

# A Cloud-based Energy-Efficient Service Architecture for Adaptive Multimedia Streaming

Jingjing Wang<sup>1, a</sup>, Jianxin Song<sup>2</sup>,

<sup>1</sup> Nanjing University of Posts & Telecommunications, Nanjing 210003, China;

<sup>2</sup> Nanjing University of Posts & Telecommunications, Nanjing 210003, China.

<sup>a</sup> iwangjingjing@139.com, <sup>b</sup> songjx@njupt.edu.cn

**Keywords:** Adaptive multimedia streaming, Cloud computing, Energy efficient, Real-time transcoding.

**Abstract.** As the technology of applying a cloud network to cloud multimedia matures, a solution to the present demand for high quality and diversified cloud multimedia can be provided. However, how to meet the users' demand for high quality and diversified cloud multimedia with handheld devices, which have limited power, is an interesting and challenging study. In this paper, a cloud based energy- efficient service architecture for adaptive multimedia streaming (EAMS-cloud) is proposed. In addition, the energy consumption of smart phones during video playback has been analyzed, based on which an optimization algorithm on energy consumption is proposed from maximizing the video quality. Several simulations are conducted, and the results turn out that the proposed approach can effectively reduce the power consumption of the terminal device.

## 1. Introduction

The focus of recent researches of adaptive multimedia streaming is mainly on how to transcode media for numerous users instantly [1,2], how to balance the power consumption of servers and the achieved QoE of end users[3], how to use proper network protocols to deliver media contents in cloud environments [4], how to schedule virtual machines for multimedia application in the cloud [5,6]. However, the specification of mobile hardware has been rarely considered [7], especially the limited power of terminal equipment. Whether the energy is sufficient for whole video play is as important as whether the current bit rate can guarantee smooth playback. The user not only want to view high quality media content, but also need to save mobile device's energy as much as possible to finish the video.

[7] proposed a cloud multimedia streaming architecture and a corresponding dynamic adjustable multimedia streaming (DAMS) algorithm, which decreased the loading of content storage servers substantially and reduced power consumption of the mobile device effectively. However, there are some deficiencies in this scheme:

(1) Although the overall power can be reduced effectively by adjusting the LCD [8], with multimedia services, users are sensitive to brightness, they dislike video brightness that repeatedly changes;

(2) As for the resolution, a particular terminal equipment supported is fixed, if there are too many optional target resolutions, the possibility that the selected resolution does not match the supported resolution will increase, and the difficulty of decoding will accordingly arise;

(3) The way they selected the optimal bitrates and resolution is so complicated that the decoding time might be longer.

For the aforementioned problems, we make some improvements to the traditional adaptive streaming transmission system combined with the cloud-based architecture proposed in [7]. The processes of policy evaluation and real-time transcoding have been added to the proposed EAMS-cloud system. To enable the system to dynamically adjust the multimedia based on the energy consumption, in the policy evaluation module, a mathematical model of influential factors of energy consumption of terminal equipment and the multimedia codec parameters has been formulated based

on a further analysis of the energy consumption during video playback. Then a scheduling algorithm is adopted to solve the presented model. Compared to the traditional DASH system and the DAMS scheme, this approach could derive more matching target bitrates and resolution for adaptive streaming with lower computational complexity.

The remainder of this paper is organized as follows. The framework design of the cloud-based energy efficient streaming system and its' workflow are described in Section II. In Section III, the energy consumption of smart phones during video playback has been analyzed, based on which an optimization algorithm on energy consumption is proposed for dynamically adjusting multimedia content. Section IV conducts experiments and evaluation, while Section V offers conclusions.

## 2. Architecture design of the cloud based energy efficient streaming system

This paper makes some improvements to the traditional adaptive streaming transmission system by using cloud computing. Fig. 1 shows the overall system architecture, which consists of two parts, namely cloud server and mobile client. The terminal device connects with the cloud server via LTE 4G network to get the processed multimedia data for decoding and displaying.

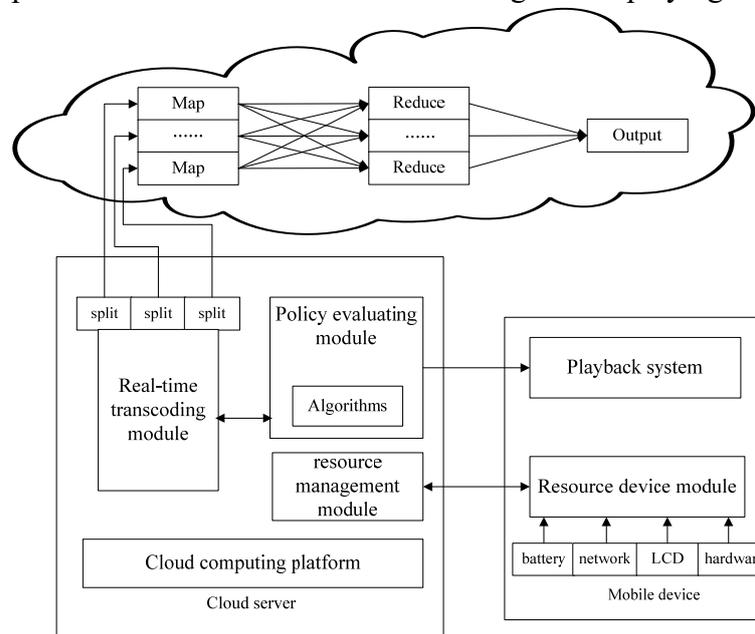


Fig. 1 The framework design of the proposed service architecture over cloud

The resource management module is in charge of receiving the configuration information of the terminal device, including the maximum power of the processor, codec types the mobile device supported and the battery capacity. The policy evaluation module is critical of algorithm realization, in which the multimedia is analyzed and evaluated based on the designed algorithm. And then the optimal bitrates and resolution can be deduced. The real-time transcoding module is responsible for obtaining the deduced parameters from the policy evaluation module and transcoding the original video segment timely and dynamically. Different from the traditional DASH technology, the real time transcoding in the cloud is adopted instead of storing transcoded video segments in advance.

Taking the downloading of the first and second video slice as an example, the workflow of the system is described in detail, as show in Fig.2.

In the initial stage, after the terminal device makes a request to the cloud server according to the address list of video slice, the cloud server would receive a request from the client, and then a list of video slice file will return to the client. With this, the initial stage is complete and the display is started.

When the first video slice is requested, there is no adaption in the beginning phase of display for the condition of the client cannot be perceived by cloud server, thus, cloud server transcodes the multimedia data according to the default bit rates and resolution through real-time transcoding module.

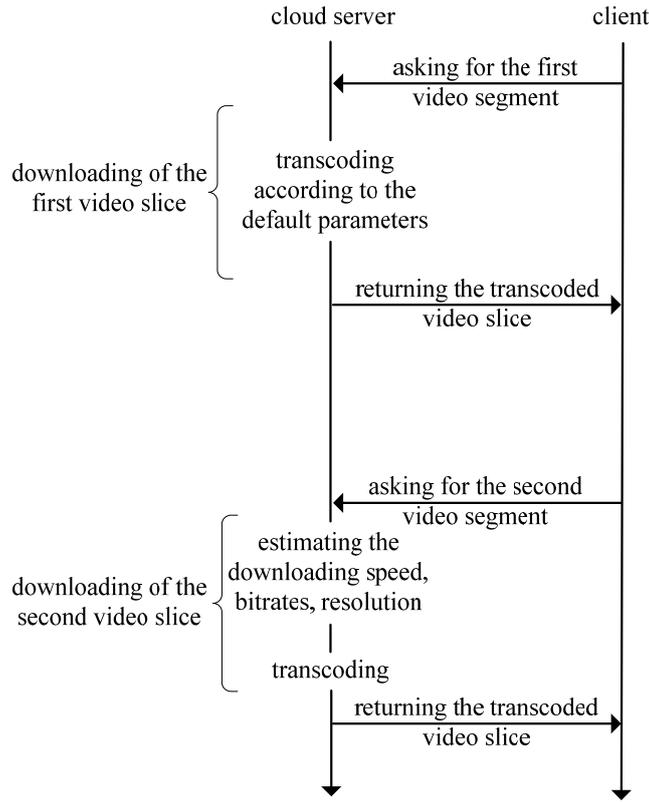


Fig. 2 The work flow of the system

After the downloading of second video slice starting, in policy evaluating module, the cloud server accesses the target bit rates of the second segment, according to the power consumption of the first video slice, and in real-time transcoding model, the next video segment is transcoded.

### 3. Problem formulation and algorithm design

#### 3.1. Problem formulation.

Regarding power consumption of static video, motion video, resolution, variable bit rate, fps, encoder, and decoder, studies have suggested that the decoding time is longer, and the power consumption is greater if the data complexity is higher [9]. Taking account of H.264, the video can be divided into I-frame, P-frame, and B-frame. A multimedia consists of many GOP (group of pictures), and a GOP comprises I-frame, B-frame, P-frame. In the paper, the multimedia data is divided into a plurality of video slices, each video slice is used as the basis to conduct estimation.

$M$  of GOP is defined as the distance between two I-frames;  $FrameSize$  denotes the frame size of each frame. In general, the bigger the  $M$  is, the greater the frame size is, and the video quality will be better, namely:

$$M \propto FrameSize \propto QoE \quad (1)$$

The power consumed by the processor for transcoding is denoted by  $E_{slice}$ , the time needed for decoding is denoted by  $T_{de}$ , greater frame size would lead to longer decoding time and higher power consumption, i.e.

$$FrameSize \propto T_{de} \propto E_{slice} \quad (2)$$

From Eq.(1) and Eq.(2), we can infer that the key to achieve the balancing between QoE and energy consumption is frame size. In the paper, the bitrates and resolution of video frames are chosen as the tunable parameters since they will affect the frame size.

As show in Fig.3, a linear relationship exists between frame size and processor power consumption [7]. A test of video sequence with different bitrates and resolution in [7] shows that the larger the bitrates, the relation curve of frame size and energy consumption move to the upper, which

means higher power consumption; Furthermore, the value of  $\alpha$  increases with the video resolution, i.e.  $\alpha$  is positively correlated to the resolution. Therefore, the analysis of the relationship between one frame decoded by a mobile device and power consumption can be deduced by Eq.(3):

$$E_{slice} = \alpha + \beta * (FrameSize) \quad (3)$$

The method of least squares is adopted in the paper to calculate  $\alpha$  and  $\beta$  follows:

$$\alpha = \bar{y} - \beta \bar{x} \quad (4)$$

$$\beta = \frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{\sum (x_i - \bar{x})^2} = \frac{s_{xy}}{s_{xx}} \quad (5)$$

With the mathematical relation of frame size and energy consumption, we can predict the frame size based on the processor power consumption while the frame size can be adjusted by resolution and bitrates.

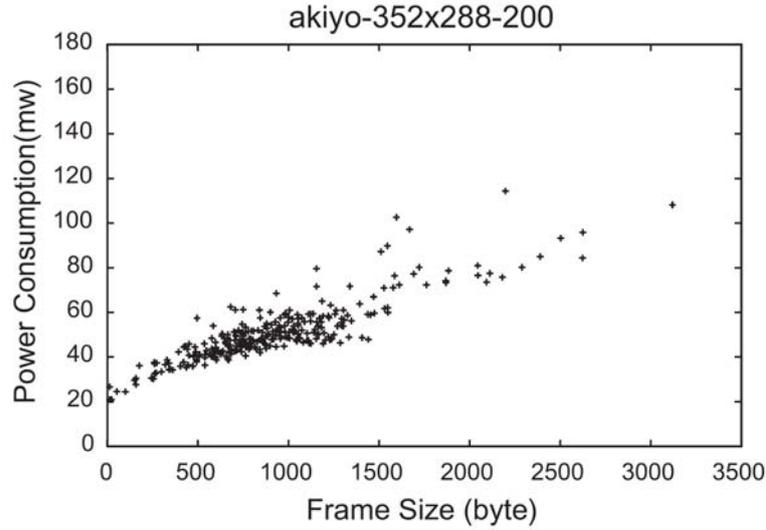


Fig. 3 Analysis between frame size and processor power consumption in [7]

### 3.2. Algorithm design

The inference algorithm was based on the following two conditions:

(1) According to the analysis of the preceding context, the LCD brightness of the mobile device is assumed to not be able to change at will during multimedia service.

(2) The configuration information of the terminal device is transmitted to the resource module of cloud in the form of the XML-scheme. XML-Schemas are a type of Metadata, which uses semantics to support and describe the data format of archive files [10]. We only select the appropriate resolution based on the data from the resource management module of cloud since the resolution that the particular terminal equipment supported is fixed.

It is assumed that  $V = \{V_1, V_2, V_3 \dots V_j\}$  is the transcoded bit rate over cloud,  $R$  is the transcoded resolution over cloud, which unit is in Kbits/s. supposing that a video segment consists of  $N$  GOPs,  $M$  of GOP can be given in encoding, and the total frame number of video segment is  $N * M$ . According to Eq.(3) each  $(V[j], R)$  has a corresponding  $\alpha_j$  and  $\beta_j$ , and the energy consumption by each frame in each  $(V[j], R)$  can be calculated as:

$$E_{X_j} = \alpha_j + \beta_j * S_{X_j} \quad (6)$$

where,  $E_{X_j}$  denotes the energy consumption of each frame, and  $S_{X_j}$  is the frame size.

Video slice is used as the unit in the paper, so we can get the expression of the total power consumption of a video segment in each  $(V[j], R)$ :

$$E_{S_j} = \alpha_j * (N * M) + \beta_j * \left( \sum_{x=1}^{N*M} S_{x(j)} \right) \quad (7)$$

Therefore, the power consumed by a processor per second can be deduced, i.e.

$$\begin{aligned}
P_x &= fps * E_{S_j} \\
&= fps * \left( \alpha_j * (N * M) + \beta_j * \left( \sum_{x=1}^{N*M} S_{x(j)} \right) \right)
\end{aligned} \tag{8}$$

where,  $fps$  is the frame rate.

From the above analysis, in case of the resolution limited, higher bit rate would lead to greater frame size, higher power consumption and better video quality, consequently, in order to get video images with high quality while saving energy, what we have to do is maximizing  $P_x$  when the power is enough for playback.

According to device profile data, the highest power consumed by a processor per second is  $P_{max}$ . If the power consumed for decoding exceeds  $P_{max}$ , it means that the processor must deal with the task at full speed. However, full speed operation of a processor may cause a failure to decode a frame and insufficient frame rate, therefore, the following condition Eq.(9) should be satisfied,

$$P_x < P_{max} - P_{current} \tag{9}$$

where,  $P_{current}$  is the energy consumption of the current CPU, it can be known through the resource management module.

The power consumption of multimedia streaming is mainly from network, processor, and LTE display[8,11], but it is more difficult to estimate the energy consumption of network. Assuming that the power consumption of the network is the time to play a multimedia, namely

$$(N * M * L) / fps \tag{10}$$

where,  $L$  denotes the number of video slices. We can get the expression of the total energy consumption

$$\begin{aligned}
E_{total} &= P_{network} * \frac{N * M * L}{fps} + P_{display} * \\
&\quad \frac{N * M * L}{fps} + E_{S_1} + E_{S_2} + \dots E_{S_m}
\end{aligned} \tag{11}$$

To ensure that there is enough electricity for whole display, the following equation Eq.(12) must be satisfied

$$E_{total} < E_{remaining} \tag{12}$$

Thus the optimization problem can be modeled as

$$\begin{aligned}
&max P_x \\
&s.t. P_x < P_{max} - P_{current} \\
&\quad E_{total} < E_{remaining}
\end{aligned} \tag{13}$$

Based on the terminal device configuration files, the target resolution is  $R_i$ , the target bitrates is arranged in descending order, and following matrix can be obtained  $V = \{V_1, V_2, V_3 \dots V_n\}$ ,

Step 1,  $R = R_i, V = V[j] | j = 1$ ;

Step 2, calculating  $E_{S_j}$  of current  $(V[j], R)$  through Eq.(7);

Step 3, getting  $P_x$  by Eq.(8);

Step 4, determining whether Eq.(9) is valid, if it is invalid, the value of  $j$  plus 1 and return to step 1, else going to Step 5;

Step 5, estimating the bitrates of the video slice which hasn't been downloaded with a current video slice. Given the Eq.(11), the total energy consumption is  $E_{total}$ ;

Step 6, determining whether Eq.(12) is valid, if satisfied, select the current bitrates for the next video slice. Else the value of  $j$  plus 1 and return to Step 1;

Step 7, the value of  $j$  continuing to accumulate until Eq.(12) becomes valid. If it remains invalid, the user should be notified that the electricity remained cannot play the multimedia and charging is recommended.

The algorithm flow chart is presented as Fig.4.

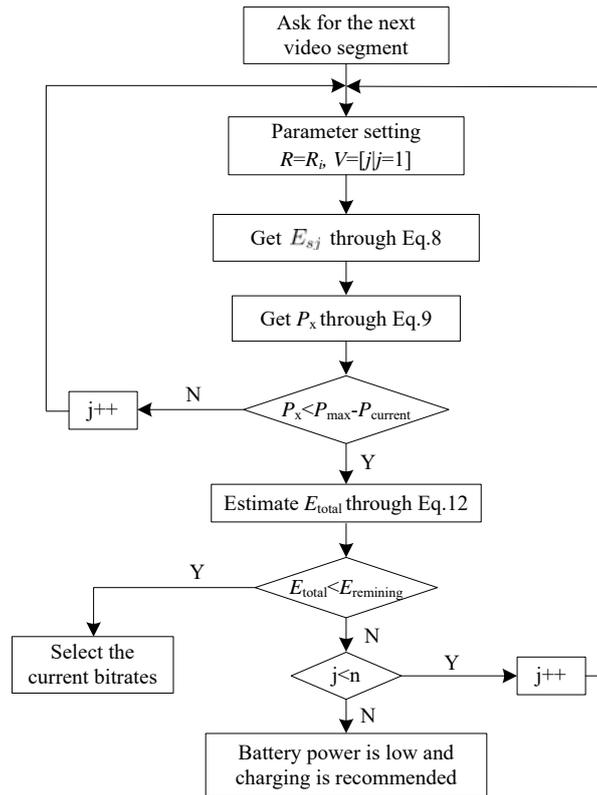


Fig. 4 The flow chart of the proposed algorithm

#### 4. Simulation results

The LTE video transmission platform based on NS3 is employed. Moreover, by adding the evalivid model [12] to the nodes, Evalivid the event of video streaming transferring through the network can be simulated, and by adding the energy model to the nodes, the event of power consuming of the client device can be modeled. A small amount of computing power is needed since there is only one client node in the simulations and the test video sequence is relatively short, therefore, to model the adaptive video streaming transmission based on cloud with the NS-3 simulator will not cause a big difference in performance.

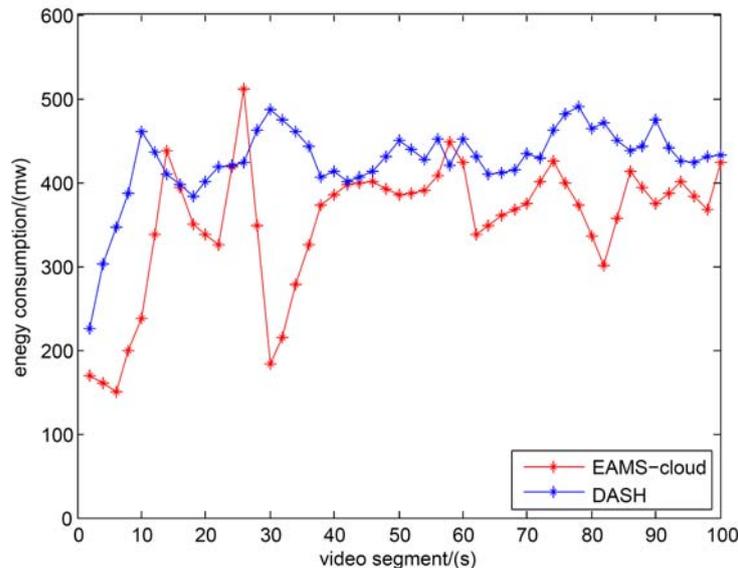


Fig. 5 The energy consumption in different schemes

The multimedia resource used for the test is a foreman video in CIF format, which lasts 19s, it is divided into 10 video slices, and each slice of length 2s, the cloud transcoded resolution is 352\*288, the cloud transcoded bitrates include 64kbps, 128kbps, 192kbps, 256kbps, 384kbps, 512kbps,

768kbps, 1024kbps, 1536kbps, 2048kbps. The value of the GOP is 30. The  $P_{max}$  of client 1 is set as 450  $mw$  while the  $P_{max}$  of client 2 is set as 400  $mw$ .

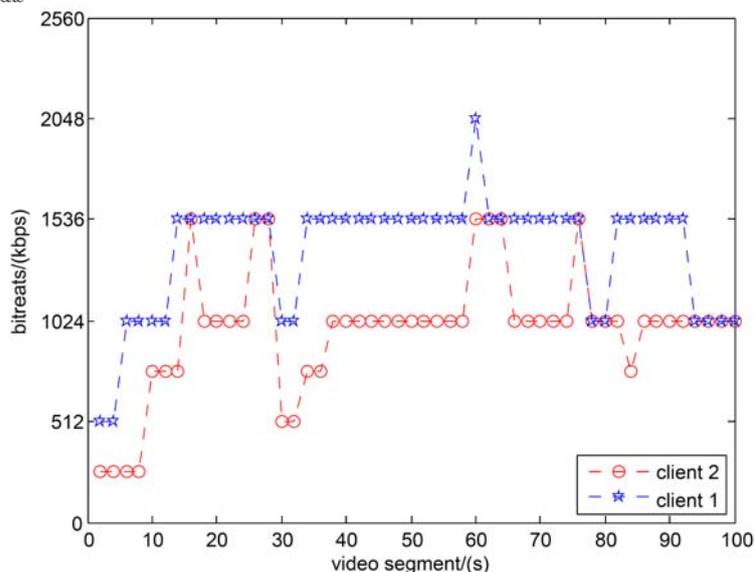


Fig. 6 The bitrates of different clients

The processes of the video streaming transmitting from the server to client have been simulated with the traditional DASH scheme and the proposed EAMS-cloud algorithm respectively, and lasted for 100s. The sufficient bandwidth is provided, and the impact of bandwidth is not considered. As shown in Fig.5, the longitudinal axis is power consumption per second, and the horizontal axis is the video segment sequence. The energy consumption curve of client 1 with DASH scheme remains relatively stable while the energy cost curve of client 2 with the EAMS-cloud algorithm fluctuates greatly which suggests that the proposed algorithm can dynamically adjust the transmission of the adaptive video stream based on energy consumption of the terminal device. The bitrates of each video slice are shown in Fig.6, from which we can see the better performance mobile select large bitrates than poor performance mobile. It is illustrated that the EAMS-cloud algorithm can match different target bitrates for client according to device configurations. In addition, the total energy consumption of client 1 with DASH algorithm is the largest, followed by client 2 with EAMS-cloud, and the smallest is the total energy consumption of client2 with EAMS-cloud, as shown is Fig.7.

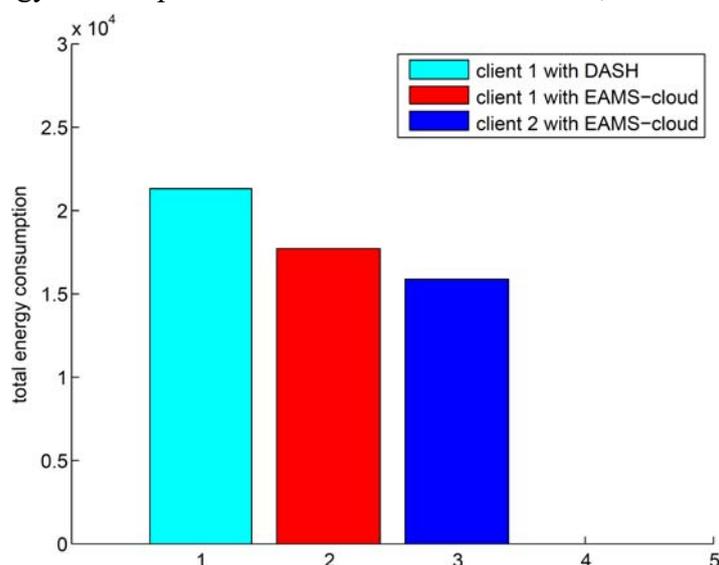


Fig. 7 The total energy consumption of different clients

Based on the above result, the tradition DASH scheme cannot dynamically adjust the video streaming according to the energy consumption of mobile devices, and thus maximize the total energy consumption for the biggest possible bitrates will be selected when sufficient bandwidth is provided. However, with the proposed EAMS-cloud algorithm, the bitrates can be selected

dynamically on the basis of energy consumption and configuration files of terminal devices, and the total energy consumption is smaller than DASH scheme. In summary, the proposed algorithm has a better performance in the whole system.

## 5. Conclusion

In this paper, an energy-efficiency problem occurs when handle devices obtain multimedia content has been investigated. We proposed a cloud based energy-efficient service architecture for adaptive multimedia streaming and formulated an optimization problem on energy consumption. An effective and practical scheduling algorithm is adopted to solve this problem. The results have demonstrated that our method can reduce the total energy consumption significantly. It can be applied to mobile network bandwidth mechanism calculation combined with energy-efficient concept to attain the goal of high efficiency and low power consumption in the future.

## References

- [1] Mahmood A, Jinnah T, Asfia Y, et al. A hybrid adaptive compression scheme for multimedia streaming over wireless networks [C]. 2008 IEEE 4th International Conference on Emerging Technologies. October 2008, p. 187-192.
- [2] Wu S, He C. QoS-aware dynamic adaptation for cooperative media streaming in mobile environments [J]. IEEE Transactions on Parallel and distributed systems. Vol.22(2011) No.3, p. 439-450.
- [3] Liu Y, Li C, Yang Z. Tradeoff between energy and user experience for multimedia cloud computing [J]. Computers & Electrical Engineering. Vol.47(2015), p. 161-172.
- [4] C. F. Lai, H. C. Chao, Y. X. Lai, J. Wan. Cloud-assisted realtime transrating for http live streaming. IEEE Wireless Communications. Vol. 20(2013) No. 3, pp. 62–70.
- [5] M. Hadji and D. Zeghlache. Minimum cost maximum flow algorithm for dynamic resource allocation in clouds. 2012 IEEE 5th International Conference on Cloud Computing (CLOUD). June 2012, p. 876–882.
- [6] S. T. Maguluri, R. Srikant, L. Ying. Heavy traffic optimal resource allocation algorithms for cloud computing clusters. Performance Evaluation, vol. 81(2014), pp. 20–39.
- [7] S.-Y. Chang, C.-F. Lai, and Y.-M. Huang, “Dynamic adjustable multimedias streaming service architecture over cloud computing,” Computer Communications, vol.35(2012) No. 15, pp. 1798–1808.
- [8] M. Kjargaard, Minimizing the power consumption of location-based services on mobile phones. IEEE Pervasive Computing, vol. 11(2012) No. 1, pp. 67–73.
- [9] D. Sostaric, D. Vinko, and S. Rimac-Drlje, “Power consumption of video decoding on mobile devices,” in ELMAR, 2010 PROCEEDINGS, Sept 2010, pp. 81–84.
- [10] A. Mathes, Folksonomies-cooperative classification and communication through shared metadata. Community, January 2004.
- [11] M. Kim and B. Noble. Mobile network estimation. Proceedings of the 7th Annual International Conference on Mobile Computing and Networking. New York, 2001, pp. 298–309.
- [12] W. A. Klaue J, Rathke B. Evalvid-a framework for video transmission and quality evaluation. Lecture notes in computer science, computer performance evaluation: modeling techniques and tools. Vol.2794(2003), pp.255-272.