Journal of Interconnection Networks, Vol. 2, No. 1 (2001) 69–84

ADAPTIVE SCHEMES FOR REDUCING POWER CONSUMPTION OF MOBILE DATA HANDSETS

RONG-JAYE CHEN, TING-YU LIN, and YI-BING LIN
Department of Computer Science and Information Engineering
National Chiao-Tung University, Hsinchu, Taiwan
E-mail: {rjchen, tylin, liny}@csie.nctu.edu.tw

Received 1 March 2000
Revised 15 January 2001

This paper describes a threshold-based wake-up mechanism to reduce the battery power consumption of a mobile data handset. The threshold approach switches the system into the sleep mode when the memory queue for arriving packets is empty, and switches on the system when the number of packets in the memory queue is above a threshold value. We propose several adaptive schemes capable of dynamically selecting the threshold value for the threshold approach. An adaptive algorithm adjusts the threshold value based on a pre-defined packet-dropping probability, for which the switch-on rate is kept reasonably small to maintain the actual packet-dropping probability as close as the pre-defined value.

Two strategies are used in the adaptive algorithm to adjust the threshold value: binary-division and fixed-amount. Two calculation strategies are considered to measure the packet-dropping probability: window-averaging and leaky-bucket integration (LBI). Our study indicates that the binary-division strategy outperforms the fixed-amount strategy in adjusting the threshold value. Furthermore, with proper setting, the LBI strategy outperforms the window-averaging strategy.

Keywords: adaptive scheme, packet-dropping probability, power consumption, sleep mode, switch-on rate, threshold, wake-up mechanism

1. Introduction

In the US today, wireline data traffic already exceeds voice traffic. Even in wireless telecommunications, data transmission promises more business opportunities than ever imagined. The convergence of personal computing, networking and wireless technologies will change the way we live forever. The Internet, specifically, the World Wide Web, has provided many data applications that can be accessed through the wireless networks. Furthermore, the advance of the third generation wide-band wireless technologies[14,15] offers large volume data transmission. Thus, research in wide-band wireless data network becomes very important.[12,13]

To support wireless data applications, a personal data device is designed to be portable, where any add-on feature needs to be power-cautious.[16] Since the operations of mobile handsets may significantly consume battery power, power consumption is an important issue in mobile data handset design. Figure 1 illustrates a mobile data handset architecture that consists of three parts.

- A *data receiving unit* is a fixed-size memory storage queue for packets received from wireless networks.
- A *data processing unit* is a processor that handles received data.
- A *user interface unit* provides input and output interfaces to the mobile user.

One approach to reduce power consumption is to design a mobile data handset that supplies power separately to the three units.[1,17] With this design, the data processing and user interface units can be shut down in the *sleep mode* while the data receiving unit is waiting for data from the network. Those two units will not be woken up until the data receiving unit receives incoming packets. The wake-up action is called *switch-on* and the frequency S_r of switch-on is referred to as the *switch-on rate*. Once the data processing and user interface units are switched on, they work until the memory queue in the data receiving unit becomes empty. Then the two units enter the sleep mode again to conserve power. In the sleep mode, only the data receiving unit is awake to receive packets. This handset architecture can be viewed as a single-server queuing system with finite capacity.[2,3,4]

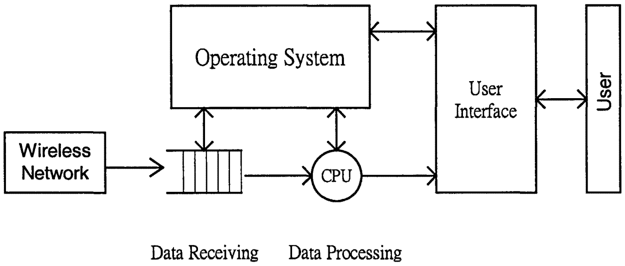


Fig. 1. A mobile data handset architecture

To wake up the data processing and user interface units immediately upon receipt of a packet may cause too many switch-on actions, and thus degrade the power efficiency.[10] To avoid this problem, the data receiving unit can be designed to wait for more arrival packets before waking up the other two units. Specifically, a switch-on action is performed only when there has been a certain number of packets in the memory queue. However, if we do not reserve enough space for the incoming packets, some packets may be dropped. Denote P_d as the *packet-dropping probability* or the probability that an arriving packet is dropped. In designing the power-saving mechanism for wireless data handsets, an important goal is to reduce the switch-on rate S_r without significantly increasing the packet-dropping probability P_d.

To achieve power saving by controlling the switch-on activities, we have proposed three wake-up approaches[11]: the threshold approach, the vacation approach, and the hybrid approach. These approaches switch the system into the sleep mode when the memory queue for arriving packets is empty. Suppose that the size of the memory queue in the data receiving unit is R. In the *threshold* approach, a threshold value r, where $1 \leq r \leq R$, is used to determine when to switch on the system. As soon as the number of packet arrivals reaches the value r, the data processing and the user interface units are woken up to start working. It is important to select the r value properly because the threshold value r affects the switch-on rate S_r and packet-dropping probability P_d at the same time.

In the *vacation* approach, a timer is utilized to count when to wake up. Every time when the memory queue becomes empty, the data processing and user interface units enter the sleep (vacation) mode immediately, and the vacation timer starts counting down. The data processing and the user interface units come back to work by a switch-on action after the vacation time has expired.[5] If there are packets waiting in the queue, then these packets will get served. After the data queue is empty, the data processing and the user interface units enter the sleep mode again. The *hybrid* approach combines both the threshold and the vacation mechanisms. In this approach, the data processing and the user interface units wake up either when the threshold r is reached or when the vacation time expires.

In the above three approaches, the selections of the threshold value r (for the threshold approach) and the vacation time (for the vacation approach) significantly affect the performance of the system. It is clear that the selection of the threshold value/vacation time depends on the incoming traffic. This paper focuses on the threshold approach. We propose adaptive mechanisms that, according to the incoming traffic, dynamically adjust the threshold value r so that packet dropping can be bounded to a pre-defined packet-dropping probability P_d^*. Two strategies are used to adjust the threshold r: *binary-division* and *fixed-amount*. Furthermore, two strategies, *window-averaging* and *leaky-bucket integration (LBI)*, are considered to calculate the actual packet-dropping probability P_d. The adaptive approach compares P_d with P_d^* to determine if r should be adjusted.

We propose an analytic model to derive the performance of the threshold approach when the incoming traffic is known. By changing the r values, optimal threshold values can be found in the analytic model. This optimal threshold will be used as a reference point for comparison. The adaptive mechanisms that do not have the knowledge of incoming traffic are investigated by simulation experiments. The performance of the adaptive mechanisms are compared with the optimal results to indicate "how good" the adaptive algorithms can be.

2. Assumptions

To model the threshold approach, we give the necessary assumptions in this section. We assume that the arrival packets to a handset are a Poisson process with rate λ. Let random variable t be the inter-arrival time of two consecutive packets. Then t

is exponentially distributed.[6]

Every arriving packet is stored in the memory queue of the data receiving unit. The arriving packet is dropped if the memory queue is full. The service times for processing the packets have an exponential distribution with the service rate μ. The packets are processed following the first-come first-served discipline. Several input parameters are considered in our model:

r: the threshold value
R: the size of the memory queue
λ: the packet arrival rate
μ: the packet service (processing) rate

Two output measures are investigated in our model:

S_r: the switch-on rate
P_d: the packet-dropping probability

In the threshold approach, if a larger threshold r is selected, then a smaller switch-on rate S_r is observed, which results in less battery power consumption of a handset. However, since the packet-dropping probability P_d increases when the threshold r becomes larger, the threshold r should be carefully selected to balance S_r against P_d.

3. The Analytic Model

An analytic model for the threshold approach is presented here. We derive the switch-on rate S_r and the packet-dropping probability P_d for a given r value. This part has been introduced in our previous work.[11] We reiterate the model for the readers' benefit.

In the threshold approach, a switch-on action is performed when the number of packets accumulated in the memory queue is above the threshold r. When $r = 1$, the handset switches on as soon as a packet arrives, and switches off when the queue becomes empty. In this case, the threshold approach can be modeled directly by an M/M/1/R queue. When $r > 1$, the state transition diagram for the stochastic process is shown in Figure 2, which consists of two groups of states. State $s(i^*)$ represents that there are i packets in the memory queue when the system is in the sleep mode, where $0 \le i \le r-1$. State $s(j)$ represents that there are j packets in the queue when the system is in the active (wake-up) mode, where $1 \le j \le R$. When the system is in the sleep mode, the stochastic process moves from state $s(i^*)$ to state $s((i+1)^*)$ for $0 \le i \le r-2$ and from state $s((r-1)^*)$ to state $s(r)$ with rate λ. When the system is in the active mode, the process moves from state $s(i)$ to state $s((i+1))$ with rate λ for $1 \le i \le R-1$, and moves from state $s(i)$ to state $s((i-1))$ for $2 \le i \le R$ and from state $s(1)$ to state $s(0^*)$ with rate μ.

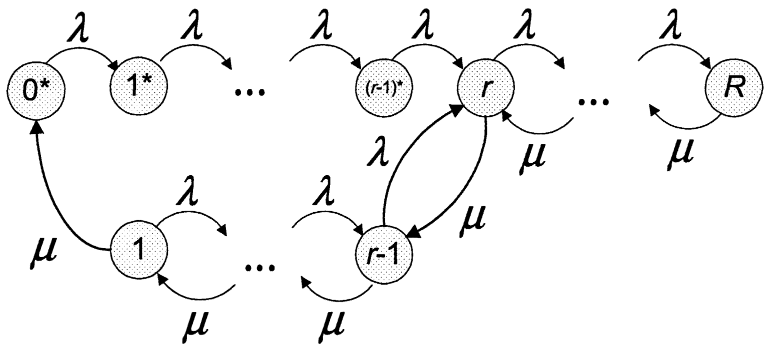


Fig. 2. The threshold model with $r > 1$

A switch-on action occurs when the process moves from state $s((r-1)^*)$ to state $s(r)$. Thus the switch-on rate S_r is computed as

$$S_r = \lambda P[(r-1)^*]$$

where $P[i]$ denotes the stationary probability of state $s(i)$. An arriving packet is dropped when the memory queue is full, i.e., when the process is at state $s(R)$. Thus, the packet-dropping probability P_d is expressed as

$$P_d = P[R]$$

Based on the state transition diagram in Figure 2, when the system is in a steady state, the following equilibrium equations hold.

$$\begin{aligned}
&\lambda P[0^*] = \mu P[1] \\
&P[i^*] = P[(i-1)^*] \text{ for } 1 \le i \le r-1 \\
&(\lambda+\mu)P[1] = \mu P[2] \\
&(\lambda+\mu)P[i] = \lambda P[i-1] + \mu P[i+1] \text{ for } 2 \le i \le r-1 \text{ or } r+1 \le i \le R-1 \\
&(\lambda+\mu)P[r] = \lambda P[(r-1)^*] + \lambda P[r-1] + \mu P[r+1] \\
&\mu P[R] = \lambda P[R-1]
\end{aligned}$$

After rearranging the above equations, the steady state probabilities for the states of the threshold approach are expressed as

$$P[i^*] = \begin{cases} \mu/\lambda P[1] & \text{if } i = 0 \\ P[(i-1)^*] & \text{if } 1 \le i \le r-1 \end{cases} \tag{3.1}$$

and

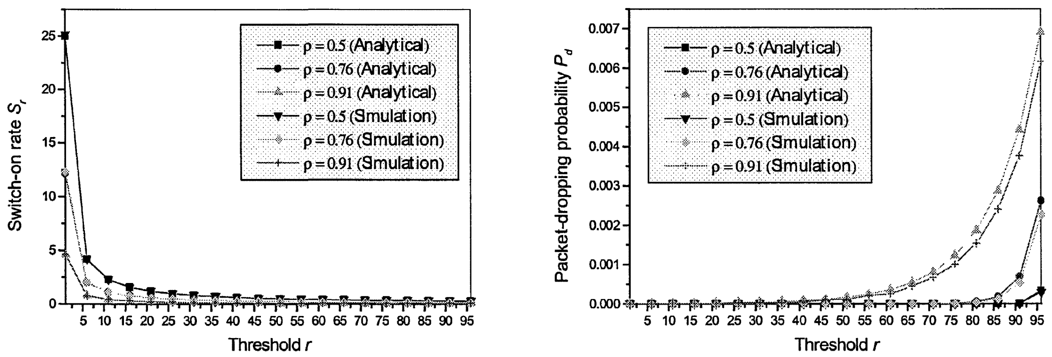
$$P[i] = \begin{cases} \mu/(\lambda+\mu)P[2] & \text{if } i=1 \\ \lambda/(\lambda+\mu)P[i-1] + \mu/(\lambda+\mu)P[i+1] & \text{if } 2 \le i \le r-1 \\ & \text{or } r+1 \le i \le R-1 \\ \lambda/(\lambda+\mu)(P[(r-1)^*] + P[r-1]) + \mu/(\lambda+\mu)P[r+1] & \text{if } i=r \\ \lambda/\mu P[R-1] & \text{if } i=R \end{cases} \tag{3.2}$$

with the constraint that

$$\sum_{i=0}^{r-1} P[i^*] + \sum_{j=1}^{R} P[j] = 1, \tag{3.3}$$

we solve the above equations using the Gauss-Seidel method[7] by initializing $P[0^*] = P[1^*] = P[2^*] = \cdots = P[1] = P[2] = \cdots = P[r] = \cdots = P[R] = 1\ /\ (R+r)$. Then both the packet-dropping probability P_d and the switch-on rate S_r can be obtained.

Based on our analytic model, numerical examples are used to illustrate how the r value affects the output measures S_r and P_d. In our examples, a memory queue of fixed size $R = 100$ is considered. Let the system workload $\rho = \lambda/\mu$. Three ρ values 0.5, 0.76, and 0.91 are considered. Simulation experiments are conducted to validate our analytic model. In Figure 3, the curves for both the analytical and simulation results are almost identical. Thus, the analytical results are consistent with the simulation experiments.



(a) Switch-on rate S_r (b) Packet-dropping probability P_d

Fig. 3. Effect of r on S_r and $P_d(R = 100)$

Figure 3 shows both the switch-on rate S_r and packet-dropping probability P_d as functions of the threshold r. Figure 3 (a) indicates an intuitive result that S_r is a decreasing function of r. The non-trivial results are that, when $r < 30$, S_r is dramatically reduced as r increases. On the other hand, when $r > 30$, S_r is insignificantly affected by r. For P_d, Figure 3 (b) indicates that P_d is an increasing function of r. When $r > 40$, P_d is dramatically increased as r increases. On the other hand, when $r < 40$, P_d is insignificantly affected by r. Thus, we conclude that for this particular workload $\rho = 0.91$, $30 < r < 40$ should be selected. Note that as ρ decreases, the range for selecting r becomes larger. For example, $25 < r < 80$ is appropriate for $\rho = 0.76$, and $15 < r < 90$ is appropriate for $\rho = 0.5$. Our example indicates that even if the system is in the heavy load situation (i.e., $\rho = 0.91$), an appropriate r range exists. However, the optimal r value cannot be determined if the traffic patterns to the handset are not known. In the subsequent sections, we show how to estimate the traffic patterns and thus the r value through adaptive mechanisms.

4. The Adaptive Schemes

This section proposes adaptive schemes to dynamically select the buffer threshold r for the threshold approach. These schemes adjust the r value based on the incoming traffic so that the packet-dropping probability can be maintained below a pre-defined value P_d^*. Figure 4 illustrates the adaptive algorithm that is described in the following steps:

Step 1. Initialize the threshold r at $\lfloor R/2 \rfloor$ where R is the size of the memory queue.
Step 2. Define a *batch* as consecutive N_p packet arrivals. When the packets in the i^{th} batch (where $i \geq 1$) have been received, the adaptive algorithm calculates the i^{th} packet-dropping probability P_d, denoted by $P_d(i)$ (to be elaborated).
Steps 3 and 4. If the calculated packet-dropping probability $P_d(i)$ is larger than P_d^*, then decrease the threshold r to reduce the packet-dropping probability.
Steps 5 and 6. On the other hand, if the calculated packet-dropping probability $P_d(i)$ is smaller than P_d^*, the threshold r is increased to reduce the number of power switching.

At Steps 4 and 6 of Figure 4, two strategies are considered to adjust the r value: the binary-division adjustment strategy and the fixed-amount adjustment strategy. The binary-division adjustment strategy uses a lower-bound value $LB(i)$, an upper-bound value $UB(i)$, and the threshold $r(i)$ for the i^{th} adjustment, where $i \geq 1$. Initially, $LB(0) = 0$, $UB(0) = R$, and

$$r(0) = \lfloor (LB(0) + UB(0))/2 \rfloor = \lfloor R/2 \rfloor \tag{4.1}$$

If the calculated packet-dropping probability $P_d(i) > P_d^*$, then set $UB(i)$ to $r(i-1)$, $LB(i)$ to $LB(i-1)$, and compute the new threshold as

$$r(i) = \lfloor (LB(i) + UB(i))/2 \rfloor \tag{4.2}$$

On the other hand, if $P_d(i) < P_d^*$, then set $LB(i)$ to $r(i-1)$, $UB(i)$ to $UB(i-1)$, and compute the new threshold as

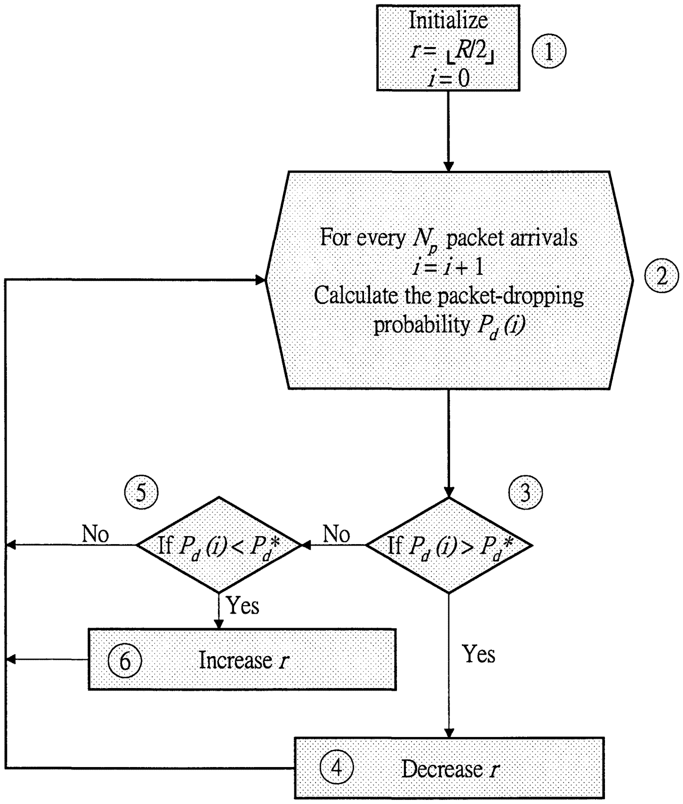


Fig. 4. The adaptive algorithm

$$r(i) = \lfloor (LB(i) + UB(i))/2 \rfloor \tag{4.3}$$

If $P_d(i) = P_d^*$, then $LB(i) = LB(i-1)$, $UB(i) = UB(i-1)$, and $r(i) = r(i-1)$. Figure 5 shows the r values after the 1^{st} and 2^{nd} adjustments.

In the fixed-amount adjustment strategy, a fixed amount X is used to adjust the r value, where $0 < X < \lfloor R/2 \rfloor$. Initially, $r(0) = \lfloor R/2 \rfloor$. If the measured packet-dropping probability $P_d(i) > P_d^*$, then

$$r(i) = \begin{cases} r(i-1) - X & \text{if } r(i-1) > X \\ r(i-1) & \text{if } r(i-1) \le X \end{cases} \tag{4.4}$$

If $P_d(i) < P_d^*$, then

$$r(i) = \begin{cases} r(i-1) + X & \text{if } r(i-1) + X \le R \\ r(i-1) & \text{if } r(i-1) + X > R \end{cases} \tag{4.5}$$

If $P_d(i) = P_d^*$, then $r(i) = r(i-1)$.

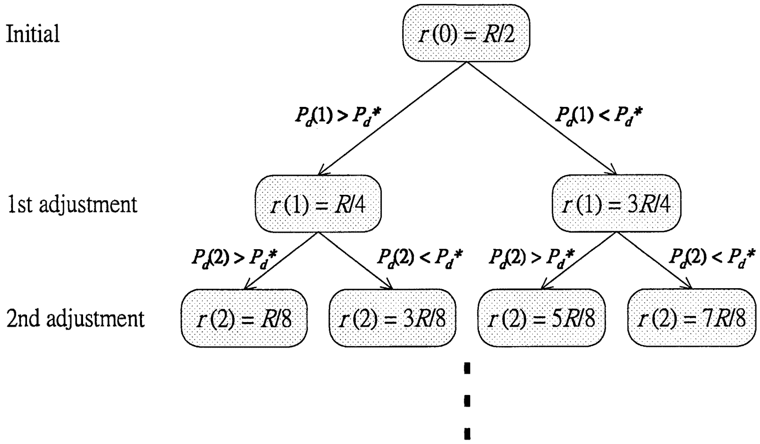


Fig. 5. The binary-division adjustment strategy

At Step 2 of Figure 4, two $P_d(i)$ calculation strategies are considered: window-averaging and leaky-bucket integration (LBI). Define $N_d(i)$ as the number of dropped packets in the i^{th} batch, where $i \geq 1$. In the window-averaging strategy, a window of size W is defined, and the portion of lost packets in the latest W batches are used to compute $P_d(i)$ at the i^{th} batch for $i \geq 1$; in other words,

$$P_d(i) = \begin{cases} \frac{\sum_{j=1}^{i} N_d(j)}{iN_p} & \text{if } i \leq W \\ \frac{\sum_{j=i-W+1}^{i} N_d(j)}{WN_p} & \text{if } i > W \end{cases} \tag{4.6}$$

The LBI strategy considers packet loss in all previous batches with different weights. A factor α, where $0 < \alpha < 1$, is introduced to reduce the impact of the "old" batches. Initially, let $P_d(1) = N_d(1)\ /\ N_p$. For the i^{th} batch, where $i > 1$,

$$P_d(i) = \alpha P_d(i-1) + (1-\alpha)N_d(i)/N_p \tag{4.7}$$

In LBI, the effect of very old batches disappears eventually.

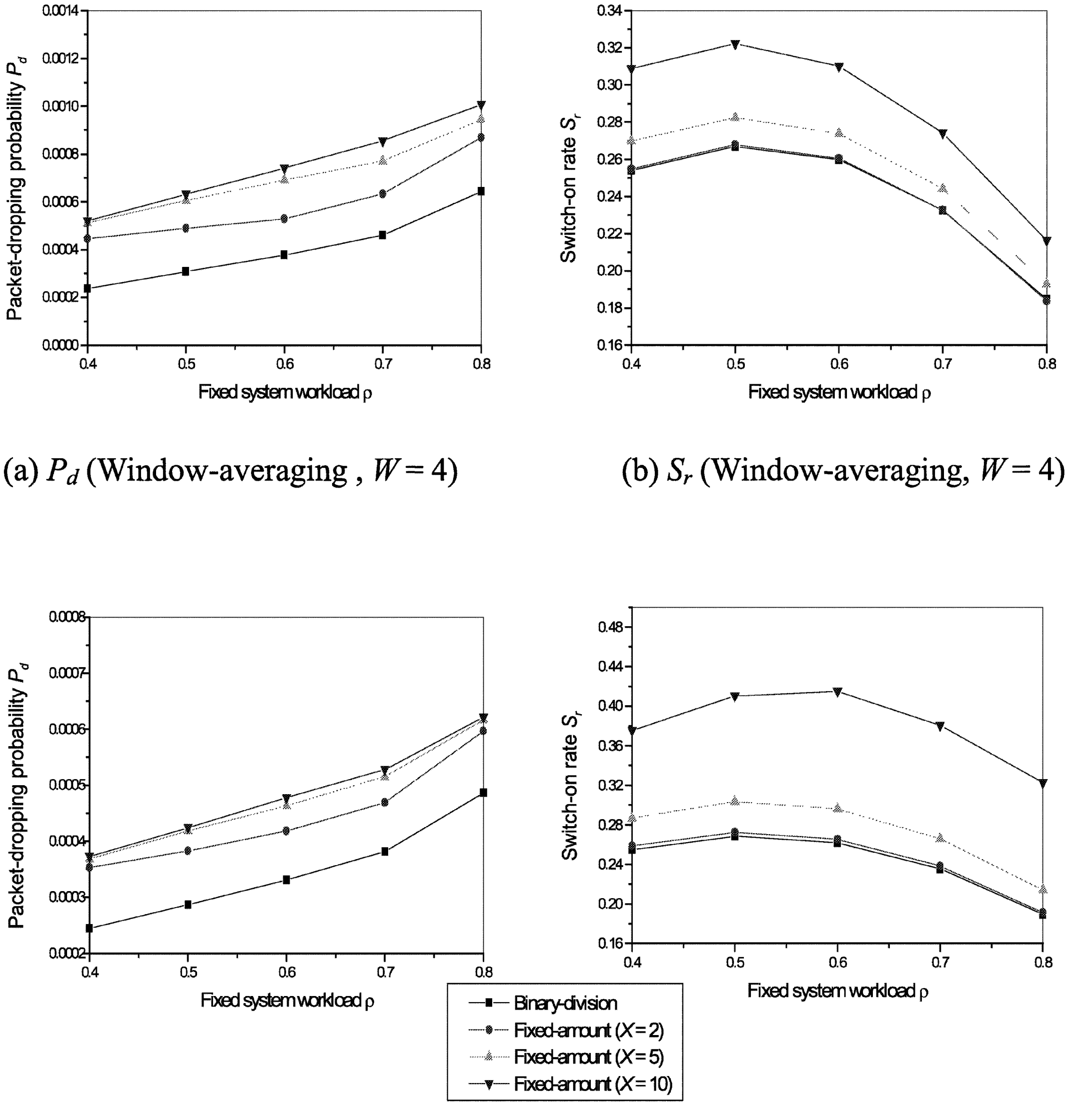
5. Performance Evaluation of The Adaptive Schemes

This section conducts simulation experiments to investigate the performance of the adaptive schemes. We first evaluate the two adjustment strategies, and then compare the two $P_d(i)$ calculation strategies.

5.1. *Performance of the adjustment strategies*

Assume that the size of the memory queue $R = 100$, $N_p = 1000$, and pre-defined packet-dropping probability $P_d^* = 0.0001$. Other values for these parameters show similar results and are not presented here.

With five values of the system workload $\rho = \lambda/\mu = 0.4$, 0.5, 0.6, 0.7, and 0.8, Figure 6 illustrates the performance of the two adjustment strategies using the two



(a) P_d (Window-averaging , $W = 4$)

(b) S_r (Window-averaging, $W = 4$)

(c) P_d (LBI, $\alpha = 0.6$)

(d) S_r (LBI, $\alpha = 0.6$)

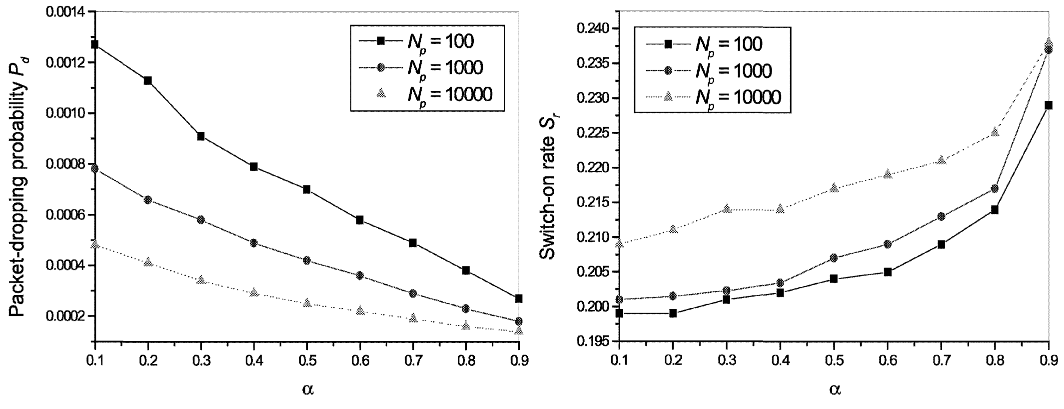
Fig. 6. Comparison of the binary-division and the fixed-amount strategies

calculation strategies. In the fixed-amount adjustment strategy, we consider $X =$ 2, 5, and 10. Figures 6 (a) and (c) plot the packet-dropping probability P_d , while Figures 6 (b) and (d) plot the switch-on rate S_r. P_d increases when ρ increases. For S_r, with light traffic ($\rho \leq 0.5$), S_r increases as ρ becomes larger. In the light traffic condition, the number of the queued packets is more likely to reach the threshold r as ρ increases. On the other hand, with heavy traffic ($\rho > 0.5$), S_r decreases as ρ increases. In the heavy traffic condition, the handset is less likely to enter the sleep mode as ρ increases. This results in a decreasing number of power switching. Therefore, the S_r curves in Figures 6 (b) and (d) are convex.

The results indicate that binary-division outperforms fixed-amount. That is, the binary-division strategy has smaller packet-dropping probability P_d and lower switch-on rate S_r compared with the fixed-amount strategy. In the remainder of this paper, we consider only the binary-division strategy.

5.2. *Performance of the $P_d(i)$ calculation strategies*

This subsection studies the performance of the $P_d(i)$ calculation schemes with the binary-division adjustment strategy. To investigate how quick our algorithms can adapt the incoming traffic, we dynamically change the packet arrival rates. Specifically, after every 20,000 packet arrivals, we randomly switch the system workload ρ to 0.2, 0.4, 0.6, and 0.8 with equal probabilities. In the experiments, P_d^* is set to 0.0001.



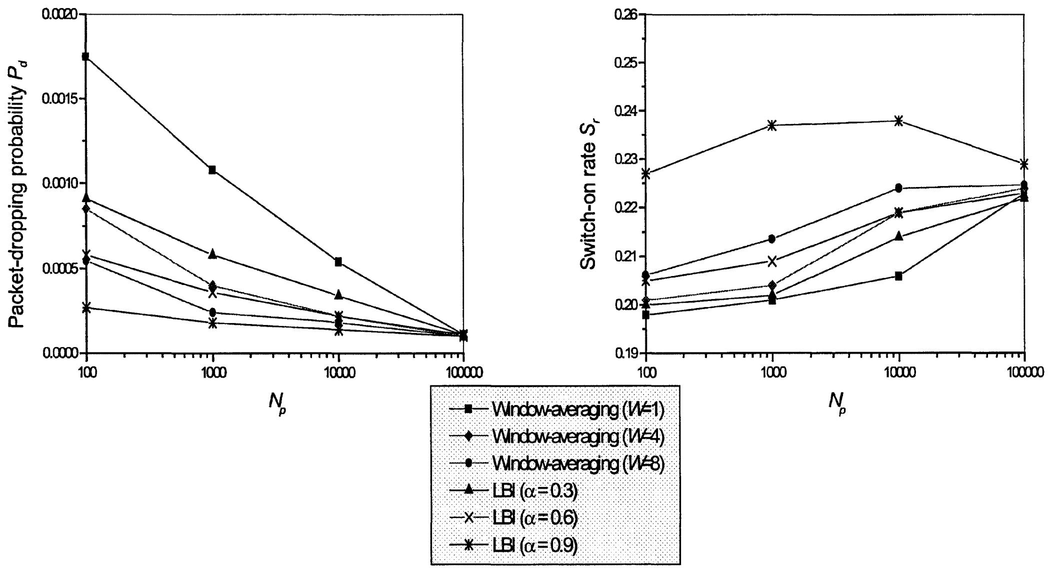
(a) Packet-dropping probability P_d (b) Switch-on rate S_r

Fig. 7. Effect of α on LBI($P_d^* = 0.0001$,varied arrival rates)

We first investigate the effect of α on the LBI strategy for the varied system workload. For $P_d^* = 0.0001$, Figure 7 plots the packet-dropping probability P_d and the switch-on rate S_r against α for the LBI strategy, where $N_p = 100$, 1000, and 10000. The figure indicates that P_d decreases as α increases, while S_r increases as

α grows. We note that P_d is close to P_d^* if large α is selected. In other words, to accurately control P_d, large α should be selected in the LBI calculation strategy. We also observe that for large α, the performance of LBI is less sensitive to the size of N_p.

Figure 8 illustrates the impact of N_p on the two calculation strategies for the varied system workload. For $P_d^* = 0.0001$, Figure 8 plots the packet-dropping probability P_d and the switch-on rate S_r against four values of N_p: 100, 1000, 10,000, and 100,000. By increasing N_p from 100 to 100000, Figure 8 (a) shows an average 82.6% change for the P_d performance, while Figure 8 (b) illustrates an average 8.9% change for the S_r performance. Thus P_d is more sensitive to N_p than S_r is. Furthermore, to accurately control the P_d value (so that it is close to P_d^*), large N_p is required for $W < 8$ in window-averaging and $\alpha < 0.9$ in LBI. On the other hand, with large W or α values, the adaptive scheme can accurately control P_d with small number of samples (i.e., small N_p).



(a) Packet-dropping probability P_d (b) Switch-on rate S_r

Fig. 8. Effect of $N_p(P_d^* = 0.0001$,varied arrival rates)

Define the packet-waiting time as the period between the time a packet arrives and when it is processed. Figure 9 plots switch-on rate and mean packet-waiting time against the packet-dropping probability where $N_p = 1000$. Figures 9 (a) and (c) illustrate the switch-on rates S_r as functions of the packet-dropping probability P_d for the fixed system workload ρ= 0.5 and varied system workload, respectively. Figures 9 (b) and (d) show the mean packet-waiting times as functions of the packet-dropping probability P_d.

For fixed system workload ρ= 0.5, Figure 9 (a) shows that the switch-on rates of the adaptive approaches are 1.5% - 3.9% higher than the optimal values derived from the analytic model. Note that the operation points optimized for the switch-on rates result in longer mean packet-waiting times. Figure 9 (b) indicates that the mean packet-waiting times of the adaptive approaches are 1.0% - 3.6% shorter than the values derived from the analytic model. Furthermore, for the fixed workload case, the calculation strategy with a lower switch-on rate results in a longer mean packet-waiting time, and vice versa.

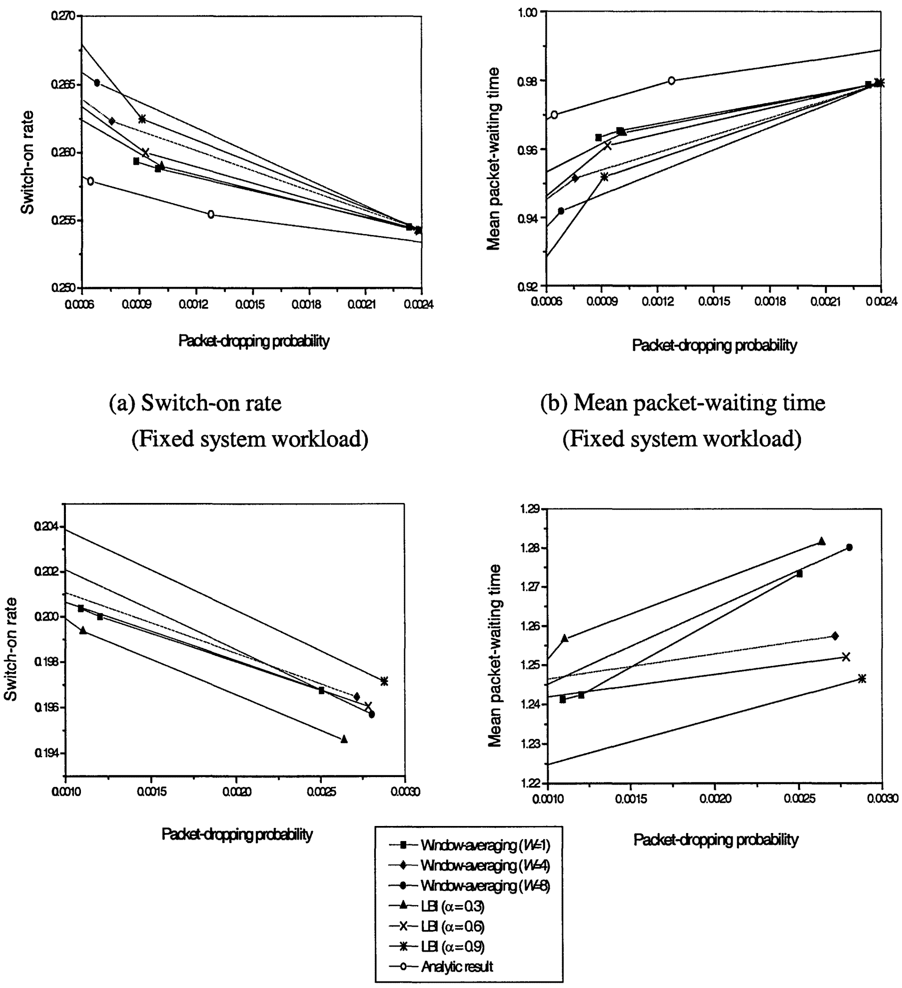
For varied system workload, Figure 9 (c) indicates that LBI with $\alpha = 0.6$ has similar or better switch-on rate performance compared with window-averaging with $W = 1$, 4, and 8. Figure 9 (d) indicates that LBI with $\alpha = 0.6$ outperforms window-averaging (with W = 1, 4, and 8) in terms of the mean packet-waiting time. We conclude that, with a proper setting of α value, the LBI strategy outperforms the window-averaging strategy. For LBI, when α increases, the switch-on rate increases and the mean packet-waiting time decreases as in the fixed workload case.

Figure 10 shows P_d as function of P_d^* for the fixed system workload ρ= 0.5 and varied system workload, respectively. This figure indicates how accurately the adaptive scheme can control P_d. The optimal values (i.e. $P_d = P_d^*$) are also drawn in the figures to show the ideal situation. The figure indicates that the window-averaging strategy with larger W values and the LBI strategy with larger α values can control P_d more accurately. This conclusion is consistent with what we observed in Figure 8.

6. Conclusions

This paper studied a threshold-based wake-up mechanism, called the threshold approach, to reduce the battery power consumption of a mobile data handset. The threshold approach switches the system into the sleep mode when the memory queue for the arriving packets is empty, and switches on the system when the number of packets in the memory queue is above a threshold value. We proposed several adaptive schemes capable of dynamically selecting the threshold value for the threshold approach. Based on a pre-defined packet-dropping probability, the adaptive algorithm adjusts the threshold value to control the actual packet-dropping probability to be close to the pre-defined value. Two strategies are used in the adaptive algorithm to adjust the threshold value: binary-division and fixed-amount. Two calculation strategies: window-averaging and leaky-bucket integration (LBI), are considered to measure the actual packet-dropping probability that is used to decide the direction for adjusting the threshold value.

The performance evaluation results showed that the binary-division adjustment strategy outperforms the fixed-amount strategy. Furthermore, with proper setting, the LBI strategy outperforms the window-averaging strategy.



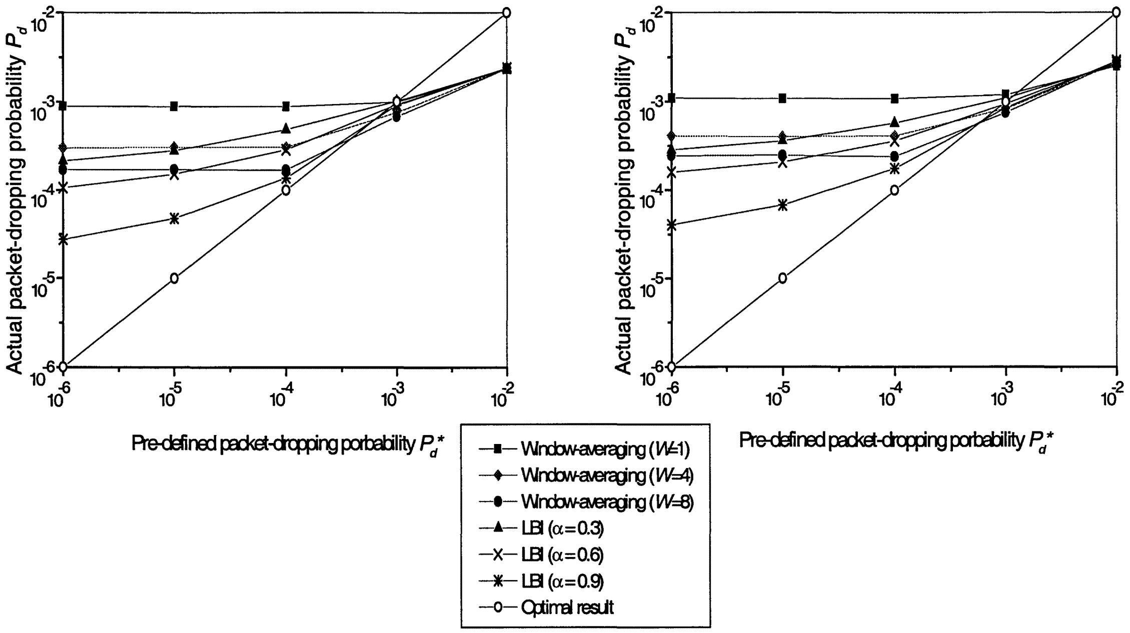
(a) Switch-on rate (Fixed system workload)

(b) Mean packet-waiting time (Fixed system workload)

(c) Switch-on rate (Varied system workload)

(d) Mean packet-waiting time (Varied system workload)

Fig. 9. Switch-on rate and mean packet-waiting time as functions of packet-dropping probability($N_p = 1000$)



(a) Fixed system workload (b) Varied system workload

Fig. 10. Actual and pre-defined packet-dropping probabilities($N_p = 1000$)

References

1. C. L. Su, C. Y. Tsui, and A. M. Despain, "Saving Power in the Control Path of Embedded Processors," *IEEE Design & Test of Computer*, pp. 24-30, Winter 1994.
2. Donald Gross and Carl M. Harris, "Fundamentals of Queueing Theory," *John Wiley & Sons, Inc.*, Second Edition, 1985.
3. Sheldon M. Ross, "Introduction to Probability Models," *Academic Press, Inc.*, Third Edition, 1985.
4. Peter G. Harrison and Naresh M. Patel, "Performance Modelling of Communication Networks and Computer Architectures," *Addison-Wesley Publishing*, 1992.
5. Leonard Kleinrock, "Queueing Systems Volume 1: Theory," *John Wiley & Sons, Inc.*, 1975.
6. Edward P. C. Kao, "An Introduction to Stochastic Processes," *Wadsworth Publishing*, 1997.
7. Eugene Isaacson and Herbert Bishop Keller, "Analysis of Numerical Methods," *John Wiley & Sons, Inc.*, 1966.
8. John M. Rulnick and Nicholas Bambos, "Performance Evaluation of Power-managed Mobile Communication Devices," *Proceedings of the International Conference on Communications*, pp. 1477-1481, 1996.
9. Raghunathan Anand, Dey Sujit, and Jha Nirajk, "Power Management Techniques for Control-Flow Intensive Designs," *Design Automation Conference*, pp. 429-434, 1997.
10. John M. Rulnick and Nicholas Bambos, "Mobile Power Management for Maximum Battery Life in Wireless Communication Networks," *Proceedings of IEEE INFOCOM*, Vol. 2, pp. 443-450, 1996.
11. Rong-Jaye Chen, Ting-Yu Lin, and Yi-Bing Lin, "Reducing Power Consumption for Mobile Multimedia Handsets," *TamKang Journal*, 1999.
12. Takehiro Murase and Minour Ohyama, "Evolution of Personal Multimedia Communications Services in Japan," *IEEE Personal Communications*, Vol. 5, Number 6, pp. 66-74, 1998.
13. Takehiro Murase and Minour Ohyama, "Personal Multimedia Communications Services," *IEEE PIMRC'96*, pp. 163-167, 1996.
14. Tero Ojanpera and Ramjee Prasad, "An Overview of Third-Generation Wireless Personal Communications: A European Prespective," *IEEE Personal Communications*, Vol. 5, Number 6, pp. 59-65, 1998.
15. Y. Ishida, "Recent Study on Candidate Radio Transmission Technology for IMT-2000," 1^{st} *Annual CDMA Eurp. Cong.*, London, UK, Oct. 1997.
16. Chatschik Bisdikian, et al., "WiSAP: A Wireless Personal Access Network for Handheld Computing Devices," *IEEE Personal Communications*, Vol. 5, Number 6, pp. 18-25, 1998.
17. Ying Wen Bai and C.H. Lai, "A Bitmap Scaling and Rotation for SH1 Low Power PC," 2^{nd} *ACM MSWiM'99*, 1999.