Research Article

Joint Optimized CPU and Networking Control Scheme for Improved Energy Efficiency in Video Streaming on Mobile Devices

Sung-Woong Jo and Jong-Moon Chung

School of Electrical and Electronic Engineering, Yonsei University, 50 Yonsei-ro, Seodaemun-gu, Seoul 03722, Republic of Korea

Correspondence should be addressed to Jong-Moon Chung; jmc@yonsei.ac.kr

Received 30 December 2016; Accepted 26 February 2017; Published 16 March 2017

Academic Editor: Karl Andersson

Copyright © 2017 Sung-Woong Jo and Jong-Moon Chung. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Video streaming service is one of the most popular applications for mobile users. However, mobile video streaming services consume a lot of energy, resulting in a reduced battery life. This is a critical problem that results in a degraded user's quality of experience (QoE). Therefore, in this paper, a joint optimization scheme that controls both the central processing unit (CPU) and wireless networking of the video streaming process for improved energy efficiency on mobile devices is proposed. For this purpose, the energy consumption of the network interface and CPU is analyzed, and based on the energy consumption profile a joint optimization problem is formulated to maximize the energy efficiency of the mobile device. The proposed algorithm adaptively adjusts the number of chunks to be downloaded and decoded in each packet. Simulation results show that the proposed algorithm can effectively improve the energy efficiency when compared with the existing algorithms.

1. Introduction

Due to rapid advances in wireless networks and data processing on mobile devices, a continuously growing number of users are accessing high quality video streaming services from mobile devices [1, 2]. According to [2], video streaming services over wireless networks will generate three-quarters of the mobile data traffic and increase ten times between 2014 and 2019. However, such video streaming services consume a significant amount of energy, resulting in a reduced battery lifetime. This is a critical problem especially for mobile devices, due to their limited energy supply. Therefore, minimizing the energy consumption of mobile video streaming services has become an essential research field.

In mobile video streaming services, the network interface and central processing unit (CPU) are two major energy consuming components, which support data communication and video data decoding [3]. Therefore, there have been many studies which focus on minimizing the energy consumption of the network interface and/or CPU. For the network interface, buffer management schemes were proposed to

reduce the energy consumption of mobile devices [4, 5]. By dynamically controlling downloading activities, these schemes improved the energy efficiency of video streaming services on mobile devices. In addition, in order to deal with both cellular and Wi-Fi networks, energy efficient methods that control both LTE and Wi-Fi simultaneously in supporting video streaming services over heterogeneous wireless networks were proposed [6, 7]. Meanwhile, for the CPU, dynamic voltage and frequency scaling (DVFS), which can save energy at the cost of a reduced processing speed, is widely used on mobile devices. Based on using the DVFS technique, to improve energy saving of video applications, power management methods were proposed in [8, 9]. These algorithms adaptively control the processing speed through predicting the decoding time of the video. In addition, several research works have focused on scheduling and DVFS in multicore processors [10–12]. Furthermore, for energy optimization among different components including the network interface and CPU, a cross-layer framework was proposed in [13]. The scheme of [13] can reduce the energy consumption of the network interface and CPU. However,

FIGURE 1: System model of video streaming services on mobile devices.

none of these papers deal with joint optimization of the network interface and CPU. Additional energy saving can be achievable from approaches that jointly consider the trade-off between the energy consumption and the processing time both on the network interface and on the CPU.

In this paper, a joint optimization scheme for improved energy efficiency supporting mobile video streaming services is proposed. First, the energy consumption of the network interface and CPU is analyzed while receiving and decoding the video chunks, respectively. Then, a joint optimization problem statement maximizing the total energy efficiency of the network interface and CPU is formulated. The proposed scheme solves the joint optimization problem and uses the solution to adaptively control the number of chunks to be downloaded and decoded in each packet by considering the video characteristics, wireless network status, and buffer status. The energy consumption of the proposed algorithm is compared with four existing algorithms based on various environments. Simulation results show that the proposed Joint optimized CPU and Video streaming Network (JCVN) control algorithm can improve the energy efficiency of video streaming services on mobile devices.

2. System Model

2.1. Mobile Video Streaming Service Model. In Figure 1, the system model is illustrated [5], where a mobile device directly obtains video streams from the streaming server through the wireless network interface. The video streaming service is assumed to be a video sequence that is compressed at a constant bit rate r bits/s. The video sequence is divided into equal-size chunks of rT_0 bits sent periodically in packets with payload durations of T_0 s. The system is assumed to operate in discrete time units based on packets $i = \{1, 2, \dots, T\}$, where T is the maximum number of packets needed to transmit the entire video sequence. By considering a time-varying wireless channel, the download data rate of the mobile device at packet i is denoted as $R_d(i)$. It is assumed that the download data rate remains constant within a packet duration but can vary across different packets.

A mobile user may experience interruptions during video playback when the download rate of the video stream is lower than the encoding rate of the video stream due to the time-varying wireless channel and network conditions. Therefore, mobile devices typically use a receiving buffer and playback buffer to satisfy the QoS requirement (i.e., no interruption) [14]. In a mobile device, the downloaded data chunks enter the receiving buffer and the chunks are decoded in the video decoder. The decoded chunks enter the playback buffer and

they are displayed corresponding to the playback rate, which is equivalent to the video encoding rate.

Let $B_r(i)$ and $B_p(i)$ denote the number of chunks in the receiving buffer and playback buffer at the beginning of packet i , respectively. Because mobile devices have limited resources, $B_r(i)$ and $B_p(i)$ are bounded by their maximum thresholds as $0 \leq B_r(i) \leq B_{r,Th}$ and $0 \leq B_p(i) \leq B_{p,Th}$, respectively. In addition, the processing delay in the video decoder should be taken into consideration when the chunks are decoded. The processing delay is dependent on the CPU frequency at which the video decoder decodes the chunks. The processing delay is inversely proportional to the CPU frequency [8, 9, 13]. Let D_p denote the processing delay required to decode a chunk at the highest CPU frequency $f_{c,max}$. Then, the processing delay required to decode a chunk at CPU frequency $f_c(i)$ in packet i is $D_p f_{c,max} / f_c(i)$.

2.2. Energy Consumption Model. The energy consumption of the wireless network interface card (WNIC) is modeled as $E = P_n t + P_d l$, where P_n is the average power consumption in active state, t is the time duration in active state, P_d is the average power consumption per downloaded data, and l is the amount of downloaded data [5]. By using the WNIC's energy consumption model, in order to evaluate the energy efficiency, the energy consumption gain of the WNIC for packet i is defined as

$$\begin{aligned}
 E_n(i) &= \left(P_n n_d(i) \frac{rT_0}{R_{\min}} + P_d n_d(i) rT_0 \right) \\
 &\quad - \left(P_n n_d(i) \frac{rT_0}{R_d(i)} + P_d n_d(i) rT_0 \right) \\
 &= P_n n_d(i) \left(\frac{rT_0}{R_{\min}} - \frac{rT_0}{R_d(i)} \right),
 \end{aligned} \tag{1}$$

where $n_d(i)$ is the number of chunks downloaded in packet i . The energy consumption gain of the WNIC $E_n(i)$ is the amount of energy saving that can be achieved when downloading $n_d(i)$ chunks at the download rate of $R_d(i)$ compared to the case when downloading equal-size chunks at the minimum download rate of $R_{\min}$. Note that, in (1), the energy consumption gain of the WNIC $E_n(i)$ cannot be a negative value due to $R_{\min} \leq R_d(i)$.

Furthermore, the power consumption of the CPU is a function of the CPU frequency. According to the model in [15], the average power consumption of the CPU at CPU frequency $f_c(i)$ in packet i can be defined as $P_p(i) = \alpha f_c(i)^3 + \beta$, where α and β are constants that depend on the mobile

device. Then, similar to the WNIC's energy consumption gain of (1), the energy consumption gain of the CPU based on packet i is defined as

$$E_p(i) = n_p(i) D_p \left((\alpha f_{c,\max}^3 + \beta) - (\alpha f_c(i)^3 + \beta) \frac{f_{c,\max}}{f_c(i)} \right), \quad (2)$$

where $n_p(i)$ is the number of chunks decoded in packet i . The energy consumption gain of the CPU $E_p(i)$ is the amount of energy saving that can be achieved when decoding $n_p(i)$ chunks at the CPU frequency of $f_c(i)$ compared to the case when decoding equal-size chunks at the maximum CPU frequency of $f_{c,\max}$.

3. JCVN Controller Design

The objective of this paper is to maximize the total energy consumption gain of the WNIC and CPU while satisfying the QoS requirements of the video streaming service. Based on the mathematical models of the previous section, the optimization problem can be stated as

$$\max_{n_d(i), n_p(i)} E_n(i) + E_p(i), \quad \forall i \quad (3)$$

$$\text{s.t.} \quad 0 \leq n_d(i) \leq \frac{R_d(i)}{r}, \quad (4)$$

$$0 \leq n_p(i) \leq \frac{T_0}{D_p}, \quad (5)$$

$$0 \leq B_r(i+1) \leq B_{r,\text{Th}}, \quad (6)$$

$$0 \leq B_p(i+1) \leq B_{p,\text{Th}}, \quad (7)$$

where the number of chunks in the receiving buffer and playback buffer can be updated as follows:

$$\begin{aligned} B_r(i+1) &= B_r(i) + n_d(i) - n_p(i), \\ B_p(i+1) &= B_p(i) + n_p(i) - 1. \end{aligned} \quad (8)$$

In (8), $n_d(i)$ is the number of chunks arriving at the receiving buffer through downloading of packet i , $n_p(i)$ is the number of chunks migrating from the receiving buffer to the playback buffer through the decoding process of packet i , and $rT_0/rT_0 = 1$ is the number of chunks to be displayed to satisfy the QoS requirements based on the encoding rate, in which the numerator term rT_0 is the amount of video data to be displayed at the encoding rate r in a packet duration T_0 and the denominator term rT_0 is the size of a video chunk.

Constraint (4) ensures that the number of chunks to be downloaded is not a negative value and cannot be greater than the maximum number of chunks which the mobile device can download at its download rate in a packet. Constraint (5) ensures that the number of chunks to be decoded is not a negative value and cannot be greater than the maximum number of chunks which can be decoded at the maximum CPU frequency in a packet. Constraints (6) and (7) ensure

that the numbers of chunks in the receiving buffer and playback buffer are not negative values and should also be less than their maximum thresholds. That is, JCVN controls the number of chunks to be downloaded and decoded while avoiding buffer overflow. In addition, constraint (7) ensures that the video chunks are displayed at their encoding rate and the viewer will not experience any interruption in the playback. This is because $B_p(i+1) \geq 0$ (i.e., $B_p(i) + n_p(i) \geq 1$) guarantees that the playback buffer will maintain more chunks than the number of chunks to be displayed at the encoding rate.

In addition, to maximize the energy consumption gain of the CPU, when the CPU decodes $n_p(i)$ video chunks, CPU frequency $f_c(i)$ in packet i should be selected to the minimum CPU frequency at which the CPU can decode $n_p(i)$ chunks in packet i . Therefore, the number of chunks to be decoded can be presented as a function of the CPU frequency in packet i as follows:

$$n_p(i) = \begin{cases} \left(T_0 - \frac{rT_0}{R_d(i)} \right) \frac{f_c(i)}{D_p f_{c,\max}} & B_r(i) = 0 \\ \frac{T_0 f_c(i)}{D_p f_{c,\max}} & B_r(i) > 0, \end{cases} \quad (9)$$

which indicates that if there is no chunk in the receiving buffer (i.e., $B_r(i) = 0$), the decoding process starts after downloading a chunk. Otherwise, if there are chunks in the receiving buffer (i.e., $B_r(i) > 0$), the chunks are decoded during the packet length T_0 . In addition, (9) satisfies constraint (5).

By solving the optimization problem (3)–(7), an optimal solution set of $n_d(i)$ and $n_p(i)$ which maximizes the total energy consumption gain of the WNIC and CPU based on each packet i can be obtained. The optimization problem is transformed into two variables x and y as follows:

$$\begin{aligned} \max_{x,y} \quad & f(x, y) = ax + (-by^3 + cy - d) \\ \text{s.t.} \quad & 0 \leq x \leq e, \\ & 0 \leq y \leq g, \\ & h \leq x - y \leq k, \\ & l \leq y \leq m, \end{aligned} \quad (10)$$

where $x = n_d(i)$, $y = n_p(i)$, and $a, b, c, d, e, g, h, k, l$, and m are (for the i th packet) constant values as

$$\begin{aligned} a &= P_n T_0 r \left(\frac{1}{R_{\min}} - \frac{1}{R_d(i)} \right), \\ b &= \frac{\alpha T_0 (D_p f_{c,\max}/T_0)^3}{(1 - r/R_d(i))^2} \quad (B_r(i) = 0), \\ b &= \alpha T_0 \left(\frac{D_p f_{c,\max}}{T_0} \right)^3 \quad (B_r(i) > 0), \end{aligned}$$

```

(1) GET input parameters of the energy consumption model
(2) COMPUTE optimal candidate solution set  $(n_{d,1}(i)^*, n_{p,1}(i)^*)$  for Case 1
(3) COMPUTE optimal candidate solution set  $(n_{d,2}(i)^*, n_{p,2}(i)^*)$  for Case 2
(4) COMPARE values of the objective function at two candidate solution sets
(5)   if  $E_n(n_{d,1}(i)^*) + E_p(n_{p,1}(i)^*) \geq E_n(n_{d,2}(i)^*) + E_p(n_{p,2}(i)^*)$ 
(6)     SET optimal solution set to  $(n_{d,1}(i)^*, n_{p,1}(i)^*)$ 
(7)   else
(8)     SET optimal solution set to  $(n_{d,2}(i)^*, n_{p,2}(i)^*)$ 
(9)   end if

```

ALGORITHM 1: JCVN algorithm.

$$\begin{aligned}
c &= (\alpha f_{c,\max}^3 + \beta) D_p, \\
d &= T_0 \beta \left(1 - \frac{r}{R_d(i)} \right) \quad (B_r(i) = 0), \\
d &= T_0 \beta \quad (B_r(i) > 0), \\
e &= \frac{R_d(i)}{r}, \\
g &= \frac{T_0}{D_p}, \\
h &= -B_r(i), \\
k &= B_{r,\text{Th}} - B_r(i), \\
l &= -B_p(i) + 1, \\
m &= B_{p,\text{Th}} - B_p(i) + 1.
\end{aligned} \tag{11}$$

Let set S denote the range of (x, y) satisfying the above constraints. Then, through the *Closed Interval Method* of [16], the maximum value of $f(x, y)$ can be obtained at the critical points in S or at the extreme values on the boundary of S . In addition, because $f(x, y)$ is a linear function with respect to x , $f(x, y)$ is maximized at the maximum value among the range of x with an arbitrarily fixed value of y . Therefore, an optimal solution set (x, y) exists on line of $x = e$ (Case 1) or $y = x - k$ (Case 2).

Case 1. From $x = e$, the objective function is represented as a function of y , which is $f(e, y) = -by^3 + cy + ae - d$ with constraint of $\max[0, e - k, l] \leq y \leq \min[g, e - h, m]$. By letting $\partial f(e, y)/\partial y = 0$, the only critical point of the equation is computed as $\sqrt{c/3b}$. In addition, because the second derivative test gives $\partial^2 f(e, y)/\partial y^2 = -6by < 0$ at $\sqrt{c/3b}$, the objective function is concave. Therefore, the objective function has a maximum value at the critical point of $\sqrt{c/3b}$, or the minimum or maximum endpoint of y [16]. If the critical point exists on the larger value than the range of y (i.e., $\min[g, e - h, m] < \sqrt{c/3b}$), the objective function has a maximum value at the maximum end point of $y = \min[g, e - h, m]$. Otherwise, if the critical point exists on the smaller

value than the range of y (i.e., $\max[0, e - k, l] > \sqrt{c/3b}$), the objective function has a maximum value at the minimum end point of $y = \max[0, e - k, l]$. Otherwise, if the critical point exists in the range of y (i.e., $\max[0, e - k, l] \leq \sqrt{c/3b} \leq \min[g, e - h, m]$), the objective function has a maximum value at the critical point of $\sqrt{c/3b}$.

Case 2. From $x = y + k$, the objective function is also represented as a function of y , which is $f(y + k, y) = -by^3 + (a + c)y + ak - d$ with a constraint of $\max[0, l] \leq y \leq \min[e - k, g, m]$. In a similar way, to solve Case 2, by letting $\partial f(y + k, y)/\partial y = 0$, the only critical point of the equation is computed as $\sqrt{(a + c)/3b}$. Then, the objective function has a maximum value at the maximum end point of $y = \min[e - k, g, m]$, or the minimum end point of $y = \max[0, l]$, or the critical point of $\sqrt{(a + c)/3b}$.

Therefore, an optimal solution set of $n_d(i)^*$ and $n_p(i)^*$ which maximizes the objective function can be obtained by comparing the values of $f(x, y)$ at the two candidate solutions from Cases 1 and 2. Note that the optimal solution set of $n_d(i)^*$ and $n_p(i)^*$ is obtained by assuming that the objective function is a continuous function of $n_d(i)$ and $n_p(i)$. Therefore, because the numbers of chunks to be downloaded and decoded are positive integer values, in the performance evaluation, the values of the optimal solution set were selected to be positive integer values that satisfy the constraints of the optimization problem and have a minimum difference with the derived optimal solution set of $n_d(i)^*$ and $n_p(i)^*$, respectively.

The pseudocode of the JCVN optimization algorithm is presented in Algorithm 1. First, the input parameters of the energy consumption model are obtained based on the optimization problem (step (1)). JCVN computes the optimal candidate solution set of the number of chunks to be downloaded and decoded for Cases 1 and 2 (steps (2) and (3)). Then, an optimal solution set is determined by comparing the values of the objective function at the two candidate solution sets from Cases 1 and 2. If the value of the objective function at Case 1's optimal candidate solution is equal to or larger than the value of the objective function at Case 2's optimal candidate solution (i.e., $E_n(n_{d,1}(i)^*) + E_p(n_{p,1}(i)^*) \geq E_n(n_{d,2}(i)^*) + E_p(n_{p,2}(i)^*)$), the optimal solution is selected as the candidate solution set for Case 1 (steps (5) and (6)). Otherwise, if the value of the objective function at Case

TABLE 1: Simulation parameter settings.

Parameters	Values
Download rate of a mobile device	A Rayleigh distributed wireless channel with the means of 0.5, 1, 1.5, and 2 Mbps
Video encoding rate	320 kbps
Entire video length	200 s
Packet length	2 s
Maximum receiving and playback buffer size	20 chunks
Average power consumption in active state (P_n)	0.897 W
Average power consumption per downloaded data (P_d)	0.134 μ J
CPU frequency	0.2, 0.4, 0.6, 0.8, 1.0, 1.2, 1.4, and 1.6 GHz
Coefficient for CPU's power consumption (α)	0.4 s
Coefficient for CPU's power consumption (β)	0.3 s
Processing density	1000 cycles/bit

l's optimal candidate solution is less than the value of the objective function at Case 2's optimal candidate solution (i.e., $E_n(n_{d,1}(i)^*) + E_p(n_{p,1}(i)^*) < E_n(n_{d,2}(i)^*) + E_p(n_{p,2}(i)^*)$), the optimal solution is selected as the candidate solution set for Case 2 (steps (7) and (8)).

4. Performance Evaluation

4.1. Simulation Settings. In order to make an accurate performance comparison, the experiment parameters of [5, 15] were applied in the simulation experiments as described in the following. The video encoding rate was set to 320 kbps and the video length was 200 s. The entire video file was divided into packets of 2 s, where the mobile device's receiving buffer and playback buffer can contain up to 20 chunks of video data. In the simulation, a Rayleigh distributed wireless channel was applied to represent the multipath fading characteristics, which results in variations of the download rate of video data with the means of 0.5, 1, 1.5, and 2 Mbps [5]. For example, the variation of the download rates with a mean of 1.5 Mbps is presented in Figure 2(a). The WNIC's power consumption parameter values applied in the simulation are $P_n = 0.897$ W and $P_d = 0.134 \mu$ J/bit. In addition, the average number of CPU cycles required per bit (when the application is processed by the CPU) is defined as the processing density, which is dependent on the application type. For the video streaming simulations, a mixture of H.264 240p, 480p, and 720p videos at 30 fps based on a closed group of pictures (GoP) without coordinated universal time (UTC) was used. In addition, a 1,000-cycle/bit processing density was applied based on [15], and the CPU's power consumption model uses the parameter values of $\alpha = 0.4$ and $\beta = 0.3$. The detailed simulation parameter settings are summarized in Table 1.

In the performance analysis, two representative WNIC downloading schemes and two of the most popular CPU processing power control schemes that are used in smart devices are compared with the proposed JCVN scheme. The two representative WNIC downloading schemes are Aggressive Buffer Management (ABM) and Periodical Buffer Management (PBM) [5]. In ABM, the video application actively downloads video chunks until the receiving buffer

of the mobile device is full, and is popularly used in mobile devices with unlimited (or very large) power supplies and has no constraints applied to the video streaming services. In PBM, the video application makes requests for a fixed number of video chunks to be periodically sent from the server and stops downloading if the requested video chunks have been all received within the time limit or if the mobile device's receiving buffer is full, which is a very popular WNIC scheme applied in mobile devices. In the simulation experiments, the PBM default period T_0 is set based on the viewer requests of two, three, four, and five video chunks per period for the mean download rates of 0.5, 1, 1.5, and 2 Mbps, respectively.

The two most popular CPU processing power control schemes are the maximum CPU frequency (MAX) scheme and the dynamic voltage and frequency scaling (DVFS) scheme. In MAX, the CPU processes video chunks at its highest frequency if video chunks to be decoded are in its receiving buffer. This strategy provides a high performance by processing data at the highest CPU frequency, while it consumes a lot of energy. In DVFS, the CPU frequency is adjusted according to the amount of video chunks to be decoded in the receiving buffer. When DVFS is used, the CPU frequency is set to maximum when the number of video chunks to be decoded is greater than a threshold, and the CPU frequency proportionally decreases (in a staircase fashion) as the decoding load diminishes. The threshold of DVFS was set to four video chunks in the experiments. Note that when the receiving buffer and playback buffer become full, the WNIC and CPU of all existing schemes and JCVN are switched to the power-saving mode in which they do not perform downloading and decoding.

In the simulations, the two WNIC schemes were each teamed up with the two CPU control schemes in the following form of PBM with MAX, ABM with MAX, PBM with DVFS, and ABM with DVFS, which are, respectively, denoted as PBM + MAX, ABM + MAX, PBM + DVFS, and ABM + DVFS in Figures 2 and 3.

4.2. Performance Analysis. Figure 2 illustrates the difference of JCVN (compared to other schemes) in how it jointly controls the number of chunks to be downloaded and

FIGURE 2: (a) Number of chunks downloaded per packet. (b) Number of chunks decoded per packet.

decoded to maximize the combined energy efficiency of the WNIC and CPU. Figures 2(a) and 2(b) show the number of chunks to be downloaded and decoded for each packet at the mean download rate of 1.5 Mbps. In Figure 2(a), ABM + DVFS and JCVN download chunks in accordance with the download rate, while PBM + DVFS has little variation in the number of downloaded chunks regardless of the download rate. Therefore, ABM + DVFS and JCVN can provide a gain in reducing the WNIC's energy consumption compared to PBM + DVFS. In Figure 2(b), it is observed that PBM + DVFS and ABM + DVFS aggressively decode chunks at the beginning of a download session (i.e., packets 1~5) quickly filling up their buffers and also use full CPU processing capability in the decoding process. Compared to this, JCVN attempts to save energy by downloading only at a sufficient rate in the initial stages (i.e., packets 1~5) and also selects a sufficient CPU operational frequency that will allow an above-QoS level of video playback.

In Figure 3, the proposed JCVN scheme is compared with the four schemes of PBM + MAX, ABM + MAX, PBM + DVFS, and ABM + DVFS for various download rates, where Figures 3(a), 3(b), and 3(c) present the mobile device's energy consumption of the WNIC, CPU, and the combined value of the WNIC and CPU, respectively. Figure 3(a) shows that, for all schemes, the energy consumption of the WNIC decreases to around 61% as the average download rate increases. This is because the energy consumption of the WNIC is inversely linear to the download rate as denoted in (1). In Figure 3(b), it is noted that PBM + DVFS and ABM + DVFS consume less energy at the CPU than PBM + MAX and ABM + MAX, which is due to the energy saving capability of DVFS, where the energy saving gain increases as the average download rate decreases. This is because the mobile device can download more video chunks at the receiving buffer in a packet when the download rate is high. Therefore, DVFS will use higher CPU frequencies in order to process the video chunks of the

receiving buffer, thereby consuming more energy. Figure 3(c) points out that the sum of the energy consumption of the WNIC and CPU for the proposed JCVN scheme is the lowest among all the algorithms under different download rates. For example, when compared with PBM + MAX and PBM + DVFS, the JCVN scheme can reduce the energy consumption by 10.59~20.05% and 4.08~11.61%, respectively. Because the JCVN scheme adaptively adjusts the number of video chunks to be downloaded and decoded according to the wireless network status as well as the receiving and playback buffer status, it can effectively reduce the total energy consumption of the WNIC and CPU.

5. Conclusion

In order to deal with the energy consumption issue on mobile devices supporting video streaming services, in this paper, a joint optimization algorithm that can reduce the combined energy consumption of the WNIC and CPU is proposed. The energy consumption of the WNIC and CPU is analyzed and the proposed algorithm adaptively adjusts the number of video chunks to be downloaded and decoded in each packet by solving an optimization problem according to the wireless network status as well as the receiving and playback buffer status. The energy consumption of the proposed algorithm is compared with four existing algorithms based on various average download rates. Results show that the proposed JCVN algorithm is more energy efficient compared to other control schemes.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

FIGURE 3: Energy consumption of video streaming service of the (a) WNIC, (b) CPU, and (c) WNIC and CPU combined.

Acknowledgments

This work was supported by a grant from the Disaster and Safety Management Institute funded by the Ministry of Public Safety and Security of the Korean Government (MPSS-SD-2015-41).

References

- [1] M. N. Ismail, R. Ibrahim, and M. F. Md. Fudzee, "A survey on content adaptation systems towards energy consumption awareness," *Advances in Multimedia*, vol. 2013, Article ID 871516, 8 pages, 2013.
- [2] CISCO, "Cisco visual networking index: global mobile data traffic forecast update, 2014–2019," White Paper, 2015.
- [3] M. Kennedy, A. Ksentini, Y. Hadjadj-Aoul, and G.-M. Muntean, "Adaptive energy optimization in multimedia-centric wireless devices: a survey," *IEEE Communications Surveys and Tutorials*, vol. 15, no. 2, pp. 768–786, 2013.
- [4] X. Li, M. Dong, Z. Ma, and F. C. A. Fernandes, "GreenTube: power optimization for mobile videostreaming via dynamic

- cache management,” in *Proceedings of the 20th ACM International Conference on Multimedia (MM '12)*, pp. 279–288, ACM, Nara, Japan, November 2012.
- [5] J. He, Z. Xue, D. Wu, D. O. Wu, and Y. Wen, “CBM: online strategies on cost-aware buffer management for mobile video streaming,” *IEEE Transactions on Multimedia*, vol. 16, no. 1, pp. 242–252, 2014.
 - [6] S. Moon, J. Yoo, and S. Kim, “Optimal multi-interface selection for mobile video streaming in efficient battery consumption and data usage,” *Mobile Information Systems*, vol. 2016, Article ID 4871629, 18 pages, 2016.
 - [7] D. H. Bui, K. Lee, S. Oh et al., “GreenBag: energy-efficient bandwidth aggregation for real-time streaming in heterogeneous mobile wireless networks,” in *Proceedings of the IEEE 34th Real-Time Systems Symposium (RTSS '13)*, pp. 57–67, December 2013.
 - [8] X. Liu, P. Shenoy, and M. D. Corner, “Chameleon: application-level power management,” *IEEE Transactions on Mobile Computing*, vol. 7, no. 8, pp. 995–1010, 2008.
 - [9] J. Cho, I. Cho, D.-Y. Kim, and B. Lee, “A combined approach for QoS-guaranteed and low-power video decoding,” *IEEE Transactions on Consumer Electronics*, vol. 57, no. 2, pp. 651–657, 2011.
 - [10] J. Luo and N. K. Jha, “Power-efficient scheduling for heterogeneous distributed real-time embedded systems,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 26, no. 6, pp. 1161–1170, 2007.
 - [11] H. F. Sheikh, H. Tan, I. Ahmad, S. Ranka, and P. Bv, “Energy- and performance-aware scheduling of tasks on parallel and distributed systems,” *ACM Journal on Emerging Technologies in Computing Systems*, vol. 8, no. 4, article no. 32, 2012.
 - [12] H. Jeon, W. H. Lee, and S. W. Chung, “Load unbalancing strategy for multicore embedded processors,” *IEEE Transactions on Computers*, vol. 59, no. 10, pp. 1434–1440, 2010.
 - [13] S. Mohapatra, N. Dutt, A. Nicolau, and N. Venkatasubramanian, “DYNAMO: a cross-layer framework for end-to-end QoS and energy optimization in mobile handheld devices,” *IEEE Journal on Selected Areas in Communications*, vol. 25, no. 4, pp. 722–737, 2007.
 - [14] M. Li, Z. Guo, R. Y. Yao, and W. Zhu, “A novel penalty controllable dynamic voltage scaling scheme for mobile multimedia applications,” *IEEE Transactions on Mobile Computing*, vol. 5, no. 12, pp. 1719–1733, 2006.
 - [15] J. Kwak, O. Choi, S. Chong, and P. Mohapatra, “Processor-network speed scaling for energy-delay tradeoff in smartphone applications,” *IEEE/ACM Transactions on Networking*, vol. 24, no. 3, pp. 1647–1660, 2016.
 - [16] J. Stewart, *Metric International Version Calculus*, Early Transcendentals, Thomson Brooks/Cols, 6th edition, 2008.

Hindawi

Submit your manuscripts at
<https://www.hindawi.com>