Sleep Scheduling That Minimizes State Transitions for IEEE 802.16e Mobile Subscriber Stations

Li-Hsing Yen
Dept. Computer Science & Information Engineering
National University of Kaohsiung
Kaohsiung, Taiwan 811, R.O.C.
Email: lhyen@nuk.edu.tw

Chih-Yuan Cheng, Chun-Hsin Wang
Dept. Computer Science & Information Engineering
Chung Hua University
Hsinchu, Taiwan 300, R.O.C.
Email: {m09502019,chwang}@chu.edu.tw

***Abstract*—IEEE 802.16e devices may enter sleep mode to conserve energy. There has been some work studying how to schedule device's packet transmissions to maximize the device's sleep period while meeting associated QoS (quality of service) requirements. This paper proposes two schemes that attempt to save more energy by further reducing the frequency of transitions from sleep to active mode (state) in a sleep schedule. Performance evaluation results indicate that the proposed schemes effectively reduce state transitions when compared with existing sleep scheduling schemes.**

***Keywords*-WiMAX; sleep scheduling; QoS; state transitions**

I. INTRODUCTION

IEEE 802.16e [1] or WiMAX is a standard for wireless Metropolitan Area Networks which provides broadband wireless access services to roaming users. Besides high-speed data transmissions, the current version of IEEE 802.16e supports handoffs of user equipments, security enhancement, and power-saving techniques.

In a WiMAX network, a mobile subscriber station (MSS) may simultaneously establish several connections with the serving base station (BS). Each connection can have its own QoS (quality of service) demand. Energy saving is a critical issue for battery-powered MSSs. To conserve energy, an MSS may turn off its transceiver (i.e., entering sleep mode) when communication services are not needed. IEEE 802.16e defines three power-saving classes (Types I, II, and III) that provide different sleep mode activation patterns to meet different QoS demands. Type I power-saving class alternates sleep and listening intervals. In listening intervals, an MSS is made aware of whether there is any packet queued at the BS that is addressed to it. If there is such packet, the MSS recovers from sleep mode to retrieve the packet. Otherwise, the MSS enters sleep mode again with a doubled sleep interval. This design aims at minimizing energy consumption on idle listening for non-realtime applications. Type II class also alternates sleep and listening intervals but it allows data exchange during the listening interval without leaving sleep mode. The ratio of listening interval to the whole is fixed to accommodate periodic packets from real-time applications. Type III class lets MSS sleep for a predefined period and enter back to active mode. It suits best for idle MSSs where only periodic ranging singaling is required.

The ratio of sleep period to the whole, referred to as *sleep ratio*, has been used to measure the efficiency of power-saving mechanisms. Achievable sleep ratio depends on packet generation/arrival patterns and power-saving class in use. When an MSS enters sleep mode, all packets destined for the MSS will be buffered in the BS serving the MSS. These packets will be sent to the MSS after the MSS goes backs to active mode. Therefore, the activation of sleep mode saves energy on one hand while increases packet delivery latency on the other hand. The length of the latency is in general proportional to the length of the sleep period.

In the literature, some studies have considered sleep scheduling for multiple MSSs [2], [3], [4], which demands that the active period of each MSS must be exclusive. Some studies have considered sleep scheduling for one or more CBR (Constant Bit Rate) connection within an MSS. Jang et al. [5] investigated the application of Type I power-saving class to a single CBR connection. Other researchers considered applying Type II power-saving class to an MSS with multiple CBR connections [6], [7]. This attempt is feasible only when the length of sleep interval is less than the minimal tolerable packet delays of all connections [7]. If connections have different characteristics, Type II power-saving class gives rise to low sleep ratio. To improve, sleep and listening intervals should be changed dynamically. Aperiodic On-Off Scheme (AS) [6] and Minimum Wakeup Time (MWT) [2] are two protocols proposed for this problem.

Both AS and MWT do not consider energy spent on transitions between sleep and active modes. When a transceiver switches back from sleep mode to working mode, it needs some time (called startup time) to scaleup its internal clock rates and optionally perform frequency calibration. Thus a mode (state) transition takes time and also extra energy.

The purpose of this work is to reduce state transitions by merging two separated active periods into one. We consider multiple CBR connections within an MSS that may have diverse traffic patterns and QoS requirements. The proposed schemes take sleep schedules yielded by AS or MWT, and attempt to further reduce state transitions while respecting packet delay constraints. We conducted extended simulations to investigate the performance of the proposed schemes. The results indicate that the proposed schemes achieve the design goal when compared with existing sleep scheduling schemes.

The remainder of this paper is organized as follows. The next section reviews related work and points out the limitation of existing approaches. Sec. III presents two approaches to state transition reductions. Experimental results are presented in Sec. IV. The last section concludes this paper.

II. PRELIMINARIES

Both BS and MSS can initiate sleep mode. If MSS initiates the sleep mode, it should send request messages to BS to coordinate related parameters such as initial sleep period and maximum sleep interval.

Transmission paths between a BS and associated MSS can be uplink (from MSS to BS) or downlink (from BS to MSS). In this work, we assume 802.16e Time Division Duplex (TDD) mode, where data transmissions within a frame are divided into two parts, one for downlink and the other for uplink. For applications that demand two-way CBR traffic such as voice over IP (VoIP), we need consider only one direction of data transmissions (either downlink or uplink). The scheduling of the other part is symmetric.

In the literature, there has been some work on energy efficiency improvement for generic BS-based [8] and IEEE 802.11 [9] wireless networks. For IEEE 802.16e networks, some researches have analyzed the performance of sleep mode operations [10], [11], [12]. Existing sleep scheduling schemes can be classified into two types. The first type considers the scheduling among multiple MSSs where only a single downlink or uplink channel is assumed. As all MSSs contend for the exclusive use of the communication channel, the problem is to arrange an non-overlapping schedule that minimizes energy consumption. In Ref. [3], an MSS in sleep mode should be periodically awaked to guarantee a minimal data rate that serves as a QoS requirement. On the other hand, an MSS in the awake state should complete its transmission as soon as possible to minimize the time and thus the energy spent in idle waiting. The authors proposed to allocate almost all the channel bandwidth to one MSS (called primary MSS) and just enough bandwidth to other MSSs (called secondary MSSs) to guarantee their minimum data rate requirements. The goal is to minimize the primary MSS's state transitions and the idle waiting time of secondary MSSs. In a follow-up work the authors considered the same setting with additional inclusion of multicast messages in the schedule [4].

The abovementioned work does not explicitly specify which power-saving class is used. MMPS (Multiple MSSs Power-Saving Scheduler) [2] considered non-overlapping sleep scheduling among multiple MSSs using power-saving class Type II. This work differs from [3], [4] in that it takes packet delay constraint rather than minimal data rate as a QoS requirement. Moreover, MMPS acquires traffic patterns from the knowledge of inter-packet arrival time. In contrast, the work in [3], [4] learns of traffic conditions from the length of the packet transmission queue. Extensions of MMPS [2] put packets from several MSSs into one frame to increase bandwidth utilization. A limitation of MMPS is the assumption of one CBR connection per MSS.

The first type of sleep scheduling demands that the transmission activation time of each MSS should be exclusive. In fact, an IEEE 802.16e network typically provides several subchannels such that several transmissions toward or from different MSSs can be done simultaneously.

The second type of sleep scheduling focuses on one or more CBR connections within an MSS. Jang et al. [5] assumed a CBR connection within an MSS and studied how to set the initial sleep interval for Type I power-saving class to maximize power saving while still meet packet delay constraint. Their experimental results indicated that packet latency is proportional to I_0 while energy consumption is inversely proportional to I_0, where I_0 is the length of the initial sleep interval. For CBR traffic, authors suggested that I_0 should be less than inter-packet arrival time (plus allowable delay jitter) to meet packet delay constraint. The weakness of this study is that only one MSS and one CBR connection were taken into account. The case of multiple connections within an MSS was not considered. Also, using Types I power-saving class to deal with CBR traffic is not appropriate.

Sleep scheduling for multiple CBR connections using Type II power-saving class have also been studied [6], [7]. When two or more CBR connections coexist in an MSS, the MSS should be awake whichever connection is served by the transceiver. Consequently, when coexisting connections share identical or proportional QoS settings (such as packet inter-arrival time), it is feasible to seek a common listening interval for these connections, and Type II power-saving class can be utilized [6], [7]. However, when coexisting connections significantly differ in QoS requirements and traffic patterns, the use of Types II power-saving class will give rise to poor bandwidth utilization and thus low sleep ratio. In that case, optimal sleep ratio can be achieved only if we could customize a sleep schedule to exactly match packet arrival patterns.

As an example, suppose that an MSS creates four CBR connections with packet interval-arrival time (in the unit of frame duration) 4, 5, 4, and 5 frames, respectively. Refer to Fig. 1(a) for the exact packet arrival pattern. If Type II power-saving class is in use, the optimal sleep schedule would be that shown in Fig. 1(c). The sleep ratio is only 0.1. If we could customize a sleep schedule to exactly match packet arrival patterns shown in Fig. 1(b), the sleep ratio can

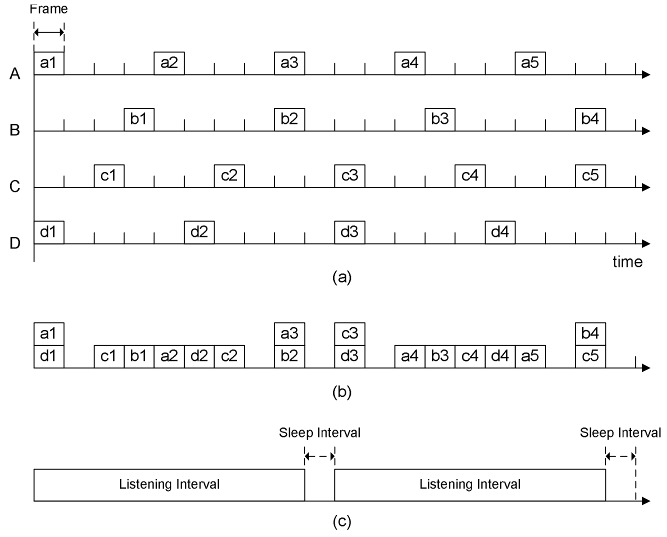


Figure 1. (a) Exact packet arrival pattern (b) Aggregate traffic pattern (c) Optimal sleep schedule using Type II power-saving class.

Table I
QOS PARAMETERS OF FOUR CBR CONNECTIONS COEXISTING IN AN MSS

Connection ID	Inter-packet arrival time (frame)	QoS delay constraint (frame)
A	1	3
B	1	3
C	2	3
D	2	4

be increased to 0.3.

AS [6] and MWT [2] are two approaches to customizing sleep schedules to accommodate multiple CBR connections with different QoS requirements and traffic patterns. AS first sorts all connections belonging to the same MSS according to the connection's packet delay constraints, and arranges packet transmission time for each connection based on the sorted order. In determining a packet's transmission schedule, AS takes the nearest possible *active frame* with priority, where an active frame is a frame that has already accommodated at least one packet's transmission. If no active frame can be found, AS chooses the latest possible frame within the delay constraint. Unlike AS, MWT aims at increasing frame utilization while minimizing packet delivery latency. Therefore, packets will be put together in the earliest possible frame such that transmissions of these packets together can run out of the frame's capacity.

The motivation behind this work is that existing sleep scheduling for multiple CBR connections do not aim at minimizing state transitions. Schedules produced by these schemes can thus be further optimized to save more energy.

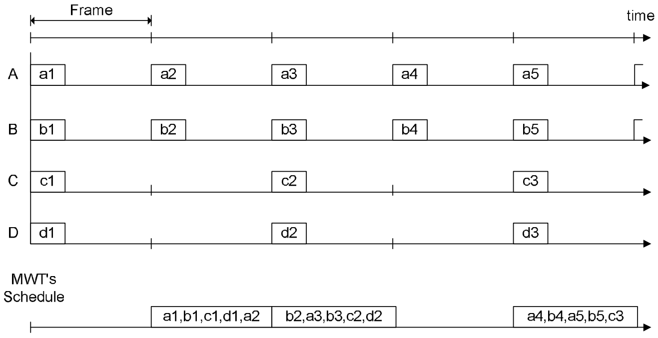


Figure 2. Schedule generated by MWT

As an demonstration, consider four coexisting CBR connections with parameters shown in Table I. If each frame is capable of accommodating five or more packets, MWT will generate a 5-frame sleep schedule with sleep ratio 0.4 as shown in Fig. 2. This schedule gives rise to four state transitions. However, packet transmissions in the second and third frames of the schedule can be postponed to the third and fourth frames, respectively, without violating the delay constraint of any connection. Such reschedule reduces state transitions while preserving the same sleep ratio.

III. MINIMIZING STATE TRANSITIONS

To ease our presentation, we introduce some conventions here. Time in this section is measured by the duration of a frame unless otherwise specified. We assume all packets are of the same length, which is typical for streaming data. The *capacity* of a frame is the number of packets that can be transmitted within the duration of a frame. A frame is *active* if it accommodates the transmission of one or more packet. An *active group* in a given schedule is a longest possible sequence of active frames. It is not hard to see that for any schedule, the number of active groups determines the number of state transitions.

A schedule is *feasible* if all packets can be transmitted by the schedule without violating the QoS delay constraint of any connection. We formulate the problem of converting a given feasible schedule into another with fewer state transitions as a state exploration problem. Our formulation demands that each schedule should have a unique state representation that contain adequate technical details for rescheduling.

For each packet in a schedule, its *transmission shift* is the time difference between the time it is generated and that it is scheduled to be sent. In a feasible schedule, the packet's transmission shift is not larger than the packet's QoS delay constraint. The difference between the latter and the former is defined to be the packet's delay tolerance (DT). Intuitively, a packet's DT indicates how long we can further postpone its transmission while still meeting its delay constraint. For an active frame f, we define f's maximum delay (MD), denoted by $\mathrm{MD}(f)$, to be the minimum DT value of all packets in f. It indicates the maximal number of frames that all packets in f together can be deferred. The MD value of each active

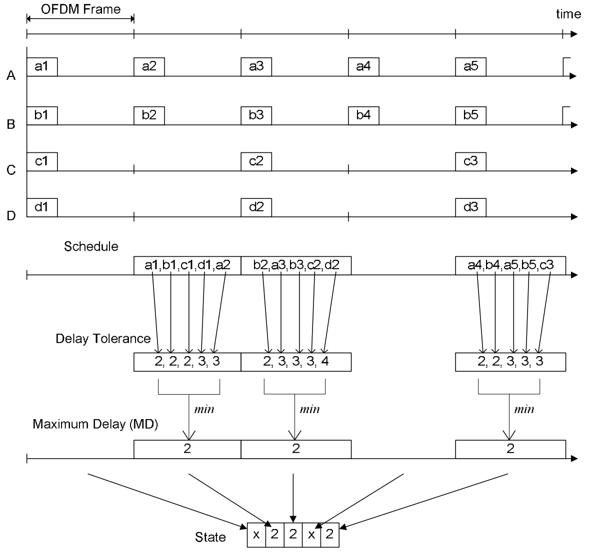


Figure 3. State representation of the schedule shown in Fig. 2

```
/* MD($G_i$) denotes the MD value of active group $G_i$ */
/* dist($G_i, G_j$) denotes the distance between $G_i$ and $G_j$ */

proc OPA(state $s$, int $j$)
    let $G_1, G_2, \ldots, G_k$ be the active groups of $s$
    if $j = k$ then output $s$
    for $i \leftarrow j$ to $k - 1$ do
        if MD($G_i$) $\geq$ dist($G_i, G_{i+1}$) then
            merge together $G_i$ and $G_{i+1}$
            let $s'$ be the new state after the merging
            OPA($s'$, $i$)
            return
        end if
    end for
end proc
```

Figure 4. algorithm OPA

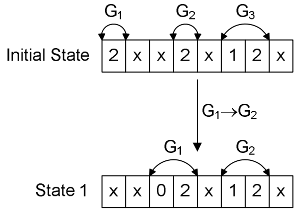


Figure 5. Result of running OPA($\langle 2, \times, \times, 2, \times, 1, 2, \times \rangle, 1$)

```
/* MD($G_i$) denotes the MD value of active group $G_i$ */
/* dist($G_i, G_j$) denotes the distance between $G_i$ and $G_j$ */
/* When $G_i$ is the last active group, dist($G_i, G_{i+1}$) denotes the
   distance from $G_i$ to the end of the schedule. */

proc ES(state $s$, int $j$)
    let $G_1, G_2, \ldots, G_k$ be the active groups of $s$
    if $j = k$ then return
    for $i \leftarrow j$ to $k$ do
        if MD($G_i$) $\geq$ dist($G_i, G_{i+1}$) then
            let $s'$ be the state when merging together $G_i$ and $G_{i+1}$
            output $s'$
            ES($s'$, $i$)
        end if
    end for
end proc
```

Figure 6. algorithm ES

group is defined to be the minimum MD value of all frames in it.

For a sleep schedule spanning n frames, we define its *state* to be an n-tuple, where the i-th element of the tuple is the MD of the i-th frame. For any frame that is not active, the corresponding element in the tuple is simply marked with a '$\times$'. Fig. 3 illustrates how the state corresponding to the schedule of Fig. 2 is defined.

We propose two state exploration algorithms. The first one termed OPA (One-Pass Adjustment) derives a single state from the initial state. OPA scans all elements in the initial state from left to right, and puts off all active groups to the most possible extent. By merging together two adjacent active groups, OPA effectively reduces state transitions.

OPA first looks for the first active group in the state, say G_i, and then determines whether G_i can be put off and merged into the next active group G_j by comparing the MD value of G_i and the distance between G_i and G_j (in frames). If the former is equal to or larger than the latter, G_i is allowed to be merged into G_j, forming a new active group G'. If G' can be formed, OPA updates MD values for all frames in G' that come from G_i, and examines G' for possible group merging with the group next to G'. Otherwise, OPA skips G_i and examines G_j instead. The examination ends when the last active group has been checked. Fig. 4 shows a recursive version of OPA and Fig. 5 illustrates the result of running OPA$(s, 1)$, where $s = \langle 2, \times, \times, 2, \times, 1, 2, \times \rangle$. Observe that OPA successfully reduces the number of state transitions from 3 to 2.

The second state exploration algorithm we proposed is called ES (Exhaustive Search). As its name suggests, ES aims at exploring *all* feasible schedules that are derivable from the initial state. In contrast, OPA yields only one schedule. Fig. 6 shows a recursive version of ES. Another difference between ES and OPA is that ES attempts moving the last active group to the end of the schedule. This attempt may further reduce state transitions as the schedule is in fact wrap-around: there is no state transition between the last active group on the end of the schedule and the first active group on the beginning of the schedule. OPA needs not consider this effect as it seldom leaves active groups on the beginning of the schedule.

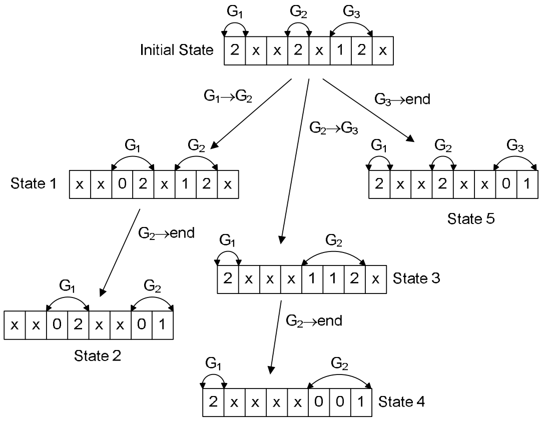


Figure 7. Result of running ES($\langle 2, \times, \times, 2, \times, 1, 2, \times \rangle, 1$)

Table II
PARAMETER SETTING

Parameter	Value/Range
Frame duration	5 ms
Frame capacity	5 packets
Number of connections (n)	1 to 10 (default 4)
Inter-packet arrival time (λ)	5 to 20 ms
QoS delay constraint (d)	30 to 150 ms
Simulation time	500 ms

Fig. 7 illustrates the result of running ES with the same initial state s. Among all states that are generated, ES picks up the one that has the minimal state transitions regarding the schedule's wrap-around property. In Fig. 7, the choice is state 4, which has only one state transition within a schedule period form a long-term perspective.

IV. PERFORMANCE EVALUATION

We conducted simulations to investigate the performance of the proposed schemes. Both MWT [2] and AS [6] were used to yield initial sleep schedules. OPA and ES were then applied to these schedules to produce alternatives. We then compared the results among all possible treatments. Table II lists the parameter setting for the simulations.

We first fixed the number of connections (n) and the QoS delay constraint (d), and observed how the number of state transitions changed with the setting of inter-packet arrival time (λ). Fig. 8 shows the result. As the results of OPA and ES were hardly distinguishable, we present only one set of data for them termed "OPA/ES". It can be seen that AS outperformed MWT in all settings and OPA/ES improved the results of both MWT and AS. The improvement on MWT is greater than that on AS. Consequently, MWT+OPA/ES exhibited fewer state transitions than AS+OPA/ES.

We observed that the performance of OPA/ES sometimes degraded with a fixed d and increasing λ. The reason might be that as long as d allows, a large λ also entails a large transmission shift of packets and thus a small DT value of frames. If the distance between active groups in the initial state is not decreased with increasing λ, the probability of merging together two adjacent active groups generally decreases.

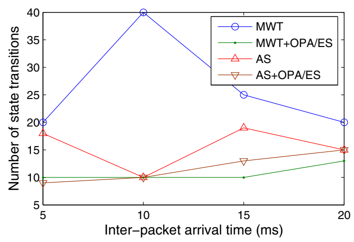


Figure 8. Number of state transitions versus λ ($n = 4, d = 50$ ms)

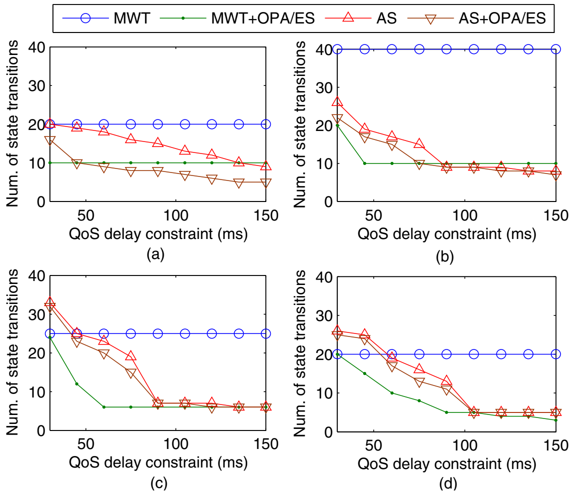
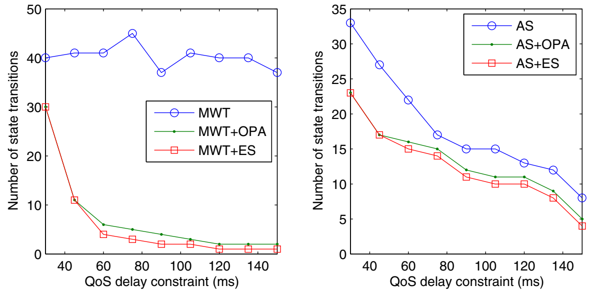


Figure 9. Number of state transitions versus d and λ ($n = 4$) (a) $\lambda = 5$ ms (b) $\lambda = 10$ ms (c) $\lambda = 15$ ms (d) $\lambda = 20$ ms

Figure 9 shows how the number of state transitions changed with increasing d and λ. We observed that with a fixed λ, AS's results improved as d increased. This was expected as a large d gives AS more flexibility to cluster packets together, which also indicates less space for OPA/ES to improve. MWT does not take advantage of loose delay constraints, which is revealed by the observation that MWT's results were independent of d. Therefore, OPA/ES significantly improved MWT's performance.

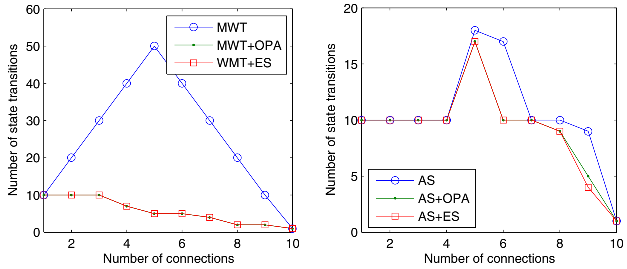
Figure 10 shows the number of state transitions versus d with λ of each connection randomly chosen from $\{5, 10, 15, 20\}$. As connections might have different values of λ, the results of OPA and ES were no longer identical. Compared with Fig. 9, the diversity of λ leaves OPA and ES more space to improve.

We also investigated the relationship between n and the number of state transitions. Fig. 11 shows the result. Here the performance of MWT+OPA/ES was stable and independent of MWT while that of AS+OPA/ES heavily related to AS. The result of MWT deserves further explanation. When



(a) (b)

Figure 10. Number of state transitions versus d ($n = 4$, λ: randomly chosen from $\{5, 10, 15, 20\}$) (a) MWT-based results (b) AS-based results



(a) (b)

Figure 11. Number of state transitions versus n ($\lambda = 10, d = 50$ ms) (a) MWT-based results (b) AS-based results

$n < 5$, each new connection caused MWT to create a new active group, where adjacent active groups were separated by one frame. That is why the number of state transitions increased linearly with n when $n \leq 5$. After five active groups had been creased, each new connection took a frame between two adjacent active groups, effectively merging together these two active groups. Therefore, the number of state transitions decreased linearly when $5 < n \leq 10$.

As the proposed schemes further postpone packet transmissions, they may increase average packet delay. We also investigate extra packet delays caused by the proposed schemes. The result is shown in Fig. 12. We can see that extra packet delays imposed on MWT were greater than those imposed on AS. This was expected as the level of performance improvement on MWT was also higher than that on AS.

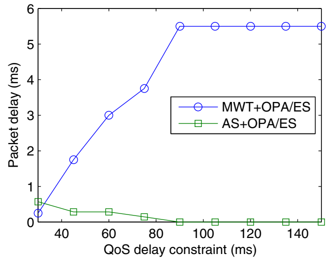


Figure 12. Extra packet delays caused by the proposed schemes ($n = 4$, λ: randomly chosen from $\{5, 10, 15, 20\}$)

V. CONCLUSIONS

We have proposed two schemes, OPA and ES. Both aim at reducing state transitions of existing sleep schedules. We have done extensive simulations to investigate the performance of OPA and ES. The results indicate that, concerning state transitions, the performance of OPA and ES is hardly distinguishable in most cases. Both improved the schedules produced by AS or MWT. The improvements on MWT were generally greater than those on AS. The proposed schemes also increase packet transmission delay. But the measured amount was not significant. In short, OPA is recommended as a light-weight augmentation to sleep scheduling schemes.

REFERENCES

[1] *Air Interface for Fixed and Mobile Broadband Wireless Access Systems*, IEEE Std. 802.16e, Dec. 2005.

[2] S.-C. Huang, R.-H. Jan, and C. Chen, “Energy efficient scheduling with QoS guarantee for IEEE 802.16e broadband wireless access networks,” in *Proc. 2007 Intl Conf. on Wireless Communications and Mobile Computing*, Honolulu, Hawaii, USA, 2007, pp. 547–552.

[3] J. Shi, G. Fang, Y. Sun, J. Zhou, Z. Li, and E. Dutkiewicz, “Improving mobile station energy efficiency in IEEE 802.16e WMAN by burst scheduling,” in *Proc. IEEE GLOBECOM*, 2006.

[4] L. Tian, Y. Yang, J. Shi, E. Dutkiewicz, and G. Fang, “Energy efficient integrated scheduling of unicast and multicast traffic in 802.16e WMANs,” in *Proc. IEEE GLOBECOM*, 2007, pp. 3478–3482.

[5] J. Jang, K. Hant, and S. Choi, “Adaptive power saving strategies for IEEE 802.16e mobile broadband wireless access,” in *Asia-Pacific Conference on Communications*, 2006, pp. 1–5.

[6] S.-L. Tsao and Y.-L. Chen, “Energy-efficient packet scheduling algorithms for real-time communications in a mobile WiMAX system,” *Computer Communications*, vol. 31, pp. 2350–2359, 2008.

[7] H.-L. Tseng, Y.-P. Hsu, C.-H. Hsu, P.-H. Tseng, and K.-T. Fen, “A maximal power-conserving scheduling algorithm for broadband wireless networks,” in *Proc. IEEE WCNC*, 2008, pp. 1877–1882.

[8] J. A. Stine and G. de Vecian, “Improving energy efficiency of centrally controlled wireless data networks,” *Wireless Networks*, pp. 681–700, 2002.

[9] R. Krashinsky and H. Balakrishnan, “Minimizing energy for wireless web access with bounded slow-down,” in *Proc. ACM/IEEE MobiCom*, 2002, pp. 119–130.

[10] J. B. Seo, S. Q. Lee, N. Park, H. Lee, and C. Cho, “Performance analysis of sleep mode operation in IEEE 802.16e,” in *Proc. VTC 2004-Fall*, Sep. 2004, pp. 1169–1173.

[11] K. Han and S. Choi, “Performance analysis of sleep mode operation in IEEE 802.16e mobile broadband wireless access systems,” in *Proc. IEEE VTC 2006-Spring*, May 2006, pp. 1141–1145.

[12] K. Lei and D. H. K. Tsang, “Performance study of power saving classes of type I and II in IEEE 802.16e,” in *Proc. IEEE Intl Conf. on Local Computer Networks*, Nov. 2006, pp. 20–27.