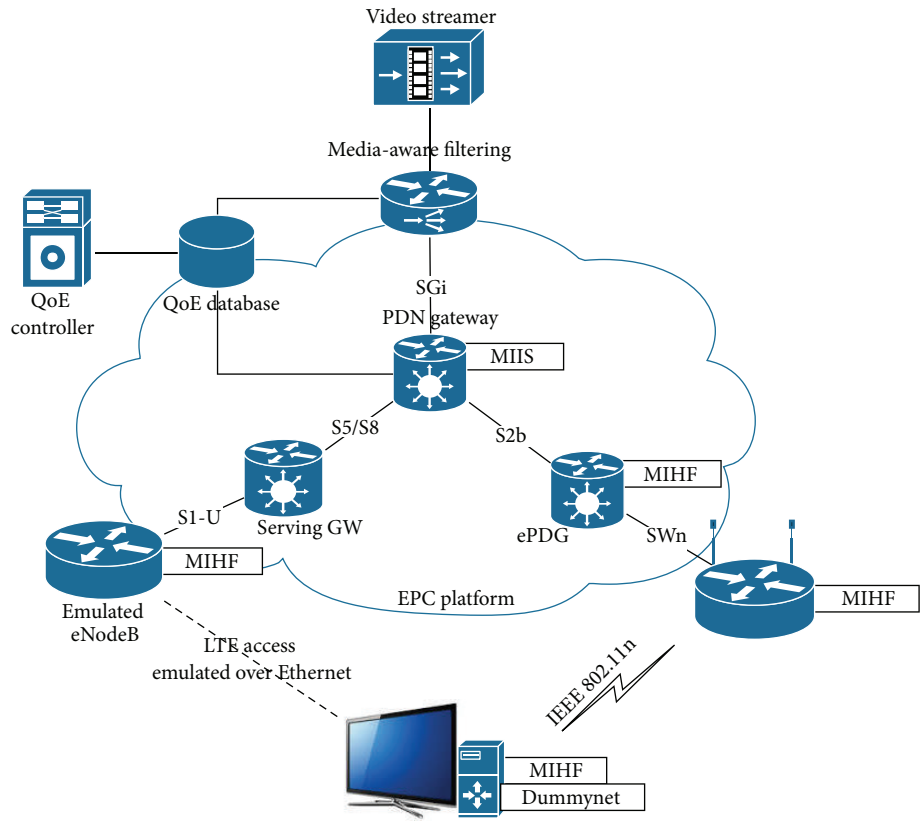


Figure 8: Test-bed platform used during the experiments.

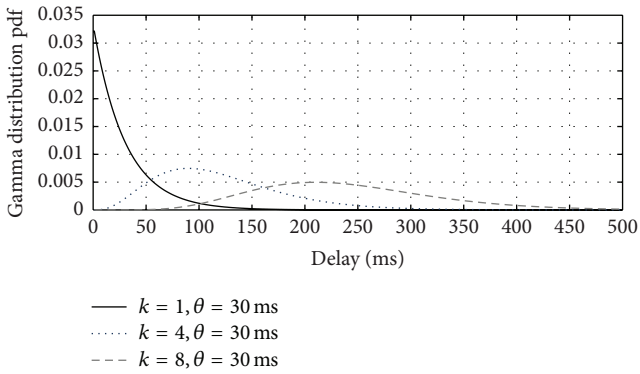


Figure 9: Delay variations in IP layer based on gamma distributions.

shape factors in the range of $k = 1, 4$, and 8. Apparently, this configuration will result in various mean delays and corresponding variances. During the experiments, three such delay variations schemes were applied, summarised in Table 4, in an effort to control the networking conditions and produce comprehensible outcomes, regarding the visual quality degradation of the decoded video.

5. Subjective Experiments and Analysis

The subjective experiments were conducted in accordance with the absolute category rating (ACR) method as defined in ITU-T Recommendations [42–45]. The perceived video quality is rated in a scale from 0 [bad] to 5 [excellent] according to the standard. In order to produce reliable and repeatable results, the subjective analysis of the video sequences has been conducted in a controlled testing environment. An LG 32LM620S LED 3D screen with a resolution of 1920 × 1080 pixels, aspect ratio 16 : 9, peak display luminance 500 cd/m^2, and contrast ratio of 5000000 : 1 along with passive glasses is used during the experiments to display the stereoscopic video sequences. The viewing distance for the video evaluators is set

Table 3: BER emulation parameters for the Rayleigh fading wireless channel.

	Bad channel	Medium channel	Good channel
P_G	$1.25e^2$	$1.29e^2$	$1.29e^2$
P_B	$4.13e^{14}$	$1.3e^{13}$	$4.1e^{12}$
P_{GG}	0.996	0.990	0.987
P_{BB}	0.336	0.690	0.740

Table 4: IP layer delay variation parameters resulting from gamma distribution.

Shape factor (k)	Mean (ms)	Variance (ms)
1	30	900
4	120	3600
8	140	7200

to 2 m, in accordance with the subjective assessment recommendations. The measured environmental luminance during measurements is 200 lux, and the wall behind the screen luminance is 20 lux, as recommended by [42]. The subjective evaluation was performed by twenty expert observers in a gender balanced way, who were asked to evaluate the overall visual quality of the displayed video sequences. A training session was included in order for the observers to get familiar with the rating procedure and understand how to recognize artefacts and impairments on the video sequences.

5.1. Statistical Measures. Initially, the collected MOSs were analysed for each subject across the different test conditions using the chi-square test [46], which verified the normality of the majority of MOS distributions. Following the normality verification of the MOS distributions, the technique of outliers' detection and elimination was applied. This technique is based on the detection and removal of scores in the cases where the difference between the mean subject vote and the mean vote for this test case from all other subjects exceeds 15%. The applied outlier detection method determined a maximum of two outliers per test case.

After removing the outliers, the remaining MOSs were statistically analysed in order to assess the perceptual quality of the received video sequences. The average MOS of each set S of tested video sequence scenarios of size K is denoted as $\overline{\mathrm{MOS}_k}$:

$$\overline{\mathrm{MOS}_k} = \frac{\sum_{k=1}^{K}\sum_{i=1}^{N}\mathrm{MOS}_{i,k}}{KM}, \tag{5}$$

where $\mathrm{MOS}_{i,k}$ is the Mean Opinion Score of the ith viewer for the kth tested scenario of one of the video sequences and N is number of evaluators of the particular tested scenario. The standard deviation of MOS_k is defined by the square root of the variance:

$$\sigma_k^2 = \frac{\sum_{k=1}^{K}\left[\sum_{i=1}^{N}\mathrm{MOS}_{i,k}/N - \overline{\mathrm{MOS}_k}\right]}{K-1}. \tag{6}$$

The confidence interval (CI) of the estimated mean MOS values, which indicate the statistical relationship between the mean MOS and the actual MOS, is computed using Student's t-distribution as follows:

$$\mathrm{CI}_j = t\times\left(1-\frac{a}{2}, N\right)\times\frac{\sigma_j}{\sqrt{N}}, \tag{7}$$

where $t\times(1-\alpha/2, N)$ is the t-value that corresponds to a two-tailed t-Student distribution with $N-1$ degrees of freedom and a desired significance level α. In this case, $\alpha = 0.05$, which corresponds to 95% significance, and σ_j is the standard deviation of the MOS of all subjects per test. Furthermore, the degree of asymmetry of the collected MOS around the mean value of its distribution of its measured samples is measured by skewness, while the "peakedness" of the distribution is measured by its kurtosis as follows:

$$\beta = \frac{m_3}{m_2^{3/2}}, \qquad \gamma = \frac{m_4}{m_2^2}, \tag{8}$$

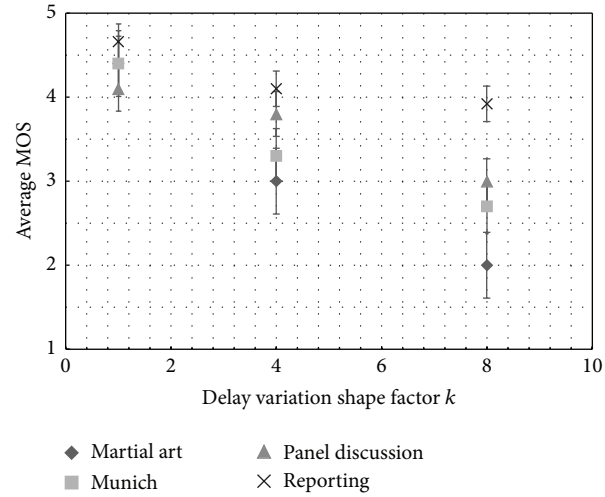


Figure 10: Average MOS measured as a result of artefacts due to delay variations.

where m_j is the jth moment about the mean given by

$$m_j = \frac{\sum_{k=1}^{K}\left[\sum_{i=1}^{N}\mathrm{MOS}_i/N - \overline{\mathrm{MOS}_k}\right]^2}{K}. \tag{9}$$

Finally the measured MOSs are subjected to an analysis of variance (ANOVA). The ANOVA tests the null hypothesis that the means between two or more groups are equal under the assumption that the sample populations are normally distributed. Although multifactor ANOVA would reveal the complex relationship of BER, MAC layer load, and delay variations with the resulting QoE as in [47], in this study the aim is to determine whether each of the three selected factors, which can be considered as statistically independent, does have a statistically significant impact on the resulting QoE. In this particular case the ANOVA is used in order to test the following null hypothesis:

(1) The mean MOSs due to the three delay variation groups (i.e., for $k = 1$, $k = 4$, and $k = 8$) independent of the video sequence, their spatial resolution, and quantization step size are equal.

(2) The mean MOSs due to the three MAC layer background loads (i.e., 2 Mbps, 3 Mbps, and 4 Mbps) independent of the video sequence, their spatial resolution, and quantization step size are equal.

(3) The mean MOSs due to the three wireless channel BERs (i.e., Good, Medium, and Bad) independent of the video sequence, their spatial resolution, and quantization step size are equal.

5.2. Analysis of Measured MOS. The averaged MOS measurements of the four test video sequences under the abovementioned experimental conditions are shown in Figures 10, 11, and 12. Each column of the figures represents the average MOS scored by the viewers due to the impact of just one of the three QoS metrics (i.e., BER, MAC load, and delay variation),

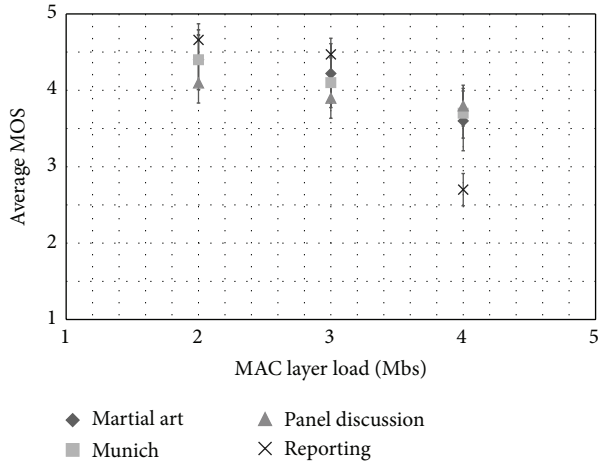


Figure 11: Average MOS measured as a result of artefacts due to MAC layer load.

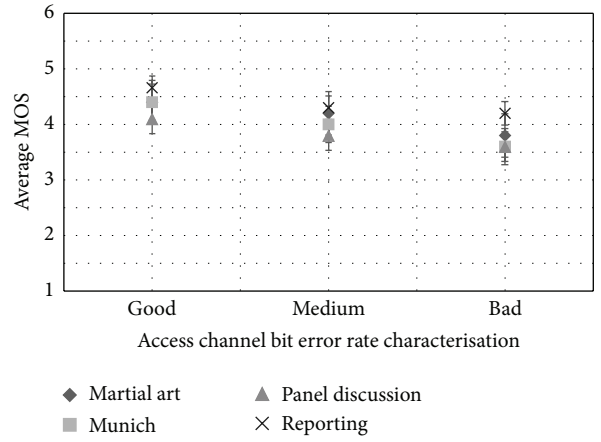


Figure 12: Average MOS measured as a result of artefacts due to bit error rate.

while the other two metrics are assigned to their "best" value (i.e., the value that results from the highest level of video delivery QoS). The comparison of the averaged MOSs provides an indication on the impact of the delay variation, load, and BER on the tested sequence. It is evident that the QoS parameters have different effect depending on the video characteristics (i.e., spatial and temporal indexes). Moreover, Table 5 summarises the statistical parameters of MOS per tested video sequence.

In order to get a better understanding of the MOS per tested video sequence the mean, the standard deviation, the variance, the skewness, and the kurtosis are studied. Through variance, the difficulty or ease of the evaluator to assess a parameter as well as the agreement between evaluators can be addressed. Hence, a lower variance may indicate a higher agreement of the overall 3D MOS among viewers. In Table 5 the low variance of the MOS around the mean in all video sequences indicates that viewers agreed on the scores that were assigned to the overall 3D video quality in each test scenario. A slightly higher variance of the "martial arts" sequence may be due to the high temporal index of the sequence that caused a more erratic behaviour of losses, thus causing more discrepancies among the viewers.

In the context of subjective evaluation, skewness shows the degree of asymmetry of the scores around the MOS value of each distribution of samples for the given parameter (in this case the video sequences). As a normal distribution has a skewness of zero, the results in Table 5 indicate a symmetry of the scores around the MOS for every tested video sequence. Similarly to the study of the variance, the "martial arts" sequence has a slight asymmetry compared to the rest of the sequences due to the fact that the viewers of the sequence approached the maximum and minimum scores more frequently than in the other cases.

The last column of Table 5 presents the kurtosis of the MOS of each video sequence. The kurtosis is the indication of how outlier-prone a distribution is. A normal distribution has a default kurtosis value of 3. The lower the kurtosis, the more prone the distribution. Nevertheless, a kurtosis two times below the standard error of the kurtosis, which is calculated as the square root of 24 divided by N (the number of MOSs) [48], could be considered as normal. In this case, as there are 20 evaluators scoring 108 different scenarios per video sequence, the kurtosis below 40 can be ignored. Therefore, the consistent low kurtosis of Table 5 for all video sequences can be regarded as an indication of lack of outliers, since all evaluators assigned similar scores to the overall 3D video quality.

Three one-way ANOVA methods have been performed in order to evaluate an equivalent number of null hypotheses regarding the three QoS related parameters under study (delay variations, load, and BER). The ANOVA results are summarised in Tables 6, 7, and 8. In more detail, assuming that the grand mean of a set of samples is the total of all the data values divided by the total sample size, then the total variation is comprised as the sum of the squares of the differences of each mean with the grand mean. In ANOVA, the between-group variation and the within-group variation are defined. By comparing the ratio of between-group variance to within-group variance ANOVA determines, if the variance caused by the interaction between the samples is much larger when compared to the variance that appears within each group, then it is because the means are not the same. Additionally, if there are n samples involved with one data value for each sample (the sample mean), then the between-groups variation has $n - 1$ degrees of freedom. Similarly, in the case of within-group variation, each sample has degrees of freedom equal to one less than their sample sizes, and there are n samples; the total degrees of freedom are n less than the total sample size: $\mathrm{df} = Nn$, where N is the total size of the samples. The variance due to the differences within individual samples is denoted as MS (Mean Square) within groups. This is the within-group variation divided by its degrees of freedom. The F-test statistic is found by dividing the between-group variance by the within-group variance. The degrees of freedom for the numerator are the degrees of freedom for the between-group variance ($n-1$) and the degrees of freedom for the denominator are the degrees of freedom for the within-group variance ($N - n$). According to the F-test the decision will be to reject the null hypothesis if the F-test statistic from

Table 5: Statistical analysis of measured MOS per video test sequence.

Video sequence	Average MOS	Standard dev.	Variance	95% CI	Skewness	Kurtosis
Martial arts	2.390	0.836	0.698	0.391	0.284	2.208
Munich	2.943	0.695	0.483	0.325	0.010	2.288
Panel discussion	3.223	0.570	0.325	0.266	0.173	2.416
Reporting	3.689	0.451	0.203	0.211	−0.024	2.278

Table 6: One-way ANOVA for the parameter delay variation.

Source	Sum of squares	Degrees of freedom	MS	F	Probability of F
Between	103.542	2	51.7711	126.95	$4.91566e^{-44}$
Within	174.945	429	0.4078		

Table 7: One-way ANOVA for the parameter MAC load.

Source	Sum of squares	Degrees of freedom	MS	F	Probability of F
Between	11.697	2	5.8486	9.4	0.0001
Within	266.79	429	0.62189		

Table 8: One-way ANOVA for the parameter BER.

Source	Sum of squares	Degrees of freedom	MS	F	Probability of F
Between	23.852	2	11.9262	20.09	$4.55639e^{-09}$
Within	254.635	429	0.5936		

Tables 6, 7, and 8 is greater than the F-critical value with $k-1$ numerator and $N-k$ denominator degrees of freedom.

By studying Tables 6, 7, and 8, it can be affirmed that all three null hypotheses, as defined previously, can be rejected; hence, there is significant statistical difference among the groups with respect to delay variations (i.e., the different shape factors k of the gamma distribution have a significant effect on the resulting 3D video MOS), MAC layer load (i.e., the three different mean values of the Poisson distributed background traffic affect significantly the 3D MOS), and the BER (i.e., the three different wireless channel conditions alter the 3D MOS significantly). The above conclusions are derived by the extremely low probability p which is much lower than the significance level of 0.05 and the F-statistic which in turn indicates that one or two means significantly differ between each other.

Therefore, one-way ANOVA affirms that the selection of the three QoS parameters (delay variation, load, and BER) as well as the values assigned to these parameters for the purposes of the subjective evaluation of 3D MOS and subsequently the definition, training, and validation of a machine learning QoE model, based on these QoS parameters, is correct.

6. ML QoE Models Comparison

The training and the analysis of the results have been performed using Weka software tool [49]. The output of the naive Bayes classification is shown in Table 9, where for every attribute of the data set the mean and standard deviation of the Gaussian distribution are shown. Furthermore, the decision tree of the C4.5 ML algorithm that is implemented as J48 in Weka is illustrated in Figure 13. In accordance with the results of the MOS comparison in the previous section, the jitter is the most important parameter; hence, it is located in the route of the tree (i.e., first splitting node).

All models are assessed in terms of predictive performance using the 10-fold cross-validation method. The confusion matrix of each of the three models is shown in Table 10. It can be seen that the naive Bayesian classifier has classified wrongly 42 instances of MOS class 2, 43 instances of MOS class 3, and all MOS class 5 instances. This is expected since the naive Bayes classifier was selected to act as the baseline of the ML performance tests. The confusion matrices of multilayer perceptron and C4.5 decision tree confirm that both methods perform similarly and better than the Bayesian classifier.

Nevertheless, the models accuracy (number of correctly classified instances) cannot be used for assessing the usefulness of classification models built using unbalanced datasets, as in this case. For this purpose the "Kappa statistic" is used, which is a generic term for several similar measures of agreement used with categorical data. Typically it is used in assessing the degree to which two or more video viewers in this case, examining the same video sequence, agree when it comes to assigning the MOSs. Similarly to the correlation coefficient, its value is zero when there is no agreement on the scores (purely random coincidences of rates) and is one when there is complete agreement. The "Kappa statistic" of the three models is shown in Table 11.

Additionally, the three ML algorithms considered in this work are compared with each other in terms of accuracy,

Table 9: Output of naive Bayes classifier.

Attributes		Class (MOS)				
		1	2	3	4	5
SI	Mean	20.21	21.71	22.28	22.67	24
	Std. dev.	0.89	1.97	1.98	1.88	0.6667
TI	Mean	17.53	14.16	12.87	12	9
	Std. dev.	2.01	4.45	4.45	4.24	1.5
QP_1	Mean	39.15	38.82	39.55	38.14	36
	Std. dev.	2.99	2.99	2.94	2.87	1
QP_2	Mean	33.15	32.82	33.55	32.14	30
	Std. dev.	2.99	2.99	2.94	2.87	1
Resolution	Mean	0.57	0.51	0.46	0.5	1
	Std. dev.	0.49	0.49	0.49	0.5	0.16
Jitter	Mean	7	5.61	2.68	0.25	0
	Std. dev.	0.58	2.00	2.17	0.90	0.5833
Load	Mean	3.42	3.13	2.98	2.64	2
	Std. dev.	0.67	0.82	0.81	0.71	0.16
Channel	Mean	0.52	0.85	1.06	1.28	2
	Std. dev.	0.67	0.77	0.83	0.76	0.16

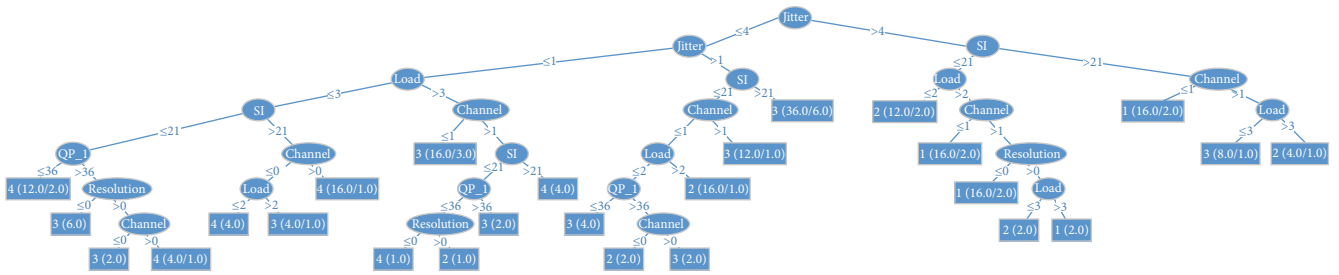


Figure 13: C4.5 based classification tree of the proposed model.

precision (p), and recall bit rate. The precision of the algorithm, which is also known as reproducibility or repeatability, denotes the degree to which repeated measurements (in this case of MOS) under unchanged conditions show the same results. Recall is defined as the number of relevant instances retrieved by a search divided by the total number of existing relevant instances. In addition to the two metrics, the algorithms are also evaluated in terms of F-measure (F), which considers both the precision (p) and the recall (r) of the test to compute the score. Let $i \in [1, 2, 3, 4, 5]$ represent the class, which in this case is derived from the MOSs. Then true positive TP_i is the number of the instances correctly classified, false positive FP_i is the number of the instances that belong to the class bit rate but have not been classified there, and false negative FN_i is the number of the instances that do not belong in class bit rate but have been classified there. Thus, (p), (r), and (F) are derived as follows:

$$p = \frac{\mathrm{TP}_i}{\mathrm{TP}_i + \mathrm{FP}_i},$$

$$r = \frac{\mathrm{TP}_i}{\mathrm{TP}_i + \mathrm{FN}_i},$$

$$F = \frac{2pr}{p + r}. \tag{10}$$

Finally, the accuracy of the models is also deduced by the area under the Receiver Operating Characteristic (ROC) curve. An area of one represents a perfect model, while an area of 0.5 means lack of any statistical dependencies. These comparison results are also shown in Table 11.

It is evident that for the particular data set, with the multiple classes and the nonlinear relationship among the attributes, the best performance is achieved by the multilayer perceptron. Nevertheless, C4.5 is a very good alternative that performs MOS classification with acceptable accuracy and with less processing effort. Moreover, since decision trees return results that have a physical significance, thus the classification can be easily interpreted and used as a decision

Table 10: Confusion matrices of the three ML models.

a	b	c	d	e	← classified as
Naive Bayesian confusion matrix					
2	17	0	0	0	$a = 1$
2	**39**	39	1	0	$b = 2$
0	11	**134**	32	0	$c = 3$
0	0	40	**109**	0	$d = 4$
0	0	0	6	**0**	$e = 5$
Multilayer perceptron confusion matrix					
11	8	0	0	0	$a = 1$
6	**63**	12	0	0	$b = 2$
0	13	**141**	23	0	$c = 3$
0	0	21	**128**	0	$d = 4$
0	0	0	6	**0**	$e = 5$
C4.5 confusion matrix					
12	7	0	0	0	$a = 1$
6	**63**	12	0	0	$b = 2$
0	14	**137**	26	0	$c = 3$
0	1	25	**121**	2	$d = 4$
0	0	0	5	**1**	$e = 5$

Table 11: Performance comparison of ML algorithms.

	Naive Bayesian	Multilayer perceptron	C4.5
Kappa statistic	0.473	0.693	0.664
Precision (p)	0.643	0.782	0.770
Recall (r)	0.657	0.794	0.773
F-measure (F)	0.641	0.788	0.771
Area under ROC	0.851	0.912	0.886

rule; C4.5 would be preferable as opposed to the neural network, as a decision engine during real-time QoE control.

7. Conclusion and Future Steps

Evidently, a new wave of IP convergence with 3GPP's LTE EPC in its heart is being fuelled by real-time multimedia applications. As new network architecture paradigms are constantly evolving, there is an ongoing effort by content providers and operators to provide and maintain premium content access with optimised QoE. This paper presents a novel media-aware framework for measuring and modelling the 3D video user experience, designed to be closely integrated with LTE EPC architecture. The proposed solution involves a synergy between a Media-Aware Proxy networking element and the IEEE 802.21 MIH protocol. The former is responsible for parsing RTP packets to collect QoE related KPIs and adapting the video streams to the current networking conditions in real time. The latter is being enhanced to act as control mechanism for providing the necessary signalling to collect KPIs across different layers and network elements. Moreover, MIH provides the central database, which stores the collected QoE related KPIs and allows a third component named QoE controller to train this dataset and model the perceived 3D video quality. Three machine learning classification algorithms for modelling QoE due to network related impairments have been investigated. Opposite to previous studies, the QoE model is a function of parameters collected from the application, the MAC, and physical layers. Hence, instead of determining the 3D QoE by measuring only the end-to-end packet loss, this paper considers a set of quality of service (QoS) parameters (i.e., bit error rate from physical layer, MAC layer load, and network layer delay/jitter), which account for the overall number of lost packets. A detailed statistical analysis of the measured 3D MOS indicated that the predominant factor of QoE degradation is the IP layer jitter. Therefore, the proposed ML scheme aims to determine more accurately the impact that different layer impairments have on the perceived 3D video experience. A comparison of the three ML schemes under study in terms of precision and accuracy indicated that the multilayer perceptron has the best performance, closely followed by the C4.5 decision tree.

This is an ongoing research. Building upon the results presented in this paper, the following steps involve the design and implementation of a QoE management mechanism that will optimise the perceived video QoE by instructing different network elements of the EPC architecture and the MAP to either adapt the delivered video traffic to the network conditions or enforce with the aid of MIH a content-aware handover to a less loaded neighbouring channel or access network. Moreover, a closer integration with the current and future LTE EPC implementations is currently under study. The aim is to reconfigure the proposed framework to incorporate the LTE policy and charging control (PCC) techniques, in an effort to upgrade it into a content-aware traffic mechanism extended over all network operator deployments.

Competing Interests

The authors declare that they have no competing interests.

Acknowledgments

This work was supported by the SIEMENS "Excellence" Program by the State Scholarship Foundation (IKY), Greece.

References

[1] A. Lykourgiotis, K. Birkos, T. Dagiuklas et al., "Hybrid broadcast and broadband networks convergence for immersive TV applications," *IEEE Wireless Communications*, vol. 21, no. 3, pp. 62–69, 2014.

[2] G.-M. Su, Y.-C. Lai, A. Kwasinski, and H. Wang, "3D video communications: challenges and opportunities," *International Journal of Communication Systems*, vol. 24, no. 10, pp. 1261–1281, 2011.

[3] C. Tselios, I. Politis, K. Birkos, T. Dagiuklas, and S. Kotsopoulos, "Cloud for multimedia applications and services over heterogeneous networks ensuring QoE," in *Proceedings of the IEEE 18th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD '13)*, pp. 94–98, Berlin, Germany, September 2013.

[4] K. Birkos, I. Politis, A. Lykourgiotis et al., "Towards 3D video delivery over heterogeneous networks: the ROMEO approach," in *Proceedings of the International Conference on Telecommunications and Multimedia (TEMU '12)*, pp. 60–65, IEEE, Chania, Greece, August 2012.

[5] ITU-T and ISO/IEC, "ITU-T recommendation H.264 (2010): advanced video coding for generic audiovisual services," Information Technology Coding of Audiovisual Objects Part 10: Advanced Video Coding ISO/IEC 14496-10:2010, 2010.

[6] A. Vetro, T. Wiegand, and G. J. Sullivan, "Overview of the stereo and multiview video coding extensions of the H.264/MPEG-4 AVC standard," *Proceedings of the IEEE*, vol. 99, no. 4, pp. 626–642, 2011.

[7] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand, "Overview of the high efficiency video coding (HEVC) standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1649–1668, 2012.

[8] 3GPP, "Technical specification group services and system aspects," 3GPP TS 23.002 V13.3.0 (2015-09), Network Architecture, 2015.

[9] H. Schwarz, D. Marpe, and T. Wiegand, "Overview of the scalable video coding extension of the H.264/AVC standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, no. 9, pp. 1103–1120, 2007.

[10] A. Vetro, P. Pandit, H. Kimata, A. Smolic, and Y.-K. Wang, "Joint draft 8 of multi-view video coding," Tech. Rep. JVT-AB204, Joint Video Team (JVT), Hannover, Germany, 2008.

[11] G. J. Sullivan, A. M. Tourapis, T. Yamakage, and C. S. Lim, "Draft AVC amendment text to specify constrained baseline profile, stereo high profile, and frame packing SEI message," Joint Video Team (JVT) Document JVT-AE204, Joint Video Team (JVT), London, UK, 2009.

[12] P. Merkle, A. Smolic, K. Muller, and T. Wiegand, "Efficient prediction structures for multiview video coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, no. 11, pp. 1461–1473, 2007.

[13] I. Politisa, L. Dounis, C. Tselios, A. Kordelas, T. Dagiuklas, and A. Papadakis, "A model of network related QoE for 3D video," in *Proceedings of the IEEE Globecom Workshops (GC Wkshps '12)*, pp. 1335–1340, Anaheim, Calif, USA, December 2012.

[14] C. Tselios, C. Mysirlidis, I. Politis, T. Dagiuklas, and S. Kotsopoulos, "On quality evaluation of 3D video using Colour-plus-Depth and MDC," in *Proceedings of the IEEE 19th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD '14)*, pp. 66–69, IEEE, Athens, Greece, December 2014.

[15] Nokia-Siemens Networks, "QoS-enabled IP networks for assured multiplay QoE," Technical White Paper, 2011.

[16] K.-T. Chen, C.-J. Chang, C.-C. Wu, Y.-C. Chang, and C.-L. Lei, "Quadrant of euphoria: a crowdsourcing platform for QoE assessment," *IEEE Network*, vol. 24, no. 2, pp. 28–35, 2010.

[17] A. Balachandran, V. Sekar, A. Akella, S. Seshan, I. Stoica, and H. Zhang, "Developing a predictive model of quality of experience for internet video," *ACM SIGCOMM Computer Communication Review*, vol. 43, no. 4, pp. 339–350, 2013.

[18] K. U. R. Laghari and K. Connelly, "Toward total quality of experience: a QoE model in a communication ecosystem," *IEEE Communications Magazine*, vol. 50, no. 4, pp. 58–65, 2012.

[19] A. K. Moorthy, L. K. Choi, A. C. Bovik, and G. De Veciana, "Video quality assessment on mobile devices: subjective, behavioral and objective studies," *IEEE Journal on Selected Topics in Signal Processing*, vol. 6, no. 6, pp. 652–671, 2012.

[20] V. De Silva, H. K. Arachchi, E. Ekmekcioglu et al., "Psychophysical limits of interocular blur suppression and its application to asymmetric stereoscopic video delivery," in *Proceedings of the 19th International Packet Video Workshop (PV '12)*, pp. 184–189, IEEE, Munich, Germany, May 2012.

[21] 3GPP, "General Packet Radio Service (GPRS) enhancements for Evolved Universal Terrestrial Radio Access Network (E-UTRAN) access," 3GPP TS 23.401, 2016.

[22] 3GPP, "3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; Network Architecture," Tech. Rep. 3GPP TS 23.002 V12.5.0, 2014.

[23] IETF, "Diameter base protocol," Tech. Rep. RFC 6733, IETF, 2012.

[24] ETSI TS 123 228 V12.9.0, "IP Multimedia Subsystem (IMS), stage 2," 3GPP TS 23.228 version 12.9.0 Release 12, 2015.

[25] Fraunhofer FOKUS OpenEPC Toolkit, http://www.openepc.net/.

[26] IEEE 802.21 Working Group, "IEEE standard for local and metropolitan area networks part 21: media independent handover," IEEE Standard 802, 2008.

[27] A. Kordelas, I. Politis, A. Lykourgiotis, T. Dagiuklas, and S. Kotsopoulos, "An media aware platform for real-time stereoscopic video streaming adaptation," in *Proceedings of the IEEE International Conference on Communications Workshops (ICC '13)*, pp. 687–691, Budapest, Hungary, June 2013.

[28] IETF, "RTP payload format for H.264 video," IETF Draft RFC 3984, IETF, 2010, draft-ietf-avt-rtp-rfc3984bis-12.

[29] IETF Draft Audio/Video Transport WG, "RTP Payload Format for SVC Video," draft-ietf-avt-rtp-svc-27, 2011.

[30] 3GPP, "Architecture enhancements for non-3GPP accesses," 3GPP TS 23.402, 2015.

[31] H. Ekström, "QoS control in the 3GPP evolved packet system," *IEEE Communications Magazine*, vol. 47, no. 2, pp. 76–83, 2009.

[32] H. Luo, S. Ci, D. Wu, J. Wu, and H. Tang, "Quality-driven cross-layer optimized video delivery over LTE," *IEEE Communications Magazine*, vol. 48, no. 2, pp. 102–109, 2010.

[33] V. Menkovski, A. Oredope, A. Liotta, and A. Cuadra Sánchez, "Predicting quality of experience in multimedia streaming," in *Proceedings of the 7th International Conference on Advances in Mobile Computing and Multimedia (MoMM '09)*, pp. 52–59, Kuala Lumpur, Malaysia, December 2009.

[34] S. B. Kotsiantis, "Supervised machine learning: a review of classification techniques," *Informatica*, vol. 31, no. 3, pp. 249–268, 2007.

[35] G. H. John and P. Langley, "Estimating continuous distributions in Bayesian classifiers," in *Proceedings of the 11th Conference on Uncertainty in Artificial Intelligence (UAI '95)*, pp. 338–345, Montreal, Canada, August 1995.

[36] J. R. Quinlan, *C4.5: Programs for Machine Learning*, vol. 1, Morgan Kaufmann, Boston, Mass, USA, 1993.

[37] M. W. Gardner and S. R. Dorling, "Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences," *Atmospheric Environment*, vol. 32, no. 14-15, pp. 2627–2636, 1998.

[38] EU-FP7-ICT-ROMEO, "D3.1 Report on 3D media capture and content preparation," May 2012, http://www.ict-romeo.eu/.

[39] Vanguard Software Solutions H.264/SVC SDK, http://www.vanguardsw.com/.

[40] S. Hemminger, "Network emulation with NetEm. , Linux conf au," in *Proceedings of the Australia's National Linux Conference (linux.conf.au '05)*, Canberra, Australia, April 2005.

[41] J. C. Bolot, "End-to-end packet delay and loss behavior in the Internet," in *Proceedings of the Conference on Communications Architectures, Protocols and Applications (SIGCOMM '93)*, pp. 289–298, 1993.

[42] ITU Radio Communication Sector, "Methodology for the subjective assessment of the quality of television pictures," Tech. Rep. ITU-R BT.500-11, 2002.

[43] ITU Radio Communication Sector, "Subjective video quality assessment methods for multimedia applications," Tech. Rep. ITU-T, Rec. P.910, 1999.

[44] ITU, "Subjective audiovisual quality assessment methods for multimedia applications," ITU-T Recommendation P.911, International Telecommunication Union, Geneva, Switzerland, 1998.

[45] ITU-T BT.2021, *Subjective Methods for the Assessment of Stereoscopic 3DTV Systems*, International Telecommunication Union, Geneva, Switzerland, 2012.

[46] DTIC, *Chi-Square Tests*, Defense Technical Information Center, 1976.

[47] A. Khan, L. Sun, and E. Ifeachor, "QoE prediction model and its application in video quality adaptation over UMTS networks," *IEEE Transactions on Multimedia*, vol. 14, no. 2, pp. 431–442, 2012.

[48] B. G. Tabachnick, L. S. Fidell, and S. J. Osterlind, *Using Multivariate Statistics*, 2001.

[49] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The WEKA data mining software: an update," *ACM SIGKDD Explorations Newsletter*, vol. 11, no. 1, pp. 10–18, 2009.

Hindawi Publishing Corporation
Mobile Information Systems
Volume 2016, Article ID 6270738, 13 pages
http://dx.doi.org/10.1155/2016/6270738

Research Article

A Low Energy Consumption Storage Method for Cloud Video Surveillance Data Based on SLA Classification

Yonghua Xiong,[1,2] Chengda Lu,[1] Min Wu,[1] Keyuan Jiang,[2] and Dianhong Wang[1]

[1]*School of Automation, China University of Geosciences, Wuhan 430074, China*
[2]*Department of Computer Information Technology & Graphics, Purdue University Calumet, Hammond, IN 46323, USA*

Correspondence should be addressed to Min Wu; wumin@cug.edu.cn

Received 29 October 2015; Revised 4 February 2016; Accepted 29 February 2016

Academic Editor: Ansgar Gerlicher

With the continuous expansion of the amount of data with time in mobile video applications such as cloud video surveillance (CVS), the increasing energy consumption in video data centers has drawn widespread attention for the past several years. Addressing the issue of reducing energy consumption, we propose a low energy consumption storage method specially designed for CVS systems based onthe service level agreement (SLA) classification. A novel SLA with an extra parameter of access time period is proposed and then utilized as a criterion for dividing virtual machines (VMs) and data storage nodes into different classifications. Tasks can be scheduled in real time for running on the homologous VMs and data storage nodes according to their access time periods. Any nodes whose access time periods do not encompass the current time will be placed into the energy saving state while others are in normal state with the capability of undertaking tasks. As a result, overall electric energy consumption in data centers is reduced while the SLA is fulfilled. To evaluate the performance, we compare the method with two related approaches using the Hadoop Distributed File System (HDFS). The results show the superiority and effectiveness of our method.

1. Introduction

Cloud computing is defined by the US National Institute of Standards and Technology (NIST) as a computing model that provides easy access to a shared pool of resources such as computing facilities, storage devices, and software applications through Internet anywhere and anytime [1, 2]. Advancement of cloud computing has been promoting the development of video surveillance and has formed a totally new cloud computing service model called video surveillance as a service [3]. According to the statistics by IMS Research, the demand for video surveillance service based on cloud computing is growing at an annual rate of 20% to 30% [4, 5]. Because of high reliability and availability, cloud video surveillance (CVS) services can help solve the problems encountered with traditional video surveillance technologies including high maintenance costs of communication and storage devices and low level of data security. For example, in [3], the architecture of typical CVS for commercial service based on virtual machine technology is designed to provide users with a kind of "plug and play" monitoring service with less cost and better convenience, and a task-oriented scheduling method is also proposed subsequently to reduce the energy consumption of the CVS. In [6], a model of video surveillance system based on the network coding distributed file system was established to save storage space.

It is estimated that global mobile data traffic will increase 13-fold between 2012 and 2017, and the video data will be two-thirds of such explosive traffic [7, 8]. Mobile devices such as laptops, tablet PCs, and smartphones are connected by networks that serve end users to get access to large service applications. The CVS system becomes a kind of mobile video application because of the wide use of mobile devices which make it convenient for users to browse, play back, or monitor in real-time video surveillance. The exponential growth of mobile video data, widespread adoption of mobile devices, and demands for video surveillance make the CVS system based on mobile networks a future direction of development.

Given its role of the main carrier of all surveillance video and multimedia data in a CVS system, the video data center is largely a contributing factor of the system performance. Some studies show that the average utilization rate of servers in data

centers is only 20% to 30% [9]. The energy consumption of idle devices is more than 50% of that in a full load [10]. This indicates that currently the power utilization rate of video data centers is low and there is room for improvement and optimization. Low energy efficiency of the data centers also translates to billions of wasted dollars every year [2]. An improvement of technology on data center energy efficiency will bring in significant saving [11]. Different from [6] whose purpose is to save storage space, we address the energy saving problem in this paper.

When users want to use the video service system, they are required to sign a service level agreement (SLA) with the providers. The SLA is used as a contract for users to agree on the cloud service items including video quality, monitoring time, volume of data storage, and pricing. An optimal resource provisioning in the SLA will help reduce the providers operational cost [10]. Our goal in this paper is to develop an energy saving storage method based on an improved SLA for CVS systems. Unlike the task-oriented scheduling method presented in another paper of ours [3], the method presented in this paper focuses on storage node scheduling for energy saving. In practice, both of our methods can be applied to the same data center.

The rest of the paper is organized as follows. Section 2 describes the CVS system and its characteristics of storage energy consumption. Section 3 presents the proposed method together with corresponding deployment and operation strategy. Experiments and conclusions are given in Sections 4 and 5.

2. Related Work

Energy saving of data centers is one of the most challenging problems in the resource-constrained networking field. This problem has attracted a lot of attention in the last few years. Currently, there are two main solutions to the data center energy consumption problem: hardware energy saving technology and software energy saving technology [12]. The hardware energy saving technology mainly reduces the energy consumption through the use of low power devices or adoption of specific hardware and device architecture [12–14]. Obviously, the hardware technology is not flexible and has low portability. In a large-scale commercial cloud video data system, there are numerous different devices. Replacing all devices that are already in use is too expensive and inconvenient. Therefore, the limitation of hardware technology approaches makes it difficult to be implemented widely.

On the other hand, the software energy saving technology focuses on utilizing software strategies to turn off some nodes or put them into sleep state with minimum performance degradation [15]. Comparing with the hardware technology, the software approach is more flexible, much cheaper, and more convenient, receiving more attention. Primarily, there are two storage management methods: dynamic data placement and static data placement.

The dynamic data placement method leverages some strategies or factors to dynamically adjust the location of data from node to node. For example, the data with a high access frequency will be migrated to some nodes together. Thus, the rest of the nodes will be set idle and can be put into low energy consumption state. In [16], dynamic consolidation of VMs which use live migration is improved. The method is the resource management strategy on "energy-performance tradeoff," that is, minimizing energy consumption while meeting the SLAs. The proposed deterministic algorithms and adaptive heuristics are claimed to have high efficiency. In [17], an energy saving algorithm is developed based on the storage structure configurations of data blocks. The frame area of data blocks is divided into active and sleep. Each data block is moved to corresponding area according to the access frequency and the blocks in sleep area are switched off to save energy. The approach is flexible and efficient. However, both methods involve the process of data migration which takes a lot of network bandwidth and results in migration delay [18]. The cost of migration will be significant especially when the methods are applied in data intensive applications. Therefore, the dynamic data placement method is not suitable for storing streaming media data and is difficult to implement in CVS system.

On the contrary, the static data placement method does not rely on dynamic resource management such as data migration. It shuts down or suspends redundant nodes by reasonable deployment according to improved schedules or policies. Usually, nodes are turned off in certain time under the premise of having a certain degree of fault tolerance or performance lost [19, 20]. A research group at Stanford University has successfully shut down more idle nodes in Hadoop by improving the strategy of replica placement [19]. But its data availability is to be verified for the CVS system. An energy conservation replica placement method is proposed in [20]. DataNodes are separated into three logical zones according to the forecast results: hot, warm, and cold. Different energy management strategies are adopted for different zones to achieve energy saving. Once the activity frequency of nodes does not match the current zone, the nodes should be transferred to the suitable zone. However, the transfer can only be done step by step among zones, without taking into account the high randomness of user access in a mobile video networking system.

In this paper, we propose an energy saving storage method based on SLA classification for CVS systems. Our approach will divide VMs and data storage nodes into different classifications according to the modified SLA with an extra parameter of access time period. VMs and nodes in different classifications will operate separately at different time period under the guidance of designed schedule. Each time period has the proper number of operating VMs and nodes while redundant ones are shut down to save energy. No data migration is required to consolidate resource. For the aforementioned reasons, our method can be classified as a static data placement method. Unlike hardware energy saving technology, our proposed method is easy to implement and more suitable for mobile video applications than dynamic data placement methods. The effectiveness of our method can be shown in comparison with other static data placement methods.

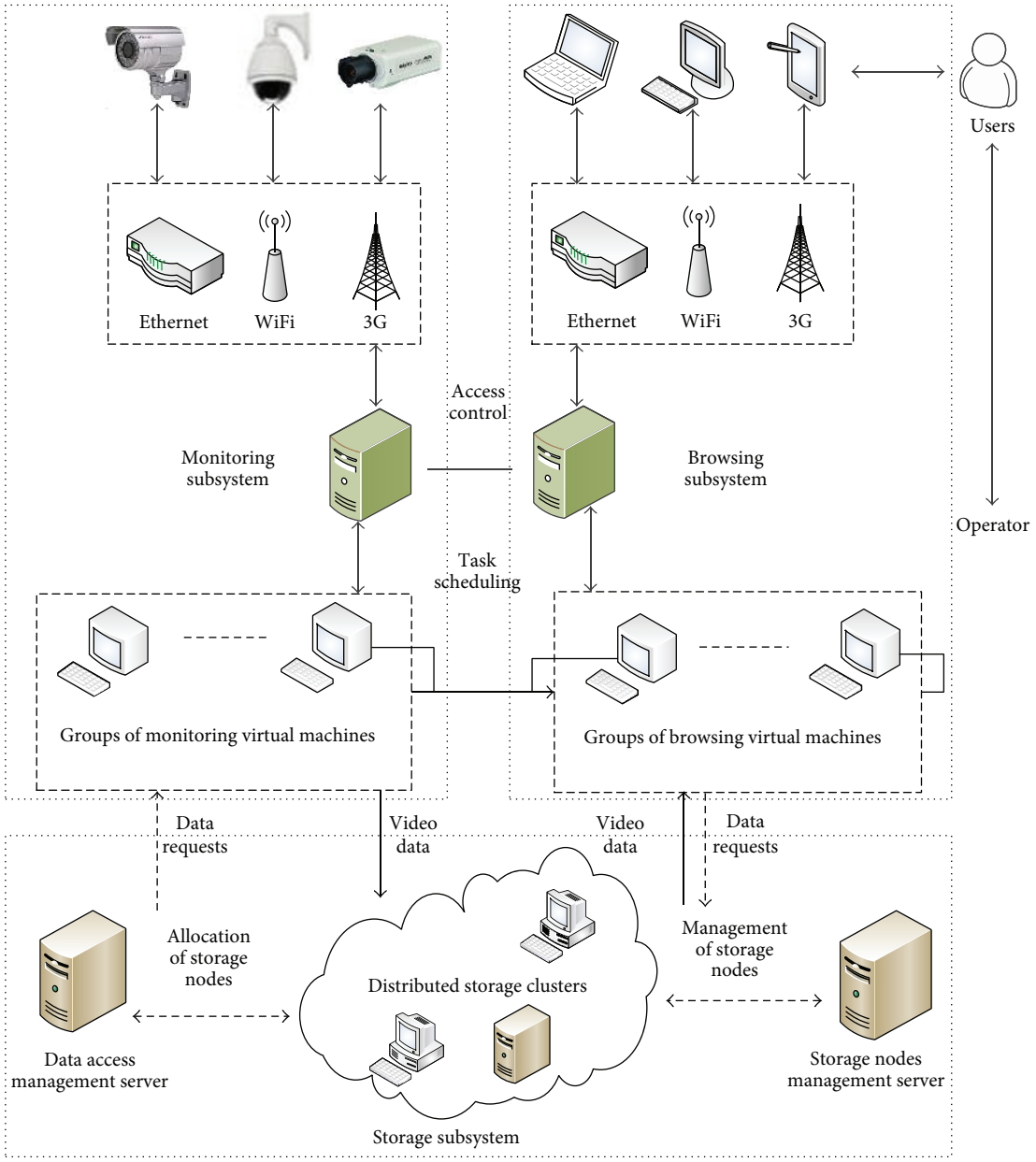


Figure 1: Architecture diagram of CVS system.

3. Cloud Video Surveillance System

The video surveillance system has been widely used in military, transportation, production, healthcare, service, and daily life. However, problems exist for an ordinary network video surveillance system: high investment and maintenance costs, limited system scalability from the space, poor security, and low reliability on stored data. These limitations have greatly led to the development of the CVS system, a video surveillance system based on cloud computing.

3.1. System Structure and Operation Mechanism. The CVS system, that is, video surveillance system based on cloud computing, consists of three parts: CVS terminals, a cloud video data center, and user browsing terminals [21, 22]. The architecture diagram of a CVS system is shown in Figure 1 which is from our prior work [23]. The monitoring subsystem, storage subsystem, and browsing subsystem are corresponding to the CVS terminal, cloud video data center, and user browsing terminal, respectively. In the figure, the webcams in the monitoring subsystem serve as surveillance terminals to obtain surveillance video data. In the browsing subsystem, user can access video data through a browser installed on desktop computers, tablet PCs, and mobile phones for obtaining video data and performing real-time monitoring.

The storage subsystem, as a data center, mainly consists of distributed storage clusters, data access management server,

and storage nodes management server. It is used to store and manage the historical surveillance video data. To make a CVS system operational, the monitoring terminals have to be installed in the areas a user wants to monitor and be registered in the CVS center.

An SLA is an agreement between two service providers or between a service provider and a service consumer. This is an important way to ensure the service quality in various complex network environments so far. An SLA includes a collection of items such as service description, priority, responsibility, guarantee and service level, the availability of services, reaction speed, price, and penalty for agreement violation. It is a document that describes the minimum performance requirements that can be met by service providers when they provide the services.

Different SLA templates are used for different demands [24]. This means that the items in SLA can be arbitrarily expanded according to the negotiation of both sides. Solutions have been developed for services with no matching SLA templates to achieve agreements [25]. There are some negotiable items such as the price and the maximum number of users. Different prices are corresponding to different SLA parameters that provide a range of choices at each level to meet users' customized requirements.

The basic structure and specification of typical SLA are shown as follows:

Property:

- Tenant.
- Validity Period.

Service description:

- Service parameters.
- Service level objectives (SLO).
- SLA parameters.
- Metrics.
- Function.
- Operand.
- Measurement service.
- Expression.
- Threshold.

Violation Handling:

- Predicate.
- Action.

Parameters of a typical SLA can be classified into three classes: property, service description, and Violation Handling. Parameters of the property mainly include the Tenant and Validity Period. The Tenant includes the information of ID, the maximum number of users, and the average number of active users. The Validity Period refers to the Validity Period of SLA specifications. Service descriptions of an SLA describe the service quality level or value range, including performance parameters of the SLA, calculation algorithm, and some necessary information about service provisions. The Violation Handling includes Predicate and Action. The Predicate specifies the operation of each SLA parameter to be calculated. The Action is what is to be taken for SLA violations.

Based on the registered information and SLA, first, the CVS center performs the video monitoring tasks through VMs that are controlled by the access management server, and later the center acquires monitoring data and stores the data in the storage subsystem. When real-time monitoring is needed, the system will directly connect the monitoring VM with the browsing VM. When monitoring historical video is needed, the system will establish a connection between the browsing VM and the storage subsystem to execute the transmission of data and commands. A user can achieve the monitoring purpose through a browser to access the CVS center.

3.2. Analysis of Storage Energy Consumption in CVS System. In the CVS system, the amount of video data increases linearly with time. Currently, the regular hardware system is unable to meet the needs for the space capacity and storage performance. Meanwhile, it is necessary for the system to deal with huge access requests from users and provide simultaneous services. Therefore, the CVS system must have a strong data storage technology that is of characteristics of scalability, high throughput capacity, high transmission rate, high reliability, and so forth.

The distributed storage technology adopts the scalable topology data organization that is of effective fault tolerance and high resistance to damage. It also includes the efficient distributed router, file cache system, and load balancing algorithm. It can avoid or alleviate the influence of the dynamic and unpredictable nature of network environment and provide consistent high performance storage services to every user in the distributed storage system. Therefore, the video data are generally stored using distributed storage method in the CVS system and the reliability is ensured by the redundancy copies.

The commonly used distributed file systems mainly include HDFS of Apache, OceanStore of UC Berkeley, CFS of MIT, and GFS of Google [26–29]. These storage systems have the characteristics of petabyte storage capacity, easier expandability, high service bandwidth, and so forth. They are effective ways for massive data storage and can meet the storage and access request needs for a CVS system.

With the growth of storage capacity, a large number of CPUs, disk arrays, and numerous cluster servers are continuously running and consuming a lot of energy in the distributed storage system. Power consumption has become the biggest expense in the continuous-operating service. The main reasons that make a CVS distributed storage system inefficient are as follows.

(1) The energy consumption at the idle state is still about 50% of that at the full-load state. An idle state means that there is no storage and access load in the system. In the CVS system, monitoring demands or video data storage may not be needed all day in some scenarios, for example, when videos are being

watched by some people or storage operation is triggered by capturing an object's movement. Numerous nodes are still wasting energy while they have no task.

(2) Redundant configuration in the system designed to meet the needs of peak application will consume a large amount of energy. In the distributed storage, copies of storage data are often added to improve the probability of successful data access. To maintain the accessibility of these data copies, some redundant nodes are required to be run at the normal state that will consume extra energy power. In the distributed storage of cloud video data, the extra energy consumption problem is even worse because more storage space and storage nodes are needed due to the large number of users and larger video data capacity.

(3) The system provided service capability is greater than what the actual applications need. In many cases, distributed systems generally have a redundancy design to ensure the system service's quality. That is, extra nodes in the system are considered as available storage nodes and used, yielding some unwanted energy consumption.

Having considered the above reasons, we add parameters such as access time period and video data storage space into the standard SLA. And a low energy consumption storage method is proposed based on the classification of access time specified in the SLA. The storage method in the CVS data center is optimized to shut down nodes not being used. The system power consumption is reduced at the premise of ensuring the SLA.

4. Low Energy Consumption Storage Method Based on SLA Classification

In addition to general parameters, the SLA of a CVS system should include characteristic parameters of a video surveillance system such as video quality, video data storage space, and scalability. Even though the video quality and storage space parameters are not specifically discussed in the paper, they are a part of the CVS SLA and can be used as reference values to other related designs of CVS. An access time period parameter, what we will mainly discuss, is added in order to limit the time range of the user's monitoring tasks.

4.1. Design of SLA with Access Time Period for Classification. The studies on CVS systems are still at the stage of exploration and trial operation. To the best of our knowledge, there is not a mature SLA for the CVS system at the moment. By analyzing the operation flow of CVS systems, the design of SLA for CVS systems proposed in the paper is shown in Figure 2. Compared to a typical cloud computing SLA shown in Section 3.1, there are the following differences.

An access time period parameter is added to the service description parameters. It is the time period for users to access the surveillance video data. In the CVS system, there is large randomness for users to access the video data because of their needs or preferences. The approximate time range of users' access to video data can be arranged with the access time parameter. According to the parameter, the 24 hours per day can be divided into several common time periods that can be chosen: "0~4 a.m." as class A, "8 a.m.~12 a.m." as class B, and so forth. When the real-time monitoring is needed, the "all-day" class can be chosen for the access time period. According to the classification, the storage nodes can be labeled correspondingly. In practice, not all users need to perform monitoring twenty-four hours a day, and users can purchase the service of corresponding access time periods according to their needs. During the purchased time period, the system can provide stable and reliable video surveillance services. Therefore, an SLA with access time period not only can, by regulating users' access behavior, make the unified management of resources easier, but also provides users with a more reasonable service with more combinations of selected access time periods.

The video quality and video storage space parameters are added to the service description parameters as well. Video quality can be classified into three grades as high (H), medium (M), and low (L) according to different video resolutions. Since different applications have their own needs such as a user's requirement and network bandwidth, it is not necessary to use the same video quality for all. Therefore, users can get a reasonable choice of the video quality level at the premise of their needs. For example, a high level may be chosen for the building monitoring, a medium level may be chosen for enterprise monitoring that requires a good video quality but is limited by the budget for service, and a low level may be chosen for the urban traffic monitoring service.

It is costly to operate the cloud data center for service providers especially when there is no restriction to users for the storage allocation. The greater the allocated storage space is, the longer the monitoring time of the historical video will be. The video storage space, in a unit of gigabyte, is a very important parameter for both CVS users and service providers. Based on the coding standard and parameters such as video quality and access time period, the storage space for a task can be approximately calculated [3]. That is, the video storage space is a reference parameter provided by the service provider to provision the minimum storage space for a user according to the access time period. It is commonly set at the maximum value for redundancy. Therefore, users cannot directly decide its value but can preset a minimum storage space according to their needs. If the storage space is not preset, it will get the default value calculated by the system. When an adjustment of minimum storage space is required by users, the parameter can be updated. However, the change is not immediate but will take effect only at the next operation time. Once users set the storage space parameter when the SLA is initialized, the overall space for tasks will be calculated. Therefore, purchasing storage space according to the need can not only control users' cost more accurately and reasonably, but also help service providers schedule resources reasonably well based on the storage space parameter in the SLA. This will improve the resource utilization and thus save the energy.

The scalability in SLA parameters is further enhanced. Although scalability is one of the basic characteristics of cloud computing and has its common specifications, it should also include the ability for users to increase or decrease the number of monitoring sites, the monitoring time range, and the video data storage space according to the needs.

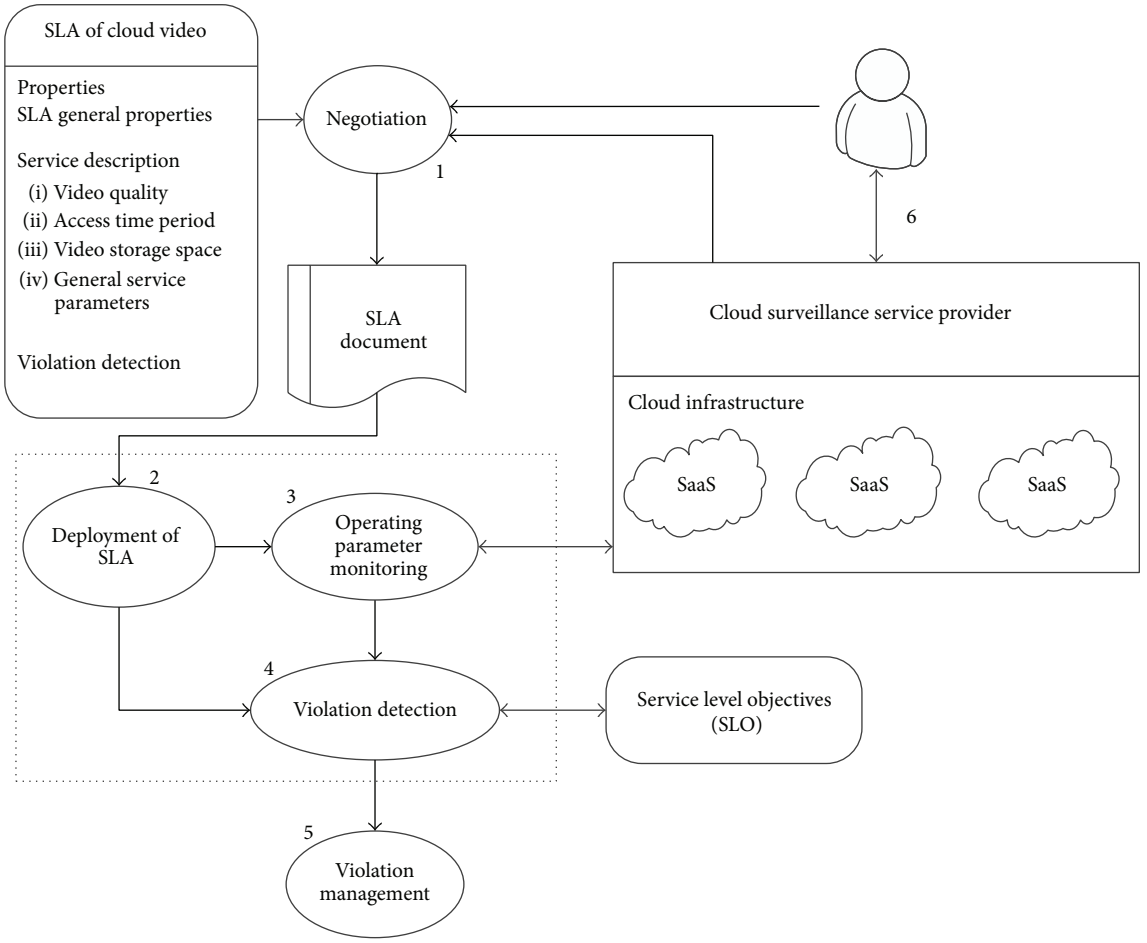


Figure 2: SLA framework of CVS.

The SLA deployment and operation management process mainly consists of the following steps.

Step 1 (negotiation). Parameters such as the monitoring time period, level of service quality, cost, and performance should be discussed between CVS service providers and users.

Step 2 (deployment). After the negotiation, SLA should be finalized and deployed to the CVS center and then the SLA monitoring, operation parameter monitoring, and violation management are activated.

Step 3 (operating parameter monitoring). Parameters in the SLA are managed and calculated in real time in the operation process of the CVS system.

Step 4 (violation check). The parameters in SLA are checked against the data measured in Step 3 to make sure that they have met the specified service level objectives.

Step 5 (violation management). The corresponding Action will be taken based on the results of violation check. Either the penalty payment as specified or replacement of other services in the SLA will be requested if violations occur in the process of the service.

Step 6 (termination of service). When the SLA valid period is over, the related instructions are sent to stop the monitoring and storage tasks and the service will be terminated as specified in the SLA.

4.2. Deployment and Resource Management for Storage Method Based on SLA Classification. In the low energy consumption storage method, the distributed storage cluster in a CVS data center is deployed to store and manage the historical video files at first. Later, different servers are deployed in the CVS center to manage the VMs and storage nodes. They include the SLA information management server, resource management server, data access management server, and storage nodes management server. Figure 3 shows the overall structure of the low energy consumption storage method.

The SLA information management server contains all users' access time period information as agreed in the SLA. On the one hand, it gets SLA service demand parameters and other information from the user registration information database server in the cloud surveillance center. On the other hand, it processes the information and passes the processed results as parameters to the resource management server, data access management server, and storage nodes management server at the same time.

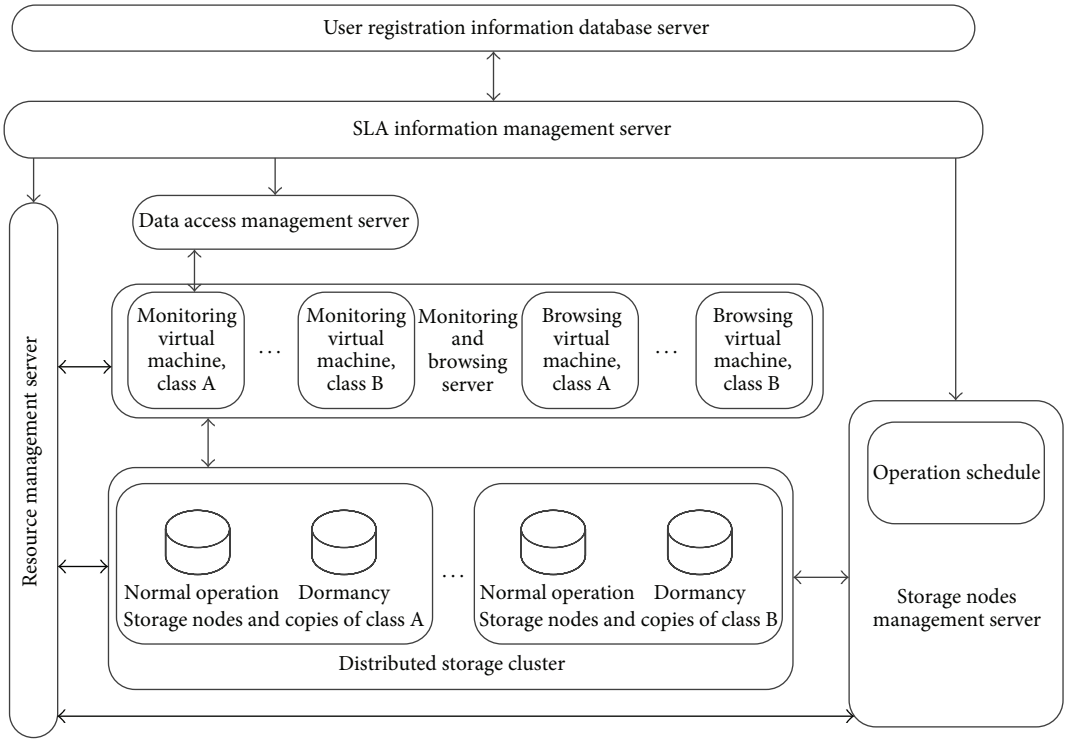


Figure 3: Overall structure of the low energy consumption storage method.

The resource management server counts the number of VMs, storage nodes, and remaining storage space in the CVS system. It receives the parameters from the SLA information management server. Afterwards, based on the parameters and the schedule from storage nodes management server, it integrates, manages, and classifies the VMs and the storage nodes in the CVS center.

The data access management server is one of the most important parts in the CVS storage subsystem. It is mainly responsible for sending data storage requests to monitoring VMs and to allocate storage nodes to the corresponding monitoring VMs.

The storage nodes management server is the key to the low energy consumption storage management method because it provides the operation schedule. The schedule is initialized by access time period parameters which are the output of the SLA information management server and the classification information from the resource management server.

Regarding the management of storage nodes, it is obviously improper to classify and manage each storage node alone by the resource management server. Each storage node can be managed through the master node in the distributed storage cluster as long as each master node is associated with the storage nodes in the same subcluster. Then, the resource management server can classify all storage nodes according to the IP address of the master node and the associated storage nodes in the subcluster. The IP address of each master node in each subcluster is combined with the same class of VMs' IP addresses as a subset to be controlled. In this way, all storage nodes can be managed through those master nodes. For each class of VMs and storage nodes after classification, further fine-grained scheduling management is possible through other management strategies that many research efforts are dedicated to. In this paper, we mainly focus on coarse-grained resource management and the further fine-grained management will be studied and developed in the future study.

4.3. Low Energy Consumption Operation Strategy. Once monitoring or browsing tasks are requested during an access time period, the access control program of the CVS center will select the classified category corresponding to current time period so that the monitoring or browsing tasks can be executed through VMs in the same category. For the historical storage data, the data access management server will periodically send requests to the monitoring VM for data storage access and adjust the categories of storage nodes based on the SLA classification information. The video data with the length up to one or two days (real-time data) are stored in the virtual disks of VMs. The minimum capacity of virtual disks can be estimated based on the maximum bandwidth, video transmission rate, and storage time. However, the historical data are stored in the specific nodes in the network distributed storage so that the static classification method can be applied for unified management.

The operation schedule of the storage nodes in the server is deployed to control the operation mode of data storage nodes in the distributed storage cluster. Based on

```
    ...
(1) Initial();    //system starts
(2) for i from 1 to r do    //i: ith class based on access time periods
(3)     chooseRandom(U_i);    //U_i: number of users in each time period.
    end for
    ...
(4) While()    //main loop of CVS system
    {
     ...
(5)  If(update)    //information updated
     {
(6)   if(current)    //new user asks for current service
      {
(7)       New VMs/Nodes added;
(8)       Validity Period recorded;
      }
(9)   else if(! current)    //new user does not ask for current service
      {
(10)      for i from 1 to r do
(11)      f_SLA(i, U_i, ...);    //classify VMs and nodes based on updated information.
          end for
      }
     }
(12) if(request)    //task to access
(13)    if(request == i)    //task is relevant to ith class.
(14)       for j from 1 to dn_i do
(15)          Node_i = normal;    //Node_i: Nodes of ith class.
           end for
        else
(16)       other nodes = sleep;
        end if
     end if
     ...
    }
```

Algorithm 1: Pseudocode of operation strategy.

the analysis in Section 3.2, the pseudocode of operation strategy is illustrated in Algorithm 1.

At first, the service parameters including access time period and number of users in each access time period are treated as inputs to initialize the resource scheduling and allocation program. Thereafter, the VMs and storage nodes are classified into several classes such as A, B, C, by the resource management server in advance. Therefore, users cannot decide when the classification starts. The storage nodes and their replicas in each class of monitoring VMs are assigned to the same category as well.

Once a new user is added and registered in the CVS center, if its access time does not cover the current time, the information management server will update the classification information based on the access time period. Based on the information and the number of users in each time period, the resource management server then reclassifies the monitoring VMs, browsing VMs, and storage nodes. Further, the allocated VMs and storage nodes are adjusted based on the number of users in each class. In addition, the IP addresses of VMs in each category are bound as a group and saved in a storage file on the resource management server. If the newly initiated user asks for an immediate service, the task will be requested at once. The number of tasks increases in the current access time period while other tasks are still in operation. New VMs and storage nodes in the same category will be open and information is recorded in the parameter of Validity Period. The cost will be different from that of a whole access time period service and adjusted according to the Validity Period parameter in the SLA specification.

An operation schedule is designed for all storage nodes and their replicas in the data center. The storage nodes are defined in two modes: normal and sleep. For example, the storage nodes of class A, between 8 a.m. and 12 a.m., are normal while the storage nodes of class B are in sleep mode.

When a task is requested, the task access control program will send the task request to monitoring or browsing VMs corresponding to the current time period. Then, control commands are sent to the storage nodes according to the operation schedule. Only nodes corresponding to the agreed access time period are set to the normal mode and others are set to the sleep mode.

Once a browsing VM receives a request in the current access time period, data from the storage nodes in the normal

mode are transmitted to the same class of browsing VMs. The service parameters and the operation schedule are updated whenever the access time period information in the SLA is changed.

Suppose the total number of VMs in the CVS center is VM, the total number of storage nodes is DN, and the total number of classifications based on access time periods is r. The number of users in each classification of access time periods is set to U_i, $i \in \{1, 2, 3, \ldots, r\}$. The fraction of users in each classification of access time period is set to P_i, $i \in \{1, 2, 3, \ldots, r\}$. The number of VMs and the number of storage nodes in each classification of access time period are vm_i and dn_i, respectively, $i \in \{1, 2, 3, \ldots, r\}$. Then,

$$P_i = \frac{U_i}{\sum_{i=1}^{r} U_i}, \quad i \in \{1, 2, 3, \ldots, r\},$$

$$\mathrm{vm}_i = \mathrm{VM} \times P_i = \mathrm{VM} \times \frac{U_i}{\sum_{i=1}^{r} U_i}, \quad (P_i \geq 0), \tag{1}$$

$$\mathrm{dn}_i = \mathrm{DN} \times P_i = \mathrm{DN} \times \frac{U_i}{\sum_{i=1}^{r} U_i}, \quad (P_i \geq 0).$$

Since there are no monitoring or browsing tasks in VMs before the CVS system starts, the resource management server can randomly select vm_i VMs and dn_i nodes as one class through function ChooseRandom() according to U_i (the number of users in each class) and continue in this way until the classification is completed. Function $f_{\mathrm{SLA}}()$ is used to classify VMs and storage nodes according to updated parameters.

The pseudocode is an outline for the system operation. The running time of the algorithm is mainly contributed by the loops from Lines (2) to (3), (10) to (11), and (14) to (15) in the pseudocode, and the computational cost is $O(n)$. Clearly, the code is concise and easy to understand.

There are other details necessary but not shown in the code. When some storage nodes in a classification stop working due to fault or other accidents during operation, the system can still guarantee the availability of data and keep working. The distributed storage system can later recover the invalid replica by duplication and the number of duplications in each category is adjusted according to the number of users in this class.

Since the storage nodes in the sleep mode cannot store historical data, a system process program is needed to receive the access time period parameter from the SLA information management center. Every day, when each class of the storage nodes is running in the corresponding access time period, the process program will initiate data storage requests to the monitoring VMs of the same class. Later, it will store the video data in the virtual disk of the distributed storage system and empty the virtual disk space.

Thus, the video surveillance or browsing tasks can be assigned to the appropriate VMs to execute. The all-day cloud video services will be divided into services in different time periods that can be handled by classification.

5. Experiments and Results

In this paper, the low energy consumption storage method is proposed to improve the power utilization rate in a data center. Therefore, experiments are mainly conducted to investigate the percentage of running storage nodes in the storage system and the average running time of the storage nodes.

For the experiment platform, we use the Hadoop Distributed File System (HDFS) to build the required storage clusters. In the traditional HDFS method, all nodes are located in one HDFS cluster and the default storage management policy, which maintains all data replicas to ensure data availability and prevent data inaccessible problem from unexpected downtime, is used [19, 30]. A low energy consumption storage management method proposed by Stanford University also uses one HDFS cluster but with an improved storage management strategy [19]. In this study, a HDFS cluster is divided into several subclusters whose operation states are managed according to the operation schedule. The storage management in each subcluster still uses the default policy. The results are compared with those from the traditional HDFS method and Stanford University's method.

5.1. Design of Experiment Platform. First of all, the hardware environment and software environment are prepared for the deployment of the storage clusters. The hardware environment includes forty-eight PCs as storage nodes and seven PCs as management nodes called "NameNode." Each node is configured in a Dell Vostro computer (OS: Ubuntu 11.04, RAM: 1 GB, and disk size: 320 GB) with software applications of Hadoop 2.1.0, jdk 1.7, Eclipse 3.7, SSH, and so forth.

The experimental system consists of seven HDFS clusters named A through G. As the reference group, G cluster is used to operate the traditional HDFS method and the Stanford method. All nodes in G cluster use one HDFS system, one NameNode, and twenty-four DataNodes (including replicas).

Clusters A to F are the experimental group that adopts the proposed method. Each cluster has 4 DataNodes (see Table 1). Each node works by following the operation schedule in the storage nodes management server. For the ease of comparison, the number of DataNodes in G and the total DataNodes' number in A~F are the same. The six clusters from A to F are classified according to the SLA access time period. The SLA access time periods are divided into 6 classes for the 24-hour period with 4 hours for each class: class A for the access time between 0 a.m. and 4 a.m., class B for the access time between 4 a.m. and 8 a.m., and so forth.

It should be noticed that, in order to make a storage node react in time, each cluster has a NameNode running continuously. Since the energy consumption of the NameNode is negligible when a large number of storage nodes are involved in an actual data center's distributed storage cluster, only the energy consumption from the running storage nodes is counted in the experiment.

The structure of the experiment platform is shown in Figure 4. The replica number of DataNodes in each cluster is set to 2. We experiment with multiple groups of video storage data (5 G, 10 G, and 15 G) with different time periods.

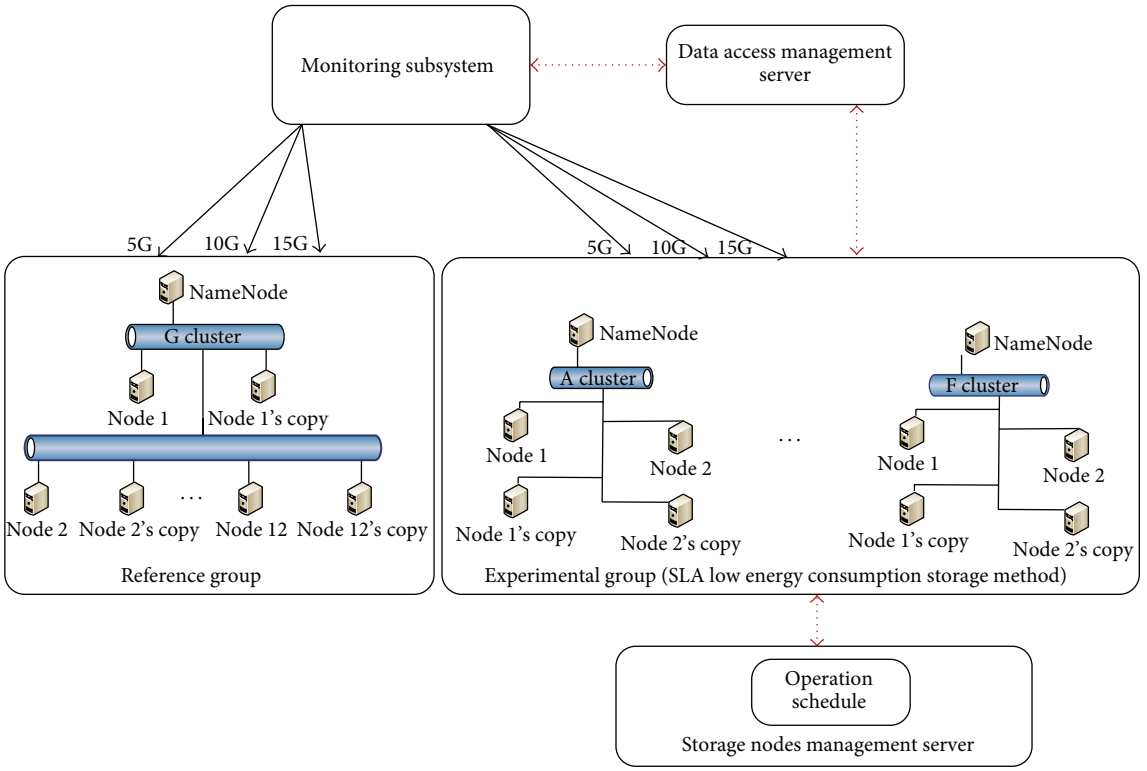


Figure 4: Structure of the experimental platform.

Table 1: Configuration number of DataNodes in clusters.

	Cluster	DataNodes
Experimental group	A (0:00–4:00)	4
	B (4:00–8:00)	4
	C (8:00–12:00)	4
	D (12:00–16:00)	4
	E (16:00–20:00)	4
	F (20:00–24:00)	4
Reference group	G	24

Reference group is used to operate the traditional HDFS method and the Stanford method while the experimental group uses the proposed method. Only experimental group is divided.

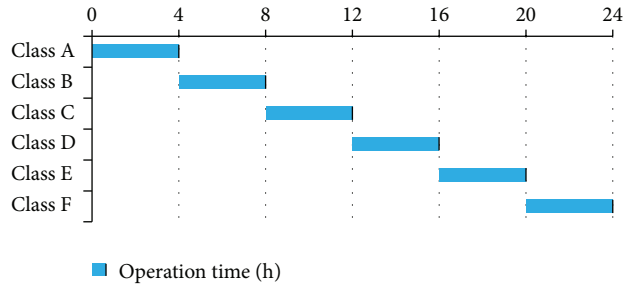


Figure 5: Operation schedule of the storage nodes.

The number of the running storage nodes (DataNode) and the running time in the multiple experiments are compared and analyzed. To facilitate the immediate verification of experimental results, one day is simulated in a three-hour experiment. In other words, the experiment time is compressed to one-eighth of the actual time, and each access time period in classification is reduced to thirty minutes. Meanwhile, we vary the SLA classification number to verify the energy saving effect at the premise of ensuring the equal number of total nodes in the experimental groups (A~F) and in the reference group (G).

5.2. Experiment Results. In the lab environment, the HDFS storage cluster is built to simulate a CVS data center. The number of storage nodes in the data center and running time of the storage nodes are recorded. Figure 5 shows the operation schedule of the storage nodes for the experiments.

It is clear to see, in Figure 5, the running state of the storage nodes from the operation schedule for the whole 24 hours. At any given time, only the class of storage nodes in access time period is in normal operation state (blue stripe) while other storage nodes not in the access time period are all in the energy saving state.

(1) Tests for Proportion of Nodes in Operating State. In the simulated experiments, with the same amount of data and loads, the initial number of storage nodes in the reference group and that in the experimental group are both set to twenty-four. All nodes are in the normal operation state. With six classifications, a day is divided into six time periods for

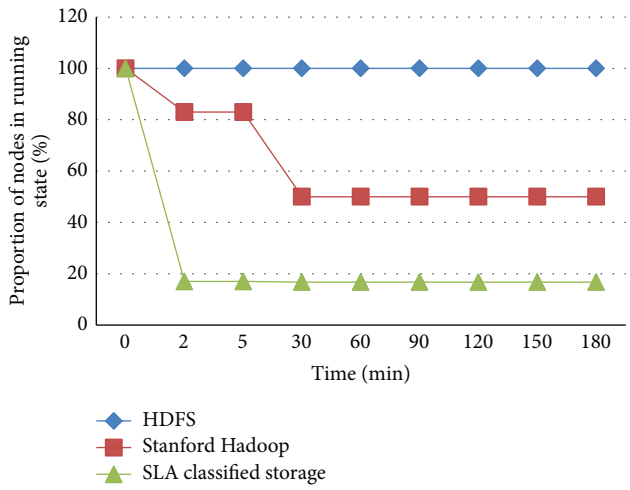


Figure 6: Comparison of storage nodes' operating rate.

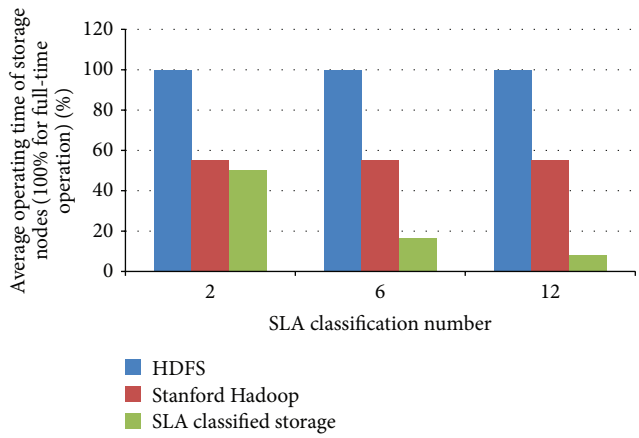


Figure 7: Average operating time of storage nodes.

data access. The storage nodes' running state changes with the execution of the storage operation as shown in Figure 6.

As illustrated in Figure 6, 100% of the storage nodes in HDFS tests (blue curve) are in the running state all the time. This proves the characteristics of the default storage management policy of the traditional HDFS method. In the experiments using the Hadoop cluster with Stanford University's method, about 50% of the storage nodes after the initial 30 minutes are in the running state (red curve). This is because only proper number of nodes is set to run according to the dynamic load analysis. Therefore, to some extent, the number of nodes in running state is reduced. Moreover, the maximum number of sleep nodes is limited to no more than 50% of the total number in order to guarantee the performance and the availability of data.

Although the low energy consumption storage method proposed in this paper needs to maintain the copies as well, the storage nodes and their copies in the data center are divided into a number of categories according to the access time period and only one class of storage nodes and their copies are running at any time. Therefore, it greatly reduces the number of storage nodes in the running state. It can be seen in Figure 6 that after the first 2 minutes only 17% of the storage nodes are needed to run (green curve), significantly lower than the other two methods. In short, our method with classifications results in significant saving in energy consumption.

(2) Tests for Average Time of Operating Storage Nodes. By varying the number of SLA classifications, we can test the effects of different SLA classification numbers on energy saving. Figure 7 shows the statistical average operating time of storage nodes versus number of classifications for the previously discussed three methods.

As one can see in Figure 7, nodes in HDFS tests (blue bar) keep running in full capacity regardless of the number of the SLA classifications, and the Stanford Hadoop follows the same trend, maintaining a constant average operating time. With the limitation of maximum number of sleep nodes not exceeding 50% of the total, Stanford Hadoop just makes some nodes sleep dynamically depending on different loads. With this in mind, it can be seen that the operating time of the Stanford Hadoop method keeps at about 55% for different SLA classification numbers. Our proposed low energy consumption storage method divides the whole day's tasks into small tasks and operates in batches at different time periods. With the increase of the number of SLA classifications, the operating time of storage nodes in different time period is reduced. Thus, the average operating time of all nodes in the data center is reduced too. As shown in Figure 7, the percentages of average operating time for the proposed method (green bar) are 50%, 17%, and 8.3% for classifications of 2, 6, and 12, respectively. In the experiment, the best case in energy saving is when twelve classifications are used, yielding 41.7% more in saving than the Hadoop method.

The higher the classification number is, the shorter the operating time of each node will be. In practice, if the operating time is too short, it is hard to ensure the completion of a task. For the aforementioned reason, the low energy consumption storage method will have a bottleneck after the classification number reaches a certain threshold. However, its energy efficiency is significantly higher than the other two methods and can be further improved.

6. Conclusion

We have proposed, in this paper, a low energy consumption storage method based on the SLA classification after carefully analyzing the structure of a CVS system, the operating mechanism, and the energy consumption problems in the distributed storage of video data. By designing an SLA with the addition of access time periods, the VMs and the storage nodes for monitoring operation and browsing tasks can be reasonably classified. Therefore, all-day CVS services are divided into various tasks associated with different time periods and processed separately. By designing an operation schedule for all storage nodes in the data center, the operation state of storage nodes can be effectively controlled to achieve energy saving. The deployment process and operation

strategy for the low energy consumption storage method are described in detail in the paper.

Experimental results show that the current method only uses 17% of the storage nodes in normal state and takes 8.3% average operating time with twelve classifications, resulting in more energy efficiency than the full load of traditional HDFS method and about half load of the Stanford improved HDFS method. In addition, the method is easy to implement and definitely offers certain application value. Our method demonstrates the following advantages:

(i) It is specially designed for a CVS system and is easy to adopt widely and deploy for other systems including cloud computing, cloud disk, or other mobile applications such as mobile UGC (user generated content) and mobile advertisement.

(ii) It can meet the performance requirement while minimizing energy consumption.

(iii) Compared with traditional and Stanford improved HDFS methods, the proposed approach proves its superiority and effectiveness.

In the future work, we will mainly aim at developing and improving a fine-grained scheduling management strategy for each storage node.

Competing Interests

The authors declare that they have no competing interests.

Acknowledgments

This work was supported in part by the National Nature Science Foundation of China under Grant no. 61202340, the International Postdoctoral Exchange Fellowship Program under Grant no. 20140011, and the Hubei Provincial Natural Science Foundation of China under Grant no. 2015CFA010.

References

[1] G. Aceto, A. Botta, W. De Donato, and A. Pescapè, "Cloud monitoring: a survey," *Computer Networks*, vol. 57, no. 9, pp. 2093–2115, 2013.

[2] T. Mastelic, A. Oleksiak, H. Claussen et al., "Cloud computing: survey on energy efficiency," *ACM Computing Surveys (CSUR)*, vol. 47, no. 2, p. 33, 2014.

[3] Y. Xiong, S. Wan, J. She, M. Wu, Y. He, and K. Jiang, "An energy-optimization-based method of task scheduling for a cloud video surveillance center," *Journal of Network and Computer Applications*, vol. 59, pp. 63–73, 2016.

[4] P. Simoens, F. De Turck, B. Dhoedt, and P. Demeester, "Remote display solutions for mobile cloud computing," *Computer*, vol. 44, no. 8, pp. 46–53, 2011.

[5] 2013, http://www.ctiforum.com/news/guandian/379338.html.

[6] Q. Zhao, K. Li, H. Li, Y. Wang, and H. You, "Model of cloud video surveillance based on network coding cloud storage system," *International Journal of Future Computer and Communication*, vol. 4, no. 4, pp. 286–289, 2015.

[7] T. De Pessemier, L. Martens, and W. Joseph, "Dynamic optimization of the quality of experience during mobile video watching," in *Proceedings of the 10th IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB '15)*, pp. 1–6, IEEE, Ghent, Belgium, June 2015.

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

[9] L. Na, M. Zhang, K. Li et al., "Green task scheduling algorithms with speeds optimization on heterogeneous cloud servers," in *Proceedings of the IEEE/ACM International Conference on Green Computing and Communications & International Conference on Cyber, Physical and Social Computing*, pp. 76–80, Hangzhou, China, December 2010.

[10] H. Goudarzi, M. Ghasemazar, and M. Pedram, "SLA-based optimization of power and migration cost in cloud computing," in *Proceedings of the 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid '12)*, pp. 172–179, IEEE, Ottawa, Canada, May 2012.

[11] Q. Zhang, M. F. Zhani, S. Zhang, Q. Zhu, R. Boutaba, and J. L. Hellerstein, "Dynamic energy-aware capacity provisioning for cloud computing environments," in *Proceedings of the 9th ACM International Conference on Autonomic Computing (ICAC '12)*, pp. 145–154, ACM, San Jose, Calif, USA, September 2012.

[12] Y.-J. Wang, W.-D. Sun, S. Zhou, X.-Q. Pei, and X.-Y. Li, "Key technologies of distributed storage for cloud computing," *Journal of Software*, vol. 23, no. 4, pp. 962–986, 2012.

[13] X. Q. Luo, Y. Qi, L. Wang, and Q. Wang, "Dynamic power dissipation control method for real-time processors based on hardware multithreading," *China Communications*, vol. 10, no. 5, Article ID 6520948, pp. 156–166, 2013.

[14] H. Shim, J.-S. Kim, and S. Maeng, "BEST: best-effort energy saving techniques for NAND flash-based hybrid storage," *IEEE Transactions on Consumer Electronics*, vol. 58, no. 3, pp. 841–848, 2012.

[15] H. Goudarzi and M. Pedram, "Energy-efficient virtual machine replication and placement in a cloud computing system," in *Proceedings of the IEEE 5th International Conference on Cloud Computing (CLOUD '12)*, pp. 750–757, IEEE, Honolulu, Hawaii, USA, June 2012.

[16] A. Beloglazov and R. Buyya, "Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in cloud data centers," *Concurrency Computation Practice and Experience*, vol. 24, no. 13, pp. 1397–1420, 2012.

[17] B. Liao, J. Yu, T. Zhang, and X.-Y. Yang, "Energy-efficient algorithms for distributed file system HDFS," *Chinese Journal of Computers*, vol. 36, no. 5, pp. 1047–1064, 2013.

[18] D. Kliazovich, P. Bouvry, and S. U. Khan, "DENS: data center energy-efficient network-aware scheduling," *Cluster Computing*, vol. 16, no. 1, pp. 65–75, 2013.

[19] J. B. Leverich, *Future Scaling of Datacenter Power-Efficiency*, Stanford University, 2014.

[20] X.-Y. Zhao and L. Wang, "An activity-based replica placement method of energy-conservation," in *Proceedings of the IEEE International Conference on Fuzzy Theory and Its Applications (iFUZZY '13)*, pp. 446–451, Taipei, Taiwan, December 2013.

[21] M. S. Hossain, M. M. Hassan, M. A. Qurishi, and A. Alghamdi, "Resource allocation for service composition in cloud-based

video surveillance platform," in *Proceedings of the IEEE International Conference on Multimedia and Expo Workshops (ICMEW '12)*, pp. 408–412, Melbourne, Australia, July 2012.

[22] June 2013, http://www.cloudsurveillance.com.

[23] Y.-H. Xiong, S.-Y. Wan, Y. He, and D. Su, "Design and implementation of a prototype cloud video surveillance system," *Journal of Advanced Computational Intelligence and Intelligent Informatics*, vol. 18, no. 1, pp. 40–47, 2014.

[24] S. Yangui, I.-J. Marshall, J.-P. Laisne, and S. Tata, "CompatibleOne: the open source cloud broker," *Journal of Grid Computing*, vol. 12, no. 1, pp. 93–109, 2014.

[25] A. Kertesz, G. Kecskemeti, and I. Brandic, "An interoperable and self-adaptive approach for SLA-based service virtualization in heterogeneous Cloud environments," *Future Generation Computer Systems*, vol. 32, no. 1, pp. 54–68, 2014.

[26] Y. N. Wang, S. D. Zhang, and H. Liu, "The design of distributed file system based on HDFS," *Applied Mechanics and Materials*, vol. 423-426, pp. 2733–2736, 2013.

[27] A. Dandoush, S. Alouf, and P. Nain, "Lifetime and availability of data stored on a P2P system: evaluation of redundancy and recovery schemes," *Computer Networks*, vol. 64, pp. 243–260, 2014.

[28] D. Wang, Z. Hu, J. Zhang, L. Zhang, and Z. Wen, "CFS: the design and implementation of a cluster file system service on Inspur AS3000," in *Proceedings of the International Conference on Computational and Information Sciences (ICCIS '11)*, pp. 847–849, Chengdu, China, October 2011.

[29] X. Y. Zhou and L.-Y. He, "A virtualized hybrid distributed file system," in *Proceedings of the 5th International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC '13)*, pp. 202–205, Beijing, China, October 2013.

[30] D. Borthakur, "The hadoop distributed file system: architecture and design," *Hadoop Project Website*, vol. 11, no. 2007, article 21, 2007.