CLUES: A Cross-Layer Energy Saving Scheme for Wi-Fi Networks

Jenn-Wei Lin[1], Chi-Yi Lin[2]

[1]Department of Computer Science and Information Engineering, Fu Jen Catholic University, Taiwan
[2]Department of Computer Science and Information Engineering, Tamkang University, Taiwan
jwlin@csie.fju.edu.tw, chiyilin@mail.tku.edu.tw

Abstract

The problem of significant energy consumption by the built-in Wi-Fi interface of mobile devices has been discussed for many years. We propose a cross-layer energy-saving scheme to maximize the sleep time and minimize the number of state transitions in the Wi-Fi interface of a mobile device. To achieve energy savings, we defer and join packet transmissions of mobile apps. However, some mobile apps may have delay constraint requirements for their packet transmissions. To avoid violating the delay constraint of an app, we also determine the final transmission interval of each delay-constrained packet. After joining the transmissions of some packets, the Wi-Fi interface can have a longer sleep duration to reduce energy consumption. To fully utilize the Wi-Fi bandwidth of the mobile device and satisfy the transmission requirements of different delay-constrained apps, the problem of selecting packets to join their transmissions is mapped to the well-known knapsack problem. We develop a greedy algorithm to efficiently solve the above mapping problem. Finally, simulation experiments are performed to demonstrate the effectiveness of the proposed scheme in energy savings and transmission delay.

Keywords: Mobile devices, Wi-Fi, Cross layer, Energy savings, Greedy algorithm

1 Introduction

In the past decade, there has been rapid growth in the use of mobile devices in Wi-Fi networks. Based on [1], more than 2 billion mobile devices (smartphones and tablets) are shipped worldwide. Mobile devices are powered by batteries to support portability. However, batteries can only store a finite amount of electrical energy. Consequently, mobile devices have a limited energy budget [2]. Additionally, the research results of [3] also showed that the Wi-Fi interface drains up to 50% of the total energy spent by a mobile device. Thus, reducing the energy consumption of Wi-Fi communication is an important research issue for the usage of mobile device [4-8].

The IEEE 802.11 standard [9] has provided a power-saving mode (PSM) to save energy consumption in Wi-Fi communications. The 802.11 PSM belongs to a static power-saving scheme, which does not adapt the sleep and wake durations according to the degree of network activity. In addition, it does not provide the delay guarantee for applications running in a mobile device. Moreover, the previous works [10-17] do not consider the following problems.

- A mobile device may concurrently execute several apps. Some apps are with video streaming and VoIP packet transmissions, which have the delay constraint requirements in their packet transmissions. Our energy-saving scheme especially considers the heterogeneous traffic (different packet types, packet arrival rates, and delay constraint requirements) in running apps. The previous schemes are usually from a single app's perspective to consider both the delay constraint and energy savings.
- Each mobile app has upstream and downstream packet transmissions. Most of the previous energy-saving schemes only focused on reducing the energy consumption of downstream packet transmissions from the access point (AP) to the mobile device.
- Within the coverage of an AP, there are usually multiple mobile devices. Each mobile device needs to share the Wi-Fi bandwidth with others. For a mobile device, if the obtained Wi-Fi bandwidth is insufficient to satisfy the transmission performance requirements of its running apps, the delay constraints of some apps will be violated. This problem is not discussed in the previous energy-saving schemes.

To consider the above energy-saving problems for mobile devices, we take the ***cross-layer utilization*** to bring the app information into the medium access control (MAC) layer for ***energy savings***. The proposed new Wi-Fi energy saving scheme is called as CLUES. We have studied the similar problem in our previous work [18]. However, the work does not elaborate the key techniques used for energy savings. In addition, it

*Corresponding Author: Chi-Yi Lin; E-mail: chiyilin@mail.tku.edu.tw
DOI: 10.3966/160792642019052003008

does not consider heterogonous traffic from different apps, and not minimize the transmission violation effect. In CLUES, we exploit two techniques: deferring transmission and joint transmission. By deferring the transmission of a packet, it is possible to find more packets in the queue of the mobile device. The joint transmission can concatenate multiple packet transmissions together. In addition, some mobile apps (e.g. video streaming, VoIP, etc.) have the delay-constrained property in their packet transmissions. To avoid violating the delay constraint of an app, we add the setting of the final transmission interval while deferring packet transmissions. As mentioned above, the Wi-Fi interface of a mobile device may not have an enough bandwidth to support all the running delay-constrained apps. In such a case, a *joint-selection problem* is introduced in the joint transmission. The problem is how to select appropriate upstream and downstream packets to join their transmissions together without exceeding the bandwidth of the Wi-Fi interface, which can be mapped into the well-known *knapsack* problem. However, the knapsack problem is known to be NP-hard [19]. In the CLUES scheme, we also propose a greedy algorithm to efficiently solve the joint-selection problem in polynomial time.

Overall, this paper makes the following contributions:

- We propose a cross-layer energy-saving scheme which uses the deferring transmission and joint transmission to achieve long sleep durations and reduce the number of state transitions in the Wi-Fi interface of a mobile device. Mobile devices can work in an energy-efficient manner.
- We consider multiple running apps in a mobile device with different traffic types. Some apps have delay constraint requirements for their packet transmissions. We present the final transmission setting to satisfy the transmission performance requirements of delay-constrained apps.
- We perform energy savings in both upstream and downstream packet transmissions.
- We also minimize the transmission violation effect if the mobile device does not have a sufficient Wi-Fi bandwidth for its running delay-constrained apps.

The rest of the paper is organized as follows. The related work is given in Section 2. Section 3 presents our CLUES scheme. Section 4 evaluates the performance of the proposed scheme. Finally, Section 5 concludes the paper.

2 Related Works

A comprehensive review of energy savings for mobile devices has been presented in [20]. There have been a number of energy-saving methodologies and techniques, which include smart batteries, energy-saving graphical user interface (GUI) design, sleep to save energy, power efficient communication, etc. Based on [20], our proposed energy-saving scheme belongs to the power efficient communication. In this section, we focus on surveying the existing schemes for discussing energy savings in Wi-Fi communication.

The 802.11 standard has defined the PSM [9] for the Wi-Fi interface of a mobile device. In the PSM of 802.11, a traffic indication map (TIM) information is attached to the beacon message. The beacon message is a control message, which is periodically transmitted between an AP and a mobile device. Upon receiving the beacon message, if the TIM indicates that there are buffered packets in the AP for the mobile device, the wireless network interface card (WINC) of the mobile device will continue to set in the active mode to receive the buffered packets. Otherwise, the WINC goes into the sleep mode. The simulation results of [10] show that the PSM can significantly reduce the amount of energy consumption spent by the WINC. However, the PSM correspondingly increases the packet transmission delay from the AP to the mobile device. In [13], it showed that PSM causes a degree of slowdown in packet transmission delay.

There is a tradeoff between the energy savings and packet transmission delay. To balance the two conflicting factors, a bounded slowdown (BSD) protocol was proposed in [10]. It can guarantee that the round-trip time (RTT) of a Web access application does not increase by more than a given factor *p*. To accomplish this, the mobile device still stays awake for a short period of time after sending a data request. However, if the delay tolerant of the packet transmission is very low, the energy consumption of the BSD will be larger than PSM. To improve the BSD, the authors of [11] proposed a new energy-saving scheme called smart power saving mode (SPSM). SPSM dynamically estimates the time instants when the mobile device wakes up to listen for beacons. The researchers of [12] used a bandwidth throttling technique to enhance the PSM. Due to bandwidth throttling at the server side, the effective data transmission rate is often much lower than the available bandwidth. The unused network bandwidth can be used to reshape the traffic into periodic bursts with an average throughput. With such periodic burst transmission patterns, idle and busy phases on the network transmissions can be distinguished. As a result, the WINC can be turned on and off at the right time in order to minimize energy consumption without degrading the user-perceived performance.

In addition, there are following PSM-improved schemes which utilize the app traffic information. The authors of [13] assumed that the app traffic behavior can be described by an alternating sequence of network activity bursts and user think times. They proposed a Cross-Layer Energy Manager (XEM) that exploits information scattered across layers to detect the beginning of network activity bursts and user think times. During a network activity burst, the XEM activates PSM. If a user think interval is detected due to no network activity, the XEM switches the WINC to the off mode to save energy. Similarly, the authors of [14] also presented a cross-layer and energy-efficient scheme, which consists of two modules deployed at the application and MAC layers, respectively. Due to assuming the bursty natural and predictable properties in data traffic, the scheme of [14] can shape the data traffic into bursts. Next, the scheme performs the adaptive sleep/wakeup scheduler to reduce unnecessary state switching in order to achieve energy savings. Palit et al. [15] studied the traffic patterns of smartphones. They observed that a good portion of packets are small in size and the generated traffic is bursty in nature. Based on the observations, the authors of [15] proposed a low energy data-packet aggregation scheme (LEADS) to aggregate a number of network-layer packets into a burst MAC-layer frame. With this scheme, a longer inactivity period can be obtained to conserve energy.

Unlike the above studies, [16] improved the PSM in the buffer management. It is known that the energy conservation of PSM is achieved by the AP buffering packets destined for mobile devices. If the AP is short of memory, some pending packets will be dropped, which produces lots of retransmissions. In [17], the authors investigated the Wi-Fi energy optimization in the multiple AP environment. The previous PSM-like schemes only considered a single AP. In the multiple AP environment, the network contention among different APs will dramatically increase the energy consumption of a mobile device. To mitigate the network contention in the energy perspective, the main idea of [17] is to make that different APs are active/inactive during non-overlapping time windows.

3 Proposed Energy Saving Scheme

In this section, we will present a new energy-saving transmission scheme, called the CLUES. Compared to the previous approaches, the CLUES scheme especially considers the following two points:

- In a mobile device, the running apps have various traffic types. Some apps have the delay constraint requirements in their packet transmissions.
- In addition to the downstream packet transmissions, each app has the upstream packet transmissions. It is also required to perform energy savings in the upstream packet transmissions.

The basic idea of the CLUES scheme is illustrated in Figure 1. With multiple apps in a mobile device, the CLUES scheme first performs the virtual queue division. The queues in the mobile device and AP are virtually divided into multiple sub-queues. The virtual queue division will be given in subsection 3.2. Each app has a corresponding sub-queue in the mobile device and AP for its upstream and downstream packet transmissions. Based on the sub-queue, we estimate the final transmission interval of each upstream (downstream) packet of an app. The estimation of the final transmission interval is for considering the delay constraint and packet lossless, which will be elaborated in subsection 3.3. In Figure 1, app_2 has the larger delay constrain for its packet transmissions than app_1. Therefore, packets p_{21} and p_{22} are set with larger final transmission intervals than packets p_{11} and p_{12}. When it is up the final transmission interval of an upstream (downstream) packet *p*, the WINC is activated to transmit the packet *p*. Here, we apply the lazy transmission concept. The WINC is not activated until the performance requirements (e.g. delay constraint, packet lossless, etc.) of a packet will be violated. When transmitting packet *p*, the WINC can simultaneously transmit other upstream and downstream packets since it owns a certain amount of bandwidth. Even if the final transmission intervals of these other packets are not due, they are transmitted together with packet *p*. In Figure 1, packets p_{11}, p_{12}, p_{22}, p_{n1}, p_{n2} are transmitted together with packet p_{21}. The details will be described in subsection 3.4. After joining multiple packet transmissions together, the WINC can go into the sleep state for a long period since all ongoing packets have been transmitted.

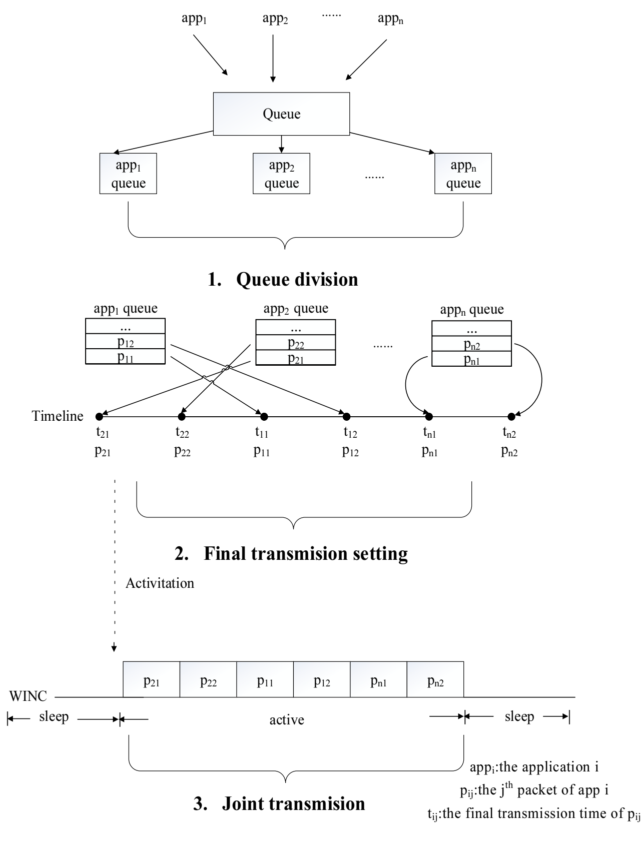


Figure 1. The basic idea of the CLUES scheme

3.1 Definitions

Before elaborating the CLUES scheme, we first make the following assumptions and definitions:

- The execution time of an app can be partitioned into a number of fixed intervals. Here, an execution interval is called a time slot. For each app, it starts at the beginning of a time slot. For example, in Android [21], the time slot is called a scheduling period which is fixed as 20msecs. Within a time slot, there may have several running apps. These apps have different traffic types.
- If an app is with the delay-constrained property (e.g. multimedia app), it requires the transmission time of a packet no greater than a specified value Δ. The transmission time consists of two parts. One is the waiting time of a packet in the queue. The other is the communication time for sending the packet from the mobile device (AP) to the AP (mobile device). The sum of the waiting time and communication time cannot exceed Δ.
- If multiple mobile devices co-exist in a network, the contention among them (packet collision) will increase the packet transmission time of a mobile device. In WLAN, the issue of collision-free packet transmissions has been extensively investigated by [22-25]. Due to assuming the collision-free property, a mobile device can hold a certain amount *B* of dedicated bandwidth to transmit data with AP.
- A basic packet size s is assumed here. Based on the basic packet size, if the upstream (downstream) packet size of an app *i* is s_i^u (s_i^d), its packet size will be normalized as $\frac{s_i^u}{s}$ ($\frac{s_i^d}{s}$).
- The traffic of an app can be modeled. In [11], the exponential moving average algorithm is applied to estimate the downstream packet distribution of an app. [14] assumed that data traffic is predictable based on historical data. Without loss of generality, we adopt the Poisson and exponential distributions [26] to model the packet arrivals and the packet transmission time of an app, respectively. If the *kth* packet of the app *i* arrives at a queue is time *t*, the (*k+1*)*th* packet will arrive at $t+exp(\frac{1}{\lambda_i^u})$, where *exp* is the function to generate a random number based on the exponential distribution, and the λ_i^u is the average packet arrival rate. In the packet transmission aspect, if the average packet transmission rate of the app *i* is μ, the average packet transmission time will be $\frac{1}{\mu_i^u}$.
- In the above traffic modelling, the Poisson and exponential distributions are used for conveniently adopting a queuing model to formulate the packet performance metrics of an app. Based on the formulated performance metrics, we can obtain the packet performance values and then feed these values into the CLUES scheme.

In addition to the above assumptions, we also define some notations for elaborating the CLUES scheme, as shown in Table 1.

Table 1. Notations

Notation	Description
n	The total number of mobile apps running in a time slot.
s_i^u (s_i^d)	The average upstream (downstream) packet size of the app *i*
s	The basic packet size for normalization
sq(aq)	The size of the whole mobile device (AP) queue
$n\_\lambda_i^u (n\_\lambda_i^d)$	The mean number of normalized upstream (downstream) packets of the app *i* in a time slot. It also represents the upstream (downstream) transmission rate of the app *i*.
$n\_\mu_i^u (n\_\mu_i^d)$	The mean service time of a normalized upstream (downstream) packet while executing the app *i*.
$dc_i^u (dc_i^d)$	The required delay constraint of a normalized upstream (downstream) packet while executing the app *i*.

3.2 Queue Division

Multiple apps may run in a mobile device concurrently. The transmission queue of the mobile device (device queue) cannot be fully occupied by only one app. The mobile device queue holds the upstream packets of apps. In the CLUES scheme, the device queue is virtually divided based on the upstream arrival rates of running apps. The device queue size qs_i^u of app *i* can be estimated using the following equation.

$$qs_i^u = sq \times \frac{n\_\lambda_i^u}{\sum_{k=1}^{n} n\_\lambda_k^u} \quad \textbf{(1)}$$

where $\sum_{k=1}^{n} n\_\lambda_k^u$ is the sum of the upstream packet arrival rates of all running apps. Each app is based on the ratio of its upstream packet arrival rate over the total arrival rate to determine the occupied device queue size. If $n\_\lambda_i^u$ is larger than $n\_\lambda_j^u$, the app *i* has more upstream packet transmissions than the app *j*. By Eq. (1), the app *i* can obtain more space of the device queue than the app *j*. For conveniently adopting Eq. (1) in later equations, we also use the maximum number $\max\_p_i^u$ of basic packets allowed to be stored to represent the device queue size of the app *i*, as follows.

$$\max\_p_i^u = \frac{qs_i^u}{s} = (sq \times \frac{n\_\lambda_i^u}{\sum_{k=1}^{n} n\_\lambda_k^u}) \times \frac{1}{s} \quad \textbf{(2)}$$

Similar to the device queue division, the AP queue can be also virtually divided by apps. The AP queue stores the downstream packets of apps. In the following, we give the maximum number $\max\_p_i^d$ of basic packets allowed to be stored in the AP queue of app *i*.

$$\max\_p_i^d = (aq \times \frac{n\_\lambda_i^d}{\sum_{k=1}^{n} n\_\lambda_k^d}) \times \frac{1}{s} \quad \textbf{(3)}$$

3.3 Final Transmission Setting

From the energy saving perspective, if an upstream (downstream) packet is sent from a mobile device (AP), this packet transmission should be deferred as long as possible to join transmissions of other packets. The above joint transmission can reduce the number of state transition (sleep to active) of the WINC to achieve energy savings. However, each app occupies the finite device (AP) queue size (see subsection 3.2).

Moreover, some apps has the delay-constrained requirements for their packet transmissions. The packet transmission of an app cannot be deferred too long. Therefore, appropriate packet transmission rates are required to be set for the upstream and downstream packet transmissions of an app. Otherwise, the packet loss and the delay-violated transmissions will occur. After deriving the packet transmission rate, we can further set the final transmission interval of each packet of an app. The packet transmission rate is a key parameter for calculating the final transmission interval of a packet. As mentioned before, the setting of the final transmission interval can consider both energy savings and transmission performance.

3.3.1 Transmission Rate

In section 3.1, the Poisson distribution and exponential distribution were exemplified to model the packet arrivals and packet transmission time of an app. Based on these two distributions, we can adopt the *M/M/1* queuing model [27] to estimate the upstream packet transmission rate. Note that if other distributions are used to model packet traffic, another appropriate queue model is selected to derive the upstream packet transmission rate. Our CLUES scheme is independent of the adopted queuing model.

$$\frac{1}{p_i^u - n\lambda_i^u} \le dc_i^u \text{ and } \frac{(n\_\lambda_i^u)^2}{n\_\mu_i^u(n\_\mu_i^u - n\_\mu_i^u)} \le \max\_p_i^u \quad \textbf{(4)}$$

In Eq. (4), the two given formulas represent the average transmission time of a packet and the average number of upstream packets in the device queue of app *i*, respectively [29]. By setting two upper bound values in the two given formulas of Eq. (4), we can derive the smallest $n\mu_i^u$ which satisfies both delay constraint and packet lossless requirements. The reasons are explained as follows.

- The transmission time of an upstream packet is not greater than a specified delay constraint. This is to ensure the delay constraint satisfaction.
- The number of the buffered upstream packets in the device queue of an app does not exceed the maximum number of packets allowed in the occupied queue. This is for preventing packet loss.

In Eq. (4), dc_i^u and $n\_\lambda_i^u$ are given in advance. The $\max\_q_i^u$ has been derived by Eq. (2). Substituting these well-known parameters into Eq. (4), we can estimate the parameter of the upstream packet transmission rate $n\_\mu_i^u$.

Similarly, the downstream transmission rate of an app can be derived as follows.

$$\frac{1}{n\_\mu_i^d - n\_\lambda_i^d} \le dc_i^d \text{ and } \frac{(n\lambda_i^d)^2}{n\mu_i^d(n\mu_i^d - n\lambda_i^d)} \le \max\_p_i^d \quad \textbf{(5)}$$

3.3.2 Final Transmission Interval

After deriving $n\_\mu_i^u$ from Eq. (4), the final transmission interval of an upstream packet can be estimated as follows. Assume that the app *i* sends an upstream packet *p* at time *t*. At the time, the packet *p* is put into the device queue of the app *i* for waiting transmission. Then, the transmission of the packet p will start at $t + \frac{1}{n\_\mu_i^u - n\_\lambda_i^u} - \frac{1}{n\_\mu_i^u}$. Note that $\frac{1}{n\_\mu_i^u - n\_\lambda_i^u} - \frac{1}{n\_\mu_i^u}$ is also a well-derived formula of the *M/M/1/n* queuing model [27] which represents the average waiting (queuing) time of the packet *p* in a queue by giving the arrival rate $n\_\lambda_i^u$, the transmission rate $n\_\mu_i^u$, and the device queue size with *n* basic packets. Next, the mobile device will take $\frac{1}{n\_\mu_i^u} \times \frac{s_i^u}{s}$ time to transmit packet *p*. In Eq. (4), we have estimated the smallest $n\_\mu_i^{u'}$ which can satisfy the delay constraint and packet lossless requirements. Therefore, the [$t + \frac{1}{n\_\mu_i^u - n\_\lambda_i^u} - \frac{1}{n\_\mu_i^u}$, $t + \frac{1}{n\_\mu_i^u - n\_\lambda_i^u} - \frac{1}{n\_\mu_i^u} + \frac{1}{n\_\mu_i^u} \times \frac{s_i^u}{s}$] can be regarded as the final transmission interval of the packet *p*. If the transmission of the packet *p* is not completed before $t + \frac{1}{n\_\mu_i^u - n\_\lambda_i^u} - \frac{1}{n\_\mu_i^u} + \frac{1}{n\_\mu_i^u} \times \frac{s_i^u}{s}$, the delay constraint of the packet *p* cannot be guaranteed. In addition, some incoming upstream packets may be lost due to deferring the transmission of packet *p* too long.

For each downstream packet of the app *i*, its final transmission interval can be estimated based on the above similar way.

3.4 Joint Transmission

In addition to the final transmission setting, the joint transmission is the other key technique proposed in our CLUES scheme. It can concatenate multiple packet transmissions together.

3.4.1 Transmission List Formation

To perform the joint transmission in the upstream and downstream packets of running apps, we first use two transmission lists to keep track the transmission information about each upstream and downstream packet of an app, respectively. The data structure of the two transmission lists is represented as a 4-tuple: *sending/receiving app identifier*, *packet identifier*, *packet size*, and *final transmission interval*. Whenever an app issues an upstream packet, the packet will be put into the device queue of the app. The 4-tuple information of the packet is extracted and kept in the upstream transmission list. Similarly, the transmission information of each downstream packet is stored in the downstream transmission list.

The upstream status list can be easily formed since the device queue is in the mobile device itself. However, since the AP queue is not stored in the mobile device, the downstream transmission list cannot be directly established by the mobile device. In section 3.1, we adopted the Poisson distribution to model the arrivals of downstream packets of an app. If the *kth* downstream packet of the app *i* arrives at the AP queue is time *t*, the *(k+1)th* downstream packet will arrive at $t+exp(\frac{1}{n\_\lambda_i^d})$, where *exp* is the function that generates a random number based on the exponential distribution with the average inter-arrival time $\frac{1}{n\_\lambda_i^d}$. Based on the packet distribution, the traffic patterns of downstream packets can be estimated by the mobile device.

In practice, the downstream packets to the AP queue do not precisely follow the adopted distribution. To reflect the practical arrival patterns of downstream packets, the downstream list is periodically updated as follows. Whenever the mobile device downloads packets from the AP, the AP determines whether or not there are un-downloaded downstream packets in the AP queue. If so, the 4-tuple transmission information of these un-downloaded packets is attached on the downloaded packets to be sent to the mobile device. Upon receiving the updated 4-tuple information, the mobile device updates its downstream transmission list.

3.4.2 Joint Packet Selection

The upstream and downstream transmission lists are used to fulfill the joint transmission, as follows. When the WINC of a mobile device transmits an upstream or downstream packet *p*, the remaining wireless bandwidth should be enough to involve other packets to join the transmission of packet *p*. The involved packets are called the *joint packets* of packet *p*, which transmissions will be joined with the transmission of packet *p* together.

The above joint transmission can enable the mobile device to have longer active and sleep durations to reduce the number of sleep-to-active state transitions. To maximize the benefit of the joint transmission, more joint packets should be involved. However, the WINC of mobile device holds a limited bandwidth capacity. Additionally, different apps have different packet sizes. The joint packet selection is a combinatorial problem. For obtaining the optimal solution, we transform the joint packet selection problem into a well-known knapsack problem.

Definition 1. The knapsack problem is a problem in which one tries to pick some items in such a way that the total profit is maximal, while ensuring that the total weight of the chosen items does not exceed the knapsack capacity.

Before describing the transformation between the joint selection problem and the knapsack problem, we first assume that packet *p* is with the earliest final transmission interval $[t_{p_1}, t_{p_2}]$ among all packets in the upstream and downstream transmission lists. The mobile device will begin to transmit packet *p* at t_{p_1}. Other packets in the upstream and downstream transmission lists are the joint packet candidates of packet *p*. Each of them is associated with a *joint value* with respect to packet *p*. Here, the joint value is used for determining the selection order to be a joint packet of packet *p*. Before formally defining the joint value, we first define the *transmission difference*.

Definition 2. Assume the packet *q* is with the final transmission interval $[t_{q_1}, t_{q_2}]$. The transmission difference with respect to packet *p* is defined as $|t_{q_2}, t_{p_1}|$, where t_{q_1} and t_{q2} are the lower bound and upper bound of the final transmission interval of packet *q*, respectively.

Definition 3. For packet *q*, its joint value with respect to packet *p* is the inverse of its transmission difference. The joint value of packet *q* is $\frac{1}{|t_{q_2} - t_{p_1}|}$.

According to Definition 3, if the joint value of the joint packet candidate *q* is larger than that of the joint packet candidate *r*, it represents that packet *q* has a smaller transmission difference with respect to packet *p*. Packet *q* is more urgent than packet *r* to join the transmission of packet *p*. Therefore, the joint packet selection order follows the descending order of the joint value.

Next, we give the transformation to the knapsack problem, as follows.

- The WINC of a mobile device holds a certain amount of bandwidth to transmit packets. The bandwidth capacity corresponds to the capacity of a knapsack.
- In the upstream and downstream transmission lists, each packet can be regarded as an item to be put into the knapsack.
- Each packet may have the different size which corresponds to the size of an item. The packet is also associated with a joint value which corresponds to the profit of an item.
- The objective of the joint packet selection problem is to maximize the total joint value of all selected joint packets. This corresponds to the maximum total profit of all chosen items in the knapsack.

Based on the above mappings, the joint packet selection problem can be transformed to the knapsack problem.

3.4.3 Greedy Algorithm

Solving the knapsack problem is known to be NP-hard [19]. Instead of finding the optimal solution, we modify a well-known knapsack greedy algorithm to solve the joint packet selection problem in polynomial time, as given in Figure 2.

In Figure 2, we first assume that a mobile device will first transmit packet p (line 1). Depending on the attribute of packet p, the identity of packet p will be put in the upstream or downstream section (lines 2-6). The upstream and downstream sections are two temporary buffers to collect the identities of packets which transmissions will be joined together. The total bandwidth demand of the joint transmission is initially set to the size of packet p (line 7), which will increase as involving joint packets. To conveniently select the joint packets, the algorithm partitions the upstream and downstream transmission lists into a number of upstream and downstream sub-lists based on the app identity (line 8). The algorithm alternatively selects two possible joint packets from the upstream and downstream sub-lists of each app (lines 9-34). The $next_{idx_i^u}$ and $next_{idx_i^d}$ are the two retrieval points used to assist the joint packet selection, which values are initialized to 1 (lines 9-11). By following the two retrieval pointers, the two possible joint packets are compared with each other using the selection metric:

Input: Packet p and the wireless bandwidth capacity of the mobile device
Output: The joint packets of packet p
1 $p \leftarrow$ Selecting the packet with the smallest beginning transmission time from the upstream and downstream transmission lists
2 **if** (p is an upstream packet)
3 $US \leftarrow$ identity of packet p; $DS \leftarrow \{\,\}$
4 **else**
5 $US \leftarrow \{\,\}$; $DS \leftarrow$ identity of packet p
6 **end if**
7 $tot\_band\_demand \leftarrow$ the size of packet p
8 Partitioning the upstream and downstream lists (UL and DL) using the app identity: $UL \rightarrow \{UL_1, UL_2, \ldots UL_n\}$ and $DL \rightarrow \{DL_1, DL_2, \ldots DL_n\}$
9 **for** each app App_i /* Initialize the retrieval pointers of joint packets */
10 $next\_idx_i^u = 1$; $next\_idx_i^d = 1$
11 **end for**
12 **while** true
13 **if** ($band\_capacity$-tot_band_demand)≤0 /* the remaining bandwidth*/
14 break
15 **end if**
16 **for** each app App_i
17 $p_i^u \leftarrow$ follow $next\_idx_i^u$ to select a possibly joint packet form UL_i;
18 $p_i^d \leftarrow$ follow $next\_idx_i^d$ to select a possibly joint packet form DL_i;
19 **if** $\left(\frac{j(p_i^u)}{s(p_i^u)} \geq \frac{j(p_i^d)}{s(p_i^d)}\right)$
20 $p_i^c \leftarrow p_i^u$
21 **else**
22 $p_i^c \leftarrow p_i^d$
23 **end if**
24 **end for**
25 Compare all p_i^c to select the p_k^c with the largest selection metric value.
26 $tot\_band\_demand \leftarrow$ tot_band_demand +the size of packet p_k^c
27 **if** (p_k^c is an upstream packet of App_k)
28 $US \leftarrow US \cup$ identifity of packet p_k^c;
29 $next\_idx_k^u \leftarrow next\_idx_k^u + 1$;
30 **else**
31 $DS \leftarrow DS \cup$ identity of packet p_k^c;
32 $next\_idx_k^d \leftarrow next\_idx_k^d + 1$;
33 **end if**
34 **end while**
35 Generating a state transition to wake up the wireless interface
36 Sending and receiving the packets specified in the US and DS including packet p.
37 Delete the corresponding status entries of the sending and receiving packets
38 Switching the wireless interface into the sleep mode

Figure 2. The joint packet selection algorithm

$\frac{joint\ value\ j(\,)}{packet\ size\ s(\,)}$. Note that the selection metric is also used in the knapsack greedy algorithm [19] to be the item selection order. The packet with the larger metric value will be a joint packet candidate (lines 19-23), which will further be compared with the joint packet candidates of other apps. After comparing the metric values of all joint packet candidates, the packet with the largest metric value will be designated formally as a joint packet (line 25). According to the attribute of the packet, the identity of the packet will be put in the upstream or downstream transmission section (lines 27-33).

If the remaining bandwidth of the mobile device is insufficient to involve one more joint packets, the joint packet selection stops (lines 13-15). After selecting all the joint packets, the mobile device wakes up its WINC from the sleep state to transmit (send and receive) packet p and its joint packets which are specified packets in the upstream and downstream transmission sections (lines 35-36). After performing the joint transmission, the entries of packet p and its joint packets are deleted from the upstream and downstream transmission lists (line 37). The WINC of the mobile device can go into the sleep state (line 38).

3.4.4 Time Complexity

The above greedy algorithm takes $O(\sum_{i=1}^{n}(n\lambda_i^u + n\lambda_i^d))$ to partition the upstream and downstream transmission lists into a number of status sub-list pairs, where $\sum_{i=1}^{n}(n\lambda_i^u + n\lambda_i^d)$ is the total number of packets in the upstream and downstream transmission lists during a time slot, and n is the number of running apps during a time slot. Then, the algorithm repeats the joint packet selection. In each selection round, two possible joint packets are first selected from the upstream and downstream sub-lists. Then, a joint packet is formally selected. Here, the joint packet selection takes $O(n)$. The maximum number of selection rounds is $\sum_{i=1}^{n}(n\lambda_i^u + n\lambda_i^d)$. In such case, all packets in the upstream and downstream transmission lists are selected as joint packets. Overall, the entire algorithm runs in polynomial time with $O(\sum_{i=1}^{n}(n\lambda_i^u + n\lambda_i^d))$

$$+O(n \times \sum_{i=1}^{n}(n\lambda_i^u + n\lambda_i^d)) = O(n \times \sum_{i=1}^{n}(n\lambda_i^u + n\lambda_i^d)).$$

4 Performance Evaluation

4.1 Simulation Setup

We used MATLAB [28] to perform simulation experiments for evaluating the performance of the proposed CLUES scheme and other related schemes. To improve the PSM with the delay constraint consideration, several schemes have been proposed in Section 2. Basically, these PSM-improved schemes wake up the WINC of the mobile device at least once during a beacon interval. Compared to these PSM-improved schemes, the energy consumption and packet transmission performance of the original PSM scheme can be regarded as the low bounds of the PSM-improved scheme in the energy consumption and packet transmission performance, respectively. If we only concern the delay constraint, the best PSM-improved scheme is the scheme that completes the packet transmission before the specified delay constraint. However, this scheme also generates the upper-bound energy consumption of all the PSM-improved schemes. In our simulation experiments, the PSM and all the PSM-improved schemes are categorized as the *PSM-type* scheme. In addition to the PSM-type scheme, the CLUES is further compared with the *non-energy-saving* (*NEE*) scheme. The NEE scheme can provide the best packet transmission performance. When a packet arrives at the device (AP) queue, the NEE scheme immediately transmits the packet between the mobile device and the AP. Here, the comparison with the NEE scheme is for quantifying the effectiveness of CLUES scheme in packet transmission in addition to the energy consumption aspect. We set the following simulation parameters in simulation experiments. First, we assume that there are five apps concurrently running in a mobile device. The five apps include two multimedia apps with different encoding schemes, namely H.264 and MPEG 4, respectively. The other three apps are Web mail, FTP, and VoIP. We refer to [29] to set different traffic types for five apps, as shown in Table 2.

Table 2. The Settings of Traffic Model and Power Consumption

	App	Multimedia	Web Mail	FTP	VoIP
Traffic Model	Traffic type	VBR	VBR	VBR	CBR
	Packet Size	[512, 1024]	[512, 1024]	[512, 1024]	512
	Packet Arrivals	Poisson	Poisson	Poisson	Fixed
Power Consumption	Sending	Idle	Sleeping	State transition	Receiving
	1.4W	0.7W	0.06W	1.85W	0.9W

In Table 2, the packet arrivals of each app follows a Poisson distribution and the mean packet size is randomly set within [512, 1024] bytes except the VoIP app. The VoIP has the fixed packet size (512 bytes) and packet arrival rate (10ms), respectively.

In the other two delay-constrained apps (two multimedia apps), the delay constrains of their upstream and downstream packet transmission are randomly set using the interval [10, 100] ms. The mobile device holds the 11 Mbps bandwidth capacity.

In the energy consumption aspect, we refer to [30-31] to set the energy consumption parameters. In [30-31], the authors measured the Wi-Fi power consumption of a mobile device (e.g. Google Nexus S) using the WINCs of Lucent 802.11 WaveLAN and Intel Wi-Fi Link 5300. The WINC has different power consumption in the four different states: sending, receiving, idle, sleeping, and state transition, as shown in Table 2. In the state transition, the WINC is waken from sleep to active, which has the largest power consumption 1.85W among the five states.

Based on the above simulation parameter settings, 50 simulation runs are performed. In each simulation run, the total number of packet transmissions for the five applications is set from 100 to 1000 steps of 100, respectively. In the next subsection, we give the average simulation results of the following metrics under the 50 simulation runs.

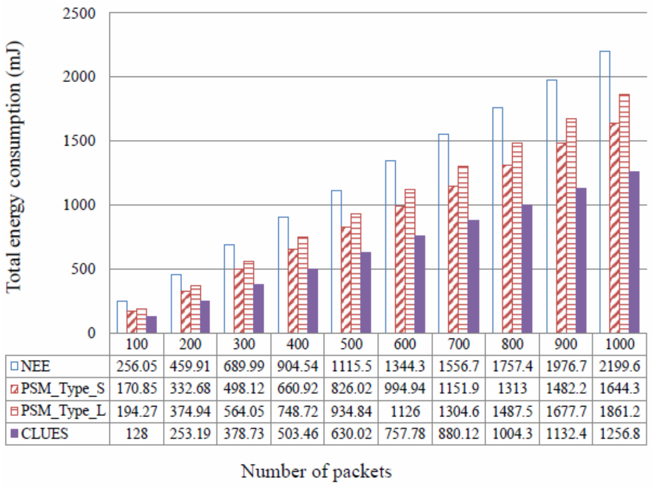
- Total energy consumption: the total amount of energy consumption taken for sending packets, receiving packets, staying in the sleep (idle) state, and launching a state transition to wake up the WINC of a mobile device.
- Average length of a sleep (idle) duration: the average time units for making the WINC of a mobile device stay in a continuous sleep (idle) period.
- Average number of state transitions: the number of waking up the WINC of the mobile device from the sleep state to the active state.
- Average satisfaction ratio: the ratio of the number of upstream and downstream packets transmitted within their respective desired delay constraints over the total number of upstream and downstream packet transmissions, such as $\frac{n_d^u + n_d^d}{n^u + n^d}$, where n^u is the number of upstream packets sent from the mobile device, n^d is the numbers of downstream packets sent from the AP, n_d^u and n_d^d are the numbers of upstream and downstream packets with the delay-constrained satisfaction, respectively.
- Average excess time: the excess time of a delay-violated packet on average. If a packet p is a delay-violated packet, the transmission of packet p is competed after its specified delay constraint, such as $(t_t^p - t_d^p) > 0$, where t_t^p is the transmission completion time of packet p, t_d^p is the specified delay constrain, and $t_c^p - t_d^p$ is the excess time of packet p.
- Average packet transmission time: the time units for sending (receiving) a packet on average from the mobile device (AP) to the AP (mobile device) after the packet generation, such as $t_t^p - t_g^p$, where t_g^p is the generation time of packet p.

4.2 Simulation Results

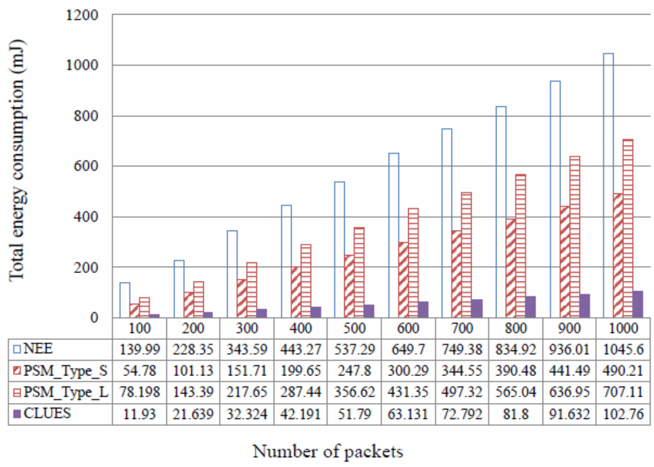
Figure 3 illustrates the comparison of total energy consumption. The PSM-type scheme includes a number of PSM-like schemes. In this subsection, each metric of the PSM-type scheme is evaluated based on the lower-bound and upper-bound manners. For example, in Figure 3, there are two bars (PSM-type-S and PSM-type-L) for the PSM-type scheme to indicate the smallest and largest energy consumption among all the PSM-like schemes, respectively. As shown in Figure 3(a), the CLUES has less energy consumption than the PSM-type-S. The average reduction ratio $\frac{(PSM_{Type_s} - CLUES)}{PSM_{Type_s}}$ is about 23%. Compared to the NEE and the PSM-type-L, larger average reduction ratios (44% and 32%) can be obtained. The basic idea of an energy-saving scheme is to make the WINC of a mobile device frequently stay in the sleep state to reduce energy consumption. The energy cost of the packet transmissions cannot be reduced regardless of adopting which energy-saving scheme. If the above irreducible energy cost is not counted into the total energy consumption, the CLUES can greatly improve the energy consumption, as shown in Figure 3(b). Compared with PSM-type-S, the average improvement ratio is at least 78% since CLUES adopts the lazy transmission and joint transmission to make longer sleep durations and fewer state transitions.

The CLUES also has longer sleep durations than other schemes, as shown in Figure 4(a). Even though the comparison is made with the original PSM, the average sleep duration of the CLUES is 14.42 times of the original PSM. The PSM periodically collects downstream packets, which usually has longer sleep durations than other PSM-like schemes. In Figure 4(a), the sleep duration of the original PSM is plotted in the PSM-type-L. For the number of state transitions, the CLUES has also fewer state transitions than all PSM-like schemes. Compared with the PSM-type-S (the PSM-like scheme with the fewest state transitions), the CLUES can reduce the number of state transitions about 90%, as shown in Figure 4(b).

In addition to energy savings, the CLUES also considers the delay constraints of packet transmissions. As shown in Figure 5, the CLUES provides a high satisfaction ratio for packet transmissions. The CLUES sets the final transmission interval for each packet transmission without violating the required delay constraint. Moreover, the joint transmission of the CLUES is also beneficial in meeting the delay constraint since the technique can move ahead of the

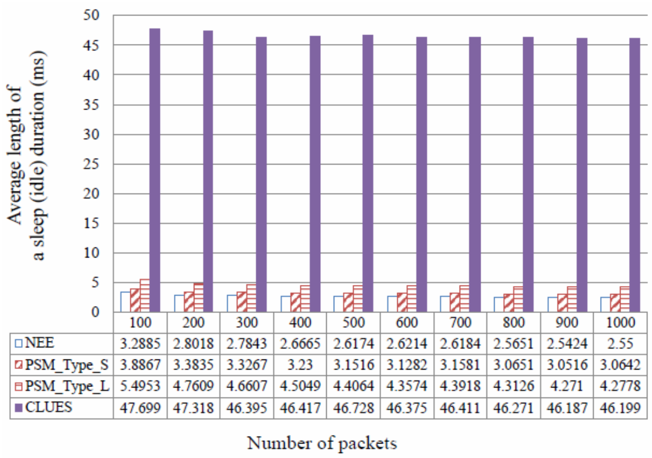


(a) The energy cost of packet transmissions included

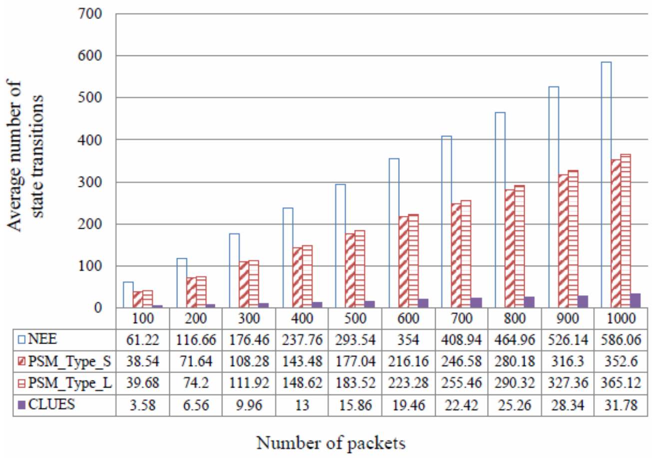


(b) The energy cost of packet transmissions not included

Figure 3. Total energy consumption



(a) The average length of a sleep (idle) duration



(b) The average number of state transitions

Figure 4. Comparisons of the sleep (idle) duration and the state transition

transmission intervals of some packets. In Figure 5, the NEE has the highest satisfaction ratio. Without energy savings, the NEE immediately sends and receives each packet. It completes almost all packet transmissions before their delay constraints. Compared to the NEE, the CLUES has similar ratio values in the satisfaction ratio metric. In this metric, the largest difference between the two schemes is 0.0086. For some PSM-type schemes, they are also concerned with the delay constraints during packet transmissions. These schemes transmit packets at the delay-satisfied threshold points. In the simulation experiments, there may be a large number of concurrent packet transmissions. Such packet transmissions may also have similar delay constraints. In such cases, the mobile device cannot provide enough bandwidth to transmit some packets at respective delay-satisfied threshold points.

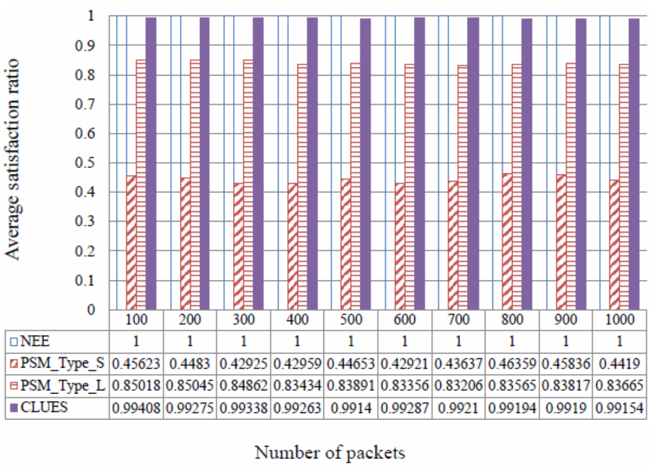
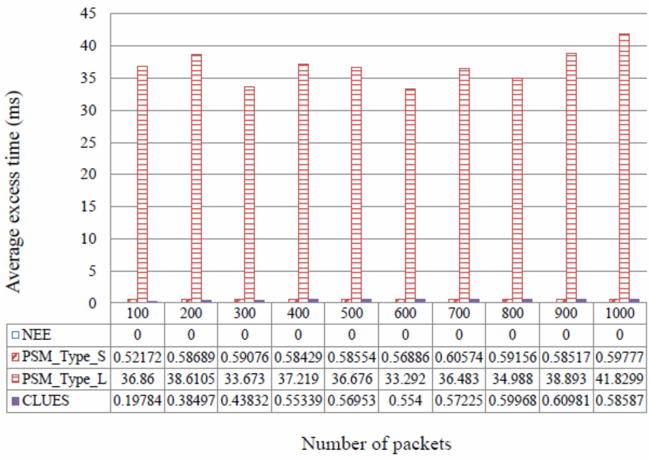


Figure 5. Average satisfaction ratio

For the delay-violated packets, the average excess time is shown in Figure 6. With the immediate transmission property, the excess time of the NEE is 0. The CLUES also has the small excess time for its delay-violated packets. Unlike the CLUES and enhanced PSM schemes, the original PSM scheme does not consider the delay constraint in the packet transmission. It significantly increases the average excess time, which can be seen from the PSM-type-L of Figure 6.



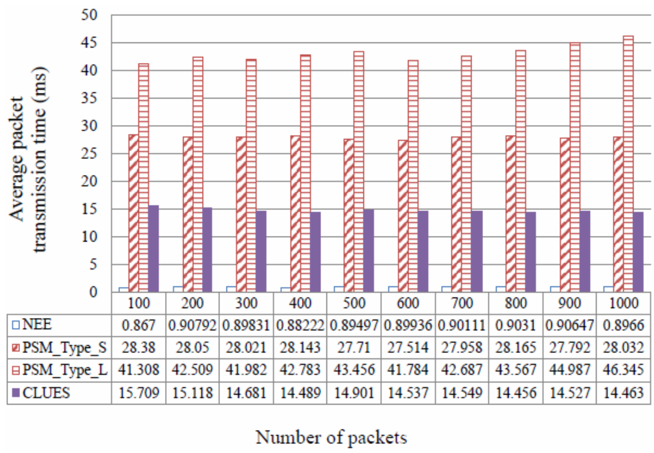
	100	200	300	400	500	600	700	800	900	1000
NEE	0	0	0	0	0	0	0	0	0	0
PSM_Type_S	0.52172	0.58689	0.59076	0.58429	0.58554	0.56886	0.60574	0.59156	0.58517	0.59777
PSM_Type_L	36.86	38.6105	33.673	37.219	36.676	33.292	36.483	34.988	38.893	41.8299
CLUES	0.19784	0.38497	0.43832	0.55339	0.56953	0.554	0.57225	0.59968	0.60981	0.58587

Figure 6. Average excess time

Although the CLUES adopts the lazy transmission, the technique does not defer a packet transmission too long since the delay constraint is considered in the setting of the final transmission interval. In addition, the CLUES also applies the joint transmission to join two or more packet transmissions. In such case, some packets are transmitted before their specified transmission intervals. Figure 7 gives the comparison of the packet transmission performance. The average packet transmission time of the PSM-type scheme is larger than that of the CLUES. Especially, since the original PSM is not concerned with the delay constraint, it increases the packet transmission time to nearly a half of a beacon interval, which can be seen from the PSM-type-L of Figure 7. Even though some PSM-like schemes are concerned with the delay constraint, the CLUES can reduce about 47% of packet transmission time in such schemes. This can be observed by comparing the CLUES with the PSM-type-S.

5 Conclusions

We have investigated the joint problem of delay constraint and energy savings for mobile devices. With efficient energy use, the lifetime of a mobile device can be extended when it concurrently performs several apps. The proposed energy-saving transmission scheme is abbreviated as the CLUES. In the scheme, we utilize the queuing model to estimate the final transmission interval of each packet without violating



	100	200	300	400	500	600	700	800	900	1000
NEE	0.867	0.90792	0.89831	0.88222	0.89497	0.89936	0.90111	0.9031	0.90647	0.8966
PSM_Type_S	28.38	28.05	28.021	28.143	27.71	27.514	27.958	28.165	27.792	28.032
PSM_Type_L	41.308	42.509	41.982	42.783	43.456	41.784	42.687	43.567	44.987	46.345
CLUES	15.709	15.118	14.681	14.489	14.901	14.537	14.549	14.456	14.527	14.463

Figure 7. Average packet transmission time

its delay constraint requirement. The final transmission is for making the CLUES achieve the lazy transmission. To further reduce energy consumption, if a packet p is transmitted during a certain time duration, the CLUES scheme involves other packets to join the transmission of packet p. After transmitting a series of packets, the WINC of the mobile device can obtain a longer sleep period to save energy consumption. However, the WINC has a limited bandwidth capacity. As a result, a joint packet selection problem is introduced, such that which packets are selected as joint packets without exceeding the bandwidth limitation. In the CLUES scheme, we modify a well-known knapsack greedy algorithm to efficiently solve the joint packet selection problem in polynomial time. The simulation results demonstrate that the proposed CLUES scheme can efficiently perform the delay-constrained and energy-saving packet transmissions in a mobile device. The CLUES has a similar satisfaction ratio with the NEE (non-energy-saving) scheme. Compared to the existing PSM-type scheme, it can reduce 27% of total energy consumption on average. If the essential energy cost of packet transmission is not counted, the reduction ratio of total energy consumption is about 81%. In the future, we plan to implement the CLUES scheme on various mobile device platforms.

References

[1] Gartner: Worldwide PC, Mobile Device Market to Grow in 2014, http://www.zdnet.com/article/gartner-worldwide-pc-mobile-device-market-to-grow-in-2014/, 2014.

[2] D. Rakhmatov, S. Vrudhula, Energy Management for Battery-Powered Embedded Systems, *ACM Transactions on Embedded Computing Systems*, Vol. 2, No. 3, pp. 277-324, August, 2003.

[3] G. Anastasi, M. Conti, E. Gregori, A. Passarella, A Performance Study of Power-Saving Policies for Wi-Fi Hotspots, *Computer Networks*, Vol. 45, No. 3, pp. 295-318, June, 2004.

[4] I. Pefkianakis, J. Chandrashekar, H. Lundgren, User-Driven

Idle Energy Save for 802.11x Mobile Devices, *2014 IEEE 11th International Conference on Mobile Ad Hoc and Sensor Systems*, Philadelphia, PA, 2014, pp. 282-290.

[5] K. Omori, Y. Tanigawa, H. Tode, Power-Saving for Wireless Stations Using RTS/CTS Handshake and Burst Transmission in Wireless LANs, *2017 14th IEEE Annual Consumer Communications & Networking Conference (CCNC)*, Las Vegas, NV, 2017, pp. 708-711.

[6] D. Liu, H. Wang, G. Zhou, W. Mao, B. Li, Arbitrating Traffic Contention for Power Saving with Multiple PSM Clients, *IEEE Transactions on Wireless Communications*, Vol. 15, No. 10, pp. 7030-7043, Oct. 2016.

[7] Y. Li, X. Zhang, K. L. Yeung, A Novel Delayed Wakeup Scheme for Efficient Power Management in Infrastructure-based IEEE 802.11 WLANs, *2015 IEEE Wireless Communications and Networking Conference (WCNC)*, New Orleans, LA, 2015, pp. 1338-1343.

[8] W. H. Jung, S. H. Rhee, Adaptive Power Saving Algorithm for 802.11 Mobile Access Points, *2017 Ninth International Conference on Ubiquitous and Future Networks (ICUFN)*, Milan, 2017, pp. 800-802.

[9] IEEE 802.11 Std-2012 - IEEE Standard for Information Technology Telecommunications and Information Exchange between Systems Local and Metropolitan Area Networks - Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, March, 2012.

[10] R. Krashinsky, H. Balakrishnan, Minimizing Energy for Wireless Web Access with Bounded Slowdown, *Wireless Networks*, Vol. 11, No. 1-2, pp. 135-148, January, 2005.

[11] D. Qiao, K. G. Shin, Smart Power-Saving Mode for IEEE 802.11 Wireless LANs, *Proceedings of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2005)*, Miami, FL, 2005, pp. 1573-1583.

[12] E. Tan, L. Guo, S. Chen, X. Zhang, PSM-Throttling: Minimizing Energy Consumption for Bulk Data Communications in WLANs, *IEEE International Conference on Network Protocols*, Beijing, China, 2007, pp. 123-132.

[13] G. Anastasi, M. Conti, E. Gregori, A. Passarella, 802.11 Power-Saving Mode for Mobile Computing in Wi-Fi Hotspots: Limitations, Enhancements and Open Issues, *Wireless Networks*, Vol. 14, No. 6, pp. 745-768, December, 2008.

[14] Y. Song, B. Ciubotaru, G. M. Muntean, Q-PASTE: A Cross-Layer Power Saving Solution for Wireless Data Transmission, *IEEE International Conference on Communications Workshops (ICC)*, Budapest, Hungary, 2013, pp. 441-446.

[15] R. Palit, K. Naik, A. Singh, Impact of Packet Aggregation on Energy Consumption in Smartphones, *2011 7th International Wireless Communications and Mobile Computing Conference (IWCMC)*, Istanbul, Turkey, 2011, pp. 589-594.

[16] Y. H. Zhu, H. C. Lu, V. C. M. Leung, Access Point Buffer Management for Power Saving in IEEE 802.11 WLANs, *IEEE Transactions on Network and Service Management*, Vol. 9, No. 4, pp. 473-486, December, 2012.

[17] J. Manweiler, R. R. Choudhury, Avoiding the Rush Hours: Wi-Fi Energy Management via Traffic Isolation, *IEEE Transactions on Mobile Computing*, Vol. 11, No. 5, pp. 739-752, May, 2012.

[18] J. W. Lin, J. M. Chang, Joint Design of Delay Constraint and Energy Saving for Efficient Packet Transmissions in Smartphones, *Proceedings of International Conference on Advances in Computer and Electronics Technology*, Hong Kong, China, 2013, pp. 1-5.

[19] S. Dasgupta, C. H. Papadimitriou, U. V. Vaziran, *Algorithms*, McGraw-Hill, 2006.

[20] K. Naik, *A Survey of Software Based Energy Saving Methodologies for Handheld Wireless Communication Devices*, Tech. Report No. 2010-13 Dept. of ECE, University of Waterloo, October, 2010.

[21] Linux Scheduler, http://oakbytes.wordpress.com/linux-scheduler, 2014.

[22] Q. Ni, L. Romdhani, T. Turletti, A Survey of QoS Enhancements for IEEE 802.11 Wireless LAN, *Wireless Communications & Mobile Computing*, Vol. 4, No. 5, pp. 547-566, August, 2004.

[23] D. J. Deng, H. C. Yen, Quality-of-Service Provisioning System for Multimedia Transmission in IEEE 802.11 Wireless LANs, *IEEE Journal on Selected Areas in Communications*, Vol. 23, No. 6, pp. 1240-1252, June, 2005.

[24] T. D. Lagkas, G. I. Papadimitriou, P. Nicopolitidis, A. S. Pomportsis, Priority-Oriented Adaptive Control with QoS Guarantee for Wireless LANs, *IEEE Transactions on Vehicular Technology*, Vol. 56, No. 4, pp. 1761-1772, July, 2007.

[25] X. Yu, P. Navaratnam, K. Moessner, Performance Analysis of Distributed Resource Reservation in IEEE 802.11e-based Wireless Networks, *IET Communications*, Vol. 6, No. 11, pp. 1447-1455, July, 2012.

[26] M. Grinstead, J. L. Snell, *Introduction to Probability*, 2nd ed. Athena Scientific, 2008.

[27] D. Gross, C. M. Harris, *Fundamentals of Queuing Theory*, 3rd Ed. New York, NY: Wiley, 1998.

[28] MathWorks - MATLAB and Simulink for Technical Computing, http://www.mathworks.com, 2014.

[29] D. Chuck, J. M. Chang, Bandwidth Recycling in IEEE 802.16 Networks, *IEEE Transactions on Mobile Computing*, Vol. 9, No. 10, pp. 1451-1464, October, 2010.

[30] X. Yi, L. Xiapu, R. K. C. Chang, Centralized PSM: An AP-Centric Power Saving Mode for 802.11 Infrastructure Networks, *2009 IEEE Sarnoff Symposium*, Princeton, NJ, 2009, pp. 1-5.

[31] N. Warty, R. Sheshadri, W. Zheng, D. Koutsonikolas, A First Look at 802.11n Power Consumption in Smartphones, *Proceedings of the First ACM International Workshop on Practical Issues and Applications in Next Generation Wireless Networks*, Istanbul, Turkey, 2012, pp. 27-32.

Biographies

Jenn-Wei Lin is a full professor in the Department of Computer Science and Information Engineering, Fu Jen Catholic University, Taiwan. He received the Ph.D. degree in electrical engineering from National Taiwan University, Taiwan, in 1999. His research interests are cloud computing, mobile computing and networks, distributed systems, and fault-tolerant computing.

Chi-Yi Lin is an associate professor in the Department of Computer Science and Information Engineering at Tamkang University, Taiwan. He received his B.S. and Ph.D. degrees in electrical engineering from National Taiwan University in 1995 and 2003, respectively. His research interests include cloud computing, ubiquitous computing, and Internet of Things.