

Proxy-Assisted Scheduling for Energy-Efficient Multimedia Streaming over Wireless LAN ^{*}

Fan Zhang^{**} and Samuel T. Chanson

Department of Computer Science,
Hong Kong University of Science and Technology,
Clear Water Bay, Kowloon, Hong Kong

Abstract. Energy is a critical resource for battery-driven mobile and wireless devices. The power-saving mode (PSM) provided by the 802.11 standard is not adequate for mobile clients running streaming media applications. We propose new power-aware scheduling strategies that enhance existing traffic shaping schemes for the purpose of minimizing the communication energy consumption of mobile clients. Scheduling decisions are made by the local proxy and executed at the access point. We consider two cases. In the first case, the proxy is unaware of the power characteristics of the clients. We present an optimal scheduling scheme that minimizes the time that mobile clients stay in high-power modes. This scheme achieves significant energy savings compared to using traffic shaping alone. In the second case, the power profiles and the residual battery capacities of the clients are taken into account in generating the schedule. Unlike most existing work, we believe these factors should be considered in the performance metric. We propose a dynamic programming approach that computes the optimal transmission schedule based on the new metric. We also present an efficient heuristic which exhibits near-optimal performance in the simulation experiments.

1 Introduction

Battery-driven mobile/wireless devices such as PDAs and smartphones need to be designed to utilize energy efficiently. As more and more mobile devices are network-aware instead of being used in the stand-alone mode, this work aims to minimize the energy consumption of the wireless network interface, which has become one of the major energy consumers in mobile systems. For handheld devices such as PDAs, communication power can be as high as 40% of the total system's power [1].

A Wireless LAN (WLAN) card typically has three high-power modes: transmit, receive and idle. Data transmission consumes the highest level of power. An idle interface consumes almost the same amount of power as when it is receiving data. Furthermore, most wireless interfaces also have a low-power sleep/doze mode which consumes significantly less power. For example, a typical IEEE 802.11b WLAN card consumes

^{*} The work described in this paper was supported by grants from the National Natural Science Foundation of China and the Research Grants Council of Hong Kong SAR, China (Project No. N_HKUST618/02 and HKUST6178/03E).

^{**} Corresponding author. Email: zhangfan@cs.ust.hk, Tel: (+852)6026-9658, Fax: (+852)2358-1477.

1400mW, 950mW and 805mW while transmitting, receiving and being idle, respectively. While sleeping, it consumes only 60mW [2].

Data communication for common mobile users in a WLAN is asymmetric, and is dominated by downlink traffic. Most of the time, a client is receiving data over the wireless link, say, watching online videos or browsing the web. The uplink traffic primarily consists of short control messages such as HTTP requests and TCP ACK packets.

Multimedia streaming applications over wireless networks are gaining popularity as wireless bandwidth increases. The IEEE 802.11b standard offers a transmission rate of 11Mbps, while the 802.11a and 802.11g can achieve up to 54Mbps. The bandwidth growth enables users to listen to music, view video clips, and even watch movies online on their mobile devices equipped with wireless interfaces. Since these applications usually run over a long period of time and involve a large volume of data, a substantial amount of energy is consumed for receiving the media streams over the wireless link.

The current Power-Saving Mode (PSM) supported by the IEEE 802.11 standards is not adequate for media streaming applications. According to the specification, all mobile clients wake up periodically and listen to the beacon frames broadcast by the Access Point (AP) at the beginning of each beacon interval (typically 100ms) [3]. If the beacon indicates data pending, then the wireless interface must stay at the high-power mode and listen to the channel until all data are received. The waiting time is not specified by the standards and can be arbitrarily long. It is obvious the energy dissipation while waiting is wasteful and should be minimized. Furthermore, existing approaches do not consider the power characteristics and the residual battery capacities of the mobile clients when scheduling streaming media traffic from the AP. If a client at a low battery level is kept waiting for a long time, its battery will be quickly depleted.

In this work, we propose power-aware traffic scheduling schemes to address the above issues. As all traffic must go through the local proxy first, in our scheme, scheduling decisions are made at the proxy and delivered to the AP for execution (see Figure 1). Instead of transmitting media frames individually, the proxy groups consecutive frames and sends them in a burst. More importantly, knowing the communication bandwidth and the workload, the proxy determines the optimal transmission schedule that minimizes the waiting times of all the clients. As the time of receiving data is fixed and cannot be changed, the total energy consumption is minimized by minimizing the sum of the waiting times. In addition, the proxy may communicate with the mobile clients and obtain useful information for scheduling such as residual battery capacity and the idle power consumption rate of their wireless interfaces. With the information, we use a weighted energy consumption function as a new performance metric, and propose an optimal scheduling algorithm that minimizes this metric.

Although proxy/server-assisted traffic shaping has been studied before, the energy saving capability of traffic shaping alone is limited without an appropriate scheduling algorithm. The power-aware scheduling algorithms we propose in this work enhance the traffic shaping technique and achieve additional energy savings. A detailed comparison with some existing schemes is presented in Section 5.

The rest of the paper is organized as follows. The system architecture and a motivating example are introduced in Section 2. We present our proxy-assisted scheduling schemes in Section 3. In Section 4, the performance of the proposed schemes is evalu-

ated and compared with some existing schemes. Following a review of related work in Section 5, Section 6 concludes this paper.

2 Background

2.1 System architecture

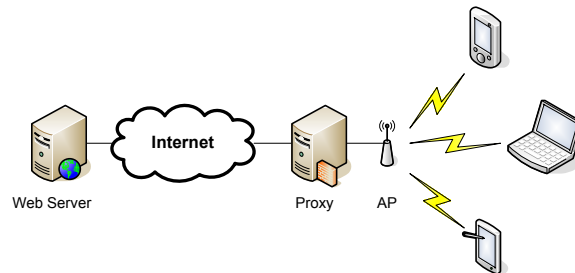


Fig. 1. System architecture.

The WLAN environment considered is illustrated in Figure 1. Heterogeneous wireless clients are associated with an AP and access the Internet via the proxy server. The 802.11b standard, which is the most prevalent setting in today's wireless deployment, is used between the mobile clients and the AP. The AP is typically connected to a proxy server by a high-speed local area network (e.g., 100Mbps or 1Gbps Ethernet). We assume the communication delay between the proxy and the AP is negligible.

Mobile clients retrieve continuous media streams from remote web servers over the Internet. Each client plays back only one stream at a time. The applications (such as stored video and audio) are assumed to be able to tolerate a relatively long startup delay. However, media frames should be delivered to the clients according to the specified frame rate to ensure smooth playback.

All the data streams go through the proxy, so the proxy server at the application layer can understand the semantics of the streams, including data rate and frame rate. The function of the proxy is two-fold. First, it receives data packets from the Internet and stores them in local buffers. Second, the proxy delivers the packets to the AP at an appropriate time so they can be sent to the mobile clients via the wireless link. As streaming media applications usually utilize the UDP protocol, uplink ACK messages are generally not required.

2.2 802.11 power-saving mode

The current 802.11b standard supports a mechanism called the Power-Saving Mode (PSM) to reduce the wireless energy dissipation of mobile clients. The PSM allows a client to switch to the low-power sleep mode when it has no outgoing or incoming traffic. The client needs to notify the AP that it is going to enter the sleep mode. The AP then buffers all the data destined to that client when it is sleeping.

All sleeping mobile clients periodically wake up and listen to the beacon broadcast at periodic *beacon intervals*. The number of beacon intervals between consecutive wake-ups is called the *listen interval*. The beacon includes a Traffic Indication Map (TIM) that informs the mobile clients of data pending. If a client has data pending, it sends a *Power-Saving poll (PS-poll)* message to the AP to indicate it is awake. The client must then actively monitor the channel until all its data are received. A client that does not have data pending can switch back to the low-power sleep mode immediately. The time and energy overhead due to transition between the active mode and the sleep mode is very short and is usually considered to be negligible.

Upon receiving a PS poll message, the AP could send data packets immediately to the client or reply with a short ACK message and delay data transmission in order to receive PS-poll messages from the other clients. The second scheme, called *deferred response*, allows the AP to determine the transmission sequence adaptively. Our proposed schemes make use of this feature, where the AP carries out the transmission schedule produced by the proxy.

3 Proxy-assisted scheduling

We propose two power-aware scheduling schemes to accompany the traffic shaping mechanism. The transmission schedule is computed at the proxy server. We assume the AP can communicate with the proxy, receive the transmission schedule and send out data packets accordingly. The AP notifies a client with the TIM only if the data of that client is scheduled to be sent in the current beacon interval. This function can be implemented in the form of application level software or firmware in the AP. Throughout this work, we do not consider transmission errors or packet retransmission, and assume the time for transmitting control packets such as RTS/CTS and PS-poll to be negligible.

3.1 Energy-efficient scheduling

We first consider the case where the power characteristics the clients are not known for scheduling. In this case, the total energy is minimized by minimizing the total time that the clients stay in the active (receive/idle) modes.

Suppose there are n streams destined to n receiving clients, respectively. The proxy groups consecutive data frames in each stream and sends them in a single burst. Each burst contains a number of media frames that can sustain continuous playback for B time units. Therefore each client only needs to receive data once every B time units. We call B the *buffer length*. Suppose the length of the beacon interval is L , then $B = mL$ where m is a positive integer. We use t_i ($1 \leq i \leq n$) to denote the time required to transmit each data burst of stream i . A feasible schedule is an assignment of the n bursts t_1, t_2, \dots, t_n into m consecutive beacon intervals I_1, I_2, \dots, I_m each of length L .

The scheduling problem is formulated as follows. We use $S_j = \{s_{j,1}, \dots, s_{j,n_j}\}$ to denote the order of transmission in the interval I_j , where $s_{j,k}$ indicates the data stream for the k -th transmission. The objective function is the total active time of all the clients

and can be expressed as:

$$\sum_{j=1}^m \sum_{k=1}^{n_j} (n_j - k + 1) \cdot t_{s_j,k}. \quad (1)$$

The above problem is equivalent to scheduling independent tasks on identical machines to minimize the sum of their finish times (i.e., completion times). The latter problem can be reduced to a min-cost-flow problem and can be solved efficiently [4].

Input: n, m, t_i ($1 \leq i \leq n$).

Output: S_j ($1 \leq j \leq m$).

```

1      Sort streams in non-increasing order of  $t_i$  ( $1 \leq i \leq n$ );
2      for  $i = 1, 2, \dots, n$ 
3           $j = i \bmod m$ ;
4          Add  $i$  to  $S_j$ ;
5      for  $j = 1, 2, \dots, m$ 
6          Sort  $S_j$  in non-decreasing order of  $t_{j,k}$ ;

```

Fig. 2. Algorithm for calculating optimal schedule.

We present a polynomial time algorithm to build the optimal schedule [5]. First the streams are sorted by the transmission time in non-increasing order. Then stream i is assigned to interval I_j where $j = i \bmod m$. Finally, in each beacon interval, streams are transmitted in non-decreasing order of the transmission time. The algorithm is formally presented in Figure 2, and the time complexity is $O(n \log n)$.

For example, suppose $m = 3$ and there are 6 streams whose receiving times are 10, 9, 8, 7, 6, 5, respectively. According to the algorithm, stream i is allocated to interval $(i \bmod m)$. As a result, streams $\{1, 4\}$ are allocated to interval 1, streams $\{2, 5\}$ to interval 2, and streams $\{3, 6\}$ to interval 3. In each interval, the allocated streams are scheduled in non-decreasing order of their receiving times. Therefore stream 4 is scheduled before stream 1 in interval 1, and so on. The total time in high-power modes of all the clients is 63.

The condition given in Theorem 1 below guarantees the schedule produced by the above algorithm is feasible. Please refer to [5] for the proof of optimality.

Theorem 1. *Suppose t_i ($1 \leq i \leq n$) are sorted in non-increasing order. If the inequality $(\sum_{i=1}^n t_i)/m + t_1 \leq L$ holds, then the schedule produced by the algorithm in Figure 2 is feasible, that is, $\forall j, \sum_{k=1}^{n_j} t_{s_j,k} \leq L$.*

Proof: We prove the theorem by contradiction.

Suppose $(\sum_{i=1}^n t_i)/m + t_1 \leq L$ holds but $\exists j, \sum_{k=1}^{n_j} t_{s_j,k} > L$. Now the last stream transmitted in interval I_j is s_{j,n_j} . Obviously $t_{s_{j,n_j}} \leq t_1$. As the schedule is optimal in minimizing the sum of finish times, moving stream s_{j,n_j} to any other interval cannot decrease the sum of finish times. Therefore, we have

$$\sum_{k=1}^{n_l} t_{s_l,k} \geq \sum_{k=1}^{n_j-1} t_{s_j,k} > L - t_{s_{j,n_j}} \geq L - t_1, \quad \forall l \neq j, l = 1, 2, \dots, n.$$

Furthermore,

$$\sum_{i=1}^n t_i = \sum_{j=1}^m \sum_{k=1}^{n_j-1} t_{s_{j,k}} > m \cdot (L - t_1),$$

which contradicts with $(\sum_{i=1}^n t_i)/m + t_1 \leq L$. \square

Given today's high wireless bandwidth and that many handheld devices support low-data-rate media streams only, the above feasible condition usually holds. In this case, the algorithm given in Figure 2 produces the optimal solution for our scheduling problem. However, if the condition does not hold, the schedule produced may not be feasible. Then the problem of determining whether all the streams can be feasibly scheduled is NP-hard [6].

In the next section, we present a dynamic programming approach to minimize the weighted sum of finish times. The algorithm has a pseudo-polynomial time complexity. The problem of minimizing the sum of finish times, as a special case, can also be solved using this dynamic programming approach.

3.2 Battery-aware scheduling

The mobile devices in a WLAN environment are likely to be produced by several different manufacturers, and their wireless interfaces may have different power characteristics. For example, the idle power of an OriNOCO PC Gold card is 805mW, while that of a Cisco AIR-PCM350 card is more than 30% higher (1080mW) [2]. In addition to power dissipation, the available battery capacities of the clients may also be different. The battery capacity is approximately proportional to the battery size. Limited by size, small mobile devices such as smart phones have lower battery capacities compared to PDAs or Pocket PCs.

The transmission schedule calculated in Section 3.1, based on minimizing the finish times, may not be optimal when the power profiles of the clients are heterogeneous. Furthermore, without considering the residual energy capacity, the schedule may require a client at a low battery level to wait for a long time and quickly deplete its battery.

Based on the above observations, we propose to use a weighted energy consumption function as the performance metric. The idea is to minimize the waiting time of the mobile clients with a high idle power and a low battery level. Denote the idle power of client i as P_i and the residual battery capacity as C_i . The weight $w_i = P_i/C_i$ is the proportion of the residual battery that will be consumed by keeping the wireless interface idle for one time unit.

Assume n_j clients receive their data in interval j , the problem is formally formulated as a linear optimization problem with 0/1 variables $x_{i,j,k}$ (see Figure 3). The binary variable $x_{i,j,k} = 1$ indicates that client i is the k -th to receive data in interval j . The first constraint specifies that the total transmission time in each interval cannot exceed the length of the interval L . The second constraint guarantees every stream is transmitted and only transmitted once. The last constraint ensures only one stream is selected in each interval and each transmission sequence in the schedule is generated.

Solving this optimization problem is NP-hard [4], so we employ a dynamic programming approach. First the streams are sorted by t_i/w_i in non-decreasing order.

$$\begin{aligned}
& \text{minimize} && \sum_{i=1}^n \sum_{j=1}^m \sum_{k=1}^{n_j} \left(w_i x_{i,j,k} \cdot \sum_{l=1}^k \sum_{i=1}^n t_i x_{i,j,l} \right) \\
& \text{subject to} && \sum_{k=1}^{n_j} \sum_{i=1}^n t_i x_{i,j,k} \leq L, \quad j = 1, \dots, m \\
& && \sum_{j=1}^m \sum_{k=1}^{n_j} x_{i,j,k} = 1, \quad i = 1, \dots, n \\
& && \sum_{i=1}^n x_{i,j,k} \leq 1, \quad j = 1, \dots, m; k = 1, \dots, n_j \\
& && x_{i,j,k} \in \{0, 1\}, \quad \forall k, 1 \leq k \leq n.
\end{aligned}$$

Fig. 3. A linear optimization formulation.

The solution procedure takes n stages. At stage i , the minimum weighted energy for transmitting the data streams to clients $1, \dots, i$ is determined. Given a schedule, let T_j ($j = 1, \dots, m$) be the latest finish time of the last transmission in interval j . We use $E_i [T_1, \dots, T_m]$ to denote the minimum weighted energy for transmitting data streams $1, \dots, i$ so that the finish times in the m intervals are T_1, \dots, T_m , respectively. The weighted energy of the optimal schedule for n clients is therefore $E_n [L, \dots, L]$.

The energy function in the initial stage is:

$$E_0 [T_1, \dots, T_m] = \begin{cases} 0 & \text{if } \forall j, T_j = 0, \\ \infty & \text{otherwise.} \end{cases} \quad (2)$$

The optimal weighted energy is obtained by iteratively computing E_1 up to E_n . The energy function in each stage i is given by:

$$E_i [T_1, \dots, T_m] = \min_{1 \leq j \leq m} \left\{ E_{i-1} [T_1, \dots, T_j - t_i, \dots, T_m] + w_i t_i \right\}, \quad 1 \leq i \leq n. \quad (3)$$

This dynamic programming solution has a pseudo-polynomial time complexity. For any given m -tuple $[T_1, \dots, T_m]$, the time complexity of computing the optimal $E_i [T_1, \dots, T_m]$ according to equation (3) is $O(m)$. As there are at most L^m different states for each stage i , the time complexity for all n stages is $O(m n L^m)$. If the data streams are not sorted, the time complexity becomes $O(m n L^m + n \log n)$. The complexity is exponential to the number of beacon intervals, so the computation time is prohibitive for large m . Therefore, we present an efficient heuristic below which has polynomial-time complexity.

We modify the optimal scheduling algorithm for minimizing the sum of finish times in Section 3.1 and use it under our new performance metric. The heuristic takes three steps to compute the transmission schedule. First, it sorts the streams in non-increasing order of t_i/w_i ($1 \leq i \leq n$). Then it schedules data stream i to interval $j = i \bmod m$. Finally, the algorithm sorts the streams scheduled in each interval in non-decreasing order of t_i/w_i ($1 \leq i \leq n$). The time complexity of the heuristic is $O(n \log n)$. In the next section, we show that this heuristic exhibits near-optimal performance.

4 Performance evaluation

Two sets of experiments have been performed to evaluate the schemes proposed in Sections 3.1 and 3.2. We have built a simulator to simulate traffic shaping and scheduling in a 11Mbps wireless LAN environment. In the simulated environment, 20 clients are simultaneously running streaming media applications. The media streams have constant data rates which are uniformly distributed between 64bps and 450bps, representing applications from music streaming to video streaming. We set the beacon interval at 100ms. Each client wakes up periodically and listens to each beacon to see if there is pending traffic. The idle/receiving power of the wireless interface is uniformly distributed between 500mW and 1500mW with a supply voltage of 5V.

4.1 Energy minimization

In the first set of experiments, we evaluated the sum of the energy consumption rates of all the clients in the idle and receiving modes, comparing the proposed scheduling scheme in Section 3.1 with a scheme that employs traffic shaping only. In the latter scheme, streams are assigned to the intervals in a round-robin manner. The buffer length B was varied from 100ms to 500ms (i.e., 1 to 5 beacon intervals). For each buffer length, 10 experiments were conducted in which the data rates of the streams were varied. The average of the total energy consumption rates in the 10 experiments was taken and presented in Figure 4. The x-axis denotes the buffer length. The y-axis specifies, under the given traffic shaping and scheduling schemes, the energy consumed by all the clients in one second.

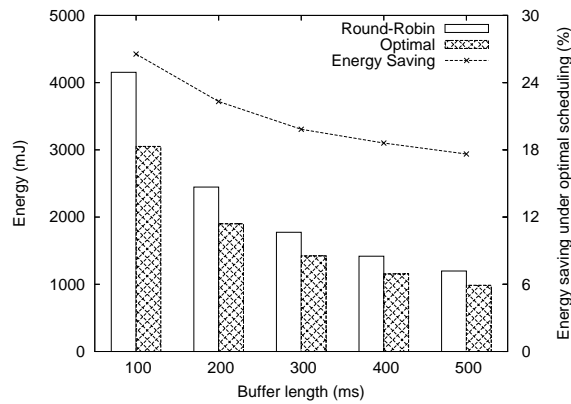


Fig. 4. Total energy consumption rate with traffic shaping and scheduling.

A few observations can be made regarding the results. First, traffic shaping with optimal scheduling achieves about 18% to 27% energy savings compared to plain traffic shaping with round-robin scheduling. Second, the energy consumption rate under both schemes decreases as the length of stream buffering increases. With a longer buffering time, the clients need to wait for data reception in fewer beacon intervals, which reduces

the energy consumption in the waiting mode. Third, as the buffer length increases, the difference between optimal scheduling and round-robin also decreases. This is because with fewer clients receiving data in each beacon interval, it is less flexible and beneficial to change the data transmission sequence.

4.2 Battery-aware performance

In the second set of experiments, we compared the performance of four algorithms in minimizing the weighted sum of finish times. In addition to round-robin, we implemented the optimal dynamic programming approach and the heuristic presented in Section 3.2. The optimal algorithm in Section 3.1 essentially minimizes the sum of finish times of all the clients disregarding their power dissipation and battery capacities. This algorithm is also evaluated under the new performance metric. Due to the exponential computation time of the dynamic programming approach, in the experiments we used only three values of the buffer length: 100ms, 200ms, and 300ms. As in the previous set of experiments, for each buffer length setting, 10 experiments were conducted and the average value was taken.

We set the maximum residual battery capacity to 5000mAh and the minimum to 100mAh. The battery capacities of the other clients were uniformly distributed between these two bounds.

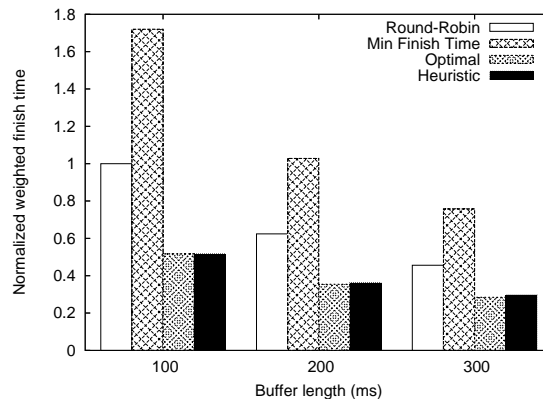


Fig. 5. Normalized weighted sum of finish times with traffic shaping and scheduling.

The simulation experiment results are presented in Figure 5. We set the baseline to the weighted sum of finish times under round-robin scheduling with a buffer length of 100ms. All results are normalized with respect to the baseline for easy comparison. Note that the algorithm that minimizes the sum of finish times performed even worse than round-robin under the new metric. The performance of the heuristic algorithm is close to that of the optimal algorithm (less than 5% difference) in all cases, and is about 50% better than that of the round-robin algorithm.

After evaluating the algorithms under the system-wide new metric, we took a closer look at the energy dissipation of each individual client. In particular, we compared the

proportion of the residual battery capacity each client spent in receiving one minute of media stream. As shown in Figure 6, we observed that the optimal algorithm for minimizing finish times created very unbalanced energy dissipation among the clients: some clients used only 0.02% while others used as much as 5% of their battery capacity.

Some clients spending a large portion of their residual energy, in particular clients 1, 13, 16 and 20, were found running at a low battery level and requesting media streams with a high data rate. The min-finish-time algorithm, disregarding their battery information, scheduled some of these clients to wait and receive data towards the end of each beacon interval. Therefore, these clients will deplete their battery power sooner than the other clients. In contrast, our proposed heuristic produced the most balanced schedule: all clients consumed less than 0.6% of their remaining battery capacities for data reception per minute. Furthermore, 70% of the clients consumed between 0.1% and 0.3%. The variance of the energy consumption among different clients was small. The battery-aware mechanism is particularly beneficial to those clients running at a low battery level. For example, client 13 consumed 1.7% and 1.2% of its residual battery capacity for receiving one minute of data under the min-finish-time and the round-robin schemes, respectively, but it consumed only 0.4% under the battery-aware heuristic.

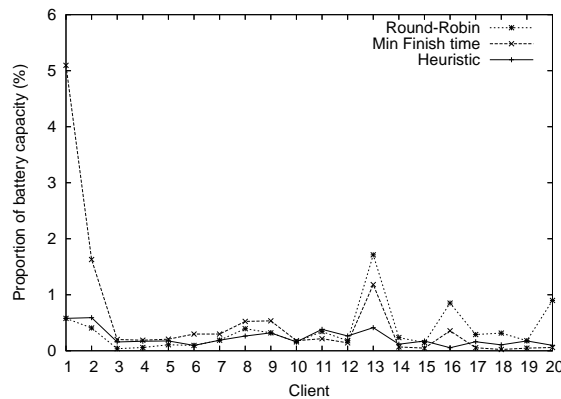


Fig. 6. Proportion of residual battery capacity to receive 1-minute of media stream.

Powered with battery-awareness, our proposed scheduling scheme effectively minimizes the waiting times of clients at a low battery level, which in turn reduces their energy consumption and prolongs their battery life.

5 Related work

As energy conservation has become an important design issue in mobile computing systems, there has been a lot of work on energy-efficient communications [7–10].

Much work has focused on minimizing energy consumption during data transmission [7,8]. Based on the Shannon equation of channel capacity, the transmission rate can be traded off for a lower transmission power. When the workload is fixed, an optimal

transmission schedule can be computed to minimize the total transmission energy [7,8]. On the other hand, when energy and time are limited, a transmission schedule can be built to maximize the data throughput or the value from data transmission [10].

Dynamic mode transition complements transmission power control by switching the wireless interface from high-power active modes to a low-power mode whenever possible. In [11], the energy consumption for accessing a web page is analyzed. Based on information such as the expected size of the page, the network round trip time (RTT) and the mode transition overhead, the mobile client can determine whether the wireless interface should be switched off before the requested page is received. Krashinsky and Balakrishnan [9] have investigated the interaction between TCP and the IEEE 802.11 power-saving mechanism. They suggest that the mobile client does not need to listen to every beacon. Instead, the client can elect to wake up adaptively based on traffic density. On the other hand, a rendezvous policy [12] requires the proxy broadcasts a transmission schedule while the clients turn on their wireless interfaces at the scheduled times. Utilizing an additional low-power transceiver at the client is studied in [2]. The low-power transceiver is always monitoring the channel. If it receives a control message indicating data pending, it wakes up the main transceiver to start receiving data. These schemes do not utilize traffic shaping technique at the proxy and require modification to the clients so they can understand the traffic characteristics and switch modes accordingly. Our approach, instead, utilizes the PSM mechanism of the IEEE 802.11 standard, which is usually already built into the system software of the clients.

Traffic shaping increases data burstiness and reduces the time intervals the wireless interface must stay at the high-power transmit/receive modes [13, 14]. Chandra [13] has studied the energy savings of server-side traffic shaping on different streaming media formats. For client-server style multimedia applications, a control channel can be established to facilitate power management [14]. The client informs the server of its expected service rate and instantaneous buffer occupancy. The server then adjusts the amount of data to be delivered in each burst. However, as the server is usually located at a distant site over the Internet, energy savings may be adversely affected due to obsolete client information and distorted data bursts. Note that none of the above schemes make use of any optimized scheduling algorithm in delivering data.

Our approach combines traffic shaping and power-aware scheduling to further reduce the energy consumption. Moreover, as the scheme is implemented at the local proxy, the delay in message exchange is low and traffic distortion is minimal.

6 Conclusion

In this work we aim to minimize the energy dissipation of mobile systems running online streaming media applications in a WLAN environment. We have analyzed the energy consumption of the wireless interface for receiving media streams over a wireless link. Note that a mobile client may be kept waiting while consuming energy for some time before it actually starts data reception. We show that the current power-saving mode (PSM) provided by the 802.11 standard is insufficient and point out traffic shaping alone does not give the best result for reducing communication energy.

We propose two power-aware scheduling schemes to enhance traffic shaping for additional energy savings. The algorithms build a transmission schedule at the local proxy server which informs the AP of the schedule that the streams should be delivered. The mobile clients simply follow the PSM specified by the 802.11 standards and do not need to be aware of the traffic scheduling scheme at the proxy. The first scheme we propose optimally minimizes the sum of finish times of all the clients running stream media applications in the WLAN. It does not require client side information and has been shown to be effective in energy minimization through simulation experiments. In the second scheme we take idle powers and residual battery capacities into consideration. We use the weighted sum of finish times as the performance metric. This has the effect of favoring the clients that are low in battery power and who idle power consumption rate is high. The optimal dynamic programming scheme we propose has pseudo-polynomial time complexity. We also present an efficient heuristic. Simulation results have shown that the heuristic achieves near-optimal performance and is able to attain significant improvement over using traffic shaping alone.

References

1. Stemm, M., Katz, R.H.: Measuring and reducing energy consumption of network interfaces in hand-held devices. *IEICE Transactions on Communications, Special Issue on Mobile Computing* **E80-B** (1997) 1125–1131
2. Shih, E., Bahl, P., Sinclair, M.J.: Wake on wireless: an event driven energy saving strategy for battery operated devices. (In: *Proceedings of MOBICOM'02*) 160–171
3. Gast, M.: *802.11 Wireless Networks: The Definitive Guide*. O'Reilly (2002)
4. Bruno, J., E. G. Coffman, J., Sethi, R.: Scheduling independent tasks to reduce mean finishing time. *Communications of ACM* **17** (1974) 382–387
5. Brucker, P.: *Scheduling Algorithms*. 4th edn. Springer-Verlag (2003)
6. Garey, M.R., Johnson, D.S.: “Strong” NP-completeness results: motivation, examples, and implications. *Journal of ACM* **25** (1978) 499–508
7. Biyikoglu, E., Prabhakar, B., Gamal, A.E.: Energy-efficient packet transmission over a wireless link. *IEEE/ACM Transactions on Networking* **10** (2002) 487–499
8. Schurgers, C., Raghunathan, V., Srivastava, M.: Power management for energy-aware communication systems. *ACM Transactions on Embedded Computing Systems* **2** (2003)
9. Krashinsky, R., Balakrishnan, H.: Minimizing energy for wireless web access with bounded slowdown. (In: *Proceedings of MOBICOM'02*)
10. Zhang, F., Chanson, S.T.: Throughput and value maximization in wireless packet scheduling under energy and time constraints. In: *Proceedings of IEEE Real-Time Systems Symposium (RTSS)*. (2003) 324–334
11. Anastasi, G., Conti, M., Gregori, E., Passarella, A.: Performance comparison of power-saving strategies for mobile web access. *Performance Evaluation* **53** (2003) 273–294
12. Gundlach, M., Doster, S., Yan, H., Lowenthal, D.K., Watterson, S.A., Chandra, S.: Dynamic, power-aware scheduling for mobile clients using a transparent proxy. In: *Proceedings of International Conference on Parallel Processing (ICPP)*. (2004)
13. Chandra, S.: Wireless network interface energy consumption: implications for popular streaming formats. *Multimedia Systems* **9** (2003) 185–201
14. Acquaviva, A., Simunic, T., Roy, S., Deolalikar, V.: Remote power control of wireless network interfaces. *Journal of Embedded Computing* (2004)