IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received April 16, 2020, accepted May 16, 2020, date of publication May 20, 2020, date of current version June 5, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2995847

Queueing Analysis of Dynamic Power Management Schemes for Mobile Devices

SUNG-HWA LIM[1], (Member, IEEE), SE WON LEE[2], (Member, IEEE), MYE SOHN[3], (Member, IEEE), AND BYOUNG-HOON LEE[4]

[1]Department of Multimedia, Namseoul University, Cheonan 31020, South Korea
[2]Division of Business Administration, Pukyong National University, Busan 48513, South Korea
[3]Department of Systems Management Engineering, Sungkyunkwan University, Suwon 16419, South Korea
[4]National Program of Excellence in Software, Chungbuk National University, Cheongju 28644, South Korea

Corresponding author: Se Won Lee (swlee@pknu.ac.kr)

This work was supported in part by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT and Future Planning under Grant NRF-2017R1E1A1A03070926, and in part by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology under Grant NRF-2019R1A2C1004102.

ABSTRACT In this paper, we analyze a queueing system under the dyadic server control, workload control and delivery deadlines of data blocks. From the viewpoint of efficient power management for the wireless interface of a smart mobile device, we first focus on deriving the mean length of an arbitrary cycle time and then investigate the mean energy expenditure during the cycle. Some numerical examples are shown with respect to several control parameters. We conduct simulations to evaluate the proposed analytic energy consumption models. Experiments are also performed by implementing a test program on an off-the-shelf smartphone. Our results can serve as quantitative guidelines for the efficient power management of wireless interfaces in mobile devices.

INDEX TERMS Dynamic power management, energy consumption, queueing approach, server control policy, mobile devices.

I. INTRODUCTION

A. BACKGROUND AND MOTIVATION

Smart mobile devices play important roles in daily life because they are the dominant media through which users connect Internet services [1]. In this field, battery power management has always been a major concern because it is one of the most crucial resources for mobile devices [2]. However, the power needs of high-end smart mobile devices cannot be met with the performance of modern batteries, and the discrepancy is increasing every year [3].

To satisfy the higher power efficiency of mobile devices, many hardware-level or firmware-level solutions have been proposed [4], [5]. Most of these works tried to reduce the power consumption of individual hardware modules [6]–[8]. However, the wasteful use of high-power hardware modules by software applications may vastly increase the power consumption of a mobile device [9], [10]. Therefore, there is a great demand for applications that can efficiently support power management by restricting the reckless use of high power modules.

In particular, efficient power managements is required for the wireless interface (WI), which is one of the most power-consuming modules in a mobile device [11].[1] To efficiently manage the power of WI modules, dynamic power management (DPM) is widely adopted. According to the operation state of the WI, DPM systems support multiple power modes, such as active modes (e.g., transmit, receive, and idle modes) and inactive modes (e.g., sleep mode and powered off) [12]. Generally, an active-mode WI consumes much more power than an inactive-mode WI, even when in idle mode. Switching the WI to an inactive mode whenever it does not have any data to send or receive is an essential scheme to reduce its energy expenditure. Therefore, most DPM systems fundamentally employ a timeout-based shutdown policy.

However, the required overhead energy and boot/cooldown time for turning on/off the WI are not negligible [13]. Therefore, there is a tradeoff between the wakeup overhead incurred by turning the WI on/off frequently and the power conserved by turning the WI off when in idle mode, which

The associate editor coordinating the review of this manuscript and approving it for publication was Yi Zhang.

[1]It is reported that the WI can account for approximately 25% of the total power budget of a smartphone [14].

is also an active mode, for some amount of time. To achieve this tradeoff, multiple sleep modes are employed in most up-to-date DPM systems [15], [16].[2]

There is a DPM scheme that provides two sleep modes: deep sleep and light sleep. In other words, in addition to a deep sleep mode, in which power consumption is very low the consuming power is very low but the wakeup overhead is high, there is a light sleep mode, in which power consumption is relatively high but the wakeup overhead is lower than that in the deep sleep mode.[3]

To reduce the wakeup overhead in a DPM scheme, we need to restrict active/inactive mode transitions efficiently. A delayed wakeup method can be a simple solution. In the suspended wakeup method, the WI will awaken only after waiting some amount of time following a transmit request instead of being activated immediately. Then, any forthcoming transmit requests submitted during the suspended time will be transmitted all at once during the next incoming WI activation. However, the method described above has two main problems. One is the overflow of the transmission queue due to increased amounts of pending data for transmission; the other is the tediously protracted delay in transmitting data.

In previous studies conducted by our research group [17], [18], a carpool-based suspended wakeup scheme was proposed that turns on the WI when either of the following two events occurs: (a) the total size of the pending data blocks exceeds the predefined workload threshold, or (b) the waiting time of any data blocks queued for the transmission exceeds the predefined time threshold (i.e., the deadline). After being activated once, the WI transmits all the pending data blocks until no data blocks are left in the transmission queue. These previously studies empirically verified that this carpool-based scheme consumes less power than the legacy DPM system and does not violate spatial or temporal constraints. However, analytical results were not provided for the scheme.

There are two main reasons that it is highly desirable to build analytic models of power consumption for power management schemes such as the legacy DPM, enhanced DPM, and carpool-based suspended wakeup schemes The first reason is that analytic models can present quantitative guidelines for the efficient power management of the WI, for example, by predicting the expected energy consumption according to various environmental parameters. The second is that an analytic power consumption model can be exploited to determine the optimal solution for various system parameter settings.

B. ANALYTICAL STUDIES ON ENERGY CONSUMPTION MODELS

Several studies have been performed on the analytic energy consumption models. Okamura and Dohi [19] proposed an analytical method to find out the optimal timeout interval to minimize energy consumption. They employed the legacy DPM method for general embedded systems (such as the hard disk drive or CPU) rather than the WI. The work in [19] considered the MAP/G/1 queueing model with vacations. Jiang *et al.* [20] developed an analytic energy consumption model considering the legacy DMP for the WI module using semi-Markovian control processes and proposed a solution to determine the optimal timeout interval by utilizing the developed energy consumption model. Luiz *et al.* [21] found the optimal timeout interval and the optimal threshold data rate for the WI under the DPM method with a timeout-based shutdown policy. They proposed an energy model to maximize the performance (i.e., response time) and minimize energy consumption; however, they did not analyze the multiple sleep modes or the carpool-based suspended wakeup scheme. Moreover, the power variations among various operating modes (i.e., Tx mode, Rx mode, and idle mode) and real-time data with delivery deadlines were not investigated.

Most previous studies focused on the optimal timeout interval to reduce energy consumption under the legacy DPM method. However, enhanced schemes such as the DPM with multiple sleep modes (i.e., the enhanced DPM) and the carpool-based suspended wakeup scheme should also be analyzed. Real-time data, which have delivery deadlines, should also be considered because most smartphone applications are subject to real-time constraints.

C. CONTRIBUTIONS

We present mathematical energy consumption models according to dynamic power management techniques, including the legacy DPM scheme and the carpool-based suspended wakeup scheme established on top of the legacy DPM method. In the Appendix, we also present analytic models of the enhanced DPM and carpool-based suspended wakeup scheme established on top of the enhanced DPM. The proposed analytic models further consider real-time characteristics (i.e., the delivery deadline for a data block queued for transmission).

The operational behavior of our system can be modeled with a queueing system that has the dyadic server control (i.e., the D-policy and deadline policy), activation/deactivation delays, and idle timeout periods. Some queueing studies have addressed the D-policy, but no study has simultaneously considered the above operational behaviors. For more details on the analysis of D-policy queues and dyadic policies, readers may reference [23]–[25]. The preliminary version of this study has been presented as a conference paper [22].

This paper features the following major contributions: 1) We modeled a complex queuing system that has the dyadic server control for the carpool-based suspended wakeup scheme, which was presented in our previous study [17]. 2) We derived the expected energy expenditure for delivering data blocks by exploiting the presented queueing system. 3) We provided numerical examples by varying several system parameters.

[2]Currently, most WI modules usually provide at least two kinds of sleep modes.

[3]In this paper, we call the DPM scheme with two sleep modes the enhanced DPM.

Our theoretical results may support the design for an energy optimization scheme under the operating characteristics of the complex queueing system, and our findings will be exploited in quantitative guidelines for the efficient power management of WIs.

The remainder of this paper is organized as follows. Section 2 presents our system model, and we analyze the queueing system for the carpool-based scheme in Section 3. The analysis of the power consumption is provided in Section 4, in which we derive the expectation of energy expenditure per data delivery. We conduct a performance evaluation in Section 5, including illustrations of several numerical examples, simulation results, and experimental measurements by implementations, and the conclusions are presented in Section 6. In the Appendix, we analyzed the enhanced DPM scheme for efficient power management.

II. SYSTEM MODEL

In this paper, we employ the system model presented in [17]. A mobile device embeds one WI and connects to the Internet through the nearest wireless access point. The WI employs the legacy DPM method, in which the WI operates in either active mode(including transmit mode, receive mode, and idle mode) or inactive mode.

Fig. 1 shows a diagram of the dynamic power modes and their transitions. The legacy DPM scheme employs the timeout-based shutdown policy, in which an activated WI is autonomously deactivated if it continues to be in idle mode for longer than the predefined time (i.e., the idle timeout value). The deactivated WI is immediately activated (i.e., awoken from sleep mode) by an event such as a Tx or Rx request. Activating or deactivating the WI requires some delay time while consuming the maximum power. We assume that the activation/deactivation delay time is constant.

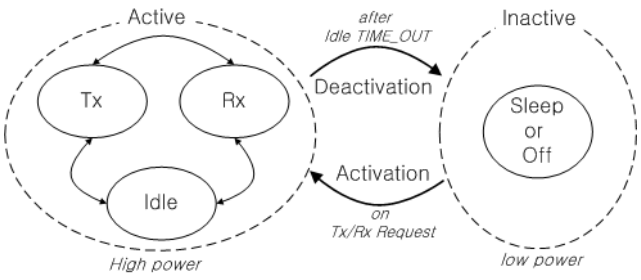


FIGURE 1. Dynamic power modes in the legacy DPM scheme.

To reduce the wakeup overhead, a carpool-based suspended wakeup scheme was proposed in our previous study [17], in which the deactivated WI wakes up only if one of the predefined trigger conditions is met, instead of being activated immediately. Then, any forthcoming transmit requests submitted during the suspended period will be transmitted all at once during the next incoming WI activation. The wakeup triggering conditions are as follows: 1. The delivery deadline of any pending Tx data blocks in the Tx queue will be missed. 2. The total size of pending Tx data blocks in the Tx queue will cause an overflow. 3. Any other events trigger the activation of the WI. Fig. 2 shows a flow diagram of the system operation of the carpool-based suspended wakeup scheme.

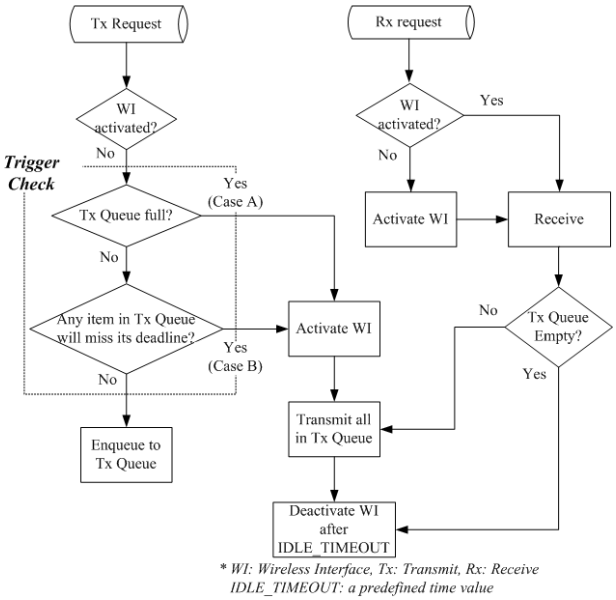


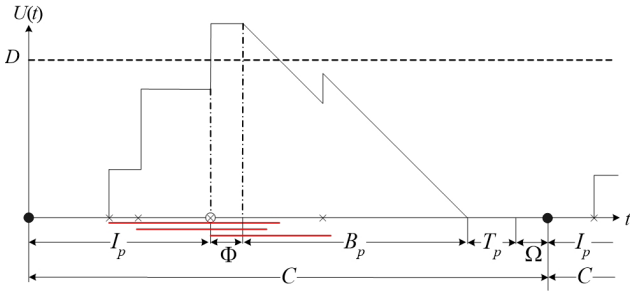
FIGURE 2. Carpool-based suspended wakeup scheme.

For clarity, let us state some important concepts regarding the terms in the general queueing literature. Hereafter, we refer to the WI, data blocks, and the size of a data block as the server, customers, and service time of a customer, respectively. We assume that the arrival process of a customer is a Poisson process with the rate λ, the service time follows an exponential distribution with the parameter μ, and the delivery deadlines of data blocks are exponentially distributed with the parameter η. The notations for the analysis are summarized in Table 1. In Table 1, the time unit is second.

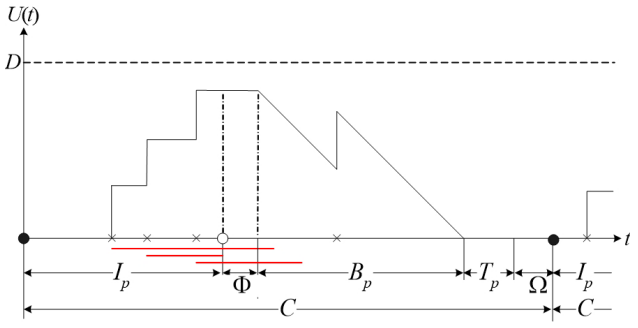
TABLE 1. Notations and functions.

Notation	Definition
Φ	predefined time length of an activation delay
Ω	predefined time length of a deactivation delay
T	predefined time length of an idle timeout
I_p	time length of an idle period
B_p	time length of a busy period
T_p	time length of an idle timeout period ($T_p \geq T$)
C	time length of a cycle
$N_{A(B)}$	number of customers at the start of the busy period in Case $A(B)$
S	service time of an arbitrary customer
S_n	nth convolution of service time S, $\sum_{i=1}^{n} S_i$
$F(x)$	distribution function of S, $Pr(S \leq x)$
$F_n(x)$	distribution function of S_n, $Pr(S_n \leq x)$
$E(\cdot)$	expectation of a random variable
$E(\cdot\|Z)$	conditional expectation given the event Z

Fig. 3 shows two possible cases of an arbitrary cycle. The horizontal red bars indicate the given deadlines of each customer. The idle server waits until one of the two conditions is met and is then activated. The two cases are described as follows. Case *A* shows that the idle server is activated because the accumulated workload during an idle period (i.e. the total sum of the service times of the three customers in Fig. 2) exceeds the predetermined workload threshold D before the deadline is missed. Case *B* demonstrates that the idle server is activated because the deadline of a waiting customer is passed before the accumulated workload exceeds D.



(a) Case A: Activated by D-policy



(b) Case B: Activated by delivery deadline a customer

FIGURE 3. Two possible cases of an arbitrary cycle.

The system operates in the following succession regardless of cases *A* and *B*: idle period, activation delay, busy period, idle timeout period, and deactivation delay. Therefore, we define a cycle as the length from the start of an arbitrary idle period to the end of the deactivation delay. We also assume that if a customer arrives during the idle timeout period, the server starts its service immediately.

III. ANALYSIS OF THE QUEUEING SYSTEM

In this section, we focus on deriving the mean length of an arbitrary cycle $E(C)$. As mentioned before, a cycle consists of the same elements in the same order regardless of cases *A* and *B* (see Fig. 2). Therefore,

$$E(C)=E(I_p)+\Phi+E(B_p)+E(T_p)+\Omega. \tag{1}$$

We assume that the activation delay (Φ) and the deactivation delay (Ω) are both constant with separate values. Therefore, we need to find the expected lengths of the idle period, the busy period, and the idle timeout period.

The expected lengths of the idle period and the busy period are, by the total expectation theorem, as follows:

$$E(I_p) = E(I_p|A)Pr(A) + E(I_p|B)Pr(B). \tag{2}$$
$$E(B_p) = E(B_p|A)Pr(A) + E(B_p|B)Pr(B). \tag{3}$$

In (2) and (3), the probability that an arbitrary cycle is Case *A* or Case *B* can be obtained by Theorem 1.

Theorem 1: *In an M/M/1 queueing system under the D-policy and deadline policy, the probabilities of Case A and Case B are as follows*:

$$Pr(A) = e^{-\mu D}\sum_{n=1}^{\infty}\frac{(\mu D)^{n-1}}{(n-1)!}\prod_{i=1}^{n-1}\frac{\lambda}{\lambda+i\eta}, \tag{4}$$
$$Pr(B) = 1 - Pr(A). \tag{5}$$

Proof: In Case *A*, the busy period always starts because the workload exceeds the threshold D (and the deadlines of every waiting customers are not missed). During an idle period, if the service time of the first customer is greater than, the server immediately starts its busy period with a probability of $e^{-\mu D}$. To start a busy period with two waiting customers during an idle period, the first customer's service time should be less than or equal to D, while the second customer's service time should be greater than $D-x$, and the second customer must arrive before the deadline of the first customer:

$$\frac{\lambda}{\lambda+\eta}(F_1(D)-F_2(D)) = \frac{\lambda}{\lambda+\eta}\mu D e^{-\mu D}.$$

To start a busy period with n waiting customers during an idle period, the sum of the foremost $n-1$ customers' service times (x) should be less than or equal to D, the nth customer's service time should be greater than $D-x$, and the customer must arrive before any of the deadlines of $(n-1)$ waiting customers are missed. This probability is expressed as

$$\prod_{i=1}^{n-1}\frac{\lambda}{\lambda+i\eta}(F_{n-1}(D)-F_n(D)) = \prod_{i=1}^{n-1}\frac{\lambda}{\lambda+i\eta}\frac{(\mu D)^{n-1}}{(n-1)!}e^{-\mu D}.$$

By considering all possible cases,

$$\begin{aligned} Pr(A) &= \sum_{n=1}^{\infty} Pr(N_A = n) \\ &= e^{-\mu D}\left[1+\sum_{n=2}^{\infty}\frac{(\mu D)^{n-1}}{(n-1)!}\prod_{i=1}^{n-1}\frac{\lambda}{\lambda+i\eta}\right]. \end{aligned}$$

After simplification, we obtain (4). Because events *A* and *B* are mutually exclusive, we have (5). □

Corollary 1: *The probability that there are n customers at the beginning of the busy period for Case A and Case B is*

$$Pr(N_A = n) = \frac{(\mu D)^{n-1}e^{-\mu D}}{(n-1)!}\prod_{i=1}^{n-1}\frac{\lambda}{\lambda+i\eta}, \tag{6}$$

$$Pr(N_B = n) = \left(\prod_{i=1}^{n-1}\frac{\lambda}{\lambda+i\eta}\right)\left(\frac{n\eta}{\lambda+n\eta}\right)\int_0^D \frac{\mu^n y^{n-1}e^{-\mu y}}{(n-1)!}dy. \tag{7}$$

Proof: From the proof of Theorem 1, $Pr(N_A = n)$ is obviously the expression in (6).

In Case B, the idle server is always activated by passing the deadline of a waiting customer before the accumulated workload exceeds D. To start a busy period with the first customer, the service time of the first customer must be less than or equal to D, and his/her deadline should be set before the second customer arrives. This probability is $(1 - e^{-\mu D})\frac{\eta}{\lambda+\eta}$.

To start a busy period with two waiting customers during the idle period in Case B, the sum of the first and second customer's service time should be less than or equal to D, and one of the deadlines of these two customers should be set before the third customer arrives. This probability can be derived as follows:

$$\begin{aligned} &Pr(N_B = 2) \\ &= \frac{\lambda}{\lambda+\eta}\left(\frac{2\eta}{\lambda+2\eta}\right)\int_0^D \left(\mu^2 y e^{-\mu y}\right) dy \\ &= \frac{\lambda}{\lambda+\eta}\left(\frac{2\eta}{\lambda+2\eta}\right)\left(1 - e^{-\mu D}(1+\mu D)\right). \end{aligned}$$

By considering the event that there are n customers at the start of the busy period, we can derive

$$\begin{aligned} &Pr(N_B = n) \\ &= \left(\prod_{i=1}^{n-1}\frac{\lambda}{\lambda+i\eta}\right)\left(\frac{n\eta}{\lambda+n\eta}\right)\int_0^D \frac{\mu^n y^{n-1} e^{-\mu y}}{(n-1)!} dy. \end{aligned}$$

After simplification, we have (7). □

Note that (5) can also be calculated by using (7) with $Pr(B) = \sum_{n=1}^{\infty} Pr(N_B = n)$.

Corollary 2: *The conditional expected lengths of the idle period for Case A and Case B are*

$$E(I_p|A) = \frac{\sum_{n=1}^{\infty}\sum_{i=1}^{n}\frac{1}{\lambda+(i-1)\eta}Pr(N_A = n)}{Pr(A)}, \quad (8)$$

$$E(I_p|B) = \frac{\sum_{n=1}^{\infty}\sum_{i=0}^{n}\frac{1}{\lambda+i\eta}Pr(N_B = n)}{Pr(B)}. \quad (9)$$

Proof: Let us consider Case A with n customers at the beginning of the activation delay. In this case, the idle period is the sum of the interarrival times of every customer who arrives when the server is idle (see Fig. 2(a)). Therefore, by using the memoryless property of the exponential distribution, the conditional expected time length of the idle period is the sum of the following:

1) Interarrival time of the first customer $1/\lambda$,
2) Interarrival time of the second customer $1/(\lambda+\eta)$,

$\vdots$

3) Interarrival time of the nth customer $1/(\lambda+(n-1)\eta)$.

Note that the expected minimum length of two or more exponential distributions is a reciprocal of the sum of parameters.

In Case B, the idle period is slightly but definitely different from that in Case A. In Case B, the mean length of the idle period is the sum of n customers' interarrival times and the length of the exponential distribution with the parameter $\lambda+n\eta$. This is because the server activates when the deadline of any customer is missed before the sum of their service times exceeds D (see Fig. 2(b)). By the definition of conditional expectation, we finally have (8) and (9). □

From (8) and (9), we can rewrite (2) as follows:

$$E(I_p) = \sum_{n=1}^{\infty}\left[\sum_{i=1}^{n}\frac{Pr(N_A = n)}{\lambda+(i-1)\eta} + \sum_{i=0}^{n}\frac{Pr(N_B = n)}{\lambda+i\eta}\right]$$

An arbitrary busy period is a delay cycle with an initial delay of the accumulated service time during an idle period. The initial delays for Case A and Case B are $E(N_A)E(S)$ and $E(N_B)E(S)$, respectively [25], [26]. In this paper, we assume that the service time distribution is an exponential distribution with the rate μ, $E(S) = 1/\mu$. Therefore, the expected length of the busy period in each case is

$$\begin{aligned} E(B_p|A) &= \frac{E(S)\sum_{n=1}^{\infty} nPr(N_A = n)}{Pr(A)(1-\lambda E(S))} \\ &= \frac{E(N_A)}{Pr(A)(\mu-\lambda)}, \end{aligned} \quad (10)$$

$$\begin{aligned} E(B_p|B) &= \frac{E(S)\sum_{n=1}^{\infty} nPr(N_B = n)}{Pr(B)(1-\lambda E(S))} \\ &= \frac{E(N_B)}{Pr(B)(\mu-\lambda)}. \end{aligned} \quad (11)$$

From (10) and (11), (3) becomes

$$E(B_p) = \frac{E(N_A) + E(N_B)}{\mu - \lambda}.$$

To complete (1), we now need to determine the expected length of the idle timeout period T_p because the length of this period is not a constant but rather a random variable. Fig. 4 shows a sample path of the workload during the idle timeout period. After a busy period ends, the server has an idle timeout period with a time length of T. If a customer arrives during T, the server immediately starts its service until there is no customer in the system; this process is exactly the same as the busy period of an ordinary $M/M/1$ queue. If no customer arrives during T, the server deactivates.

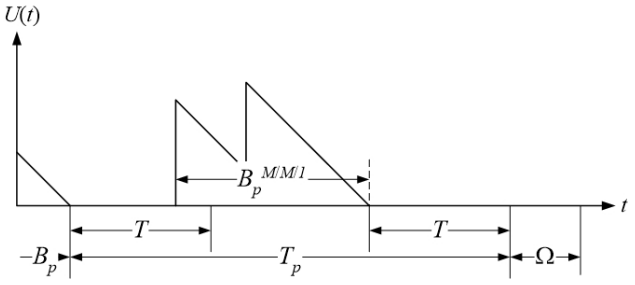


FIGURE 4. Sample path of the idle timeout period.

The expected length of the idle timeout period can be obtained from Theorem 2.

Theorem 2: *The expected length of the idle timeout period for an M/M/1 queue is as follows*:

$$E(T_p) = \frac{1 - e^{-\lambda T}}{(1 - \lambda E(S))\lambda e^{-\lambda T}} = \frac{(e^{\lambda T} - 1)\mu}{\lambda(\mu - \lambda)}. \quad (12)$$

Proof: Let us define Q_0 as the event in which no customer arrives during T and Q_0^c as the complementary event of Q_0. By conditioning the number of arrivals during T, we can obtain the mean length of the idle timeout period T_p. If no customer arrives during T, the conditional expectation of T_p is obviously T; otherwise, the conditional expectation is the sum of the following: (i) the interarrival time, which is less than T, (ii) the busy period of an ordinary M/M/1 queue ($B_p^{M/M/1}$ in Fig. 3), and (iii) the expected value of T_p. Therefore,

$$\begin{aligned} E(T_p) &= Pr(Q_0)E(T_p|Q_0) + Pr(Q_0^c)E(T_p|Q_0^c) \\ &= e^{-\lambda T}T + (1 - e^{-\lambda T})\left[\frac{1}{\lambda} - \frac{Te^{-\lambda T}}{1 - e^{-\lambda T}}\right. \\ &\quad \left. + \frac{E(S)}{1 - \lambda E(S)} + E(T_p)\right]. \end{aligned}$$

After simplification, we have (12). □

IV. ANALYSIS OF POWER CONSUMPTION

In this section, we derive the amount of energy expended per cycle $\mathbf{\Gamma}$ and the mean time length for data delivery per cycle U.

Let us define γ_t as the energy expenditure (J, joules) per unit time during a time period t, and generally, $\gamma_\Phi > \gamma_\Omega = \gamma_{B_p} > \gamma_T \gg \gamma_{I_p}$. From Corollary 2 and Theorem 2,

$$\mathbf{\Gamma} = E(I_p)\gamma_{I_p} + \Phi\gamma_\Phi + E(B_p)\gamma_{B_p} + \gamma(T_p) + \Omega\gamma_\Omega. \quad (13)$$

Note that the fourth term on the right-hand side of (13) is not simply $E(T_p)\gamma_{T_p}$ because T_p is composed of some time lengths that need different types of energy expenditure.

The energy expenditure during an idle timeout period $\gamma(T_p)$ can be obtained by the same argument in Theorem 2 as follows:

$$\begin{aligned} \gamma(T_p) &= e^{-\lambda T}T\gamma_T + (1 - e^{-\lambda T}) \\ &\quad \times \left[\left(\frac{1}{\lambda} - \frac{Te^{-\lambda T}}{1 - e^{-\lambda T}}\right)\gamma_T \right. \\ &\quad \left. + \frac{E(S)}{1 - \lambda E(S)} \cdot \gamma_{B_p} + \gamma(T_p)\right]. \end{aligned}$$

After simplification, we have

$$\gamma(T_p) = (e^{\lambda T} - 1)\left(\frac{\gamma_T}{\lambda} + \frac{\gamma_{TX}}{\mu - \lambda}\right). \quad (14)$$

Let us denote T_p^U as the data delivery time during the idle timeout period within the cycle; then T_p^U can be obtained as follows:

$$\begin{aligned} T_p^U &= (1 - e^{-\lambda T})\left[\frac{E(S)}{1 - \lambda E(S)} + T_p^U\right] \\ &= \frac{e^{\lambda T} - 1}{\mu - \lambda}. \end{aligned} \quad (15)$$

Because U is the sum of the expected length of the busy period and T_p^U,

$$\begin{aligned} U &= E(B_p) + T_p^U \\ &= \frac{E(N_A) + E(N_B) + e^{\lambda T} - 1}{\mu - \lambda}. \end{aligned} \quad (16)$$

We finally have the total energy expenditure per delivery of a data block[4] $\psi = \mathbf{\Gamma}/U$ from (13) and (16).

V. PERFORMANCE EVALUATION

A. ANALYTICAL RESULTS

In this section, we illustrate simple numerical examples and then discuss the computational results.

We assume that $\Phi = 2$ s, $\Omega = 1$ s, $\gamma_{I_p} = 0.1728$ mW, $\gamma_\Phi = \gamma_\Omega = 997.2$ mW, $\gamma_{B_p} = 997.2$ mW and $\gamma_T = 169.2$ mW, which were also employed in [17].

Fig. 5 shows multiple graphs of the energy expended per data delivery (i.e., ψ) for varying parameters under the above-mentioned conditions. In each graph in Fig. 5, the dashed line depicts the result of M/M/1 queue under the legacy DPM system, and the solid line shows the result of the carpool-based suspended wakeup scheme established on top of the legacy DPM system. We can see that the carpool-based suspended wake up scheme (i.e., the D-policy and deadline policy) is much more efficient than the legacy DPM policy from an energy conservation perspective.

In Fig. 5(a), we varied D while keeping $\lambda = 0.3$, $\mu = 0.5$, $\eta = 0.05$, and $T = 2$. As we expected, ψ decreases as D (i.e., the size of the Tx queue) increases. This is because an increase in D may allow more data blocks awaiting transmission to stay in the transmit queue before the WI is activated, which increases the effect of our carpool policy. Note that the slope gradually decreases as D increases, and almost becomes saturated when D is greater than 14.

In Fig. 5(b), we varied the delivery deadline while keeping $\lambda = 0.3$, $\mu = 0.5$, $T = 2$, and $D = 10$. ψ decreases as the delivery deadline of each data block increases. This is because an increase in deadline may allow the data blocks awaiting transmission to stay longer in the transmit queue. Fig. 5(a) and Fig. 5(b) show that ψ under the legacy DPM scheme is a constant because D and deadline cannot affect the system performance.

In Fig. 5(c), we varied λ while keeping $\mu = 0.5$, $\eta = 0.05$, $T = 2$, and $D = 10$. As λ increases, the energy expended per data delivery (i.e., ψ) decreases because the WI takes less time in its idle mode. In the idle mode, the WI is active; thus, the WI consumes a large amount of energy but does not transmit any data.

In Fig. 5(d), we varied T while keeping $\lambda = 0.3$, $\mu = 0.5$, $\eta = 0.05$, and $D = 10$. As T increases, power consumption decreases.

[4]For simplicity, the total energy expenditure per delivery of a data block will hereafter be interchangeably referred to as the energy expended per data delivery.

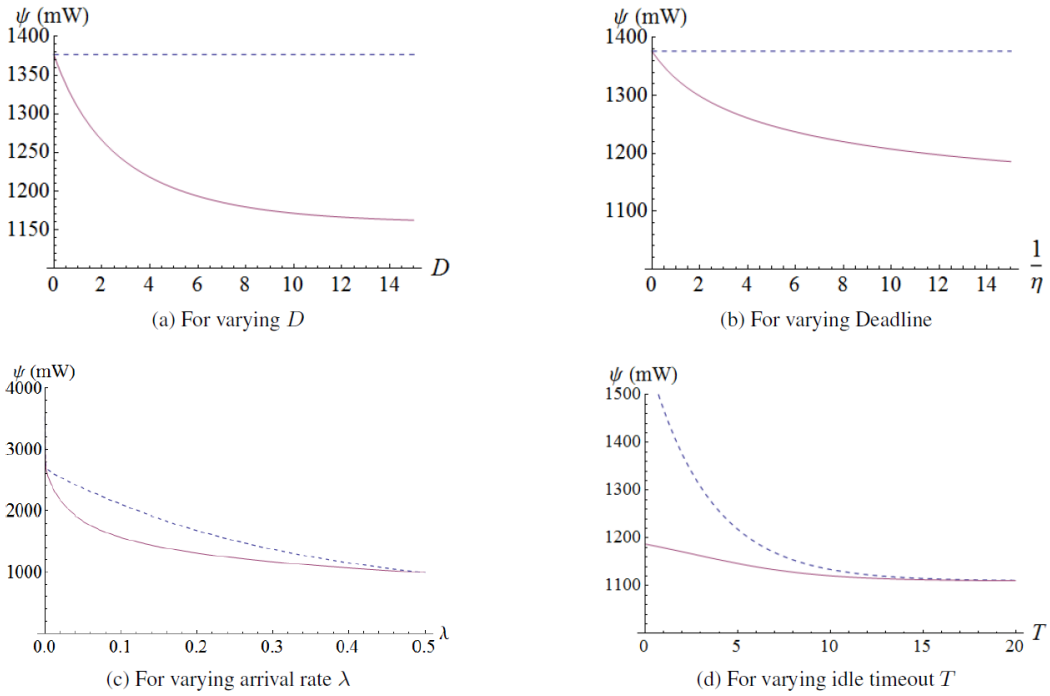


FIGURE 5. Energy expended per data delivery.

Fig. 6 shows the values of ψ for varying D values and mean deadline $1/\eta$ simultaneously.

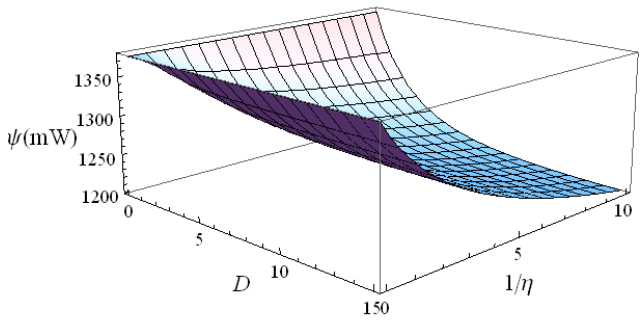


FIGURE 6. Energy expended for varying *D* values and deadlines.

Our system is a versatile M/M/1 queue; therefore, we can obtain the results for various systems as special cases by setting several system parameters as follows:

- M/M/1 queue (basic): $\Phi = \Omega = T = 0, D \to 0$ or $\eta \to \infty$
- M/M/1 under the D-policy: $\Phi = \Omega = T = 0, \eta \to 0$
- M/M/1 under the deadline policy: $\Phi = \Omega = T = 0, D \to \infty$
- M/M/1 with an idle timeout: $\Phi = \Omega = 0, D \to 0$ or $\eta \to \infty$
- M/M/1 under the legacy DPM system: $D \to 0$ or $\eta \to \infty$
- M/M/1 under the DPM and D-policy: $\eta \to 0$
- M/M/1 under DPM and deadline policy: $D \to \infty$

Although the analysis of this system is based on an M/M/1 queue, extra complexities are involved in calculating some measures in (4),(8),(9),(10), and (11). Therefore, we tried to provide refined mathematical results for rapid calculations. This is one of the main contributions of this paper.

B. SIMULATIONS

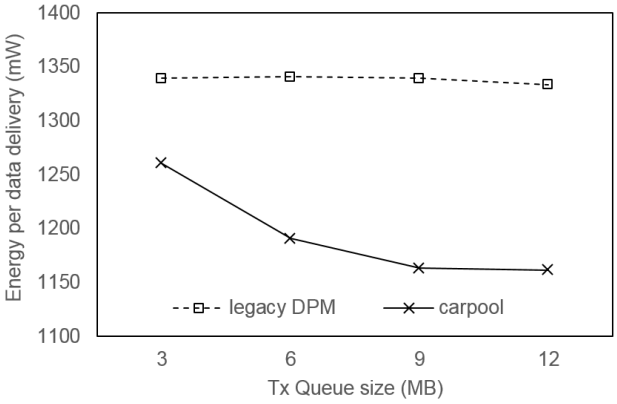
For a practical performance evaluation, we conducted simulations with practical power consumption parameters, by referring to the data sheet of an off-the-shelf Wi-Fi module as shown in Table 2 [27]. For the simulation, the same assumptions and system parameters used in the Analytical Results section are employed. We assume that the WI's transmission speed is 11 Mbps, the wakeup time (i.e., the activation delay) is 2.0 s, the deactivation delay is 1.0 s, and the idle timeout T is 2.0 s. The size of a data block is set to 1.75 MB, and the data transmission requests occur as a Poisson process with a rate λ. The energy expended per data delivery is measured over the total simulation time of 3,600 s. *Legacy DPM* stands for the legacy dynamic power management scheme introduced in the System Model section, and *carpool* refers to the carpool-based suspended wakeup scheme illustrated in Fig. 2. We used MATLAB v.8.5 as a simulation tool, and each simulation was repeated 1,000 times and averaged.

Fig. 7(a) shows the average energy expended per data delivery while varying the Tx queue size when λ is 0.3 and the delivery deadline is 20 s. As we expected, the energy expended per data delivery in *carpool* decreases as the Tx queue size increases, which is similar to the analytical results

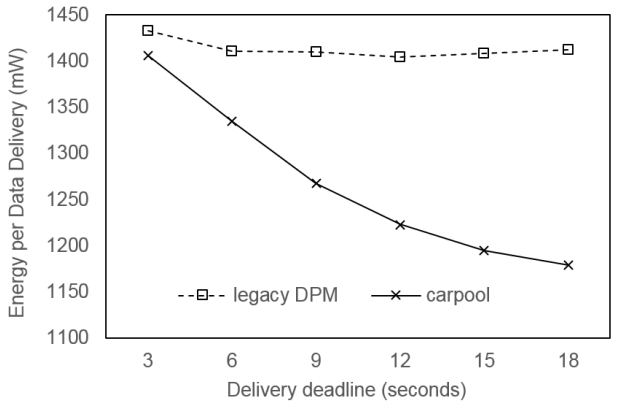
TABLE 2. Energy expenditure parameters.

Interface	Energy expenditure (mW)
Wi-Fi (2.4GHz)	Idle = 169.2, Sleep (i.e., inactive) = 0.17
	Rx = 219.6, Tx = 997.2

∗idle: *idle* mode, Rx: *receive* mode, Tx: *transmit* mode



(a) For varying the Tx queue size

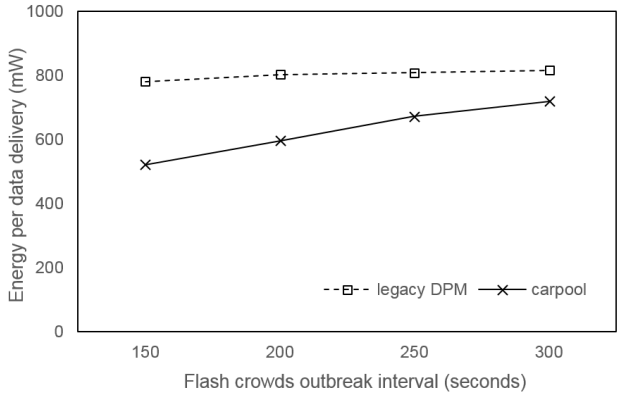


(b) For varying the delivery deadline

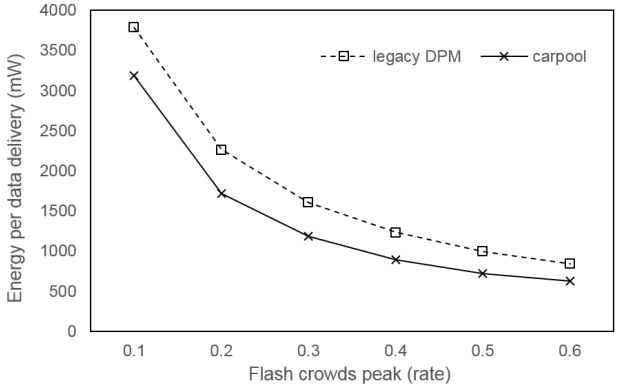
FIGURE 7. Simulation results of the energy expended per data delivery.

shown in Fig. 5(a). Fig. 7(b) shows the average energy expended per data delivery while varying the delivery deadline length when Tx queue size is 20MB, and λ is 0.3. As we expected, the energy expended per data delivery in *carpool* decreases as the delivery deadline length increases, which is similar to the analytical results shown in Fig. 5(b).

To show the results with a different traffic arrival model, we conducted simulations where the traffic occurs by following the flash crowds traffic model, which is suitable for emulating a burst of traffic [28], and we added the results to Fig. 8. Fig. 8(a) shows the average energy expended per data delivery while varying the flash crowds outbreak interval where the flash duration is 150 s, flash peak (i.e., maximum Tx request arrival rate) is 0.6, the maximum queue size is 100 MB, and the delivery deadline of each data block is set to 20 s. As flash crowds occur more often, *carpool* shows less energy expenditure than *legacy DPM*. Fig. 8(b) shows



(a) For varying the flash crowds outbreak interval



(b) For varying the flash crowds peak

FIGURE 8. Simulation results of the energy expended per data delivery on the flash crowds traffic model.

the average energy expended per data delivery while varying the flash crowds peak when the flash crowds interval is 200 s and the flash duration is 150 s. As the flash peak increases, *carpool* shows less energy expenditure than *legacy DPM*.

C. EXPERIMENTAL MEASUREMENTS

For the experimental measurements, we implemented a test program on a test bed, which consisted of an off-the-shelf smartphone, a Wi-Fi access point, and a server, as shown in Fig. 9. The smartphone was embedded with a 2.2 GHz 64-bit Snapdragon 820 quad-core processor, 4 GB of memory, an 802.11ac Wi-Fi module, and a 2,800 mAh battery, and the phone operated on the Android platform. The test application was developed in JAVA. To obtain more accurate results, we did not execute any applications beyond the test program and system-related processes, and we disabled all wireless interface modules of the smartphone other than Wi-Fi during the experiment.

The total energy expended during 1,800 s was measured as the percentage of the battery capacity of the smartphone for *legacy DPM* and *carpool*. The requests for data transmission occurred following a Poisson process with a rate λ. The experiments were repeated 10 times for each scenario, and the results were averaged.

FIGURE 9. Test bed for the experimental measurements.

Fig. 10 shows the total energy expended while varying the size of the Tx queue when the deadline of each data delivery is 30 s and λ is 0.8. As shown in the result, *carpool* incurs less energy expenditure than *legacy DPM*, and the energy expended decreases as the Tx queue size increases.

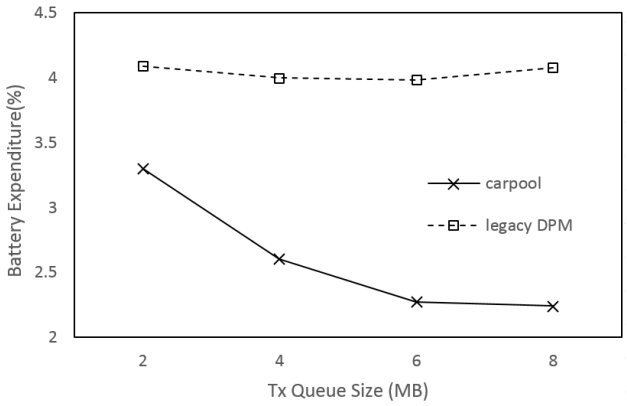


FIGURE 10. Experimental results for varying the Tx queue size.

VI. CONCLUSION

In this paper, queueing analyses for the legacy DPM, enhanced DPM, and carpool-based suspended wakeup schemes were proposed. For the efficient power management of the WI, we derived theoretical results that can support the power optimization for a complex queueing system that is operated under the dyadic server control, workload control and delivery deadlines of data blocks. We derived all the results in the exact and refined form with system parameters for practical uses and rapid calculations. The results derived can be exploited to provide quantitative guidelines for the efficient power management of the WI. To evaluate the proposed analytic energy consumption models, simulations and experiments on an off-the-shelf smartphone were also conducted. In some performance evaluations, it is shown that the carpool-based suspended wakeup scheme reduces the energy expended per data delivery by up to 30% compared to legacy schemes (i.e., the legacy DPM or enhanced DPM schemes).

APPENDIX

In this section, we introduce an enhanced DPM that provides two sleep modes and present an analytic power consumption model using queueing theory. We also present an analytic power consumption model of the carpool-based delayed wakeup scheme established on top of the enhanced DPM. To achieve a tradeoff between wakeup overhead and energy conservation, multiple sleep modes are introduced into the DPM scheme [15], [16].

Fig. 11 illustrates a diagram for the dynamic power modes and their transitions in the enhanced DPM with multiple sleep modes. The WI in an active mode (i.e., Tx, Rx or Idle modes) switches to the light_sleep mode if it continues to be in idle mode longer than light_idle_TIMEOUT. The mode transition from active mode to light_sleep mode requires a certain time (i.e., the light_deactivation delay), which is far smaller than the heavy_deactivation delay.

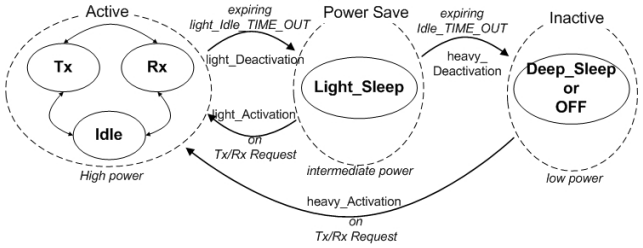


FIGURE 11. Dynamic power modes in the enhanced DPM.

We considered an extended idle timeout period with two phases. Fig. 12 illustrates a sample path of the idle timeout period in the enhanced DPM. From the same argument of Theorem 1 and Section IV, we can obtain (17)-(20) corresponding to (12), (14), (15), and (16).

$$\begin{aligned} E(T_p) &= e^{-\lambda T_1}\cdot e^{-\lambda T_2}\,(T_1+\Omega_1+T_2) \\ &\quad +(1-e^{-\lambda T_1})\left[\frac{1}{\lambda}-\frac{T_1 e^{-\lambda T_1}}{1-e^{-\lambda T_1}}\right. \\ &\quad \left.+\frac{E(S)}{1-\lambda E(S)}+E(T_p)\right] \\ &\quad +e^{-\lambda T_1}(1-e^{-\lambda T_2})\,[T_1+\Omega_1 \\ &\quad +\frac{1}{\lambda}-\frac{T_2 e^{-\lambda T_2}}{1-e^{-\lambda T_2}}+\frac{E(S)}{1-\lambda E(S)}+E(T_p)\Big] \\ &= \frac{\mu(e^{\lambda(T_1+T_2)}-1)}{\lambda(\mu-\lambda)}+e^{\lambda T_2}\Omega_1 \end{aligned} \tag{17}$$

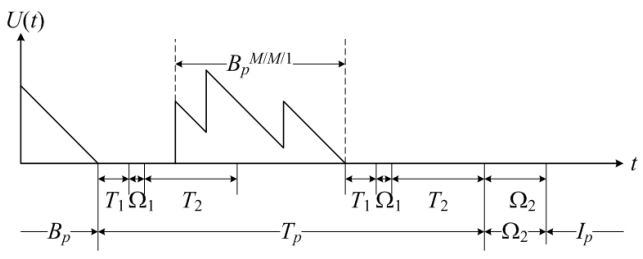
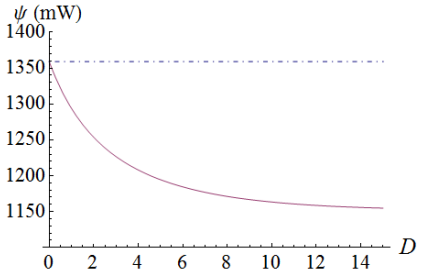
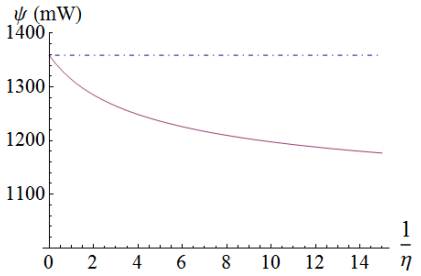


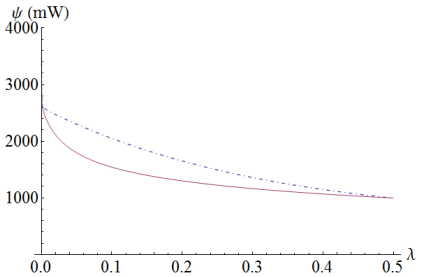
FIGURE 12. Sample path of the idle timeout period in the enhanced DPM.



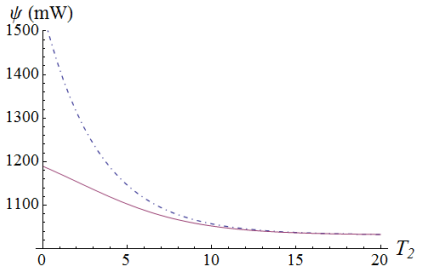
(a) For varying D



(b) For varying Deadline



(c) For varying arrival rate λ



(d) For varying idle timeout T_2

FIGURE 13. Energy expended per data delivery on the enhanced DPM system.

$$\begin{aligned}\gamma(T_p) &= e^{-\lambda T_1}\cdot e^{-\lambda T_2}(T_1\gamma_{T_1}+\Omega_1\gamma_{\Omega_1}+T_2\gamma_{T_2})\\ &\quad+(1-e^{-\lambda T_1})\times\left[\left(\frac{1}{\lambda}-\frac{T_1e^{-\lambda T_1}}{1-e^{-\lambda T_1}}\right)\gamma_{T_1}\right.\\ &\quad\left.+\frac{E(S)}{1-\lambda E(S)}\cdot\gamma_{B_p}+\gamma(T_p)\right]\\ &\quad+e^{-\lambda T_1}(1-e^{-\lambda T_2})\left[T_1\gamma_{T_1}+\Omega_1\gamma_{\Omega_1}\right.\\ &\quad+\left(\frac{1}{\lambda}-\frac{T_2e^{-\lambda T_2}}{1-e^{-\lambda T_2}}\right)\gamma_{T_2}\\ &\quad\left.+\frac{E(S)}{1-\lambda E(S)}\cdot\gamma_{B_p}+\gamma(T_p)\right]\\ &= \gamma_{T_1}\frac{(e^{\lambda T_1}-1)e^{\lambda T_2}}{\lambda}+\gamma_{B_p}\frac{e^{\lambda(T_1+T_2)}-1}{\mu-\lambda}\\ &\quad+\gamma_{T_2}\frac{e^{\lambda T_2}-1}{\lambda}+\gamma_{\Omega_1}e^{\lambda T_2}\Omega_1 \qquad (18)\end{aligned}$$

$$\begin{aligned}T_p^U &= (1-e^{-\lambda T_1}e^{-\lambda T_2})\left[\frac{E(S)}{1-\lambda E(S)}+T_p^U\right]\\ &= \frac{1-e^{-\lambda T_1}e^{-\lambda T_2}}{e^{-\lambda T_1}e^{-\lambda T_2}}\cdot\frac{E(S)}{1-\lambda E(S)}\\ &= \frac{e^{\lambda T_1}e^{\lambda T_2}-1}{\mu-\lambda}. \qquad (19)\end{aligned}$$

$$\begin{aligned}U &= E(B_p)+T_p^U\\ &= \frac{E(N_A)+E(N_B)+e^{\lambda(T_1+T_2)}-1}{\mu-\lambda}. \qquad (20)\end{aligned}$$

Fig. 13 shows multiple graphs of the energy expended per data delivery for varying parameters. $T_1 = 0.5$, $T_2 = 1.5$, and other parameters are the same as those for Fig. 5. The dashed line depicts the results for the enhanced DPM system, and the solid line shows the results for the carpool-based suspended wakeup scheme established on top of the enhanced DPM system. The results also show similar trends to those shown in Fig. 5.

The power expended in the light_sleep mode is lower than that in the active mode but higher than that in the deep_sleep mode. The WI in the light_sleep mode is immediately activated whenever any Tx/Rx event occurs, which requires a light_Activation delay. The light_Activation delay is far smaller than the deep_Activation delay. The WI in the light_sleep mode switches to the deep_sleep mode if it continues to be in idle mode longer than idle_TIMEOUT. The behavior in the deep_sleep mode is the same as that in the legacy DPM.

REFERENCES

[1] Canalys, "Smart phones overtake client PCs in 2011," Canalys, Singapore, Tech. Rep. 2012/021, Feb. 2012.

[2] D. H. Bui, Y. Liu, H. Kim, I. Shin, and F. Zhao, "Rethinking energy-performance trade-off in mobile Web page loading," in *Proc. 21st Conf. MOBICOM*, Paris, France, 2015, pp. 14–26.

[3] S. Udani and J. Smith, "Power management in mobile computing," Univ. Pennsylvania, Philadelphia, PA, USA, Tech. Rep. MS-CIS-98-26, Aug. 1996.

[4] L. Benini, A. Bogliolo, and G. D. Micheliy, "Dynamic power management of electronic systems," in *Proc. ICCAD*, San Jose, CA, USA, 1998, pp. 696–702.

[5] T. Simunic, L. Benini, A. Acquaviva, P. Glynn, and G. De Micheli, "Dynamic voltage scaling and power management for portable systems," in *Proc. DAC*, Las Vagas, NV, USA, 2001, pp. 524–529.

[6] S. Liu, K. Fan, and P. Shinha, "CMAC: An energy-efficient MAC layer protocol using convergent packet forwarding for wireless sensor networks," *ACM Trans. Sensor Netw.*, vol. 5, no. 4, pp. 29–34, 2009.

[7] T. S. Muthukaruppan, M. Pricopi, V. Venkataramani, T. Mitra, and S. Vishin, "Hierarchical power management for asymmetric multi-core in dark silicon era," in *Proc. DAC*, Austin, TX, USA, 2013, pp. 1–9.

[8] C.-Y. Li, C. Peng, S. Lu, and X. Wang, "Energy-based rate adaptation for 802.11n," in *Proc. MOBICOM*, Istanbul, Turkey, 2012, pp. 340–341.

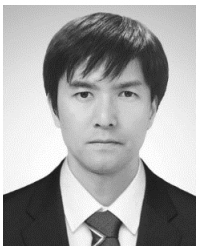
[9] C. Hwang, S. Pushp, C. Koh, J. Yoon, Y. Liu, S. Choi, and J. Song, "RAVEN: Perception-aware optimization of power consumption for mobile games," in *Proc. MOBICOM*, Snowbird, UT, USA, Oct. 2017, pp. 422–434.

[10] N. Peters, S. Park, D. Clifford, S. Kyostila, R. McIlroy, B. Meurer, H. Payer, and S. Chakraborty, "API for power-aware application design on mobile systems," in *Proc. MOBILESoft*, Gothenburf, Sweden, 2018, pp. 90–91.
[11] A. Carroll and G. Heiser, "An analysis of power consumption in a smartphone," in *Proc. USENIX*, Berkeley, CA, USA, 2010, pp. 1–21.
[12] V. Raghunathan, C. Schurgers, S. Park, and M. B. Srivastava, "Energy-aware wireless microsensor networks," *IEEE Signal Process. Mag.*, vol. 19, no. 2, pp. 40–50, Mar. 2002.
[13] *IEEE Standard—Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, IEEE Standard 802.11-2007, IEEE-SA Standards Board, 2007.
[14] F. Xia, C. Hsu, X. Liu, H. Liu, F. Ding, and W. Zhang, "The power of smartphones," *Multimedia Syst.*, vol. 21, no. 1, pp. 87–101, 2015.
[15] C. Hou and Q. Zhao, "A new optimal algorithm for energy saving in embedded system with multiple sleep modes," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 24, no. 2, pp. 706–719, Feb. 2016.
[16] R. Jurdak, A. G. Ruzzelli, and G. M. P. O'Hare, "Radio sleep mode optimization in wireless sensor networks," *IEEE Trans. Mobile Comput.*, vol. 9, no. 7, pp. 955–968, Jul. 2010.
[17] S.-H. Lim, J. Oh, B.-H. Lee, S. W. Lee, and M. Sohn, "Energy-efficient carpool policy for wireless interfaces of mobile devices in ubiquitous environments," *Math. Comput. Model.*, vol. 58, nos. 5–6, pp. 1301–1312, Sep. 2013.
[18] S.-H. Lim, J. Oh, B.-H. Lee, and M. Sohn, "Threshold-based energy-efficient data transmission policy for mobile devices," in *Proc. IMIS*, Italy, Palermo, Jul. 2012, pp. 138–142.
[19] H. Okamura and T. Dohi, "Dynamic power management with optimal time-out policies," *IEEE Syst. J.*, vol. 11, no. 2, pp. 962–972, Jun. 2017.
[20] Q. Jiang, H.-S. Xi, and B.-Q. Yin, "Adaptive optimisation of timeout policy for dynamic power management based on semi-Markov control processes," *IET Control Theory Appl.*, vol. 4, no. 10, pp. 1945–1958, 2010.
[21] S. O. D. Luiz, A. Perkusich, B. M. J. Cruz, B. H. M. Neves, and G. M. da S. Araujo, "Optimization of timeout-based power management policies for network interfaces," *IEEE Trans. Consum. Electron.*, vol. 59, no. 1, pp. 101–106, Feb. 2013.
[22] S. W. Lim and S.-H. Lim, "Analysis of a queueing system under deadline and workload controls," in *Proc. EEECS*, Phuket, Thailand, 2016, pp. 1–2.
[23] J. R. Artalejo, "On the M/G/1 queue with D-policy," *Appl. Math. Model.*, vol. 25, pp. 1055–1069, Dec. 2001.
[24] H. W. Lee, W. J. Seo, S. W. Lee, and J. Jeon, "Analysis of the MAP/G/1 queue under the Min(N,D)-policy," *Stochastic Models*, vol. 26, no. 1, pp. 98–123, Feb. 2010.
[25] H. W. Lee, S. W. Lee, W. J. Seo, S. H. Cheon, and J. Jeon, "A unified framework for the analysis of M/G/1 queue controlled by workload," in *Computational Science and Its Applications—ICCSA* (Lecture Notes in Computer Science), vol. 3982. Berlin, Germany: Springer-Verlag, 2006, pp. 718–727.
[26] H. Takagi, *Queueing Analysis: A Foundation of Performance Evaluation: Vacation and Priority Systems*, vol. 1. Amsterdam, The Netherlands: North Holland, 1991.
[27] *CW1200: 802.11n Dual-Band WLAN System-on-Chip, Data Sheet*, ST Ericsson, Plan-les-Ouates, Switzerland, Oct. 2009.
[28] B. Zhang, A. Iosup, J. Pouwelse, and D. Epema, "Identifying, analyzing, and modeling flashcrowds in BitTorrent," in *Proc. IEEE P2P*, Aug. 2011, pp. 240–249.

SUNG-HWA LIM (Member, IEEE) received the B.S., M.S., and Ph.D. degrees in computer engineering from Ajou University, South Korea, in 1999, 2001, and 2008, respectively. He was a Postdoctoral Researcher with the Coordinated Science Laboratory, University of Illinois, Urbana Champaign (UIUC), from 2008 to 2009. He is currently an Associate Professor with the Department of Multimedia, Namseoul University. His research interests include the Internet of Things, power-aware computing, and real-time systems.

SE WON LEE (Member, IEEE) received the B.S., M.S., and Ph.D. degrees in industrial engineering from Sungkyunkwan University, South Korea, in 1998, 2003, and 2008, respectively. He is currently an Associate Professor with the Division of Business Administration, Pukyong National University. He is also a Visiting Scholar with the Mathematical Science Division (Brain Korea 21 Plus), Department of Mathematics, Korea University, South Korea. His research interests include queueing theory, operations research, and applied stochastic processes.

MYE SOHN (Member, IEEE) received the M.S. and Ph.D. degrees from the Korea Advanced Institute of Science and Technology (KAIST). She is currently a Professor with the Department of Systems Management Engineering, Sungkyunkwan University. Her main research interests include machine learning, ontology, web of things, and semantic web.



BYOUNG-HOON LEE received the B.S. and M.S. degrees in computer engineering from Chungbuk National University, Cheongju, South Korea, in 1998 and 2000, respectively, and the Ph.D. degree in information and communication from Ajou University, Suwon, South Korea, in 2009. He is currently a Visiting Professor with the National Program of Excellence in Software, Chungbuk National University. His research interests include the IoT programming, distributed systems, and embedded programming.