



Adaptive sleep mode management in IEEE 802.16m wireless metropolitan area networks

Sunggeun Jin ^{a,b,*}, Xi Chen ^b, Daji Qiao ^b, Sunghyun Choi ^c

^a ETRI, Daejeon 305-700, Republic of Korea

^b Department of Electrical and Computer Engineering, Iowa State University, Ames, IA 50011, United States

^c School of Electrical Engineering and INMC, Seoul National University, Seoul 151-744, Republic of Korea

ARTICLE INFO

Article history:

Received 15 May 2010

Received in revised form 20 December 2010

Accepted 1 March 2011

Available online 11 March 2011

Keywords:

IEEE 802.16m

Sleep cycle

Listening window

Sleep window

ABSTRACT

The emerging IEEE 802.16m standard provides a new sleep mode operation for Mobile Stations (MSs). It evolves from the sleep mode operation in the IEEE 802.16 standard but with more advanced features, e.g., the listening window may be extended and the sleep cycle length is adjustable. To fully exploit these advancements, we conduct a comprehensive analytical study on the power consumption and average packet delay under the new sleep mode operation. Then, based on the analytical results, we propose a novel adaptive sleep mode management scheme called Adaptive Sleep Mode Management (ASMM), which adjusts an MS's sleep cycle and listening window in an adaptive manner based on online monitoring and estimation of the traffic condition. The goal is to minimize the power consumption by the MS while limiting the average packet delay under a user-specified level. The effectiveness of ASMM is demonstrated via simulation-based performance evaluation.

© 2011 Elsevier B.V. All rights reserved.

1. Introduction

In IEEE 802.16m [2] wireless metropolitan networks (WMANs), the sleep mode is designed to save Mobile Stations (MSs)' energy consumption when they are serviced with lightly-loaded traffic. Basically, an MS in the sleep mode wakes up to receive downlink data packets and/or exchange signaling messages with the Base Station (BS) during a *listening window*, and powers down its transceiver during a *sleep window* to reduce unnecessary power consumption. The interleaved listening and sleep windows repeat every *sleep cycle* as long as the MS is in the sleep mode. While the 802.16m sleep mode inherits most features from the 802.16 [1] sleep mode, it also introduces a number of new advanced features [3–6]. Among them, two of the key enhancements are that (1) the listening window may be extended upon new packet arrivals; and

(2) the sleep cycle length may be reconfigured to better service the dynamic traffic condition.

The 802.16/m sleep mode has been well-studied in the previous works [3–10]. Specifically, [6–10] deal with the 802.16 sleep mode operations. The authors of [4] introduce an analytical model for the 802.16m sleep mode operation. However, their numerical analysis is based on an estimation obtained from an observation of simulation results, and hence, the numerical model may not be accurate. The authors of [3] provided a more accurate model for the case that realtime and non-realtime packets are serviced simultaneously. They assume that non-realtime packet service times follow an exponential distribution. In comparison, we provide a more general model in this paper, where non-realtime packet service times may follow any general distribution. Then, based on this model, we design a novel adaptive algorithm for the 802.16m sleep mode management.

Concretely, to fully exploit the new features available in the 802.16m sleep mode, we analyze the relation between traffic condition, sleep cycle length and power consumption

* Corresponding author at: ETRI, Daejeon 305-700, Republic of Korea.
E-mail address: sunggeun.jin1@gmail.com (S. Jin).

as well as average packet delay in the 802.16m sleep mode. We derive closed-form expressions for the power consumption and average packet delay, based on which we find the optimal value of the sleep cycle length that minimizes an MS's power consumption while satisfying a user-specified delay constraint. Then, we propose a novel adaptive sleep mode management (ASMM) scheme for 802.16m MSs. The basic idea of ASMM is to dynamically adjust the length of the sleep cycle of an MS based on online monitoring and estimation of the packet arrival rate. Specifically, ASMM doubles the sleep cycle length when there is no traffic in the previous cycle; otherwise, ASMM reconfigures the sleep cycle towards the optimal value obtained from the theoretical analysis.

The rest of the paper is organized as follows: In Section 2, we describe the 802.16m sleep mode operation in detail. In Section 3, we provide a comprehensive numerical analysis for the 802.16m sleep mode. In Section 4, we propose a management scheme for the 802.16m sleep mode operation, called ASMM, which is designed based on the numerical analysis derived in Section 3. The effectiveness of ASMM is demonstrated in Section 5 via simulation results. Finally, the paper concludes in Section 6.

2. IEEE 802.16m sleep mode

In this section, we summarize the key advancements in the 802.16m sleep mode. Firstly, in the 802.16 sleep mode, it is possible that the MS maintains a separate power saving class for each of its connections. As a result, multiple listening windows and sleep windows may overlap, which degrades the power saving efficiency of the system and defeats the purpose of having multiple power saving classes. For this reason, the 802.16m simplifies the sleep mode operation by offering a single power saving class for each MS. It implies that the MS maintains a single power saving class for all of its connections. Note that, different MS's in the 802.16m network may have different power saving classes. Traffics belonging to different connections are mixed together and served within the same power saving class.

In the 802.16 sleep mode, the listening window size is fixed. The second advancement in the new 802.16m sleep mode is that the listening window may be extended if there are data packets buffered at the BS for the MS. For example, during the second cycle shown in Fig. 1, the packets arriving during the initial listening window trigger the

window extension, while the packets arriving during the first extension of the listening window extend the window further.

The third advancement is related to sleep mode reconfiguration. As specified in the 802.16m standard, when there is no traffic destined for an MS, its sleep cycle length doubles every cycle till reaching and remaining at a maximum value. In this case, the listening window length remains the same. For example, in Fig. 1, the sleep cycle length doubles after the first cycle as there is no traffic during the first cycle. However, whenever there is traffic destined for the MS, the 802.16m sleep mode allows the sleep cycle to be either reset to a minimum value or reconfigured to an arbitrary value. The third cycle shown in Fig. 1 is an example of such reconfiguration. In 802.16m, sleep mode reconfiguration may be done without disrupting the ongoing power saving class. In other words, the 802.16m sleep mode allows the BS to convey the reconfiguration information to the MS via special signaling messages (e.g., AAI-SLP-REQ) during the regular sleep mode operation (e.g., at the beginning of a sleep cycle). This is different from the 802.16 sleep mode where the BS can only reconfigure the operational parameters of an MS after it deactivates the power saving class and via more complicated signaling exchange.

In this paper, we study the following typical operation of the 802.16m sleep mode:

- An AAI-TRF-IND signaling message is sent at the beginning of each listening window to inform an MS of the packets buffered at the BS for it.
- The listening window is extended as long as the transmission queue at the BS is nonempty.
- The entire listening window is fully utilized for packet transmissions.
- Sleep mode reconfiguration is done via special signaling messages without disrupting the ongoing power saving class, and the maximum length of the sleep cycle is a configurable parameter.

3. Analytical study

3.1. Problem statement

The power consumption of an 802.16m MS may be reduced via prolonging its sleep cycle. However, as shown in [3], a longer sleep cycle usually incurs higher packet

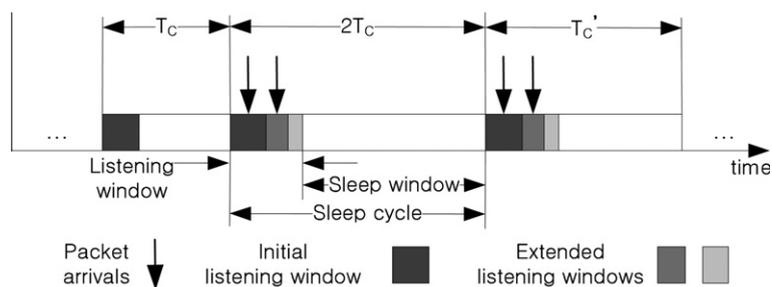


Fig. 1. An example of IEEE 802.16m sleep mode.

delays. So there is a tradeoff between power conservation and packet delay. The goal of this work is to design an adaptive sleep mode management scheme to adjust an 802.16m MS's sleep cycle (T_C) so that its power consumption (P) can be minimized while the average packet delay (D) is bounded below D_T – a delay upper bound specified by user or application. We first study the following optimization problem in this section:

$$\begin{aligned} \text{Given} \quad & \lambda, \tau, \sigma, P_L, P_S, \\ \text{find} \quad & T_C^\dagger, \\ \text{to minimize} \quad & P(T_C), \\ \text{subject to} \quad & D(T_C) \leq D_T, \end{aligned} \tag{1}$$

where (1) λ is the packet arrival rate, which can be monitored and estimated by the BS; (2) τ is the mean of packet transmission time; (3) σ is the deviation of packet transmission time; (4) P_L is the MS's power consumption during the listening window; (5) P_S is the MS's power consumption during the sleep window. Next, we derive closed-form expressions for $P(T_C)$ and $D(T_C)$ in Section 3.2, then discuss how to determine T_C^\dagger in Section 3.3, followed by numerical results in Section 3.4.

3.2. Derivation of $P(T_C)$ and $D(T_C)$

In order to determine T_C^\dagger , we need to derive the power consumption $P(T_C)$ and the average packet delay $D(T_C)$. We make the following assumptions in the analysis: (1) the network is in the steady state and the sleep cycle T_C remains constant; (2) packet arrivals follow a Poisson distribution with an expected packet arrival rate of λ ; (3) packet transmission time follows a general distribution with a mean of τ and a variance of σ^2 . For convenience, we summarize the notations used in the numerical analysis in Table 1.

3.2.1. Derivation of $P(T_C)$

To help analyze $P(T_C)$, we first define a random variable $t_p^{(i)}$. As shown in Fig. 2, $t_p^{(i)}$ represents the overall time for completing the transmissions of packet i and all subsequent packets that arrived during the transmissions of previously arrived packets. It is recursively defined as follows:

$$t_p^{(i)} = t_x + \sum_{j \in \mathcal{K}_i} t_p^{(j)}, \tag{2}$$

where t_x is the packet transmission time (for packet i) and \mathcal{K}_i is the set of packets that arrived during the transmission of packet i . $t_p^{(j)}$'s are mutually independent random variables with the same distribution. Let $k = |\mathcal{K}_i|$. Then, we have the Laplace transform of $t_p^{(i)}$ by:

$$\begin{aligned} E[e^{-st_p^{(i)}} | t_x = t, |\mathcal{K}_i| = k] &= \mathcal{E} \left[\exp \left(-s \left(t + \sum_{j \in \mathcal{K}_i} t_p^{(j)} \right) \right) \right] \\ &= e^{-st} \left(\mathcal{F}_p^*(s) \right)^k, \end{aligned} \tag{3}$$

where $F_p^*(s)$ is the Laplace–Stieltjes transform of t_p . Based on our assumptions on packet arrival and packet transmission time, we have:

Table 1
Notations used in the numerical analysis.

Notation	Meaning
t_L	Listening window size (ms)
t_S	Sleep window size (ms)
t_X	Packet transmission time (ms)
T_C	Length of sleep cycle (ms)
$P(T_C)$	Average power consumption with sleep cycle T_C (mW)
$D(T_C)$	Average packet delay with sleep cycle T_C (mW)
D_T	User-specified upper bound for average packet delay (ms)
$F_L(t)$	c.d.f. and L–S transform of random variable t_L
$F_L^*(s)$	
$F_S(t)$	c.d.f. and L–S transform of random variable t_S
$F_S^*(s)$	
$F_X(t)$	c.d.f. and L–S transform of random variable t_X
$F_X^*(s)$	
P_S, P_L	Power consumption during sleep window and listening window (mW)
τ	Mean of packet transmission time (ms)
σ^2	Variance of packet transmission time
λ	Packet arrival rate (1/ms)
ρ	Traffic intensity: $\rho = \lambda\tau$

$$\begin{aligned} F_p^*(s) &= E[e^{-st_p^{(i)}}] = \int_0^\infty \sum_{k=0}^\infty e^{-st} F_p^*(s)^k \frac{(\lambda t)^k}{k!} e^{-\lambda t} dF_X(t) \\ &= \int_0^\infty \exp(-t(s + \lambda - \lambda F_p^*(s))) dF_X(t) \\ &= F_X^*(s + \lambda - \lambda F_p^*(s)), \end{aligned} \tag{4}$$

where $F_X(t)$ and $F_X^*(s)$ are cumulative distribution functions and L–S transform of random variable t_X , respectively. Furthermore, since $E[t_X] = -dF_X^*(s)/ds|_{s=0} = \tau$ and $F_X^*(0) = F_p^*(0) = 1$, we have

$$E[t_p^{(i)}] = -dF_p^*(s)/ds|_{s=0} = \frac{\tau}{(1 - \rho)}. \tag{5}$$

The listening window, denoted by t_L , is the sum of the transmission time of all packets buffered during the previous sleep window. Hence, t_L can be written as:

$$t_L = \sum_{j \in \mathcal{N}} t_p^{(j)}, \tag{6}$$

where \mathcal{N} is the set of packets buffered during the previous sleep window t_S , and $t_p^{(j)}$ has the same physical meaning as in Eq. (2). Let $n = |\mathcal{N}|$. Since an AAI-TRF-IND signaling message is transmitted at the beginning of each listening window, the total number of packets to be transmitted is $n + 1$. Similar to Eqs. (3) and (4), we can derive the L–S transform of t_L as follows:

$$\begin{aligned} E[e^{-st_L} | t_S = t, |\mathcal{N}| = n] &= E \left[\exp \left(-s \left(t_p^{(\text{AAI-TRF-IND})} + \sum_{j \in \mathcal{N}} t_p^{(j)} \right) \right) \right] \\ &= \left(F_p^*(s) \right)^{n+1}, \end{aligned} \tag{7}$$

and

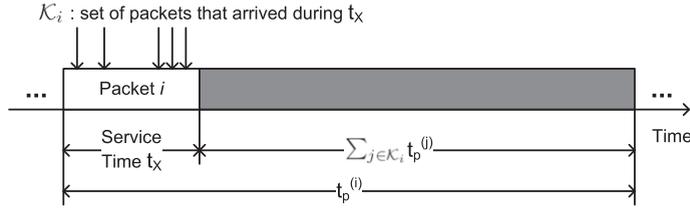


Fig. 2. Illustration of $t_p^{(i)}$.

$$\begin{aligned}
 F_L^*(s) &= E[e^{-st_L}] = \int_0^\infty \sum_{n=0}^\infty F_p^*(s)^{n+1} \frac{(\lambda t)^n}{n!} e^{-\lambda t} dF_S(t) \\
 &= \int_0^\infty F_p^*(s) \exp(\lambda - \lambda F_p^*(s)) dF_S(t) \\
 &= F_p^*(s) F_S^*(\lambda - \lambda F_p^*(s)),
 \end{aligned} \tag{8}$$

where $F_S(t)$ and $F_S^*(s)$ are cumulative distribution function and L-S transform of sleep window size t_S , respectively. From Eq. (8), we can obtain the expected listening window size $E[t_L]$ by:

$$E[t_L] = -dF_L^*(s)/ds|_{s=0} = \frac{\tau}{(1-\rho)}(1 + \lambda E[t_S]), \tag{9}$$

where τ and $E[t_S]$ are the expectations of random variables t_x and t_S , respectively. In the steady state, the length of the sleep cycle is $T_C = t_L + t_S$, which, together with Eq. (9), gives the following results:

$$E[t_L] = \rho T_C + \tau, \tag{10}$$

$$E[t_S] = (1 - \rho)T_C - \tau. \tag{11}$$

The average power consumption $P(T_C)$ can then be derived as:

$$\begin{aligned}
 P(T_C) &= (P_L E[t_L] + P_S E[t_S])/T_C \\
 &= (P_L - P_S)\rho + (P_L - P_S)\tau/T_C + P_S,
 \end{aligned} \tag{12}$$

where P_L and P_S are power consumption during listening window and sleep window, respectively, as listed in Table 1.

3.2.2. Derivation of $D(T_C)$

We now consider the average packet delay $D(T_C)$. Let us first introduce a Markov chain with a random variable Q defined as the number of packets in the transmission queue at the end of each packet transmission. The one-step

state transition probability for this Markov chain is $p_{jk} = Pr(Q_{n+1} = k - Q_n = j)$, which, when $k \geq j - 1$, is indeed the probability that there are $k - j + 1$ packet arrivals between the ends of the previous and current packet transmissions. As shown in Fig. 3, the time interval between the ends of two adjacent packet transmissions may include (1) a sleep window and one packet transmission time (in this case, the previous state is $Q = 0$ since there is no packet in the queue at the end of a listening window when the network is in the steady state); or (2) one packet transmission time only. Note that there is at least one packet to be transmitted at each listening window: the AAI-TRF-IND signaling message. Therefore, if there are no packets arriving during the previous sleep window as well as the AAI-TRF-IND packet transmission time, the queue length is zero at the end of the AAI-TRF-IND transmission. Based on the above analysis, p_{jk} can be derived as:

$$p_{jk} = \begin{cases} \sum_{m=0}^k \alpha_m \beta_{k-m}, & \text{if } j = 0, k \geq 0, \\ \beta_{k-j+1}, & \text{if } k \geq j - 1, \\ 0, & \text{if } j \geq 1, 0 \leq k < j - 1, \end{cases} \tag{13}$$

where α_m is the probability that m packets arrive during the previous sleep window while β_m is the probability that m packets arrive during a packet transmission time. In the following we provide some analysis on α_m as well as β_m , then find the steady-state probabilities of the defined Markov chain. Firstly, α_m can be written as:

$$\alpha_m = \int_0^\infty \frac{(\lambda t)^m}{m!} e^{-\lambda t} dF_S(t), \tag{14}$$

where $F_S(t)$ is the cumulative distribution function of the sleep window size t_S . We define $\alpha(z) = \sum_{m=0}^\infty \alpha_m z^m$. From this equation, $\alpha(z)$ is derived by:

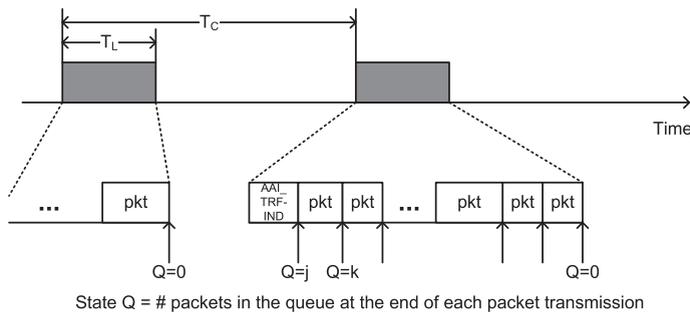


Fig. 3. Derivation of p_{jk} – the probability that the number of packets in the transmission queue changes from j to k between the ends of two adjacent packet transmissions.

$$\begin{aligned}\alpha(z) &= \sum_{m=0}^{\infty} \alpha_m z^m = \int_0^{\infty} \sum_{m=0}^{\infty} \frac{(\lambda t)^m}{m!} e^{-\lambda t} z^m dF_S(t) \\ &= F_S^*(\lambda - \lambda z),\end{aligned}\quad (15)$$

where $F_S^*(t)$ is the L-S transform of the sleep window size t_S . Similarly, we can define $\beta(z) = \sum_{m=0}^{\infty} \beta_m z^m$ and obtain $\beta(z) = F_X^*(\lambda - \lambda z)$ where $F_X^*(t)$ is the L-S transform of a single packet transmission time t_X .

Now, we analyze the steady-state probability π_k – the probability that k packets are left in the queue at the end of a packet transmission. π_k can be calculated with the state transition probabilities in the following way:

$$\pi_k = \pi_j \sum_{j=0}^{\infty} p_{jk} = \pi_0 \sum_{m=0}^k \alpha_m \beta_{k-m} + \sum_{j=1}^{k+1} \pi_j \beta_{k-j+1}. \quad (16)$$

We define $\Pi(z) = \sum_{k=0}^{\infty} \pi_k z^k$ and it is derived by:

$$\begin{aligned}\Pi(z) &= \pi_0 \sum_{m=0}^{\infty} \alpha_m z^m \sum_{k=m}^{\infty} \beta_{k-m} z^{k-m} + \sum_{j=1}^{\infty} \pi_j z^{j-1} \\ &\quad \times \sum_{k=j-1}^{\infty} \beta_{k-j+1} z^{k-j+1} \\ &= \pi_0 \alpha(z) \beta(z) + \frac{\Pi(z) - \pi_0}{z} \beta(z).\end{aligned}\quad (17)$$

By substituting $\alpha(z)$ and $\beta(z)$ with $F_S^*(\lambda - \lambda z)$ and $F_X^*(\lambda - \lambda z)$, respectively, we have:

$$\Pi(z) = \frac{\pi_0 [1 - z F_S^*(\lambda - \lambda z)] F_X^*(\lambda - \lambda z)}{F_X^*(\lambda - \lambda z) - z}. \quad (18)$$

Since $\Pi(1) = 1$, π_0 can be obtained by applying the L'Hôpital's rule:

$$\pi_0 = \frac{1 - \rho}{1 + \lambda E[t_S]}. \quad (19)$$

From Eqs. (18) and (19), the average queue length $E[L]$ is derived by:

$$\begin{aligned}E[L] &= \left. \frac{d\Pi(z)}{dz} \right|_{z=1} \\ &= \frac{(1 - \rho)(\lambda^2 E[t_S^2] + 2\lambda E[t_S]) + 2\rho(1 - \rho)\tilde{N} + \lambda^2 \tilde{N} E[t_X^2]}{2\tilde{N}(1 - \rho)},\end{aligned}\quad (20)$$

where $\tilde{N} = 1 + \lambda E[t_S]$, $E[t_S^2] = E[t_S]^2 + \text{Var}(t_S)$, and $E[t_X^2] = \tau^2 + \sigma^2$.

Since $T_C = t_S + t_L$ in the steady state, we can derive the variance of t_S (i.e., $\text{Var}(t_S)$) as:

$$\text{Var}(t_S) = \text{Var}(-t_L + T_C) = \text{Var}(t_L). \quad (21)$$

Moreover, since in the steady state, the total number of transmitted packets is the same as the total number of packet arrivals during a sleep cycle, the variance of t_L is:

$$\text{Var}(t_L) = (\lambda T_C + 1) \text{Var}(t_X) = (\lambda T_C + 1) \sigma^2. \quad (22)$$

Now, we have $E[t_S^2]$ as:

$$E[t_S^2] = E[t_S]^2 + (\lambda T_C + 1) \sigma^2. \quad (23)$$

According to Little's Law, $D(T_C)$ can be expressed by:

$$D(T_C) = E[L]/\lambda. \quad (24)$$

Now we have completed the derivation of the closed-form expressions for $P(T_C)$ and $D(T_C)$, which are the foundation of our proposed sleep mode management scheme that will be discussed in detail in Section 4.

3.3. Calculation of optimal sleep cycle T_C^\dagger

With $P(T_C)$ and $D(T_C)$ obtained in the previous section, we can calculate T_C^\dagger with a typical approach using the Lagrangian dual function. In other words, we can (1) take the first-order derivative of $P(T_C) + \nu(D(T_C) - D_T)$ with regard to T_C ; (2) obtain a real-value solution (T_C') to the Lagrangian function by setting the derivative to zero; and then (3) obtain T_C^\dagger by rounding down T_C' to the nearest integer value that is a multiple of 5 ms, since the 802.16m standard mandates that T_C is in the unit of frames and each frame is 5 ms. However, it is difficult, if not impossible, to derive a closed-form expression for the first-order derivative of the Lagrangian function with regard to T_C (particularly the $D(T_C)$ part).

In this work, we instead take a numerical approach to obtain T_C^\dagger by enumerating all possible T_C values and finding the one that yields the lowest power consumption while satisfying the delay constraint. Since T_C can only take discrete values, the number of choices for T_C is limited. Hence, such a numerical approach is simple and efficient.

3.4. Numerical results

In this section, we provide some numerical results for $P(T_C)$ and $D(T_C)$, and then show how the values of T_C^\dagger and $P(T_C^\dagger)$ are affected by the traffic intensity ρ and the delay upper bound D_T .

Fig. 4 plots the results for $P(T_C)$ and $D(T_C)$ with varying ρ and T_C values. We have the following observations. Firstly, when ρ increases (under a fixed T_C), the power consumption increases however the packet delay decreases. This is because a larger ρ indicates a higher traffic rate, hence the listening window gets extended more to serve the data packets. As a result, the sleep window becomes smaller during which less packets are queued up. Secondly, as T_C increases (with a fixed ρ), the power consumption decreases but the packet delay increases. This is because a larger T_C reduces the per-sleep cycle AAI-TRF-IND signaling overhead, hence conserving more power; however it also yields a larger sleep window during which more packets are queued up and have to wait for a longer time before being served.

Fig. 5 plots the results for T_C^\dagger and $P(T_C^\dagger)$ with varying ρ and D_T values. We can see that T_C^\dagger increases with ρ . This is because a higher ρ indicates a higher traffic rate, hence the MS spends more time serving the data traffic. To achieve this goal, the MS needs to increase the sleep cycle to reduce the signaling overhead. Another observation is that T_C^\dagger also increases with D_T . This is because a less stringent delay constraint allows the MS to operate with a larger sleep cycle (hence a larger sleep window), which in turn reduces the signaling overhead and the average power consumption.

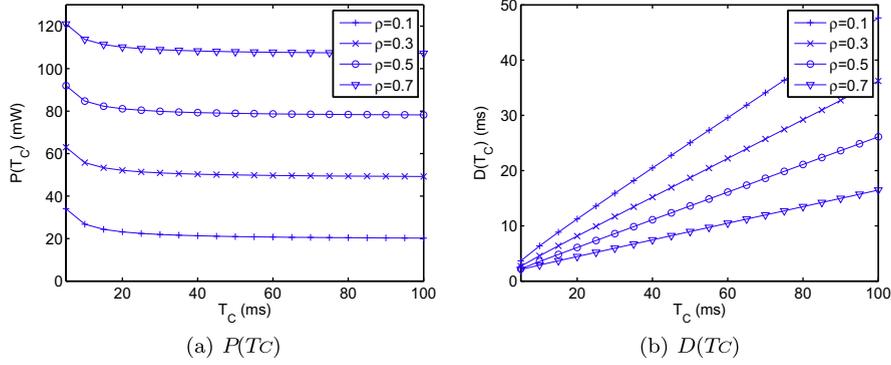


Fig. 4. Numerical results for $P(T_C)$ and $D(T_C)$ with varying ρ and T_C values. We assume that $\tau = 0.5$ ms, $\sigma = 0.25$ ms, $P_A = 150$ mW and $P_S = 2$ mW.

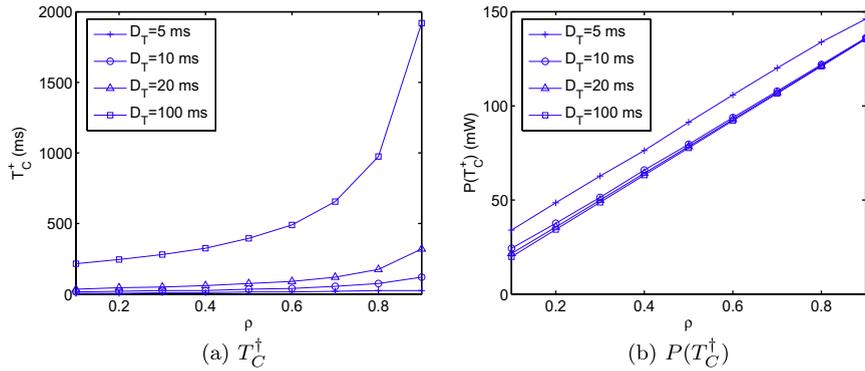


Fig. 5. Numerical results for T_C^* and $P(T_C^*)$ with varying ρ and D_T values. We assume that $\tau = 0.5$ ms, $\sigma = 0.25$ ms, $P_A = 150$ mW and $P_S = 2$ mW.

4. The proposed ASMM scheme

In this section, we present our proposed adaptive sleep mode management scheme, called ASMM. ASMM is executed at the BS. It (1) monitors downlink packets for each MS and estimates the packet arrival rate online, based on which (2) it adjusts the sleep cycle for the MS and uses the AAI-SLP-REQ signaling message to notify the station of the new sleep cycle length. Once the MS receives AAI-SLP-REQ, it starts the next sleep cycle following the BS's instruction.

4.1. Estimation of the packet arrival rate

ASMM employs a modified moving average algorithm to estimate the packet arrival rate based on the observed packet arrival times. Let T_n denote the time instance when the n th downlink packet for the MS arrives at the BS. Let λ_n denote the estimation of the packet arrival rate at time T_n . ASMM estimates λ_n in the following manner:

$$\lambda_n = \begin{cases} \lambda_{n-1}(1-w) + \frac{1}{T_n - T_{n-1}}w, & \text{if a packet arrives within} \\ & \text{the last } \frac{1}{\lambda_{n-1}} \text{ time,} \\ \frac{\lambda_{n-1}}{2}, & \text{if no packet arrives within} \\ & \text{the last } \frac{1}{\lambda_{n-1}} \text{ time,} \end{cases} \quad (25)$$

where $w(0 < w < 1)$ is a weighting factor. Basically, upon a packet arrival, λ is updated with a weighted moving average. If there is no packet arrival during a certain time period (i.e., $\frac{1}{\lambda}$), λ is reduced to one half of its previous value. Initially, we set $\lambda_0 = 0$ and $T_0 = 0$.

In the case when there exist multiple connections with different delay requirements, we set D_T to be the most stringent delay requirement in our proposed scheme since the 802.16m standard only allows a single power saving class for each mobile station.

4.2. Adjustment of the sleep cycle

At the end of each sleep cycle, ASMM makes an adjustment to the sleep cycle length and the BS notifies the MS of the adjustment via the AAI-SLP-REQ message. As shown in Fig. 6, there are two operating states for adjusting the sleep cycle length:

- S_1 : when there are no packet arrivals during the previous sleep cycle, the sleep cycle length will be doubled; if the sleep cycle length has reached the maximum value T_C^{\max} , it remains unchanged.
- S_2 : when some packets arrive during the previous sleep cycle, the sleep cycle length will be reconfigured. We assume that τ and σ of the packet transmission time as well as the power consumptions P_A and P_S are known

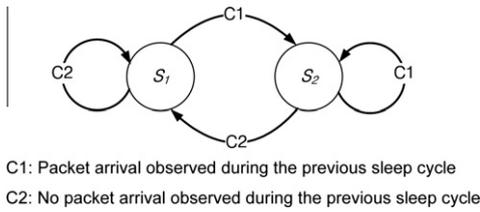


Fig. 6. State transition diagram in ASMM.

a priori. Then the length of the next sleep cycle is set to the optimal value T_C^i based on the most recently estimated packet arrival rate λ , i.e., the solution to the optimization problem specified by Eq. (1) in Section 3. Moreover, the maximum value for the sleep cycle length is updated to $T_C^{\max} = \gamma T_C^i$, where $\gamma \geq 1$ is a design parameter. We will evaluate the effects of γ in Section 5.

To help further understand how ASMM works, we show an example scenario in Fig. 7. In this example, there are no packet arrivals during the first three cycles; therefore, the sleep cycle length keeps doubling (from 5 ms to 10 ms to 20 ms to 40 ms). In comparison, since packets arrive during the next three cycles, the sleep cycle length is reconfigured at the end of each one (to 15 ms, 10 ms and 30 ms, respectively). Then the sleep cycle length is doubled at time 130 ms as no packets arrive during the seventh cycle.

5. Performance evaluation

We evaluate the effectiveness of our proposed ASMM scheme using simulations. We compare ASMM with a naive scheme which (1) doubles the sleep cycle T_C when there is no traffic in the previous cycle; and (2) resets T_C to 10 ms otherwise. T_C^{\max} is set to 5.115 s (= 5 ms \times 1023) – the maximum allowed by the IEEE 802.16m standard. The following parameters are used in the simulations: $\tau = 0.5$ ms, $\sigma = 0.25$ ms, $P_A = 150$ mW and $P_S = 2$ mW, which are also used in [11].

5.1. Effects of γ in ASMM

We evaluate the effects of γ – the design parameter in ASMM that determines the maximum sleep cycle for an

MS. Fig. 8 compares the performances of ASMM with different γ values under different traffic arrival rates (λ) and delay constraints (D_T).

It is very interesting to see that, when γ has a large value (i.e., ≥ 2), the average packet delay is indeed higher than the delay bound, although ASMM is designed to converge to the optimal point where the delay can be bounded while the power consumption is minimized. This counterintuitive observation is surprising at first sight, but rather reasonable for the following reason. In ASMM, when there are no packet arrivals during a sleep cycle, the length of the sleep cycle will be doubled. As a result, when packets start arriving after a long idle period, the sleep cycle and hence the sleep window could be very large, which in turn results in large delivery delay for these newly-arrived packets. As shown in the figure, this phenomenon is particularly salient when the traffic load is light, since the MS may experience a longer idle period under light traffic conditions than when the traffic load is heavy. On the other hand, when γ takes a smaller value less than two, the delay constraint can be better satisfied, since the maximum length of the sleep cycle and hence the sleep window is effectively limited. Based on these observations, we empirically set γ to 1.5 in ASMM.

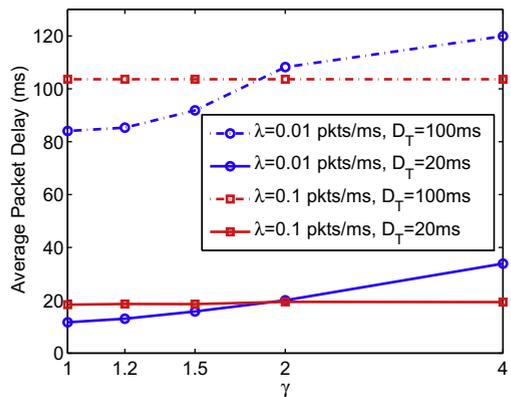


Fig. 8. The effects of γ in ASMM.

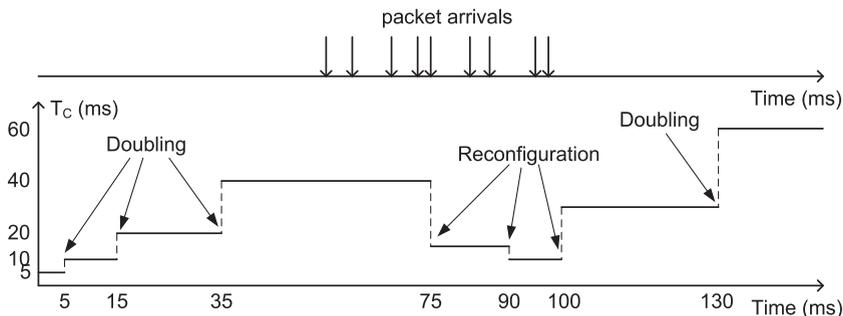


Fig. 7. An example of the operation of the proposed ASMM scheme.

Table 2
Performance comparison with different λ values.

λ (packets/ms)	0.01		0.1	
	P (mW)	D (ms)	P (mW)	D (ms)
Naive scheme	3.33189	120.907	11.1246	13.5286
ASMM with $D_T = 100$ ms	2.87478	91.8521	9.34492	103.607
ASMM with $D_T = 20$ ms	4.82035	15.7767	9.6383	18.5469

5.2. Performance comparison

We simulate ASMM and the naive scheme under different traffic conditions: $\lambda = 0.01$ (light traffic) and $\lambda = 0.1$ (heavy traffic). For ASMM, we set different delay upper bounds: $D_T = 100$ ms (loose delay constraint) and $D_T = 20$ ms (tight delay constraint). Results are summarized in Table 2 and we have the following observations.

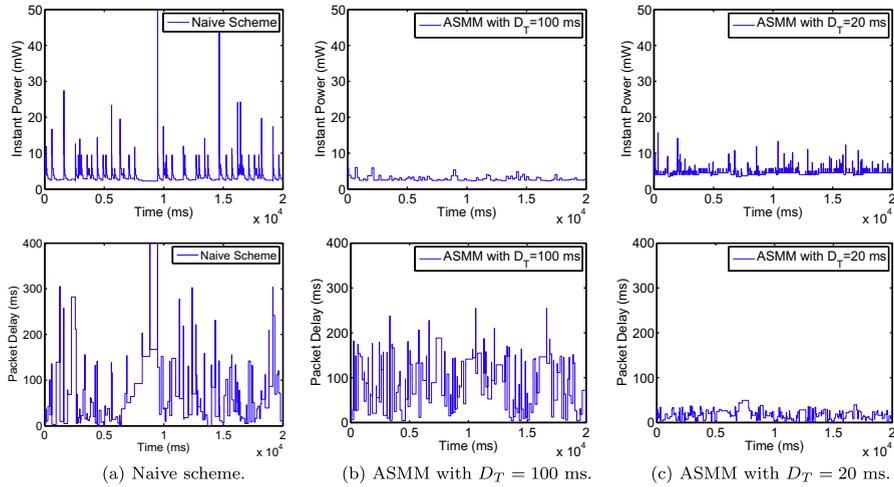


Fig. 9. Performance comparison with $\lambda = 0.01$ packets/ms.

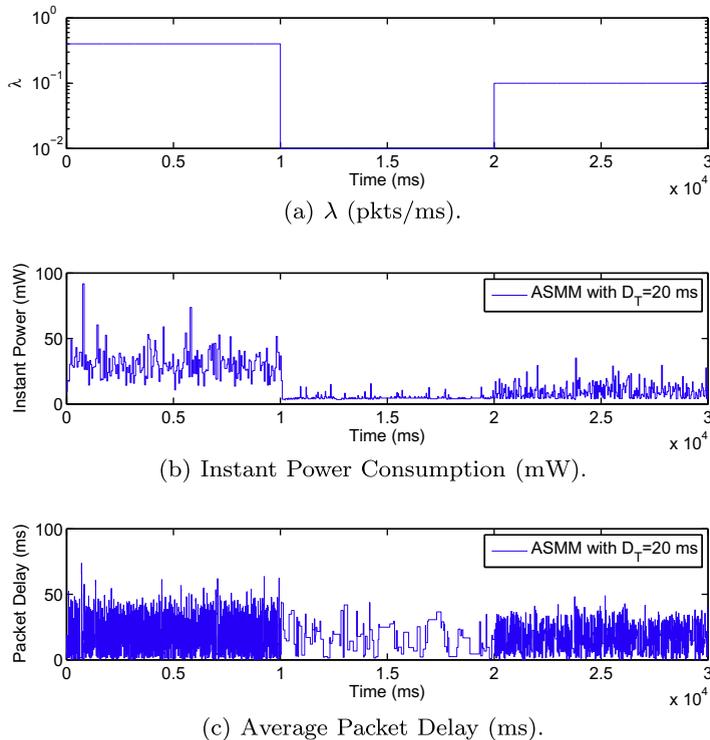


Fig. 10. ASMM behavior with λ varying from 0.4 to 0.01 to 0.1 over time.

When traffic load is light, the naive scheme yields a high packet delay (around 120 ms). This is because the naive scheme simply doubles or resets the sleep cycle length without considering the delay performance. In comparison, ASMM takes into consideration the delay constraint when adjusting the sleep cycle. Hence, the delay constraint is always satisfied. We can see that, when the delay constraint is tight at $D_T = 20$ ms, more power is consumed as the tradeoff. However, when the delay constraint is loose at $D_T = 100$ ms, ASMM yields a better performance in both power consumption and packet delay than the naive scheme. This is due to the relatively stable and large sleep cycle used by ASMM, comparing to the large variation range of the sleep cycle (from 10 ms to 5.115 s) in the naive scheme.

When traffic load is heavy, the naive scheme is busy serving data packets and rarely has the chance to double the sleep cycle. As a result, the packet delay is low while the power consumption is high. In comparison, ASMM carefully adjusts the sleep cycle for the MS so that the MS may sleep for a longer period (with a larger sleep cycle hence a large sleep window) to save more power while making sure that the delay constraint is *satisfied*. As shown in the table, ASMM allows a higher packet delay (than the naive scheme) to be around the delay constraint while saving about 15% power.

We investigate the behavior of ASMM and the naive scheme further by plotting their instant power consumption and average packet delay (in each sleep cycle) when $\lambda = 0.01$ in Fig. 9. In this figure, we can clearly observe (1) high power consumption and unpredictable delay performance with the naive scheme; (2) ASMM's packet delay around the desired delay constraint; and (3) low power consumption by ASMM.

Finally, we study the behavior of ASMM when the traffic rate varies over time. As shown in Fig. 10, λ is initially set to 0.4 in the simulation, then reduces to 0.01 at time 10 s, and then changes to 0.1 at time 20 s. Traces for instance power consumption as well as average packet delay show that ASMM is able to track the variation of λ well and adjust the sleep cycle accordingly. Over the entire 30 s period, the average packet delay is always around the target 20 ms, while the sleep cycle is adjusted over time to conserve as much power as possible. For example, between time instances 10 s and 20 s, larger sleep cycles are used (which is evidenced by the sparse delay trace in the period) to deal with the light traffic condition and allow the MS to sleep longer to save more power (which is evidenced by the low power trace in the period).

6. Conclusion

In this paper, we investigate the sleep mode management issues in IEEE 802.16m WMANs. We conduct a comprehensive analytical study on the power consumption and average packet delay for 802.16m MSs. Based on the analytical results, we propose a novel adaptive sleep mode management scheme called Adaptive Sleep Mode Manage-

ment (ASMM) to minimize the power consumption of an 802.16m MS while satisfying a user-specified packet delay constraint. Simulation results show that ASMM performs well under various traffic conditions and delay constraints, particularly when the traffic load is light and the delay constraint is less stringent.

Acknowledgements

This work was in part supported by the MKE/KEIT, Korea, under both the ITRC support program (IITA-2008-C1090-0801-0013) and the IT R&D program (KI002139, Development of cooperative operation profiles in multicell wireless systems). The research reported in this paper was supported in part by the Information Infrastructure Institute (iCube) of Iowa State University and the National Science Foundation under Grant CNS 0831874.

References

- [1] IEEE 802.16-2009, Part 16: Air Interface for Broadband Wireless Access Systems, May 2009.
- [2] IEEE 802.16m/D7, Part 16: Air Interface for Broadband Wireless Access Systems: Advanced Air Interface, July 2010.
- [3] S. Jin, M. Choi, S. Choi, Performance analysis of IEEE 802.16m sleep mode for heterogeneous traffic, IEEE Commun. Lett., May 2010.
- [4] E. Hwang, K.J. Kim, J.J. Son, B.D. Choi, The power-save mechanism with periodic traffic indications in the IEEE 802.16e/m, IEEE Trans. Veh. Tech., January 2010.
- [5] S. Baek, J.J. Son, B.D. Choi, Performance analysis of sleep mode operation for IEEE 802.16m advanced WMAN, in: Proc. ICC'09, June 2009.
- [6] R.K. Kalle, M. Raj, D. Das, A Novel Architecture for IEEE 802.16m subscriber station for joint power saving class management, in: Proc. COMSNETS'09, January 2009.
- [7] T.-C. Chen, J.-C. Chen, Y.-Y. Chen, Maximizing unavailability interval for energy saving in IEEE 802.16e wireless MANs, IEEE Trans. Mobile Computing, April 2009.
- [8] L. Kong, D.H.K. Tsang, Optimal selection of power saving classes in IEEE 802.16e, in: Proc. IEEE WCNC'07, March 2007.
- [9] Y. Zhang, Performance modeling of energy management mechanism in IEEE 802.16e mobile WiMAX, in: Proc. IEEE WCNC'07, March 2007.
- [10] K. Han and S. Choi, Performance analysis of sleep mode operation in IEEE 802.16e mobile broadband wireless access systems, in: Proc. IEEE VTC'06-Fall, September 2006.
- [11] <<http://www.wavesat.com/pdf/OD-8500-IC-PB.pdf>>.



Sunggeun Jin is a senior engineer working for ETRI, which he joined in 1998, Korea. Prior to joining ETRI, he received his B.S. and M.S. degrees in School of Electrical Engineering and Computer Science at Kyungpook National University (KNU), Korea, in 1996 and 1998, respectively. He received his Ph.D. at School of Electrical and Computer Engineering, Seoul National University (SNU), Korea, August, 2008. He has participated in standard developments including IEEE 802.11v, IEEE 802.16j, IEEE 802.16m, and IEEE 802.11ad. He has served as a TPC member for IEEE WCNC 2008, ICUFN 2009, ICST BROADNETS 2010, and IEEE GLOBECOM 2011. Also, he completed peer reviews for journals and conferences such as IEEE TMC, IEEE INFOCOM, IEEE ICC, IEEE GLOBECOM, and IEEE WCNC.



Xi Chen received the B.S. degree in Electrical Engineering and Information Science from University of Science and Technology of China, Hefei, China, in 2006. He is currently working toward a Ph.D. degree in the Department of Electrical and Computer Engineering, Iowa State University, Ames, Iowa. His research interests include protocol design and performance evaluation for 802.11-based wireless/mobile networks. He is a student member of the IEEE.



Daji Qiao is currently an Associate Professor in the Department of Electrical and Computer Engineering, Iowa State University, Ames, Iowa. He received his Ph.D. degree in Electrical Engineering: Systems from The University of Michigan, Ann Arbor, Michigan, in February 2004. His current research interests include algorithm and protocol innovation and implementation for IEEE 802.11 wireless local area networks, system modeling and performance analysis of wireless sensor networks, and pervasive computing applications. He is a

member of the IEEE and ACM.



Sunghyun Choi is an associate professor at the School of Electrical Engineering, Seoul National University (SNU), Seoul, Korea. Before joining SNU in September 2002, he was with Philips Research USA, Briarcliff Manor, New York, USA as a Senior Member Research Staff and a project leader for three years. He was also a visiting associate professor at the Electrical Engineering department, Stanford University, USA from June 2009 to June 2010. He received his B.S. (summa cum laude) and M.S. degrees in electrical engineering from

Korea Advanced Institute of Science and Technology (KAIST) in 1992 and 1994, respectively, and received Ph.D. at the Department of Electrical

Engineering and Computer Science, The University of Michigan, Ann Arbor in September, 1999.

His current research interests are in the area of wireless/mobile networks with emphasis on wireless LAN/MAN/PAN, next-generation mobile networks, mesh networks, cognitive radios, resource management, data link layer protocols, and cross-layer approaches. He authored/coauthored over 140 technical papers and book chapters in the areas of wireless/mobile networks and communications. He has co-authored (with B. G. Lee) a book "Broadband Wireless Access and Local Networks: Mobile WiMAX and WiFi," Artech House, 2008. He holds 19 US patents, 10 European patents, and 11 Korea patents, and has tens of patents pending. He has served as a General Co-Chair of COMSWARE 2008, and a Technical Program Committee Co-Chair of ACM Multimedia 2007, IEEE WoWMoM 2007 and IEEE/Create-NetCOMSWARE 2007. He was a Co-Chair of Cross-Layer Designs and Protocols Symposium in IWCMC 2006, 2007, and 2008, the workshop co-chair of WILLOPAN 2006, the General Chair of ACM WMASH 2005, and a Technical Program Co-Chair for ACM WMASH 2004. He has also served on program and organization committees of numerous leading wireless and networking conferences including ACM MobiCom, IEEE INFOCOM, IEEE SECON, IEEE MASS, and IEEE WoWMoM. He is also serving on the editorial boards of IEEE Transactions on Mobile Computing, IEEE Wireless Communications, ACM SIGMOBILE Mobile Computing and Communications Review (MC2R), Journal of Communications and Networks (JCN), Computer Networks, and Computer Communications. He has served as a guest editor for IEEE Journal on Selected Areas in Communications (JSAC), IEEE Wireless Communications, Pervasive and Mobile Computing (PMC), ACM Wireless Networks (WINET), Wireless Personal Communications (WPC), and Wireless Communications and Mobile Computing (WCMC). He gave a tutorial on IEEE 802.11 in ACM MobiCom 2004 and IEEE ICC 2005. From 2000 to 2007, he was a voting member of IEEE 802.11 WLAN Working Group.

He has received a number of awards including the Young Scientist Award awarded by the President of Korea (2008); IEEE/IEEE Joint Award for Young IT Engineer (2007); the Outstanding Research Award (2008) and the Best Teaching Award (2006) both from the College of Engineering, Seoul National University; the Best Paper Award from IEEE WoWMoM 2008; and Recognition of Service Award (2005, 2007) from ACM. Dr. Choi was a recipient of the Korea Foundation for Advanced Studies (KFAS) Scholarship and the Korean Government Overseas Scholarship during 1997-1999 and 1994-1997, respectively. He is a senior member of IEEE, and a member of ACM, KICS, IEK, KIISE.