

Energy Efficient Schedulers in Wireless Networks: Design and Optimization^{*,**}

Jeongjoon Lee¹, Catherine Rosenberg², and Edwin K. P. Chong³

¹*School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47907-1285.*

²*Dept. of Electrical and Computer Engineering, University of Waterloo, Canada.*

³*Dept. of Electrical and Computer Engineering, Colorado State University, Fort Collins, CO 80523-1373.*

Abstract. Minimizing energy consumption is crucial for portable wireless stations because they operate on a limited battery supply. For example, the IEEE 802.11 standard includes a mechanism called power-saving mode (PSM), which allows a network interface on a mobile station to enter a sleep state whenever possible to reduce its energy consumption. We consider a generic wireless system composed of an access point (AP) and several stations that offer a PSM to its users. Our PSM is AP-centric (i.e., gives control to the AP) to save more energy. We formulate a downlink scheduling optimization problem aimed at saving energy and propose two heuristic scheduling policies to solve it. One of these policies is non-work-conserving, and it offers an interesting tradeoff between energy consumption and user performance.

We also study and show how the length of the Beacon Period (BP) has a significant impact on the energy and the delay performance of wireless stations. For each of our two scheduling policies, we derive simple approximate formulas for the length of the BP that minimizes the energy consumption and for the relationship between the delay performance and the length of the BP. Assuming the maximum allowable average packet delay is given by the users as a QoS requirement, we illustrate how to dimension the length of the BP for the two schedulers we have proposed and show that we can achieve significant energy savings while meeting the delay constraint with the non-work conserving one in many cases. Extensive simulations show that a fine-tuning of the length of the BP as well as well-designed scheduling disciplines is essential to saving energy in wireless stations.

Keywords: energy efficiency, power-saving mode, optimization, packet scheduling algorithm, wireless LAN

1. Introduction

Wireless stations have become an integral part of our day-to-day lives. A key concern in the design and development of such wireless stations is energy consumption. Although advances in battery technology and low-power circuit design have been significant, dramatic improvements in battery technology to meet the energy needs for future mobile stations are not expected in the near future [1]. As a result, minimizing power consumption is a major factor driving the design of mobile stations and of the protocols therein. As the authors of [2] show, a large part of the power drain of a mobile station connected to a wireless LAN can be attributed to its wireless LAN card.

* *A part of this paper was presented at WiOpt'04 (Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks), March 24-26, 2004, Cambridge, UK.*

** *This work was supported in part by the Indiana Twenty-First Century Fund through the Indiana Center for Wireless Communication and Networking and NSF through grants 0087266, ECS-0098089, ANI-0099137, and ANI-0207892.*

While this paper is not specifically on IEEE 802.11, it is interesting to discuss this wireless LAN technology as it is the most important example of data-oriented wireless technology to date. In IEEE 802.11, a wireless network interface can be in either the *awake* or *doze* states. In the awake state, there are three different modes—TX (transmit), RX (receive), and IDLE (idle)—and each consumes different amounts of energy. In the doze state, there are two different modes—SLEEP (sleep) and OFF (power off)—and the wireless network interface consumes much less energy in this state than in the awake state. Many researchers ([2], [3], [4], [5]) have reported that the power consumption of a wireless LAN card in idle mode is significant, and this is the reason why the IEEE 802.11 standard [6] addresses the issue of power saving and defines a mechanism called PSM (Power-Saving Mode) that offers an opportunity for stations (STAs) to reduce their power consumption by going into sleep mode on a regular basis.

In the PSM specified in IEEE 802.11, the Access Point (AP) sends a Traffic Indication Map (TIM) containing information about its downlink queues (one per PSM station) at the beginning of each Beacon Period (BP). The TIM is the way to inform the PSM stations that care to listen if the AP has packets to deliver to them. It is then left to the station to decide how often to wake up (i.e., a STA can decide to wake up every n beacon periods) and when to request the sending of its packets by using PS-POLL messages. In this sense, the PSM in IEEE 802.11 is station-centric and gives little control to the AP to optimize its scheduling. Note that while the mechanism related to ListenInterval and PS-POLL in IEEE 802.11 gives the stations flexibility to partially control when the buffered packets will be sent so as to save energy, sending a PS-POLL does not guarantee the AP's rapid response. The AP may be congested, in which case the STA has to remain awake (i.e., in the IDLE mode) until the packet is eventually delivered without knowing if the delivery will be made in the next BP or in a subsequent BP. Hence, the energy saved by having a larger ListenInterval may be offset by increased IDLE-mode power consumption.

We consider a PSM for a generic wireless LAN, called the Generic Power Management Model, which is different in many respects from that of IEEE 802.11. In particular, we give a more central role to the

AP since we believe that this will be necessary for offering QoS to some users and for taking full advantage of a power-saving mode. We show how being in PSM yields considerable energy savings for a station as compared to the Constantly Awake Mode (CAM), when an efficient scheduling mechanism is used at the AP to manage the access to the downlink. In addition, we study how the length of the BP has a significant impact on the energy and the delay performance of the PSM STAs. The length of the BP is typically fixed and usually related to the QoS parameters such as throughput and delay. However, there have been few studies on the relationships between the length of the BP and the energy performance of the PSM STAs. These are the main topics of this paper.

This paper is organized as follows. The related work is described in Section 2. The Generic Power Management Model (GPMM) and the general assumptions we make for our study are introduced in Section 3. A problem formulation and our schedulers to solve it are presented in Section 4. The impact of the length of the BP on the energy and delay performance is presented in Section 5. A strategy to fine tune the length of the BP for energy efficient schedulers is illustrated in Section 6, and our conclusions follow.

2. Related Work

In IEEE 802.11, every station (STA) that is associated with the AP in the PSM is assigned an association identifier (AID) during the association process, and the AID maps to a single bit in the TIM that indicates if there is any packet queued in the AP for that station. Stine et al. [7] study alternative protocols that use multiple bits of the TIM to indicate the order of transmissions, the number of packets queued or to be transmitted, or the time when the station should awake or go into sleep mode. At the cost of these additional bits, the nodes may doze more to save energy consumption. Note that the energy saving through this way does not come for free; it costs more in energy for all the PSM stations to listen to the longer TIM than the single bit TIM, and it increases the overall complexity. While they also mention some scheduling issues, they mainly focus on protocols and simply use the SPT (Shortest Processing Time) scheduler [8], and hence their work is different from ours.

The IEEE 802.11 Working Group recommends the length of the BP to be 100ms, and this recommendation has been reinforced by the simulation results in [9]. The authors study the dependencies of different sizes of the BP on the throughput of an ad-hoc network. Their simulation results show that optimum throughput is achieved for a BP length at around 95ms. Though their work is different from ours since their results are for the ad-hoc mode while ours are for the infrastructure mode, it was a first attempt at understanding the impact of the length of the BP and the tradeoff between throughput and energy saving.

Krashinsky et al. [10] also study, via simulation, the performance-energy tradeoff when IEEE 802.11 PSM is used. They find that the power consumption used to listen for TIMs can dominate the total energy consumption if the network is accessed only sporadically, and hence they propose to adapt the sleep durations to reduce the energy consumed to listen to beacons. This work is different from ours in spirit. They focus on reducing the power consumption due to listening to beacons when the STAs are very sporadically active, while we focus on minimizing the overall power consumption for all levels of activities of the STAs, and we study the impact of the length of the BP on energy and delay performance.

3. Generic Power Management Model (GPMM) and General Assumptions

In this section, we briefly introduce the GPMM (first introduced in [11]). It is important to remember that the GPMM is not a model based on IEEE 802.11.

In GPMM, we consider a system composed of an AP and several STAs. To simplify, we assume that the downlink and the uplink are separated, and that the AP is in full control of the downlink. A STA can associate with the AP either in normal mode or in PSM (generic Power-Saving Mode). When a STA is in normal mode, it stays awake all the time (i.e., never goes into SLEEP mode). In the following, we assume that all the STAs are PSM-enabled. Before the beginning of a new Beacon Period (BP) composed of $\Lambda(=L+I)$ timeslots (i.e., one timeslot for transmitting the TIM, L for scheduling data packets), the AP decides which packets to send in the coming BP using its own scheduling policy. The AP then prepares

and sends the TIM in the first timeslot of the BP. The TIM is a bitmap consisting of one bit for each PSM STA indicating if there is at least one packet **scheduled** during the upcoming BP for the corresponding STA. That is, if there is at least one packet ready to be delivered by the AP to a PSM STA during the BP, then the corresponding bit is 1; otherwise, it is 0.

We assume that the PSM STAs **have to** wake up at the beginning of **each** BP to listen to the TIM. If the bitmap for the STA is not set in the TIM, then the STA goes back to sleep immediately. Otherwise, the STA has to remain awake until the last packet scheduled for it in this BP is delivered. The STA knows if a packet it has received is the last one in the BP by inspecting the MORE DATA bit field in the packet header. If the MORE DATA bit is set to zero, it indicates that the packet is the last one to be scheduled in this BP, and hence after receiving the packet from the AP, the corresponding STA goes back into the SLEEP mode and stays in this mode until the beginning of the next BP.

We assume that the STAs are always synchronized in time with the AP. Without loss of generality, we assume that the power consumption in IDLE mode, P_{IDLE} , is equal to the power consumption in RX mode, P_{RX} , and that it takes one unit of energy consumption per timeslot. We assume that the power consumption in SLEEP mode is negligible. We also neglect the transition delays and the energy consumption when a STA changes its states. We assume that the packets are of fixed size and that a timeslot is designed to serve exactly one packet. The number of PSM STAs within the coverage area of the AP, M , is also fixed. The packets arrive continually over time, and the service is assumed to be *gated*, i.e., packets arriving in a BP are served only in or after the next BP (this is a reasonable assumption since the AP has to prepare the TIM in advance). We focus on the downlink and hence on the energy spent listening or waiting (idle) at the stations.

4. Energy Efficient Scheduler Design

We first formulate our scheduling problem: find an optimal scheduling discipline U^* that minimizes the average total power consumption of all the PSM STAs per BP, i.e.,

$$\text{minimize } \bar{E} = \lim_{P \rightarrow \infty} \frac{1}{P} \sum_{q=1}^P \sum_{j=1}^M E_j^q \quad (1)$$

where E_j^q denotes the energy consumption of the j^{th} STA during the q^{th} BP.

In the q^{th} BP, a station j would spend at least one energy unit to receive the TIM and of at most $L+I$ energy units if the AP has scheduled one of its packets to be sent in the last timeslot of the BP (i.e., $\forall j, q, 1 \leq E_j^q \leq L+1$). Hence the total energy (in units of energy) spent in a BP by a STA does not only depend upon the number of packets that have been scheduled by the AP for that station but also upon the position of the last packet scheduled for that station in the BP.

Finding a scheduling policy that minimizes \bar{E} is not easy, and we developed insights by studying the static case in [11]. The static case is formulated as follows: the number of PSM STAs is fixed, the number of packets buffered at the AP per STA is known (the total number of packets is N) and we assume that no more packets will arrive at the AP for any STA. Hence, the problem becomes to decide how to allocate the N packets to the $Q = \lceil N/L \rceil$ required BPs (each of size $\Lambda = L+I$) so as to minimize the total power consumption. A discipline is defined as *work-conserving* if it does not allow the server to be idle when there are packets in the queue. Otherwise, the discipline is *non-work-conserving*. We defined in [11] a *semi-work-conserving* discipline as a non-work-conserving policy that allows the server with N packets in queue to be idle even if there are packets in the queue, as long as those N packets are scheduled within Q BPs.

We showed that this problem is NP hard in general and we could only derive the optimal solution in some special cases. We then designed a very efficient heuristic called EES (Energy Efficient Semi-work-conserving) for the general case. In particular, we realized that it was very much to our advantage to allocate the empty timeslots (we have $QL-N$ of them) in a smart way among the Q BPs to minimize the power consumption. This motivates us to consider non-work-conserving disciplines for the dynamic case.

Based on what we had learned in the static case, we develop two heuristic scheduling disciplines and compare their performances with more traditional scheduling disciplines. The first of the two heuristic policies is a work-conserving scheduling discipline called LPTSPT (Longest Processing Time in selecting stations, Shortest Processing Time in scheduling packets). We show that without increasing the complexity of the algorithm compared to SPT, which is known to perform fairly well in terms of total energy consumption [7], LPTSPT performs at least as well as SPT. The second is a non-work-conserving scheduling discipline called DEES (Dynamic Energy Efficient Semi-work-conserving), an extension to the dynamic case of the EES algorithm. We describe these policies in the following subsections and then illustrate their performance by simulation.

A. LPTSPT, a work-conserving scheduling discipline

We use the term *cluster* for a group of packets that are queued and destined for one STA. From the static case for which we showed that in some special cases, the SPT algorithm was optimal, we have learned two important rules on how to schedule in a way to minimize energy consumption: 1) Allocate adjacent timeslots to the packets that belong to a STA (i.e., treating them as a cluster). 2) Schedule the clusters within a BP in the order of non-decreasing length. What LPTSPT does further as compared to SPT is to try to minimize the number of STAs to be scheduled during a BP so as to reduce further the IDLE mode energy consumption.

The basic idea is as follows. Assume that the number of packets queued at the beginning of a BP, say N , is larger than the number of timeslots to schedule during a BP, say L (i.e., $N > L$). Then, the scheduler needs to select L out of N packets to schedule during this BP. Note that any work conserving scheduler must schedule L packets. In this case, we have two decisions to take: 1) How to select L packets (in particular from which stations)? 2) After having selected L packets, how to schedule them? For the second decision, we already have an answer: schedule the clusters in the order of non-decreasing length.

Let us first discuss what SPT does. SPT would order the clusters in non-decreasing order (i.e., $b_1 \leq b_2 \leq \dots \leq b_M$) and would index them such that STA₁ is the one with the smallest cluster. SPT would then choose STAs 1 to j where j is chosen such that: $\sum_{i=1}^{j-1} b_i < L \leq \sum_{i=1}^j b_i$. Moreover, if necessary, SPT would truncate b_j , and then allocate timeslots to schedule the selected clusters in non-decreasing length (the cluster for STA_j could be anywhere because its cluster has been truncated).

For the first decision, LPTSPT does the following. While any work-conserving scheduler has to schedule exactly L packets (since $N > L$), to save more power it is key to reduce the number of STAs spending IDLE-mode energy. To this end, LPTSPT selects clusters in the order of non-increasing length; i.e., it will select (assuming the same ordering as above) stations M to k where k is chosen such that: $\sum_{i=k+1}^M b_i < L \leq \sum_{i=k}^M b_i$. Then, if necessary, LPTSPT would truncate cluster b_k and would then allocate timeslots in the same way as SPT. When $N \leq L$, LPTSPT does the same as SPT, since all the packets currently queued are to be scheduled.

For example, let's assume that there are 5 STAs that have 1, 2, 3, 4, and 5 packets respectively queued at the AP. Let's also assume that $L=10$. SPT would select the first four STAs (i.e., those with 1, 2, 3, and 4 packets) and would schedule the clusters in this increasing order during the current BP. Then in that case, the total energy consumption is $(5)+1+3+6+10=25$ units (note that the first 5 units are due to the energy consumption spent by the 5 STAs to listen to the TIM). On the other hand, LPTSPT would select the stations with 5, 4 and 3 packets, truncate the cluster of the third station to 1 packet, and schedule them in non-decreasing order, i.e., in the order of 1, 4, 5. This time the total energy consumption is $(5)+1+5+10=21$ units, which is less than that of SPT.

We present below the details of our LPTSPT algorithm. This algorithm is run at the AP before the beginning of a new BP to prepare the TIM.

LPTSPT algorithm

Step 1: Group all the packets for a given station j in a cluster b_j .

Step 2: If the total number of packets currently queued is less than or equal to L , then schedule the clusters in the order of non-decreasing length; done. Else, go to Step 3.

Step 3: Sort the clusters in the order of non-increasing length. Take as many clusters as necessary to get L packets or more starting from the largest cluster. If necessary, truncate the smallest selected cluster. For the selected clusters, schedule the clusters in the order of non-decreasing length. Done.

Note that LPTSPT requires $O(N)$ effort in Step 1 as does SPT. The LPTSPT algorithm involves at most one more sorting than SPT algorithm in Steps 2 and 3, and hence the overall complexity for both algorithm is $O(N) + O(M \log M)$.

B. DEES, a non-work-conserving discipline

In our previous work [11], we have developed a heuristic non-work-conserving scheduling algorithm called EES (Energy-Efficient Semi-work-conserving) for the static case described earlier. (2)

EES determines a schedule over Q BPs, i.e., the number of BPs required to send the N packets without putting any restriction on where the empty timeslot can be. By generalizing EES to the dynamic case, we are allowing our scheduler to not necessarily fill up the current BP even if $N > L$. This makes DEES a non-work-conserving scheduler, and this will clearly impact the delay performance. However, by nature we believe that the PSM stations consider that saving energy is of prime importance and hence are ready to trade off QoS parameters such as delay for energy saving. Hence, non-work-conserving schedulers are also worth looking at (refer to [11] for more detailed background and simulation results on EES).

Let us now present DEES. Before the beginning of a BP, DEES decides the schedule for all the packets currently queued as if it needed to schedule all of them during $Q(= \lceil N/L \rceil)$ BPs starting from the current BP (in fact it will redo the same thing before each BP). Hence out of this schedule, only what has been decided for the first BP will be used. We describe DEES as follows. (Note that Steps 1 to 6 are the same as our EES algorithm shown in [11]).

DEES algorithm

Step 1: Group all the packets for a given station j in a cluster b_j . If the total number of packets currently queued is less than or equal to L , then schedule the clusters in non-decreasing order (SPT); done. Else, go to Step 2.

Step 2: Sort the clusters so that they are in non-decreasing order of lengths (i.e., $b_{[1]} \leq b_{[2]} \leq \dots \leq b_{[M]}$). Note that we will use square brackets $[.]$ to denote the position in the order, i.e., the symbol $[l]$ in the expression $b_{[l]}$ means the cluster in first position in the order. Since we sort the clusters in non-decreasing order, we have $b_{[k]} \leq b_{[l]}$ if $k < l$. Partition the set of clusters ordered as above to obtain the sets R_r where

$$R_r = \{b_{[M-(r-1)Q]}, b_{[M-(r-1)Q-1]}, b_{[M-(r-1)Q-2]}, \dots, b_{[M-(r-1)Q-(Q-1)]}\} \text{ and } r \in \{1, 2, \dots, \lceil M/Q \rceil\} \quad (3)$$

We define this set as the set of clusters of *rank* r , i.e., the set of rank 1 contains the Q largest clusters, the set of rank 2 contains the Q largest clusters among the remaining clusters, and so on.

Step 3: For each cluster $b_j \in R_r$, $\forall r$, calculate the length difference $d_j = b_j - \min_k \{b_k; b_k \in R_r\}$.

Step 4: Assign clusters to the Q BPs according to the following rules.

- (i) Assign the cluster with the largest length difference to the BP with the smallest sum of the length differences as long as the next rule is not violated.
- (ii) The cluster $b_j \in R_r$ can only be assigned to BP l , if l has no other cluster from R_r already assigned to it.
- (iii) If there is more than one BP that has the smallest sum of the length differences and that meets the rule (ii), then assign the cluster to the BP with the smallest sum of lengths of clusters, to break the tie.

Step 5: Once every cluster is assigned, check if each BP has its sum of the lengths of its clusters equal to at most L . If yes, go to Step 7. Else, go to Step 6.

Step 6: Consider the BPs that have the sum of the lengths of their clusters greater than L .

- (i) Sort the clusters in the BP in non-increasing order of length.

- (ii) Reassign the clusters in that order to that BP until the sum of lengths of the assigned clusters is equal to L . Split a cluster if it is needed. Add unassigned (possibly split) clusters to a temporary set of clusters W . Do this for all the BPs that have the sum of lengths of clusters greater than L .
- (iii) If $W = \emptyset$, then go to Step 7. Otherwise, for the clusters currently in W do the following.
- a. Take the cluster with the largest length in W .
 - b. Assign this cluster, say b_d , to the BP, say B_i , that has the smallest number of clusters among the BPs that have the sum of lengths of clusters less than L . If there is more than one BP with the same smallest number of clusters, then try the one with the smallest sum of lengths of clusters first, to break tie. If, by assigning b_d to B_i , the sum of lengths of clusters for B_i , say $Z(B_i)$, becomes greater than L , then split the cluster b_d into clusters $b_{d(c)}$ and $b_{d(\sim c)}$. The split clusters $b_{d(c)}$ and $b_{d(\sim c)}$ have their lengths $b_d - (Z(B_i) - L)$ and $Z(B_i) - L$ correspondingly, and $b_{d(c)}$ is assigned to B_i . Add the remaining cluster $b_{d(\sim c)}$ to W . Go to (iii).

Step 7: Among the Q BP schedules to which the packets have been allocated, choose the BP with the longest length. Schedule the clusters in that BP schedule in the SPT order; done.

Let us illustrate how the DEES algorithm works through an example. Suppose we have 9 PSM STAs that have $b_j = j$, $j = 1, 2, \dots, 9$, and $L = 15$. Then the total number of packets, N , is 45, and we have $Q = \lceil 45/15 \rceil = 3$, i.e., we should prepare 3 BPs. As we have described in Step 2, we partition the set of clusters to obtain the sets R_i 's. We obtain the 3 following sets of different ranks: $R_1 = \{b_9, b_8, b_7\}$, $R_2 = \{b_6, b_5, b_4\}$, and $R_3 = \{b_3, b_2, b_1\}$. We compute the length difference d_j for each cluster in each rank. For example, for the clusters of rank 1, we compute the length differences $d_9 = b_9 - b_7 = 2$, $d_8 = b_8 - b_7 = 1$, and $d_7 = b_7 - b_7 = 0$. Then, we sort the clusters in the order of non-increasing length differences and start to assign each of them in order to the one of 3 BPs with the smallest sum of length differences. For example, we first assign b_9 , b_6 , and b_3 to each BP since the corresponding differences are equal to 2. The sum of the length differences for each BP is then updated to 2. Then b_8 , b_5 , and b_2 , each of which has a length

| | | | |
|-----|-------|-------|-------|
| BP1 | b_1 | b_5 | b_9 |
| BP2 | b_2 | b_6 | b_7 |
| BP3 | b_3 | b_4 | b_8 |

Figure 1. A power-consumption-optimal schedule obtained after Step 4 in the DEES algorithm.

difference of 1, are assigned to the 3 BPs. When we assign b_8 , we cannot assign it to BP1 since b_9 , which is of the same rank as b_8 , is already assigned to it. Thus, b_8 could only be assigned to BP2 or BP3. Since both have the sum of the length differences of 2, there is a tie. To break the tie between BP2 and BP3, we choose the BP with the smaller sum of the lengths of clusters; hence, BP3 is chosen since it has the sum of the length of its clusters as 3 while BP2 has 6. Next, to assign b_5 , we choose BP1 since BP1 has the smallest sum of the length differences among the BPs that meet the rank-restriction rule, which requires all the clusters to retain the original rank restriction so that each BP is assigned exactly one cluster from each rank. We continue to assign the clusters in this way, and after assigning all the clusters, we schedule the clusters in each BP in non-decreasing order. This schedule corresponds to the one shown in Figure 1. Note that the length of this schedule is 15, which is less than or equal to L , and hence this schedule is feasible. (If it is not, we start cluster splitting as detailed in Step 6.)

Now following step 7, we need to choose the BP with the longest length among 3 BPs. Since each BP schedule has the same length, we randomly choose one BP schedule in this example. The reason we adopt the strategy described in Step 7 is as follows. At the end of Step 6, we have allocated packets to the Q BPs. Only one out of these BP schedules will be used since the scheduler will start the process anew at the beginning of the next BP. There are several ways to select that BP schedule. In general, if we select the BP schedule with the least number of packets, then we should achieve lower energy consumption. However, this strategy could severely aggravate the delay performance and cause buffer overflows, i.e., it threatens the whole system stability. In contrast, choosing the BP schedule with the largest number of packets limits the impact on delay and on the system stability. For example, if the number of packets currently queued is $2L < N \leq 3L$, then $Q=3$ and we have to choose one BP schedule among 3 BP schedules.

Since we choose the one with the largest number of packets, the number of packets that will be served is at least $2L/3$. Likewise, if $(Q-1)L < N \leq QL$, by choosing the one with the largest number of packets, the scheduler serves at least $(Q-1)L/Q$ packets in a BP. As the queue length increases, Q increases, and the number of packets to be scheduled in a BP (i.e., $(Q-1)L/Q$) increases as well. Hence if the system becomes backlogged, the scheduler becomes less and less non-work-conserving.

C. Performance Evaluation

We evaluate the performance of the LPTSPT and the DEES algorithms by simulation. We generate the traffic for each STA using a Poisson process with rate λ (packets/timeslot), and all the STAs are assumed to be i.i.d. We assume that the number of STAs (M) is fixed throughout a simulation run. We assume the system to be gated; however, the packets arrive continually over time. We simulate 20 different cases per value of λ , each case being generated from different random seeds. Each case comprises 200,000 timeslots. We calculate the average total energy consumption by summing up the energy consumption of all the STAs over the 200,000 timeslots for each case, and then averaging over the 20 cases. We use the average total energy consumption and the average packet delay as performance measures. Note that throughout this paper, we have found that the 95% confidence intervals are narrow for most of our results and hence concluded that our data are statistically precise. The 95% confidence intervals do not show up clearly in the figures and compromise their readability, and therefore we have chosen not to display them in the figures.

We compare the performance of our algorithms with those of three other scheduling disciplines: FIFO (First In First Out), RR (Round Robin), and SPT. Figure 2(a) and (b) show the performance comparison in terms of the average total energy consumption and the average packet delay as a function of the offered load $\rho = \lambda M$ when $M=10$, and $\Lambda=L+1=20$ is the length of the BP. Let us focus on the performance of LPTSPT first. As we can see from Figure 2(a), LPTSPT and SPT outperform FIFO and RR significantly in total energy consumption. In addition, LPTSPT outperforms SPT especially when the offered load is high. Note that when the offered load is low (up to around 0.6), LPTSPT performs at least equal to or slightly better than SPT. The reason is that in most BPs, the number of packets queued is less than or equal to L , so that all the packets can be scheduled within a BP, and hence LPTSPT reduces to SPT. The delay performance for all the work-conserving schedulers including LPTSPT are the same as expected, and they are shown in Figure 2(b). From these results, we conclude that if delay is the main issue, then LPTSPT is a better discipline than the other work-conserving schedulers introduced in this paper, since it yields lower average total energy consumption.

As for the DEES algorithm, it significantly outperforms the other disciplines studied in this paper in total energy consumption, while trading it off for some amount of delay. When the offered load is lower than 0.6, the performance of the DEES algorithm in both the total energy consumption and the average packet delay is very close to that of LPTSPT. Again, this is because all the packets queued at the

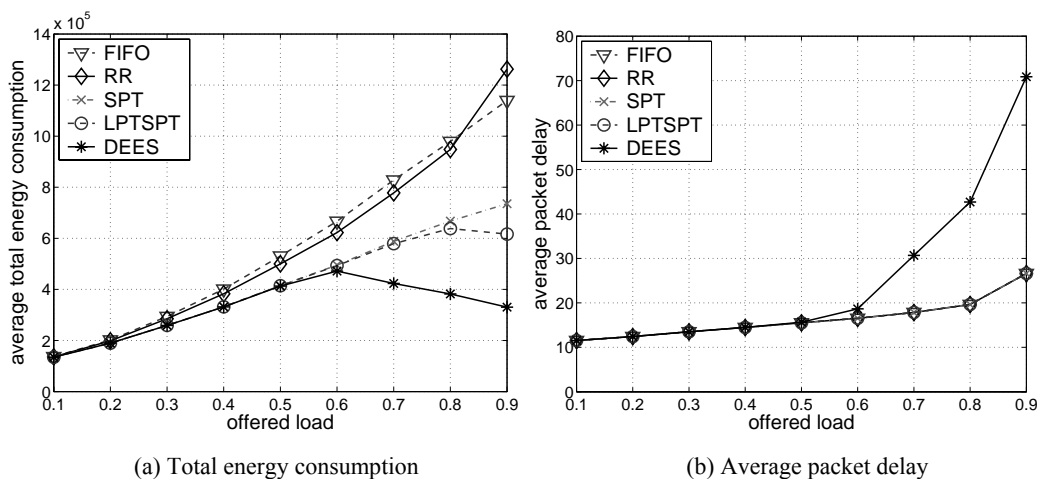


Figure 2. Total energy consumption and average packet delay versus offered load (when $M=10$, $\Lambda=20$)

beginning of a BP are scheduled during that BP in most cases, and in this case, DEES does exactly the same as LPTSPT. As the offered load increases, we can achieve significant energy savings by using DEES. We have run extensive simulations for different M 's and Λ 's, and have seen the same trend in all cases.

However, the delay of DEES increases dramatically as the offered load goes close to 1. When the offered load is around 0.6~0.8, which is where a typical network operating point usually is, DEES saves up to 40% of energy relative to LPTSPT, which is the best discipline among the work-conserving disciplines used in our simulation, at the cost of increasing the average packet delay by one more BP duration. Recall that since energy saving is of prime importance for energy-deprived mobile stations, DEES may indeed be an attractive option. Note that DEES and LPTSPT strike different delay/energy trade-offs but that we are not able to control these trade-offs with these schedulers. One may wonder if there exists a scheduler that can control the tradeoff between energy and delay performance. Extending our approach to this issue is part of our future work.

5. The Impact of the Length of the BP on Energy and Delay Performance

In our generic model, recall that all the PSM STAs have to wake up at the beginning of every BP to listen to the TIM, and this costs a unit of energy consumption per BP for each PSM STA. Intuitively, if the length of the BP (Λ) is too small, then the energy consumption to listen to the TIM dominates the total energy consumption, which we define as the sum of energy (in units of energy) spent by all the STAs in a BP. That is to say, the PSM STAs have to wake up unnecessarily frequently. On the contrary, if Λ is too large, then first the delays become large because the service is gated and the contribution from listening to the TIM to the total energy consumption becomes smaller. From the above discussion, a natural question arises: is there an optimal length for the BP so as to minimize the total energy consumption? To answer this question, we first study the relationships between Λ and energy, and between Λ and delay for both LPTSPT and DEES.

To examine the relationships between Λ and energy and between Λ and delay, we first perform some simulations for both scheduling disciplines. Then, we analyze the relationships both between the total energy consumption and the length of the BP and between the average packet delay and the length of the BP. More precisely we derive a formula for the optimal length of the BP, i.e., the length that achieves the minimum total energy consumption for each scheduling discipline, given M and ρ . We also derive, for each discipline, a formula for the average packet delay as a function of the length of the BP. We validate our formulas by simulation.

The way we simulate and the performance measures we use in this section are the same as those in the previous section. In addition to the homogeneous case, we run some simulations for the heterogeneous case. For the heterogeneous case, we divide the STAs into two groups; the STAs in the first group have their packet generation probability five times greater than that of the STAs in the second group, i.e., $\lambda_1=5\lambda_2$.

A. Analyzing the impact of Λ for LPTSPT

Figure 3 shows the total energy consumption and average packet delay versus the length of the BP for the homogeneous case when $M=50$ and when we use LPTSPT as our scheduler. As we can see from Figure 3(a), for each offered load, there exists a length of the BP that minimizes the total energy consumption. Note that when the offered load is low, the total energy consumption is less sensitive to the

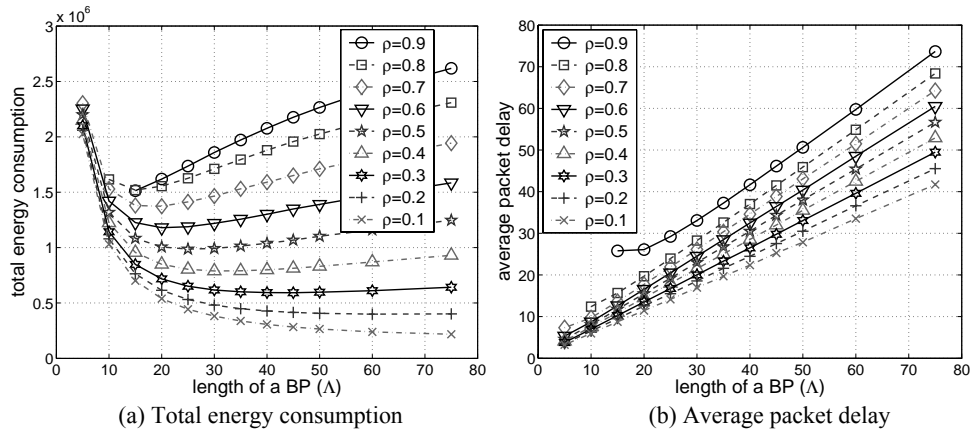


Figure 3. Total energy consumption and average packet delay vs. length of the BP for different offered loads (LPTSPT; homogeneous case; $M=50$)

choice of Λ . For example, when $\rho=0.1$, it may not matter whether we choose Λ as 50 or 200, since the difference in energy consumption is not significant. However, as the offered load increases, energy consumption is more and more sensitive to the choice of Λ . One may wonder why the total energy consumption increases as the length of the BP, Λ , increases especially when ρ is high. We think that the main reason for that is that the number of STAs that are likely to be active is larger when Λ is larger, and some of them have to wait longer to receive their packets, which increases the total energy consumption. As far as the relationship between the average packet delay and the length of the BP is concerned, we see from Figure 3(b) that the average packet delay is linearly increasing with respect to the length of the BP for each offered load.

Assuming that M and ρ are fixed, we proceed as follows to derive an approximate formula for the optimal length of the BP, i.e., the length that minimizes the total energy consumption. In steady state, $\rho\Lambda$ packets are assumed to arrive and be scheduled on average in each BP. We analyze both the cases when $\rho\Lambda \leq M$ and $\rho\Lambda > M$ as follows.

When $\rho\Lambda \leq M$ (i.e., $\Lambda \leq M/\rho$), since the number of packets currently queued is less than or equal to the number of STAs, we assume that $\rho\Lambda$ STAs have one packet each to be scheduled and the other stations have no packets, and in this case, the average energy consumption per timeslot (denoted by E_{TS}) can be calculated as

$$E_{TS} = \frac{1}{\Lambda} (M + (1+2+3+\dots+\rho\Lambda)) = \frac{M}{\Lambda} + \frac{\rho(\rho\Lambda+1)}{2} \quad (4)$$

Note that the first term in the parenthesis on the right hand-side is the amount of energy consumed by the M STAs to listen to the TIM at the beginning of the BP. By differentiating (4) with respect to Λ , we obtain Λ_{opt} that minimizes the energy consumption per timeslot:

$$\Lambda_{opt} = \frac{\sqrt{2M}}{\rho} \quad (5)$$

We also derive E_{TS} for the case when $\rho\Lambda > M$ (i.e., $\Lambda > M/\rho$) in a similar way. We can show that

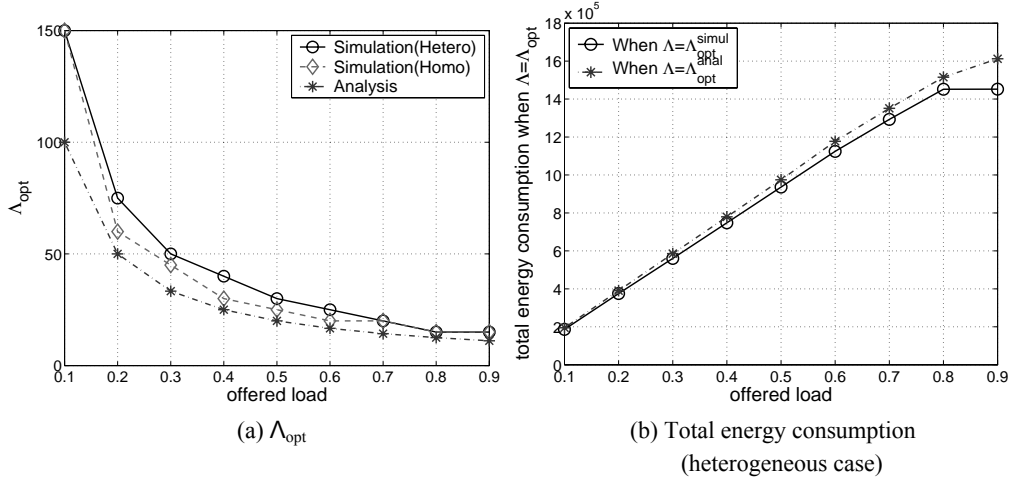


Figure 4. Comparison of Λ_{opt} obtained by analysis and through simulation (a); and comparison of the corresponding total energy consumption (b) (for LPTSPT).

$E_{TS}(\Lambda_{opt})$ obtained in (5) is either smaller than $E_{TS}(\Lambda)$ for all $\Lambda > M/\rho$ or larger by an amount of at most $\rho/8$, which is negligible. Hence, choosing Λ_{opt} as in (5) is still valid.

To validate our approximate and very simple formula for Λ_{opt} , we compare it with the values obtained by simulation. First, we validate it with respect to ρ . To do this, we fix M and study the variations of Λ_{opt} and the corresponding energy consumption while we vary ρ . Note that while we use a Poisson packet-arrival model in our simulation, we do not need any assumption on the traffic model in our analysis; what we only need is the knowledge of the aggregated offered load regardless of the traffic model. Figure 4(a) shows the comparison of the optimal lengths of the BP obtained by our analysis (as in (5)) and our simulation results for the homogeneous and the heterogeneous cases. Figure 4(b) shows the energy consumption by simulation when using Λ_{opt} obtained from our analysis and Λ_{opt} obtained from the simulation of the heterogeneous case. As we can see, the optimal length of the BP obtained from our analysis is very close to the simulation results of both the homogeneous and the heterogeneous cases, especially when the offered load is greater than 0.6. Note that when the offered load is low, we may have large differences in the optimal length of the BP obtained by analysis and by simulation, but the difference in energy consumption is not at all significant as shown in Figure 4(b).

Next, we justify our formula with respect to M . We vary the number of STAs while fixing the offered load. As we can see from Figure 5, with Λ_{opt} obtained from our approximate analysis, we achieve energy

performance that are very close to those obtained by simulation. In conclusion, choosing Λ_{opt} as in (5) is valid and its simplicity makes its very easy to use. Note that while Λ_{opt} is dependent on both M and ρ , being proportional to \sqrt{M} means that the number of STAs does not have a large impact if it does not change drastically.

To analyze the relationship between the average packet delay and the length of the BP, we proceed as follows. Since our measure for delay is the average packet delay for all the packets, the scheduling discipline does not affect it as long as the scheduling is work conserving. Since our system is gated, we can think of packet arrivals during the last BP as the arrival of one batch at the beginning of the current BP. The average packet delay from the arrival time of a packet to the beginning of the new BP is $\Lambda/2$. Using the analysis of the queuing system with batch arrivals given in [12], assuming that the packet arrivals are Poisson with rate ρ and that all the packets that arrived in a batch at the beginning of a BP are scheduled during the next BP, we can calculate the mean order of a schedule of a randomly selected packet (see [12] for more details). Note that since we assume that the packet size is fixed and designed so that a packet can be served in exactly one timeslot, the order of a packet in a BP is the exact number of timeslots the packet has to wait to be scheduled; i.e., if a packet is in the j th order, it would experience a delay of j timeslots.

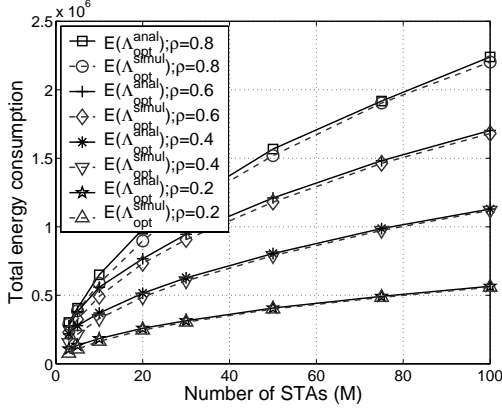


Figure 5. Justification of Λ_{opt} with respect to the number of STAs (M) for LPTSPT.

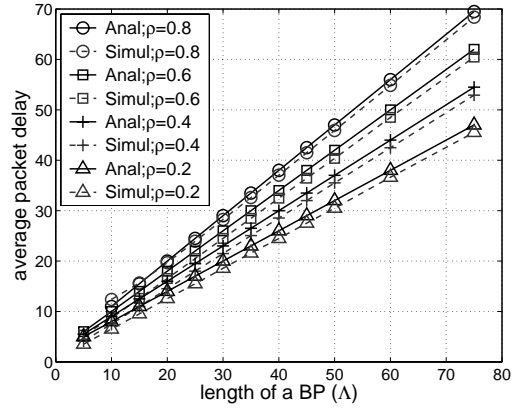


Figure 6. Average packet delay comparison between the analysis and the simulation for LPTSPT.

Using an analysis similar to the one in [12], we obtain the following approximate value, $E[D]$, for the average packet delay:

$$E[D] = \frac{\Lambda}{2} + 1 + \frac{\rho\Lambda}{2} + 1 = \frac{(\rho+1)}{2} \Lambda + 2 \quad (6)$$

As we can see from (6), in our approximation, the average packet delay increases linearly with respect to Λ . Figure 6 shows the comparison between the results obtained from our approximate analysis in (6) and that from the simulations. This validates our derivation for the average packet delay for LPTSPT.

B. Analyzing the impact of Λ for DEES

To analyze the relationships between Λ and the energy consumption and between Λ and delay when we use DEES as our scheduler, we use a similar approach as the one we used for LPTSPT.

Figure 7 shows the total energy consumption and average packet delay versus the length of the BP when we use DEES as our scheduler. As we can see from Figure 7, the total energy consumption and the average packet delay for DEES when the offered load is less than 0.6 are very close to the counterparts for LPTSPT. This is because DEES does the same as LPTSPT if all the packets currently queued can be scheduled during the next BP. However, as the offered load increases, we have seen that we can achieve significant energy savings while trading off delay performance, by using the non-work-conserving DEES discipline. As we did for LPTSPT, we derive simple approximate formulas for the length of the BP that

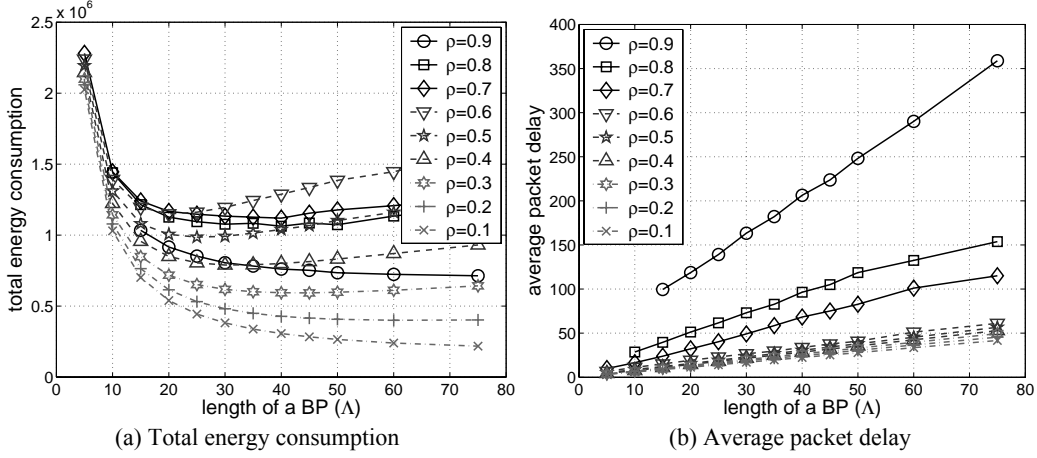


Figure 7. Total energy consumption and average packet delay vs. length of the BP for different offered loads (DEES; homogeneous case; $M=50$)

achieves the minimum total energy consumption and for the relationship between the length of the BP and the average packet delay, as follows.

Let us consider a steady state where the $\rho\Lambda$ packets are assumed to arrive and to be scheduled on average in each BP. We assume that $Q = \lceil N/L \rceil$ remains constant, i.e., the queue length N stays in the range $((Q-1)L, QL]$. Note that DEES serves at least $(Q-1)L/Q$ packets in a BP when Q is determined, and hence to maintain Q in steady state, the offered load must be in the range of $(Q-1)/Q \leq \rho < Q/(Q+1)$, and from this relationship we obtain Q in steady state:

$$Q = \left\lceil \frac{1}{1-\rho} \right\rceil \quad (7)$$

When $\rho\Lambda \leq M$, we assume that the $\rho\Lambda$ packets that arrive at the beginning of the current BP are distributed evenly to $\rho\Lambda$ STAs. Since we have Q BP schedules, there are $\rho\Lambda/Q$ STAs in each BP schedule, and for $\rho\Lambda$ packets to be scheduled to maintain equilibrium, each of $\rho\Lambda/Q$ STAs is assumed to have Q packets. In this case, the average energy consumption per timeslot can be calculated as

$$E_{TS} = \frac{1}{\Lambda} \left(M + Q(1 + 2 + 3 + \dots + \frac{\rho\Lambda}{Q}) \right) = \frac{M}{\Lambda} + \frac{\rho(\rho\Lambda + Q)}{2Q} \quad (8)$$

By differentiating (8) with respect to Λ , we obtain Λ_{opt} that minimizes the energy consumption per timeslot:

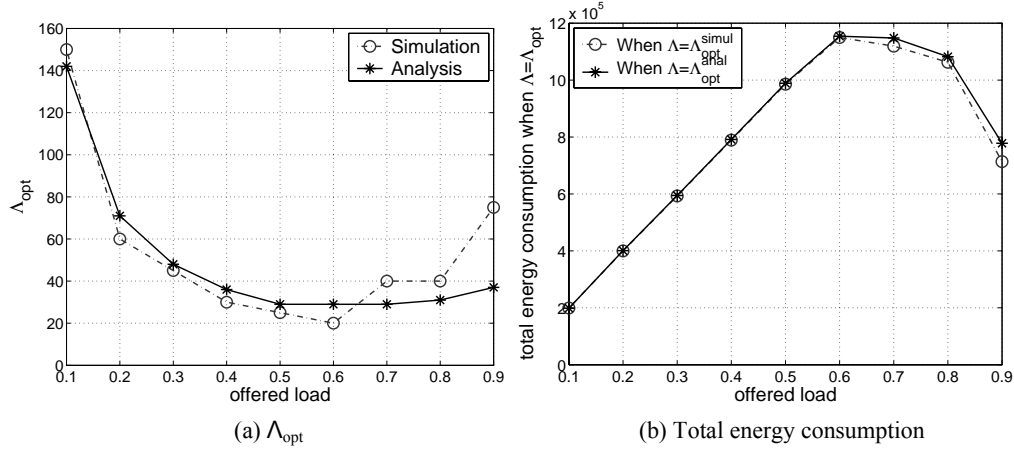


Figure 8. Comparison of Λ_{opt} obtained by analysis and through simulation (a); and comparison of the corresponding total energy consumption (b) (for DEES; $M=50$).

$$\Lambda_{opt} = \frac{\sqrt{2MQ}}{\rho} \quad (9)$$

We also derive E_{TS} for the case when $\rho\Lambda > M$ (i.e., $\Lambda > M/\rho$) in a similar way; as was the case for LPTSPT, choosing Λ_{opt} as in (9) is valid when $\rho\Lambda > M$. Figure 8(a) shows the comparison of Λ_{opt} obtained by (9) and the one obtained by simulation, and Figure 8(b) shows the difference in the corresponding energy consumption when using Λ_{opt} from both cases. As we can see, with Λ_{opt} we have obtained from our approximate formula, we achieve an average total energy consumption that is very close to the one obtained by simulation, and hence choosing Λ_{opt} as in (9) is valid.

We proceed as follows to analyze the relationship between the average packet delay and the length of the BP. We assume steady state and that $\rho\Lambda$ packets arrive and are scheduled in each BP while $Q = \lceil N/L \rceil$ remains constant. As we did for LPTSPT, we think of the $\rho\Lambda$ packet arrivals during the last BP as the arrival of one batch at the beginning of the current BP. According to the DEES algorithm, we can assume that these $\rho\Lambda$ packets are evenly distributed onto Q BP schedules, and hence $\rho\Lambda/Q$ packets are scheduled per BP starting from the current BP. Then, for a packet that is to be scheduled during the k^{th} BP, we can think of its delay as a sum of the $(k-1)\Lambda$ timeslots of delay incurred by postponing its schedule to the k^{th} BP and the delay incurred in the k^{th} BP. Note that regardless of in which BP a packet is

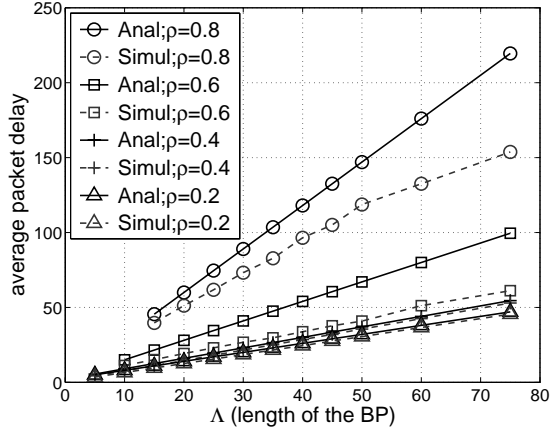


Figure 9. Average packet delay comparison between the analysis and the simulation (for DEES).

scheduled, we can calculate the delay incurred by the scheduling in the BP in which it will be sent in the same way as we did for LPTSPT. We obtain the following approximation for the average packet delay $E[D]$ for DEES:

$$E[D] = \frac{(\rho+1)}{2} \Lambda + 2 + \frac{1}{Q} (\Lambda + 2\Lambda + \dots + (Q-1)\Lambda) = \frac{(\rho+Q)}{2} \Lambda + 2 \quad (10)$$

Figure 9 shows the comparison between the average packet delay obtained from our approximate analysis in (10) and that from the simulation. Note that when $\rho < 0.6$, the result from our analysis is very close to that from the simulation. However, when $\rho \geq 0.6$, the difference between the average packet delays obtained by analysis and by simulation is somewhat larger than that for LPTSPT. This can be explained as follows. In our analysis for DEES, while Q remains constant, we have assumed that the N packets currently queued are evenly distributed onto Q BP schedules, and hence N/Q packets are allocated to each BP schedule. In fact, since DEES selects the BP schedule with the longest length among Q BP schedules for the current BP, more than N/Q packets are always scheduled, and as a result the average packet delay is smaller than that from analysis. Therefore, we can think of the result in (10) as a bound on the average delay.

6. Fine Tuning the Length of the BP

In the previous sections, we have developed two heuristic scheduling disciplines to minimize the total

energy consumption: a work-conserving discipline called LPTSPT and a non-work-conserving discipline called DEES. We have shown that at the expense of delay, DEES can achieve up to 40% of energy saving over LPTSPT, which already saves a significant amount of energy compared to more traditional scheduling disciplines such as FIFO and RR. We have also studied the impact of choosing the length of the BP on the energy and delay performance for both schedulers, and established that the energy and delay performance is heavily dependent on the length of the BP, at least for medium to heavy loads.

Assuming that we have the flexibility to choose the length of the BP, let us go back to our fundamental question: which scheduling discipline should we use to consume less energy while meeting some delay constraint? To answer this question, we proceed as follows.

If we compare the energy and the delay performance characteristics shown in Figure 3 for LPTSPT and in Figure 7 for DEES, we note that: (1) When $\rho \leq 0.6$, the energy and delay performance characteristics for both the LPTSPT and the DEES schedulers are very close. (2) When $\rho > 0.6$, the length of the BP that minimizes the total energy consumption for DEES is different from the one that minimizes the total energy consumption for LPTSPT, i.e., $\Lambda_{opt}^{LPTSPT} \neq \Lambda_{opt}^{DEES}$, and the corresponding energy performance for DEES is better than that for LPTSPT, i.e., $E(\Lambda_{opt}^{DEES}) \leq E(\Lambda_{opt}^{LPTSPT})$. (3) On the contrary, the delay performance for LPTSPT is better than that for DEES. Therefore, if saving energy is of prime importance and the delay constraint may be relaxed, then using DEES with $\Lambda = \Lambda_{opt}^{DEES}$ is the best option, while we could choose LPTSPT if the delay constraint is tight.

To quantify the above observation and compare the two schedulers on a fair ground, we formulate the following problem for each scheduler:

$$\begin{aligned} & \text{find } \Lambda \text{ s.t.} \\ & \text{it minimizes } \bar{E}, \text{ while satisfying the delay constraint } D \leq D_{max}, \end{aligned} \quad (11)$$

where D_{max} is the maximum allowable average packet delay given by the users.

To solve this problem, assuming that M and ρ are fixed and known and D_{max} is given, we tune the

parameters for each scheduler independently by the formulas we have derived previously. For LPTSPT, we calculate Λ_{opt}^{LPTSPT} by the formula given in (5). We also calculate the maximum allowable length of the BP, Λ_{max}^{LPTSPT} , by the relationship we have obtained in (6), i.e.,

$$\Lambda_{max}^{LPTSPT} = \frac{2(D_{max} - 1)}{\rho + 1} \quad (12)$$

Note that due to the fact that the first timeslot in each BP is used for transmitting the TIM, for the system to be stable, the service rate has to be limited to $(\Lambda - 1)/\Lambda$, and hence for a given ρ , this leads to a minimum length of the BP, Λ_{min} :

$$\frac{\Lambda - 1}{\Lambda} > \rho, \text{ and hence, } \Lambda_{min} = \left\lceil \frac{1}{1 - \rho} \right\rceil + 1 \quad (13)$$

Then, we choose the length of the BP for LPTSPT as follows:

$$\Lambda_*^{LPTSPT} = \max(\Lambda_{min}, \min(\Lambda_{opt}^{LPTSPT}, \Lambda_{max}^{LPTSPT})) \quad (14)$$

Similarly, we can obtain Λ_{max}^{DEES} and Λ_*^{DEES} as follows:

$$\Lambda_{max}^{DEES} = \frac{2(D_{max} - 1)}{\rho + Q} \text{ where } Q = \left\lceil \frac{1}{1 - \rho} \right\rceil \quad (15)$$

and

$$\Lambda_*^{DEES} = \max(\Lambda_{min}, \min(\Lambda_{opt}^{DEES}, \Lambda_{max}^{DEES})). \quad (16)$$

Note that by the non-work-conserving nature of DEES, sometimes it may not be possible to find any Λ for DEES to meet the given D_{max} . For example, when ρ is 0.9, Λ_{min} is 11 and the corresponding delay from our formula in (10) is 62. Hence, any D_{max} ¹ smaller than 62 cannot be met by DEES.

We compare the energy consumption, i.e., $E(\Lambda_*^{DEES})$ and $E(\Lambda_*^{LPTSPT})$, for the two schedulers dimensioned with their respective optimal length of the BP obtained in (14) and (16). Figure 10 shows the relative energy gain of DEES over LPTSPT for $\rho=0.7\sim 0.9$ as a function of D_{max} , calculated by

¹ The unit for D_{max} is the timeslot.

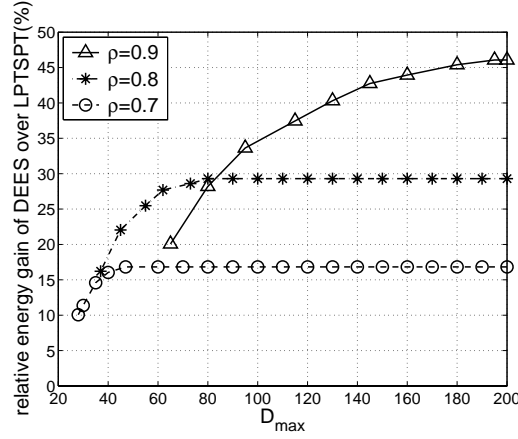


Figure 10. Relative energy gain of DEES over LPTSPT (in %)

$100 * \{E(\Lambda_*^{LPTSPT}) - E(\Lambda_*^{DEES})\} / E(\Lambda_*^{LPTSPT})$. Note that for each ρ , DEES is better than LPTSPT only for $D_{max} \geq D_{max}^*(\rho)$ (with $D_{max}^*(0.9) = 62$ for example). This implicitly means that using DEES may not be possible or advantageous for $D_{max} < D_{max}^*$.

As we can see from Figure 10, we achieve significant energy savings if we choose DEES over LPTSPT whenever $D_{max} \geq D_{max}^*$. One may worry that such a significant gain in energy savings may be obtained only if the delay constraint D_{max} is quite loose. For example, around 30% of energy saving is achieved by DEES over LPTSPT if D_{max} is 80 timeslots or more when $\rho=0.8$. In general, the end-to-end delay should be kept less than 200ms for real-time video and voice conversation, since people find larger delay uncomfortable [13]. If we consider IEEE 802.11b wireless LAN as our wireless medium, which offers a maximum bit-rate of 11Mbps, then a simplified calculation leads us to roughly a timeslot duration of 1ms excluding overhead if we assume that the packet size is 1000 bytes each. In this case, 80 timeslots (and hence 80ms) of delay for one hop (i.e., the access network) may be too high in view of an end-to-end delay constraint of 200ms. In the near future, we can expect the timeslot duration to decrease to say 0.1ms (e.g., the IEEE 802.11n working group is considering a new radio that offers speeds of up to 108Mbps, and that would make 80 timeslots equal 8ms in duration), and in this case a D_{max} of 80 timeslots could be acceptable.

7. Conclusions

Based on a generic wireless system that offers a PSM to its users, we have formulated the problem of minimizing the total power consumption of all the portable stations that are in PSM as a downlink scheduling problem. We have developed two heuristic algorithms and evaluated their performance by simulation. We also have studied the impact of the length of the BP on the energy and delay performance characteristics of PSM stations. We have identified the parameters that affect the energy and delay performance and derived relationships between the length of the BP and the energy and delay performance through very simple approximate formulas. Assuming that M and ρ are fixed and that D_{max} is given, we have illustrated how to dimension the optimal length of the BP for each scheduler and shown that through this dimensioning we can achieve significant energy savings while meeting the delay constraint. The results of this paper show that a fine-tuning of the length of the BP as well as well-designed scheduling disciplines are essential to reducing the energy consumption of mobile stations.

References

1. J. Flinn and M. Satyanarayanan, "Energy-aware adaptation for mobile applications," in Symposium on Operating Systems Principles, December 1999, pp. 48–63.
2. M. Stemm and R. H. Katz, "Measuring and reducing energy consumption of network interfaces in handheld devices," IEICE Transactions on Fundamentals of Electronics, Communications, and Computer Science, Aug. 1997, pp. 1125–1131.
3. R. Kravets and P. Krishnan, "Power management techniques for mobile communication," Proc. of Mobicom 98, Dallas, TX, October, 1998, pp. 157–168.
4. L. M. Feeney and M. Nilsson, "Investigating the energy consumption of a wireless network interface in an ad hoc networking environment," Proc. Of IEEE Infocom 2001, Vol. 3, pp. 1548 –1557.
5. J.-P. Ebert, B. Burns, and A. Wolisz, "A trace-based approach for determining the energy consumption of a WLAN network interface," In proc. of European Wireless 2002, Florence, Italy, Feb. 2002, pp. 230-236.
6. The editors of IEEE 802.11, *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification*, 1999.
7. J. A. Stine and G. de Veciana, "Energy efficiency of centrally controlled transmission of fixed size packets," Proc. of WCNC 2000, pp. 765-770.
8. W. E. Smith, "Various optimizers for single-state production," Nav. Res. Log. Quart. 3, No. 1, March 1956.
9. H. Woesner, J-P. Ebert, M. Schlager, and A. Wolisz, "Power-saving mechanism in emerging standards for wireless LANs: The MAC level perspective," IEEE Trans. on Personal Communications, Jun. 1998, pp. 40-48.
10. R. Krashinsky and H. Balakrishnan, "Minimizing energy for wireless web access with bounded slowdown," Proc. of Mobicom'02, Sep. 23-28, 2002, Atlanta, GA, pp. 1–12.
11. J. Lee, C. Rosenberg, and E. K. P. Chong, "Energy efficient scheduler design in wireless networks," Proc. of Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WiOpt'04), Mar. 24-26, 2004, Cambridge, UK, pp. 237-246.
12. R. W. Wolff, *Stochastic Modeling and the Theory of Queues*, Prentice-Hall, 1989.
13. J. Walrand and P. Varaiya, *High Performance Communication Networks*, 2nd Ed., Morgan Kaufmann Publishers, 2000



Jeongjoon Lee received the B.S. and the M.S. degrees in Electrical Engineering from Seoul National University, Seoul, Korea, in 1988 and 1990. He is currently pursuing the Ph.D. degree at the School of Electrical and Computer Engineering in Purdue University, Indiana. Before he joined Purdue University, he has worked for LG Industrial Systems, Co., Seoul, Korea. His research interests include wireless PAN, wireless LAN, wireless sensor networks, ubiquitous computing, and home networking.

Email: jjlee@purdue.edu.



Catherine Rosenberg Born and educated in France (Ecole Nationale Supérieure des Télécommunications de Bretagne, ‘Diplôme d'Ingénieur’ in EE in 1983 and University of Paris, Orsay, ‘Doctorat en Sciences’ in CS in 1986) and in the USA (UCLA, MS in CS in 1984), Dr. Rosenberg has worked in several countries including USA, UK, Canada, France and India. In particular, she worked for Nortel Networks in the UK, AT&T Bell Laboratories in the USA, Alcatel in France and taught at Purdue University (USA), Ecole Polytechnique of Montreal (Canada). Dr. Rosenberg is currently Chair of the Department of Electrical and Computer Engineering at the University of Waterloo, Canada, where she also holds a University Research Chair. Agencies and industries that have supported her research include NSF (National Science Foundation) NSERC (Natural Sciences and Engineering Research Council of Canada), FCAR (The Quebec counterpart of NSERC), CRC (Canadian Ministry of Communications), EEC (European Commission), ESA (European Space Agency), France-Telecom, CISCO, Bell Canada, and Nortel Networks. Her research interests are in broadband networks (IP and ATM), in wireless networking, in network security, in peer-to-peer networks, and in traffic engineering (Quality of Service, Network Design, and Routing). She has authored over 70 papers and has been awarded six patents in the USA.

Email: cath@ece.uwaterloo.ca



Edwin K. P. Chong received the B.E.(Hons.) degree with First Class Honors from the University of Adelaide, South Australia, in 1987; and the M.A. and Ph.D. degrees in 1989 and 1991, respectively, both from Princeton University, where he held an IBM Fellowship. He joined the School of Electrical and Computer Engineering at Purdue University in 1991, where he was named a University Faculty Scholar in 1999, and promoted to Full Professor in 2001. Since August 2001, he has been a Professor of Electrical and Computer Engineering, and Professor of Mathematics, at Colorado State University. His current interests are in communication networks and optimization methods. He coauthored the best-selling book, *An Introduction to Optimization, 2nd Edition*, Wiley-Interscience, 2001. He received the NSF CAREER Award in 1995 and the ASEE Frederick Emmons Terman Award in 1998. Professor Chong is a *Fellow* of the IEEE. He was founding chairman of the IEEE Control Systems Society Technical Committee on Discrete Event Systems, and was an IEEE Control Systems Society Distinguished Lecturer. He has been on the editorial board of the *IEEE Transactions on Automatic Control*. He is currently on the editorial board of the journal *Computer Networks*. He has also served on the organizing committees of several international conferences, including the IEEE Conference on Decision and Control, the American Control Conference, the IEEE International Symposium on Intelligent Control, IEEE Symposium on Computers and Communications, and the IEEE Global Telecommunications Conference. He was the Conference (General) Chair for the Conference on Modeling and Design of Wireless Networks, part of SPIE ITCOM 2001. An up-to-date vita is available at www.engr.colostate.edu/~echong.

Email: echong@engr.colostate.edu