Etiquette Protocol for Ultra Low Power Operation in Sensor Networks

Samir Goel and Tomasz Imielinski
{gsamir, imielins}@cs.Rutgers.EDU
Department of Computer Science, Rutgers University, NJ

October, 2004

(Revises and Updates DCS-TR-552)

Abstract

In this paper, we propose a novel Etiquette protocol for enabling extremely energy-efficient communication among nodes in a multi-hop sensor network. Our key idea for energy efficiency is to allow dynamic, flexible scheduling of inter-node communication, thereby minimizing energy wasted due to idle-listening. Scheduling communication is commonly used in single-hop networks for significant energy savings (e.g. GSM). However, scheduling communication is a *hard* problem in a *multi-hop* network where packet load may not only vary in different parts of the network, but it may also fluctuate over time. The proposed Etiquette protocol allows nodes to schedule their communication in a completely distributed manner, while also allowing them the ability to change their communication schedule in response to the fluctuating packet load in the network. Etiquette protocol allows many-to-many communication, including ability to perform local broadcast. Our simulation results show that Etiquette protocol significantly reduces the average energy expended in delivering each bit as compared to S-MAC [25]. We show that Etiquette protocol allows a network designer to trade increased latency for reduction in energy consumption. The proposed protocol is simple and intuitive.

1 Introduction

In a multi-hop sensor network, nodes are typically powered by battery. In a number of applications, replacing the battery of the sensor nodes is an expensive operation, exceeding the cost of the node itself. This is especially true for sensor networks that are deployed in remote terrains (e.g., mountain top), or in hazardous environments (e.g., battlefields). For such applications, energy efficiency is a key requirement for protocols running on sensor nodes.

For a battery powered node, the most energy consuming component is the radio. The costs of receiving or transmitting a packet is an order of magnitude greater than that for performing computation. Between the two, transmitting a bit consumes twice as much energy as receiving a bit [10]. When a radio is ON but idle it still consumes the same amount of energy as when it is receiving data. This "idle-listening" is the major source of inefficiency in inter-node communication (e.g., a Berkeley Mote transmitting and receiving one packet per second would drain 90% of energy in idle-listening if its radio is always ON). Ideally, a node should turn its radio ON only when it is going to transmit or receive data; at all other times it should keep its radio OFF. However, the asynchronous nature of communication between nodes makes it impossible for a node to know when its neighbor is going to communicate with it. As a result, it is not possible to know the exact instant at which the node should turn ON its radio for receiving. In this paper, we propose a novel Etiquette protocol that attempts to minimize the idle-listening time for the nodes. This minimization is achieved through dynamic scheduling of communication among nodes. TDMA style of scheduling is known to be extremely energy-efficient [24] as the nodes know the precise times to turn ON/OFF the radio, removing idle-listening altogether. Scheduling protocols are quite popular in one-hop networks [3, 8, 9], and have been successfully used in commercial wireless communication systems such as GSM [15]. However, in a multi-hop network where packet load can vary in different parts of the network, and over time, scheduling communication in a way that is responsive to the packet load is a difficult problem. In fact, optimal scheduling of communication in a multi-hop network is known to be a NP-complete problem [6].

The proposed Etiquette protocol allows nodes in a multi-hop network to schedule their communication in a completely distributed manner while also offering flexibility to dynamically adapt their schedule in response to changes in packet load in the network. Etiquette protocol makes this possible by requiring nodes to communicate with their neighbors by *appointments*. With this style of operation, a node knows the precise times at which its neighbors are going to communicate with it. At all other times, it can keep its radio OFF. In this paper, we describe the proposed protocol and different optimizations performed to efficiently solve this distributed scheduling problem.

Recently, a number of distributed protocols have been proposed in literature [5, 11, 14, 17, 22, 25] with the goal of supporting energy efficient communication in multi-hop sensor network. Etiquette differs from S-MAC [25], arguably the most popular among these protocols, in the following important ways:

- Adapts the radio ON time dynamically based on the packet load in the network.
- S-MAC (and T-MAC) concentrates traffic in the active portion of the cycle resulting in significant packet losses at high packet load. In contrast Etiquette spreads the traffic over the entire cycle thereby achieving substantially higher packet delivery ratio at high packet load.

Our simulation results show that Etiquette protocol significantly reduces the average energy consumed in delivering each bit compared to S-MAC protocol while also achieving higher packet delivery ratio.

We start by describing our assumptions about the system. In section 4 we provide details of the Etiquette protocol. In section 3, we briefly review S-MAC protocol. In section 5, we present simulation results showing performance of Etiquette protocol. In section 6, we discuss related work and, finally in section 8, we conclude the paper.

2 System Model

We consider a multi-hop sensor network where each node runs on a battery, thus having limited lifetime. A Berkeley Mote is a good example of a typical node in the network — runs on a battery and therefore energy constrained, has limited memory and computation capability, and has effective communication bandwidth of $\sim$ 19.2Kbps. Our target applications are ones where latency constraints are relaxed (in tens of seconds), and typical rate of *generation* of data at a sensor node is a small fraction of the communication bandwidth (e.g., in Great Duck Island Project [20], sensor nodes transmitted their readings once every 5 mins). Data gathering applications monitoring the environment (natural habitat monitoring, battlefield surveillance, etc) are good examples of such applications. As is typically the case in these applications, we assume that the packet load near the sink nodes may be substantially higher than in other parts of the network.

3 Background

In this section, we briefly review S-MAC [25] protocol. S-MAC introduced a novel technique for making basic CSMA/CA mechanism energy-efficient. In S-MAC, all nodes cycle through LISTEN and SLEEP state. Nodes exchange messages to synchronize with each other so that they are all in active state (LISTEN) at the same time and in sleep state (SLEEP) at the same time. While in active state, nodes use CSMA/CA for communication. This co-ordinated sleeping reduces the energy wasted due to idle listening. All nodes

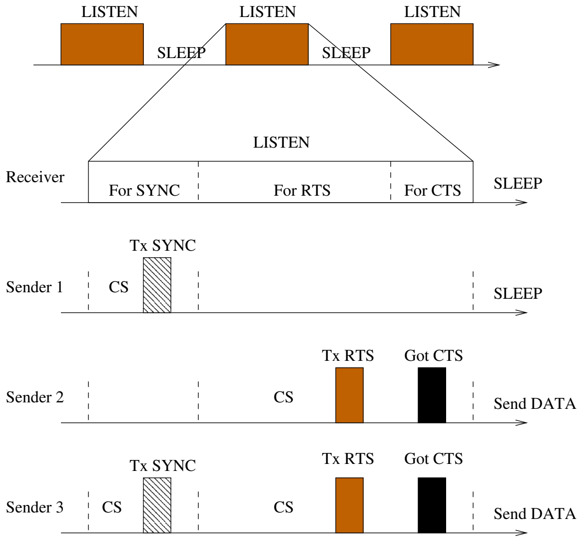


Figure 1: Illustration of timing relationship between a receiver and different senders [25]. CS stands for carrier sense.

operate at the same pre-set duty cycle which dictates the amount of time spent in the two state. The duty cycle is not adjusted dynamically.

The active portion of the cycle is divided into two parts: one for exchanging SYNC messages, and one for communicating data packets (figure 1). Each part accommodates a contention window with multiple time slots to allow multiple senders to compete for the channel. The purpose of SYNC messages is to keep the nodes synchronized. The data part of the active portion is just enough to accommodate contention window and a RTS/CTS exchange. If this completes successfully, the node stays up to complete the exchange of data packet. Otherwise, it sleeps at the end of the active portion.

The duration of LISTEN state is fixed. It's exact duration depends on the channel bandwidth and the contention window size. Based on the duty cycle, the duration of SLEEP state is adjusted.

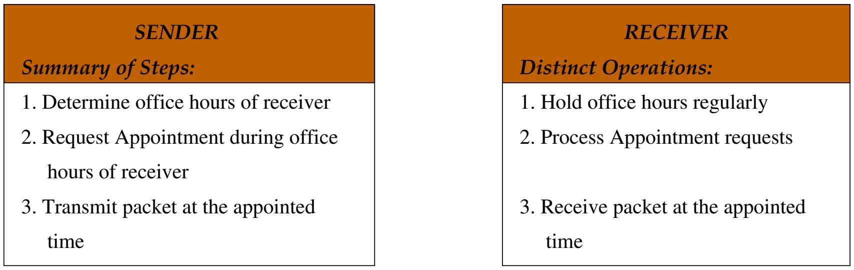
4 Etiquette protocol

4.1 Analogy

We explain the intuition behind Etiquette protocol and its working with the help of an analogy. Consider a lazy teaching assistant living in pre-modern time without amenities like e-mail and telephone. One of her jobs is to be available to help students individually, for which she maintains office hours. However, she does not want to remain in the office for more than what is strictly required to get a good review from the students. To get a good review from her students she should try to meet them as often as they want. Also, she should try to meet as soon as possible, and for the duration they want. She can achieve her goal by announcing regular office hours in which students come and make appointments for one-on-one interaction. She can then close up her office and open it again only when she has an appointment.

4.2 Basic idea

In our target class of applications packet load is small compared to the channel bandwidth. For such applications, *bandwidth is effectively "free"*; the *"expensive"* commodity in the network is the energy of sensor nodes. This simple observation suggests that a strategy that is able to reduce the radio ON time of sensor



SENDER

Summary of Steps:

1. Determine office hours of receiver
2. Request Appointment during office hours of receiver
3. Transmit packet at the appointed time

(a)

RECEIVER

Distinct Operations:

1. Hold office hours regularly
2. Process Appointment requests
3. Receive packet at the appointed time

(b)

Figure 2: Summary of operations performed by sender and receiver in Etiquette

nodes (thereby reducing their energy consumption), even at the expense of consuming extra bandwidth, is a winner. Etiquette protocol is based on this observation.

In Etiquette protocol each sensor node "holds *office hours"* at regular intervals. During this time a sensor node keeps its radio ON. Their purpose is to give an opportunity to neighboring nodes to request *"appointments"* for communicating data packets. When a node receives an *appointment-request* packet during its office hours, it has the option of responding by *granting* an appointment or *denying* the request. *Grant-appointment* packet specifies the time and duration of appointment. At the appointed time both the sender and receiver turn ON their radio to perform the communication. Figure 2 summarizes the basic operations performed by sender and receiver for communication. Each of these operations are by design simple and intuitive. Figure 3 illustrates the interaction between sender and receiver for communicating a data packet.

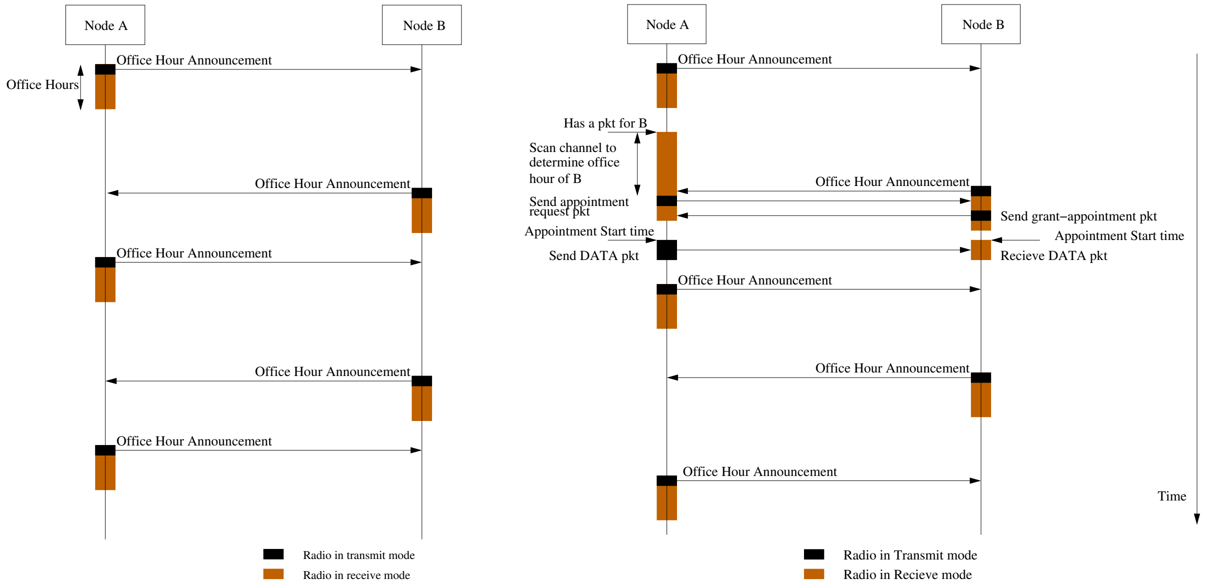
Nodes choose their own office hours independent of the office hours of their neighbors. The office hours repeat at regular intervals. The only requirement is that the *office hour period* — the interval between consecutive office hours, cannot exceed a certain pre-defined maximum, P_{max} for the network. This forces each node to hold office hours at least once within P_{max} units of time. If a node chooses, it can change the start time, duration, and period of its office hours at any point in time. At the start of its office hours, a node sends a small *office-hours announcement* message containing the duration and period of its office hours.

A node determines the office hours of a neighboring node (say, B) by simply turning its radio ON and listening on the channel. The node is guaranteed to hear the office-hours announcement message from node B within P_{max} units of time (modulo packet losses) because of the above constraint. The node caches office hours information in order to avoid scanning the channel on a per-packet basis. This information is valid at least in the short-term as nodes hold their office hours at regular intervals and change them only rarely.

Before we fill the details in this sketch of Etiquette protocol, we would like to emphasize several important aspects of the protocol:

1. In Etiquette protocol, the onus of communication lies with the sender. The sending node needs to determine the office hours of the receiver, and request appointment with it. This is an important aspect of the protocol because it allows the receiver to select its office hours independent of the office hours of its neighbors, and also change the office hours without the need for informing any of its neighbors. The requirement that sender take an appointment with the receiver before communicating a data packet, limits the intervals in which a node may be idle-listening.

2. Etiquette protocol operates on top of existing MAC layer protocol (e.g., CSMA, CSMA/CA, etc). This allows it to control the radio ON/OFF times at a macro-level. The existing MAC layer protocol still handles the micro-level issues such as channel contention, hidden terminal problems, etc. It is possible for neighboring nodes to assign overlapping appointments as they act on local knowledge. We



(a)

(b)

Figure 3: Schematic diagram illustrating the basic protocol. (a) All nodes hold office hours at regular intervals. (b) Steps performed by node A in communicating a data packet to node B.

avoid collisions simply by using larger appointment slots than strictly required for communicating the payload, and letting MAC layer handle channel contention (section 4.3.3).

3. Etiquette protocol does not require fine clock synchronization with neighbors for its operation. All times (e.g., appointment time, start time of next office hours) are specified in terms of offset from the "current" time (time at which packet was sent/received). In order to take care of micro-timing issues, both sender and receiver introduce guard bands before and after the scheduled activity (section 4.3.4).

4. The maximum office hour period, P_{max}, parameter in Etiquette protocol allows a network designer to choose an operating point in the latency-energy tradeoff (section 4.3.2). The larger the value of P_{max} the less frequent the office hours, leading to increase in per-hop communication latency. At the same time, holding office hours less frequently reduces the energy consumption of each node.

5. Etiquette protocol spreads the traffic over the entire office hour period by allowing nodes to choose their office hours independently, and by spreading appointments for a node over the entire office hour period.

6. Etiquette protocol is *"slot-free"* — it does not divide the time in slots. This *avoids* any need for nodes to synchronize on slot boundaries.

These aspects are key to the simplicity and efficiency of the protocol.

4.3 Details

In this section, we provide the key details of each of the basic steps involved in communication. For ease of explanation, in our description we have assumed that nodes A and B are one-hop neighbors and that node A would like to transmit a message to node B.

4.3.1 Establishing office hours

When a node is powered ON, its immediate goal is to establish office hours so that its neighbors are able to communicate with it. For this purpose, it scans the channel for $c * P_{max}$ units of time in order to gather information about office hours of its one-hop neighbors. c is a small pre-defined constant for the network chosen to minimize the probability that a node is not aware of office hours of some of its neighbors due to loss of their office hours announcement messages. Node uses the gathered information in selecting a time-slice for its office hours. It selects a time-slice that has little overlap with the office-hours of its neighbors. At the start of its office hours node sends an *office hour announcement* containing the following information: *duration* and *period*.

4.3.2 Determining a node's office hours

In order to determine the office hours of node B, node A simply scans the channel for office hour announcement from node B. As mentioned earlier, office hour period of a node cannot exceed P_{max}, a pre-defined constant for the network. In the absence of packet losses, P_{max} represents an upper bound on the scanning time for node A.

Since holding office hours depletes the energy reserves of a node, it would like to set the office hour period to as large a value as possible. However, this increases the latency of communicating with it. P_{max} is chosen by the network designer to place an upper bound on this latency.

Optimizing the energy cost of determining a node's office hours

Assuming that the office hours of node B has a period of T^B_{off} ($T^B_{off} \leq P_{max}$), the expected scanning time for node A is $E[T^A_{scan}] = \frac{T^B_{off}}{2}$. In applications where the latency requirement is relaxed, T^B_{off} can be quite large (in minutes or even hours). Consequently, the expected scanning time for node A would also be very large leading to significant energy wastage. This suggests that the individual interest of a node to conserve energy may be at odds with the goal of the network — to conserve energy of all the nodes. We propose a simple solution to this problem.

In our solution, a node with long office hour period sends out *blurbs* at regular intervals. A *blurb* is a short message indicating the next time the node is going to hold its office hours. When node A receives a blurb from node B, it terminates its scanning operation, turns OFF its radio, and waits for the start of office hours of node B. By sending a blurb in between consecutive office hours, node B reduces the expected scanning time of node A ($E[T^A_{scan}]$) by half. Since a blurb is a short message, sending it saves much more energy than it costs. If node B were to send two equally spaced blurb messages, it would reduce $E[T^A_{scan}]$ to one-third of its original value. In general, if B sends n equally spaced blurbs during its office hour period, it would reduce the expected scanning time to $\frac{T^B_{off}}{2(n+1)}$.

As n increases, although the energy consumed in determining the office hours reduces, the energy consumed by node B in sending blurbs increases. There is an optimal value of $n = n^*$ beyond which sending blurbs increases the overall energy consumed in the network rather than reducing it. We derive this optimal value below:

During one office hour period of node B, let K be the average number of neighbors scanning the channel for its office hours. Let J_{recv} be the energy consumed by the radio per second in listening on the channel, J_{blurb} be the energy consumed by a node in transmitting a blurb. Both sending blurbs and scanning the channel contribute to the overhead. The total energy wasted in the overhead during one office hour period is given by:

$$O_{overhead/office-hr} = Cost(Blurbs) + Cost(Scanning) = nJ_{blurb} + K\frac{T^B_{off}}{2(n+1)}J_{recv} \tag{1}$$

From this equation we obtain the optimal number of blurbs, n^* that would minimize $O_{off-hr-disc}$

$$n^* = \sqrt{\frac{KT_{off}^B J_{recv}}{2J_{blurb}}} - 1 \tag{2}$$

Note that all the parameters involved in equation 2 except K are known locally at a node. A node can estimate parameter K by maintaining average statistics on the number of nodes that have scanned the node's office hours in recent past. To enable this, a neighbor sets a flag in its first appointment request after obtaining the node's office hours by scanning. Assuming that recent past is a good predictor for future, node can use this estimate of K in equation 2 to obtain the number of blurbs it should transmit during each office hour period.

From equation 1 and 2, we obtain the minimum value of the energy overhead contributed by node B per office hour period:

$$O^*_{overhead/office-hr} = \sqrt{2KT_{off}^B J_{recv} J_{blurb}} - J_{blurb} \tag{3}$$

The total energy overhead over a long time interval T is given by:

$$\begin{aligned} O_{overhead/node}(T) &= \tfrac{T}{T_{off}^B}(O^*_{overhead/office-hr}) \\ &= T\left(\sqrt{\tfrac{2KJ_{recv}J_{blurb}}{T_{off}^B}} - \tfrac{J_{blurb}}{T_{off}^B}\right) \end{aligned} \tag{4}$$

Equation 4 captures an interesting aspect of the system. Equation 3 shows that with increase in office hour period the energy overhead contributed by a node per office hour period increases. However, as the office hour period increases, the node holds fewer office hours. Equation 4 shows that the combination of these two factors results in a overall decrease in the energy overhead. Thus, it is not only in the node's individual interest but also in the overall network's interest to have as long office hour period as possible. Latency constraints places an upper bound on the maximum length of the office hour period. Thus, in choosing the maximum office hour period P_{max}, a network designer has to decide on an operating point in the energy-latency tradeoff. Etiquette protocol allows the designer to choose an operating point that best fits the needs of the applications.

An important side-effect of using blurbs is that it makes the protocol more robust to loss of office hour announcement messages.

4.3.3 Negotiating appointment

Sender Behavior
An appointment request specifies four parameters:

1. *Appointment length:* This is a function of the size of the payload that the sender would like to communicate

2. *Appointment type:* Appointments may be *one-time* or *periodic*. Periodic-appointments are especially useful for the periodic sources commonly seen in data gathering applications. Since communication requirements of such sources is predictable, they can avoid requesting appointment for each packet.

3. *Appointment window:* The requesting node may *optionally* specify a time interval in which the appointment must lie, based on its own appointment schedule.

4. *Number of appointments:* The is an *optional* field that allows requesting node to specify the number of such appointment slots needed. In wireless networks, the probability of getting a packet through is inversely proportional to the size of packet. If a node has more data than the maximum packet size defined for the network, the node can request multiple appointments.

Pseudo-code 1 Psuedo-code for processing response to appointment request

```
Function ProcessResponseToAppointmentRequest
Input: Response, APPT

if (APPT.Type = GRANT-APPOINTMENT) {
    record appointment APPT
    schedule wake-up timer at APPT.start-time
}
else if (APPT.Type = DENY-APPOINTMENT) {
    ResendAppointmentRequest
}
return
```

Pseudo-code 2 Psuedo-code for processing response to appointment request

```
Function ResendAppointmentRequest
Input: Appointment Request, REQ
     Receiver's office hours, OffHrs

if (REQ.attempts > retry-threshold) {
    discard packet
    return
}
timeRemaining = OffHrs.endTime - currentTime
if (timeRemaining > $\frac{\texttt{request-timeout}}{2}$) {
    send appointment request
    return
}
else {
    resend during next office hours
}
return
```

Sender (node A) should tune into the office hour announcement message of the receiver (node B) to determine the duration of its current office hours (unless receiver's office hours are already going on).

Pseudo-code 1 shows the steps performed by node A on receiving a response to its appointment request. It is possible for node A to not hear any response from node B because of packet losses. Node A waits for a certain pre-defined `request-timeout` interval before sending the request again. Psuedo-code 2 shows the steps performed when re-sending the appointment requets. Note that the request is re-sent only when there is "enough"[1] time left in the current office hours to communicate the request packet. Otherwise, node A waits until the next office hours of node B before requesting again. If the number of attempts exceed the `retry-threshold`, node A discards the packet.

Note that it is quite possible that the appointment slot of a node overlaps with that of its neighbor(s). The existing MAC layer handles channel contention in such a case. In order to allow MAC layer time to recover from this condition, the sender always requests an appointment slot that is `APPT_LEN_MULT` times the size strictly needed for communicating the packet, where `APPT_LEN_MULT` is a small number[2]. Requesting an appointment slot greater than what is strictly needed may seem inefficient at first. However, it eliminates the need for collecting schedule of neighbors thereby making the protocol scalable to dense networks. The performance of this technique is a function of the channel load — the lighter the load, the less the chances of overlaps, and hence, better the performance.

Receiver Behavior

When processing an appointment request, the node searches for an appointment slot that satisfies the following constraints:

The appointment slot should:

1. be at least of the requested size
2. lie within the appointment window if mentioned in the appointment request
3. not overlap any existing appointments of the node
4. end before the next office hours of the node

Among the set of appointment slots that satisfies the above criteria, the node gives *preference* to the *longest* one that does not overlap with its office hours. Any overlap effectively reduces the duration of its office hours, potentially preventing some neighbors from seeking an appointment with the node.

If number of appointment requested, d, is more than one, the node gives as many appointments (up to d) as can fit back-to-back in the chosen slot. This has the advantage that grant-appointment message can be conveyed in a compact form.

4.3.4 Communicating at the appointed time: Handling micro-timing issues and clock drifts

Ability to send/receive a packet at a specified time is critical to the operation of Etiquette protocol. Nodes specify all times in future as offset from the "current" time — the time of transmission of packet. However, before reaching the next-hop receiver, all packets incur a random delay at the MAC layer. This introduces uncertainty about the specified time at the receiver. Clock drifts also contribute to the uncertainty. To address this problem, nodes introduce a *"guard band"* around the appointments. The size of guard band is dictated by the maximum uncertainty. One can substantially reduce the uncertainty caused by the random delay at MAC layer by having it timestamps the packet just before transmitting it [7]. Measurements show that clock drifts between two sensor nodes are quite small — on average 0.2 msecs in every sec [25]. A small guard band would be able to address this.

---

[1] The time left in the current office hours should exceed $\frac{\texttt{request-timeout}}{2}$.

[2] We expect value of `APPT_LEN_MULT` to be between 2 and 6 depending on the expected level of contention on the channel

4.3.5 Support for local broadcast/multicast

There are two possible mechanisms for supporting broadcast in Etiquette — using multiple unicasts, or by having all nodes agree on a common broadcast slot — a small fixed size slot that repeats periodically at a pre-defined fixed rate. The mechanism used by nodes to agree on a common broadcast slot is similar to the way in which nodes synchronize their ACTIVE state in S-MAC. Nodes decide on a common broadcast slot when establishing their office hours, and include their chosen common broadcast slot in their office hour announcement message. When deciding on a common broadcast slot, a node attempts to select the same common broadcast slot as one the indicated in the office hour announcement messages of its neighbors. As in S-MAC, it is possible for a node to have neighbors that have different notion of common broadcast slot. A node needs to adopt the common broadcast slots of all its neighboring nodes.

In sensor network, typically control packets and queries are broadcasted. Such packets have latency constraints that are different from that for communicating data packets. These latency constraints dictate the period of the common broadcast slot. The period of common broadcast slot can be set independently of the office hour period of nodes.

We leave the precise details and performance analysis of broadcast/multicast in Etiquette as a subject for another paper.

4.4 Optimizations

4.4.1 Short-circuiting appointments

For a small packet, instead of requesting an appointment, it may be more efficient to directly communicate it during the neighbor's office hours. This is especially true when the packet size is comparable to the combined size of appointment-request and grant-appointment messages. For this purpose, we define a threshold on the minimum packet size above which node should request appointment. This is very similar to the idea of `RTSThreshold` in 802.11 [13], where a node does not send RTS/CTS messages for packets whose size is below the `RTSThreshold`.

4.4.2 Dynamically adapting the duration of office hours

When a node starts its operation, it sets the duration of its office hours to $d_{off}^{initial}$, a pre-defined constant for the network. It dynamically adapts the duration based on its observed load of *appointment-request*s. If a node idle time during its office hours exceeds a certain threshold τ_{high}, it reduces the duration of its subsequent office hours. On the other hand, if this fraction is drops below τ_{low}, it indicates crowded office-hours and the node increases the duration of subsequent office hours. Note that a node considers average idle time during last k office hours when taking this decision. The specific strategy for changing the duration of office hours, and the choice of value of the parameters, $d_{off}^{initial}$, τ_{low}, τ_{high}, and k is highly dependent on the requirements of the application.

4.4.3 MAC layer feedback

When a node sends an appointment request, it sets a timer to expire after `request-timeout` secs (section 4.3.3). The value of `request-timeout` is set based on the estimate of time required by MAC layer to transmit the packet. Under high channel load, estimating transmission delay at MAC layer can be tricky if it performs some number of re-transmissions (similar to 802.11 [2]) because of the high variance in the delay at the MAC layer. If value of `request-timeout` falls below the MAC layer delay, it would cause the timer to expire prematurely, resulting in unnecessary re-transmission of the appointment request, exacerbating the already high channel load. We propose to address this problem by making use of a simple feedback from MAC layer to dynamically adjust `request-timeout`. A MAC layer using ARQ can indicate to the higher layer if attempts to transmit a packet succeeded or failed. This is a standard feature in typical MAC layer protocols (e.g., 802.11 [2], 802.15.4 [4]). Etiquette makes use of the MAC layer failure notification in updating the estimate of `request-timeout,` using exponentially weighted moving average.

MAC layer feedback is also used by sender during appointment. As mentioned in section 4.3.3, the duration of appointment is `APPT_LEN_MULT` times the size strictly needed for communication. When the sender receives feedback from the MAC layer that the transmission was successful, it can switch OFF its radio immediately without waiting for the appointment to get over. Note that the receiver cannot turn OFF its radio even though it has received the packet because it can never be sure that the MAC layer ACK packet has successfully reached the sender.

4.4.4 Caching information about office hours of neighbors

Channel scanning is an expensive operation. A node caches office hours information of its neighbors to avoid repeated scanning. The cached information is aged unless renewed. There are two reasons for aging:

1. *Clock drifts:* Clock drifts make the cached information inaccurate. These inaccuracies may grow with time, exceeding the size of guard band used. The maximum allowable age of the cached entry is a function of the accuracy of the clock and the size of guard band used.
2. *Node dynamics:* The neighboring node may change its office hours, may simply run out of battery, or may move out of range of its neighbor(s). To handle these dynamics a node should age the cached office hours information.

5 Performance Evaluation

We compare the performance of Etiquette protocol with S-MAC, and IDEAL-802.11. IDEAL-802.11 is 802.11 protocol coupled with an *ideal* radio — it consumes 0 energy when it is idle. Performance of IDEAL-802.11 tells us how far we are from the optimal performance.

For now, our comparison of these three protocols is based on simulations. We have used ns-2 (ver 2.26) [1] simulator for this purpose. We have used the most recent version of S-MAC code available for ns-2[3].

We simulated IDEAL-802.11 protocol by using 802.11 in ns-2, and computing the total energy consumed from the number of MAC and application layer packets transmitted during the simulation. In our calculations, we assumed that for each packet transmitted, there is one receiver that has its radio ON strictly for the time required to receive the transmission.

5.1 Performance Metrics

We use the following performance metrics to characterize the performance:

1. *Energy expended/bit:* It is the ratio of the total energy consumed to the total number of bytes successfully transmitted. It represents the average energy expended in transporting a bit.
2. *Average queuing delay per packet:* It represents average time spent by a packet in a node's packet queue. It measures how quickly a node is able to forward a packet to its next hop.
3. *Packet delivery ratio:* It is the ratio of the number of packets received successfully to the number of packets generated.

5.2 Simulation setup and parameters

In our simulations, we have used a 50 node sensor network, where nodes are uniformly distributed over a 100mx100m region (fig 4). Transmission range of each node is set to 20m, and interference range is set

---

[3] S-MAC code was taken from daily-snapshot of ns-2 of 07-01-04. It was modified appropriately so that energy consumption of each node can be tracked accurately.

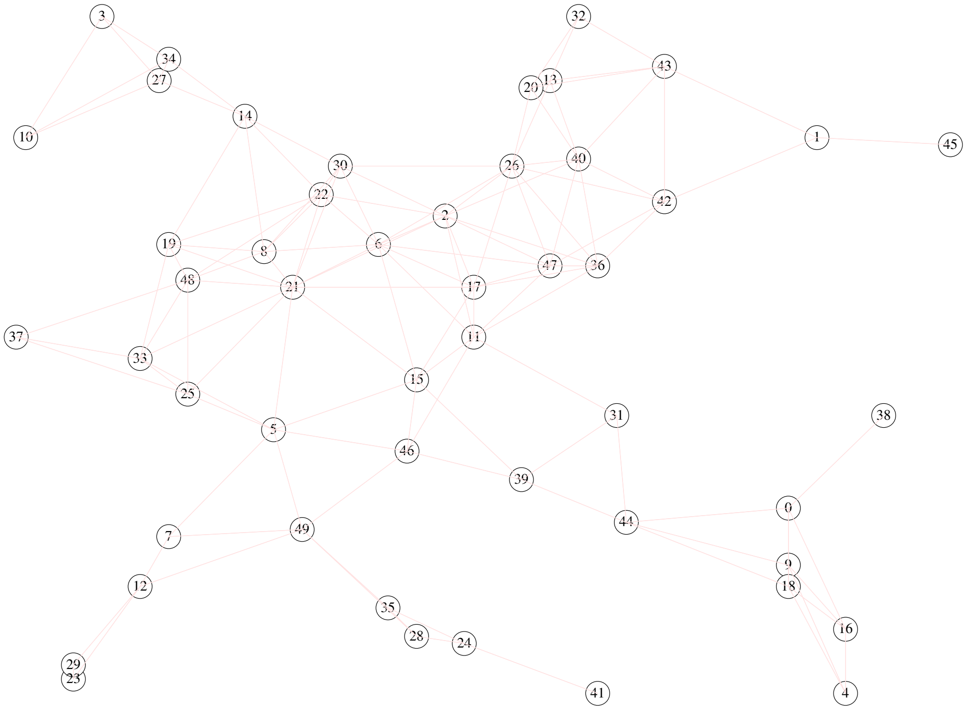


Figure 4: Simulation Topology

Data Packet Size	55 bytes		Transmission Power	24.75 mW
Channel Bandwidth	38.4 Kbps		Receiving Power	13.5 mW
Size of common broadcast slot	0.2 secs		$d_{off}^{initial}$	2 secs
τ_{low}	0.5		τ_{high}	0.8
APPT_LEN_MULT	6		GUARD_BAND_SIZE	6 msecs

Table 1: Parameters for the simulations

to 40m. In the simulated network, each node has 6 neighbors on an average, and average size of two-hop neighborhood is 17. These characteristics are similar to that of the topology used in [14].

In our simulations, we have varied the duty cycle for S-MAC from 5% to 50%, and we have simulated Etiquette for two different office hour periods: 10 secs and 20 secs. The chosen value of various parameters used in the simulation is shown in table 1. The parameters for the physical layer were based on the characteristics of Chipcon CC1000, radio used in Mica Motes [10]. The parameters for transmit and receive power are based on those used in [25]. We assume that sensor nodes use Manchester encoding [25], using 2 bits to represent each bit.

As explained in section 4.4.2, nodes adapt the duration of their office hours. In this performance study, we have chosen to simulate an aggressive strategy that doubles the duration of office hours when fraction of idle time drops below τ_{low}, and reduces it linearly by 0.5 when the fraction exceeds τ_{high}. Analysis of effect of different strategies and choice of related parameters on performance is a subject for another paper.

For all simulated protocols, the length of MAC layer control packets and the length of MAC layer header was set to the same value.

We present the simulation results for *homogeneous traffic* scenario [14, 22]. Here each node generates packets based on exponential distribution. The mean of the distribution is a simulation parameter. For each

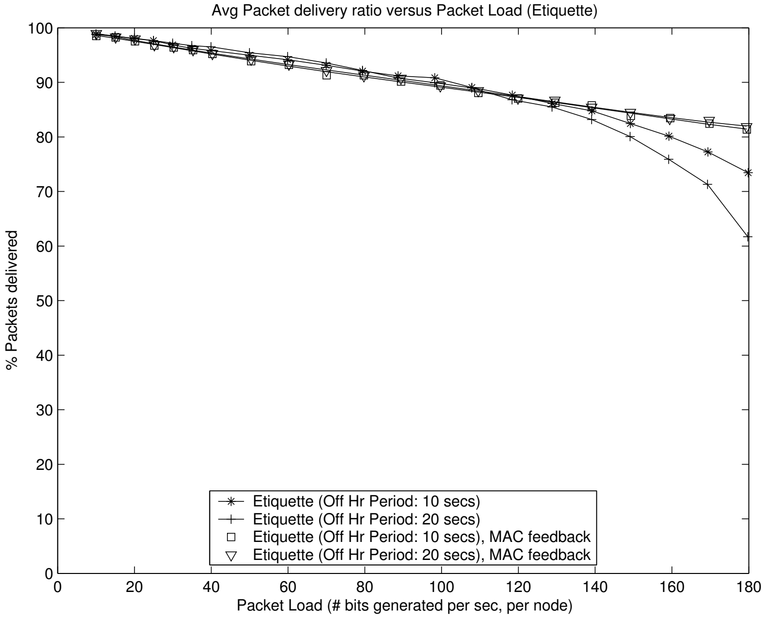


Figure 5: Simulations results illustrating the effect of using MAC layer feedback on performance

generated packet, a node randomly selects one of its one-hop neighbors as the destination. The simulation results for data gathering scenario were qualitatively similar. We have omitted them for lack of space.

5.3 Simulation Results

5.3.1 Effect of MAC layer feedback

We start by investigating the effect of MAC layer feedback on the performance of Etiquette protocol. Fig 5 shows the packet delivery performance of Etiquette protocol with and without MAC layer feedback for two office hour periods: 10 secs and 20 secs. In the variation without feedback, the `request-timeout` parameter is set conservatively and is not changed. While all four performance graphs overlap at light load, the performance of Etiquette without MAC layer feedback drops considerably at higher load (by 10% and 20% respectively for the office hour period of 10 secs and 20 secs). As explained in section 4.4.3, this is because of premature timeouts at the Etiquette layer. These timeouts cause Etiquette protocol to retransmit appointment request, exacerbating the channel load. As the performance graph shows adjusting `request-timeout` based on MAC layer feedback helps correct this problem.

5.3.2 Comparison between Etiquette, S-MAC, and IDEAL-802.11

Average Packet Delivery ratio

We first compare the performance of Etiquette, S-MAC, and IDEAL-802.11 with respect to the average packet delivery ratio (averaged over all the nodes in the network). Figure 6 shows that as the packet load increases beyond a certain threshold, the percentage of packets delivered in S-MAC drops sharply. This is because in S-MAC a certain duty cycle determines the maximum packet load that can be handled by the nodes. On the other hand, the percentage of packets delivered in Etiquette protocol degrades gracefully with increase in packet load. At the highest packet load simulated, the packet delivery ratio for Etiquette protocol stayed above 80%. Notice that the performance for two different office hour period is almost identical. This illustrates that office hour periods in Etiquette protocol does not constrain the maximum packet

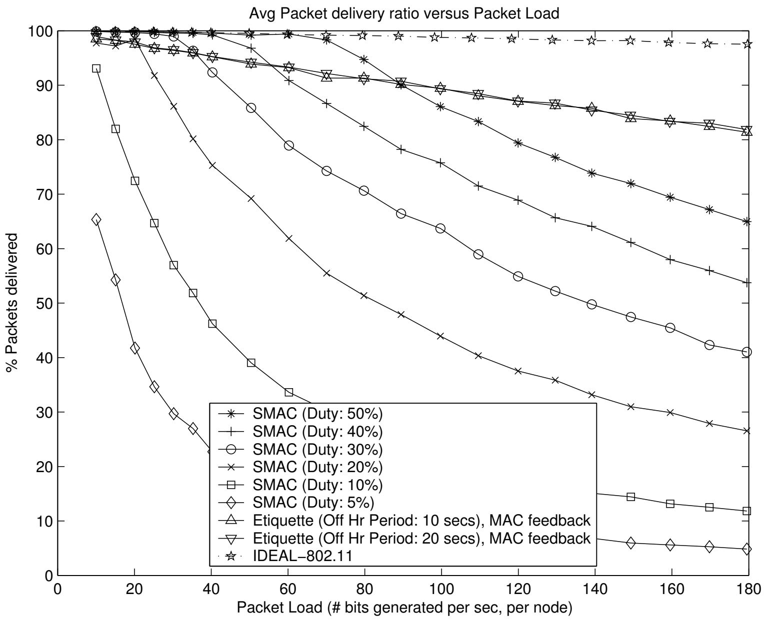


Figure 6: Simulations results comparing the percentage packets delivered for S-MAC, Etiquette, and IDEAL-802.11 as the packet load in the network is increased

load that can be handled by a node. Also notice that for lower packet load, the performance of Etiquette protocol is not 100%. This is because for the same packet load, more packets are transmitted in Etiquette than in S-MAC or IDEAL-802.11, leading to slightly higher collisions. This can be improved by introducing link-layer re-transmissions.

Average energy expended per delivered bit

Figure 7 compares the performance of S-MAC, Etiquette, and IDEAL-802.11 with respect to the average energy expended by the protocol in delivering each bit (averaged over all the nodes in the network). We do not extend any graph beyond the point when the corresponding packet delivery ratio falls below 70%.

The energy per delivered bit in all cases is high for light load This is because in both S-MAC and Etiquette protocol there is a certain minimum energy overhead associated independent of the packet load in the network. The larger the duty cycle in S-MAC, the higher the overhead. Similarly, the more frequent the office hours in Etiquette, the larger the overhead. As the packet load increases, this overhead gets amortized over the increasing number of packets delivered in the network resulting in decrease in energy per delivered bit. The performance of Etiquette protocol and S-MAC differs the most at low packet load and that it starts to converge at higher packet load. In the region of low packet load, the effectiveness with which different protocols cut down the energy consumed in the network is tested. In this region, Etiquette performs significantly better than S-MAC. As packet load increases, nodes need to have their radio ON for a greater fraction of the cycle, and the average cost of delivering each bit for all simulated protocols starts to converge.

In Etiquette protocol, the energy cost of delivering each bit reduces with the increase in office hour period. This supports our analysis in section 4.3.2.

Using the results for S-MAC, one can derive the performance of *Optimal-S-MAC*. In Optimal-S-MAC, the duty cycle is controlled by an Oracle. Given a target packet delivery ratio and the packet load in the network, the Oracle adjusts the duty cycle to the smallest one that possible. The performance of Optimal-S-MAC is shown in figure 7 using solid lines. Note that the energy per delivered bit of even Optimal-S-MAC is

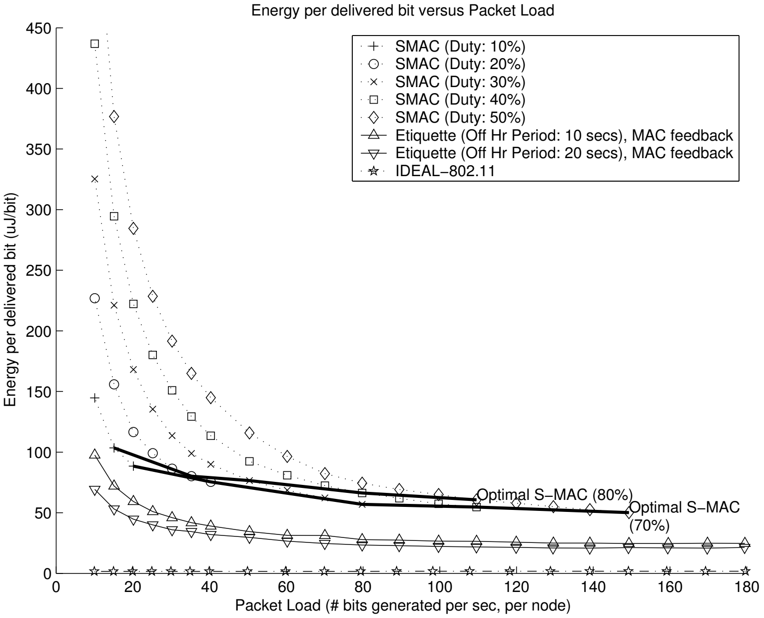


Figure 7: Simulations results comparing the average energy expended/bit for S-MAC, Etiquette, and IDEAL-802.11 as the packet load in the network is increased

higher than that of Etiquette by a factor of more than 2, even when Etiquette has higher packet delivery ratio.

Key Advantages of Etiquette over S-MAC

To better understand the reason for better performance in Etiquette, we examine the average time for which radio is ON in S-MAC and Etiquette protocol for a specific node[4] in the network (figure 8). The result were qualitatively similar for all the nodes in the network. Average duration for which radio is ON has a direct relationship with energy consumption. The results for S-MAC show that for a specific duty cycle, the average radio ON time remains almost constant, decreasing very slowly with the increase in packet load. This decrease is because nodes in S-MAC turn OFF their radio when they overhear a transmission. In contrast, in Etiquette the radio ON time is a function of the packet load. For light load, it is much smaller than that for any of the simulated duty cycle for S-MAC. For high packet load it is close to the radio ON time for S-MAC with duty cycle of 30%. And yet, it does better than S-MAC in terms of packet delivery ratio.

The primary reason S-MAC performs worse than Etiquette is that it synchronizes the active period of all nodes, thereby concentrating traffic in a small portion of the cycle. As a result, the nodes in Optimal-S-MAC experience a very busy channel (even though the packet load may be substantially smaller than the raw channel bandwidth). It is well known that CSMA/CA, the medium access scheme used in S-MAC, is not efficient at high load, leading to high number of collisions. In contrast, Etiquette spreads the traffic over the entire cycle, and thus, incurs significantly smaller number of collisions. Another important reason for better performance of Etiquette is that S-MAC needs to dedicate a considerable portion (∼ 38%) of its active period for exchanging SYNC messages even though it does not need fine synchronization. This leads to higher overhead in S-MAC than that in Etiquette (in holding office hours).

Average queuing delay

---

[4] The node under consideration is in a relatively sparse section of the network and has 5 one-hop neighbors.

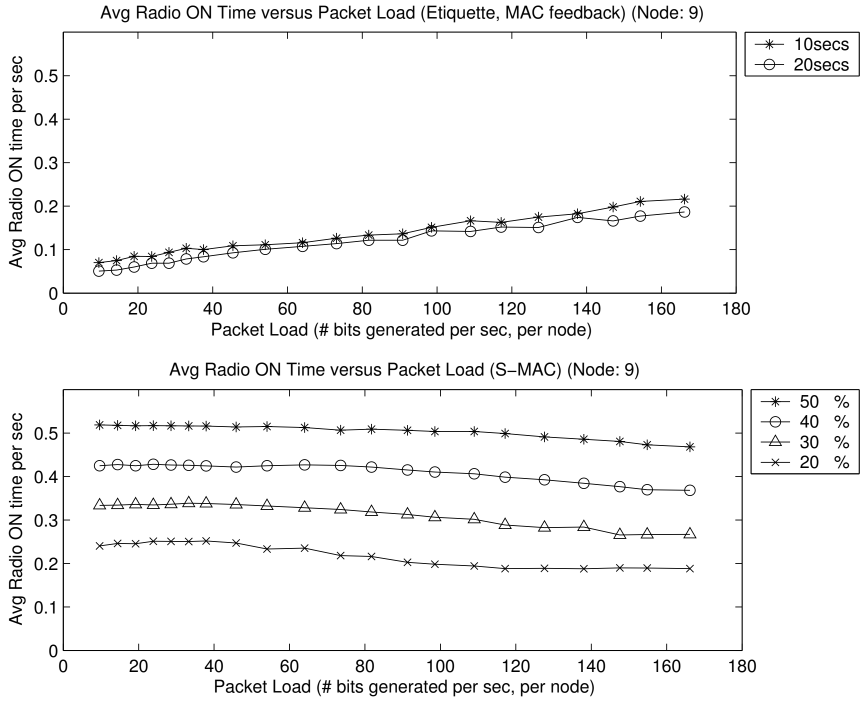


Figure 8: Avg radio ON time per sec in Etiquette and S-MAC protocol for a specific node

Figure 9 compares the performance of S-MAC, Etiquette, and IDEAL-802.11 with respect to the average queuing delay per packet. The average queuing delay is a function of the chosen cycle length (office hour period in Etiquette protocol) in both S-MAC and Etiquette protocol. As expected, in S-MAC the latency for smaller duty cycle is slightly higher than that for larger duty cycle. When the packet load reaches a certain threshold, the latency in S-MAC sky-rockets. The points at which this happens corresponds to the points at which the percentage packets delivered by S-MAC drops sharply (figure 6). As explained earlier, this shows that S-MAC has reached its maximum capacity for handling packet load for the chosen duty cycle.

The average queuing delay for Etiquette protocol is a function of the office hour period. Notice that the average queuing delay for a particular office hour period does not change with the increase in packet load. This demonstrates that the mechanism for adjusting the office hour duration is effective in responding to changes in instantaneous packet load.

In summary, in Etiquette protocol increasing the office hour period reduces the energy consumed per bit at the cost of increased latency. The duty cycle does not limit the maximum packet load that can be handled by the node. Thus, in order to minimize their energy consumption, all nodes would set their office hour period to P_{max}, the maximum office hour period defined for the network. By setting P_{max} to a suitable value, network designer can choose an appropriate operating point for the network on the energy-latency tradeoff curve that best suits the needs of the application.

5.3.3 Energy breakdown in Etiquette Protocol

We now examine the breakdown of the energy expenditure for Etiquette for a specific node in the network. Figure 10 shows the breakdown of energy cost in transmitting a packet (normalized by the number of bits the that were successfully received by the neighbors). The cost of transmitting a bit has three components: *cost of determining office hours*, *cost of requesting appointments*, and the *cost of sending packet during the appointment*. Note that among these, the cost of determining the office hours is the highest, and is a

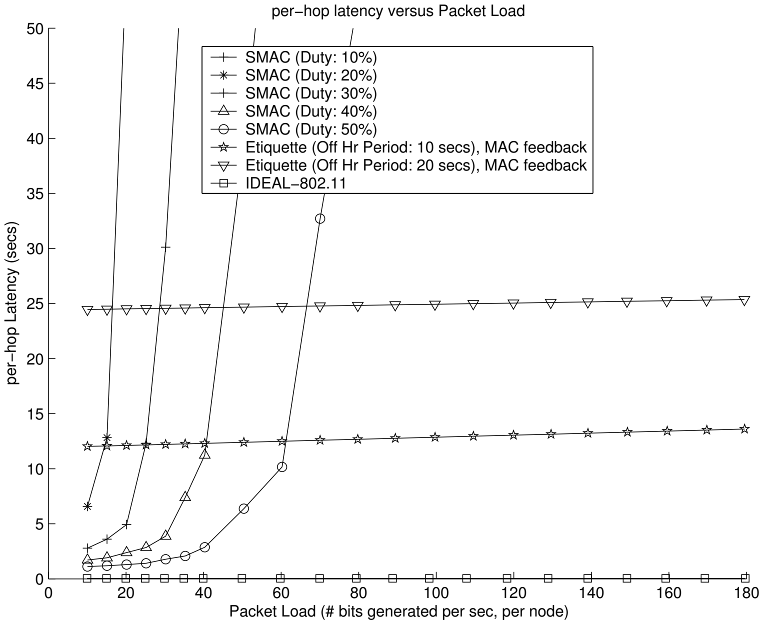


Figure 9: Simulations results comparing the average queuing delay in S-MAC, Etiquette, and IDEAL-802.11 as the packet load in the network is increased

significant fraction of the total cost. This illustrates the importance of using blurbs in reducing the scanning time (section 4.3.2). This cost reduces with increase in packet load because a node is communicating with each of its neighbors more often. As a result, with a greater probability it is able to find their office hours in its cache. The other two energy costs remain almost constant. They increase slightly with the increase in packet load, because of the decrease in the corresponding packet delivery ratio.

Figure 11 show the breakdown of energy cost in receiving a packet (normalized by the number of bits received successfully by the node). This cost has three components: *cost of holding office hours*, *cost of transmitting blurbs*, and *cost of receiving packet during the appointment*. Among these, the cost of holding office hours is the highest and is a significant portion of total cost. It is high for low packet load because a node needs to hold office hours even when there is little or no traffic. It decreases with the increase in packet load, becoming almost constant beyond a certain packet load. The cost of transmitting blurbs also reduces with the increase in packet load. As explained above, this is because fewer number of nodes scan the channel at high load. This illustrates the effectiveness of the mechanism for controlling the number of blurbs transmitted by a node. The cost of receiving a packet during an appointment remains constant as expected. Note that this cost is bigger than that the corresponding cost for sender in transmitting a packet during an appointment. This is because the sender is able to turn OFF its radio in the middle of an appointment once it receives MAC layer feedback about successful delivery (section 4.4.3).

6 Related Work

Energy-efficient communication is a hotly pursued topic of research in sensor networks. The work on MAC layer protocol for sensor networks is perhaps most closely related to our work. In this section, we briefly review the protocols that are most relevant to our work. Interested readers are referred to [24, 12] for an excellent survey of MAC protocols for network of energy-constrained nodes. Broadly, the MAC layer protocols can be classified into two categories: contention-based and schedule-based.

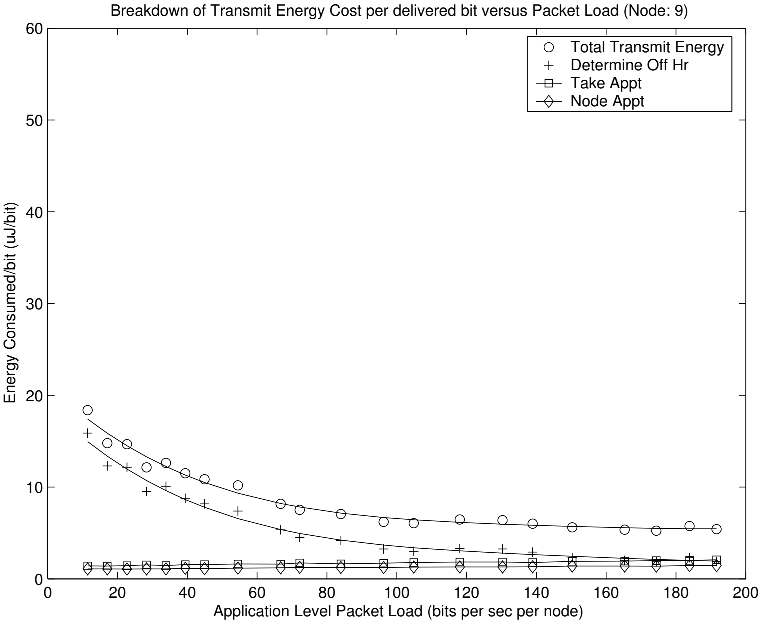


Figure 10: Breakdown of the transmit cost in Etiquette (for a specific node)

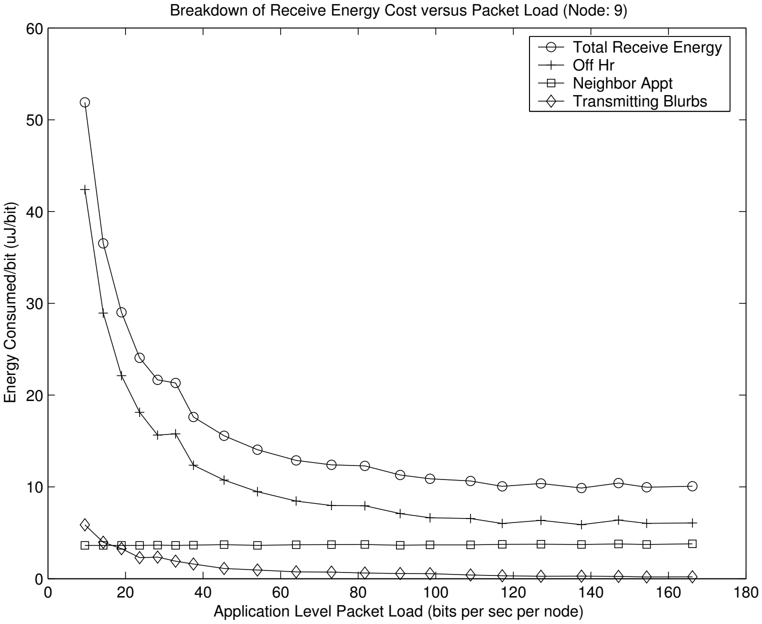


Figure 11: Breakdown of the receive cost in Etiquette (for a specific node)

Arguably the most popular contention-based protocol is the distributed coordination function (DCF) of the IEEE 802.11 standard [2]. It uses CSMA/CA to allow nodes to contend for access to the medium, while reducing the chances of collision. However, the asynchronous nature of the protocol requires nodes to always keep their radio ON, resulting in high energy consumption. IEEE 802.11 standard has defined power-saving mode of operation for infrastructure and ad-hoc mode. However, both these modes only work for single-hop network [21].

PAMAS [18] was among the earliest energy-efficient contention-based protocol. It used out-of-band signaling and proposed cutting down on idle listening by turning radio OFF when neighboring nodes participate in a communication. Woo and Culler [23] have proposed rate based control mechanism for achieving fair bandwidth allocation among contending sensor nodes.

S-MAC [25] introduced a novel technique for making basic CSMA/CA mechanism energy-efficient. As pointed our earlier, the two main drawbacks of S-MAC are: Firstly, it uses same pre-set duty cycle for all the nodes, forcing the network designer to select a value based on the worst-case estimate of the packet load in the network. This is clearly wasteful because not only worst case requirements can be significantly higher than average case, some nodes may have substantially higher duty cycle requirements than others (e.g., in data gathering sensor networks, packet load close to the sink node is higher). Secondly, by virtue of its design, this protocol concentrates traffic during the ACTIVE portion of the cycle. Since CSMA/CA is not efficient at higher packet load, it leads to higher collisions.

T-MAC [22] protocol modifies S-MAC in order to allow individual nodes to adapt their duty cycle based on their packet load. By default a node is in ACTIVE state for the entire cycle (i.e., 100% duty cycle). It switches to SLEEP state during the cycle if it does not sense *any* radio activity[5] in its neighborhood for TA units of time (TA was set to 15 msecs in [22]). The main drawback of T-MAC is that it aggressively concentrates the traffic in the beginning of the ACTIVE cycle. However, it uses CSMA/CA for medium access, which is not efficient at high load, resulting in higher number of collisions. As a result, while T-MAC works well for light packet load, it breaks down at higher packet load. Detailed study in [12] illustrates this characteristics.

IEEE 802.15.4 standard [4] defines the PHY and MAC specification for battery constrained nodes in low rate wireless personal area networks. However, the standard does not address the idle-listening problem in peer-to-peer communication mode. Distributed Mediation Device (MD) protocol [17] attempts to address this problem. In this protocol, nodes communicate with each other via mediation device (MD) by leave "intent-to-communicate" for intended receivers. This mode of operation forces nodes to regularly poll any MD device. This puts the onus of communication on the receiver and is wasteful for a node when no neighbors intends to communicate with it.

The protocol proposed by Sohrabi and Pottie [19] is an example of schedule based protocol that uses fixed slot assignment. Clearly this is not adpative to the needs of each node, and idle slots waste bandwidth and energy. T-RAMA [14] adapts slot assignment to the needs of the node. However it requires each node to maintain a consistent view of two-hop topology information, and knowledge of schedules of all its neighbors. This is costly, especially in dense networks. Also, each node needs to maintain state that is proportional to the density of nodes in the network. These two aspects affect the scalability and robustness of the protocol.

In ReSync [5] protocol nodes periodically send out “intent” messages indicating the address of the destination node and the time at which they would communicate. All nodes to keep track of and tune into the “intent” messages of each of its neighbors. This is wasteful for a node when only a small fraction of its neighbors are communicating with it in every cycle. This problem gets exacerbated in dense networks.

Flexible power scheduling [11] shares some of the features of Etiquette protocol in that it places the onus of communication on the sender and it operates above existing MAC layer. However, Flexible Power Scheduling divides time into slots whereas Etiquette protocol is “slot-free”. This avoids the need for nodes to synchronize on slot boundaries. It also allows nodes to request for appointments of variable lengths. The other important difference is that Etiquette protocol supports many-to-many communication whereas Flexible Power Scheduling does not.

---

[5] Radio activity refers to high RSSI value irrespective of whether the node is the target of communication or not.

7 Future Work

In the short term, we are working on filling the details for supporting local broadcast in Etiquette protocol. We would also like to extend Etiquette protocol to effectively support periodic sources with different periodicity in the same network.

In the long term, we see a number of interesting and challenging issues when adapting Etiquette protocol for use in networks other than sensor networks:

1. *Mobile ad hoc networks:* The problem of idle listening also plagues multi-hop mobile ad hoc networks. Although IEEE 802.11 has defined power-saving operation in ad hoc mode, the mechanism works only in single-hop network where all nodes can hear each other. Tseng, et al [21] have proposed three schemes for extending power-saving mode of IEEE 802.11 to multi-hop network. Among these, the Quorum-based scheme is the most efficient. However, even in this scheme, a node needs to keep its radio ON during the entire quorum interval leading to higher overhead. The period of such quorum intervals is a function of the tolerable latency. A challenging research issue is to reduce the energy overhead (for the same delay) while at the same time ensuring that the node is able to inter-operate with radios running IEEE 802.11. We would like to investigate if Etiquette protocol can be adapted for this purpose.
2. *Cognitive radio [16]:* It has been envisaged that as the number of wireless devices increases, the nodes would be required to negotiate use of spectrum to avoid interfering with each other and for better spectrum efficiency. Recent work on cognitive radio at Winlab [16] is exploring use of common signaling channel (Common Spectrum Coordination Channel) for ad hoc spectrum negotiation. An interesting research issue is to explore if Etiquette protocol can be adapted to support such a signaling channel energy-efficiently.

8 Conclusions

Etiquette protocol proposed in this paper reduces idle-listening by allowing nodes to dynamically and flexibly schedule their communication in a multi-hop network. This completely distributed protocol allows nodes to adapt their schedule in response to fluctuating packet load. We have shown through simulations that Etiquette protocol significantly reduces the average energy consumed in delivering each bit when compared with S-MAC, especially at low packet load. The specific performance improvement is dependent on the packet load in the network. We also showed that Etiquette protocol allows a network designer to trade increased latency for reduction in energy consumption. Etiquette protocol is simple and intuitive. This makes it suitable for use in sensor nodes with small on-board memory.

9 Acknowledgments

We thank Prof. Roy Yates for helpful discussions that improved the quality of this work. We thank Prof. Richard P. Martin, Andrea Passarella, Budhaditya Deb, Jaewon Kang, Pandurang Kamat, and K.V. Vihari for their insightful comments that greatly helped us in improving the quality of this paper.

References

[1] The Network Simulator - ns-2. http://mash.cs.berkeley.edu/ns.

[2] IEEE Std. 802.11a - Wireless LAN Medium Access Control (MAC) and Physical layer (PHY) layer specifications: High Speed Physical Layer in the 5 GHz Band, 1999.

[3] Specification of the Bluetooth system: Core, 2001. http://www.bluetooth.org.

[4] *802.15.4-2003 IEEE Standard for Information Technology-Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) specifications for Low Rate Wireless Personal Area Networks (LR-WPANS).* IEEE, New York, New York, NY, 2003.

[5] W. Steven Conner, Jasmeet Chhabra, Mark Yarvis, and Lakshman Krishnamurthy. Experimental Evaluation of Synchronization and Topology Control for In-building Sensor Network Applications. *Proc. 2nd ACM International Conference on Wireless Sensor Networks and Applications*, September 2003.

[6] Arikan E. Some Complexity Results About Packet Radio Networks. *IEEE Transactions on Information Theory*, 30(4):681–685, July 1984.

[7] Saurabh Ganeriwal, Ram Kumar, and Mani B. Srivastava. Timing-Sync Protocol for Sensor Networks. In *Proceedings of the first Int'l Conference on Embedded Networked Sensor Systems (SenSys)*, pages 138–149. ACM Press, 2003.

[8] J.C. Haartsen. The bluetooth radio system. *IEEE Personal Communications*, 7:28–36, February 2000.

[9] W. Heinzelman, A. Chandrasekaran, and H. Balakrishnan. Energy-efficient communication protocol for wireless microsensor network. In *Proceedings of the 33rd Hawaii International Conference on System Sciences (HICSS '00)*, January 2000.

[10] J. Hill, R. Szewczyk, A. Woo, S. Hollar, and K. Pister D. Culler. System architecture directions for networked sensors. In *Proceedings of the 9th International Conference on Architectural Support for Programming Languages and Operating Systems*, November 2000. http://tinyos.millennium.berkeley.edu/papers/tos.pdf.

[11] Barbara Hohlt, Lance Doherty, and Eric Brewer. Flexible Power Scheduling for Sensor Networks. Technical Report UCB/CSD-03-1293, University of California, Berkeley, January 2003.

[12] K. Langendoen and G. Halkes. *Energy-Efficient Medium Access Control.* Book Chapter in the Embedded Systems Handbook, CRC Press, 2004. To Appear.

[13] LAN MAN Standards Committee of the IEEE Computer Society. *Wireless LAN MAC and PHY Specifications, IEEE Std 802.11-1999 Edition.* IEEE, New York, New York, NY, 1999.

[14] Venkatesh Rajendran, Katia Obraczka, and J. J. Garcia-Luna-Aceves. Energy-efficient collision-free medium access control for wireless sensor networks. In *Proceedings of the first Int'l Conference on Embedded Networked Sensor Systems (SenSys)*, pages 181–192. ACM Press, 2003.

[15] Theodore S. Rappaport. *Wireless Communications: Principles and Practice.* Prentice Hall, 2001.

[16] D. Raychaudhari. Adaptive Wireless Networking Using Cognitive Radios as a Building Block. *Keynote speech at Tenth Annual Int'l Conference on Mobile Computing and Networking (Mobicom)*, October 2004.

[17] Qicai Shi and Edgar Callaway. An Ultra-Low Power Protocol for Wireless Networks. *Proc. 5th World Multi-Conference on Systemics, Cybernetics and Informatics*, 4(5):321–325, 2001.

[18] S. Singh and C.S. Raghavendra. PAMAS - Power Aware Multi-Access protocol with Signalling for Ad Hoc Networks. *ACM Computer Communication Review*, 28(3), July 1998.

[19] K. Sohrabi, J. Gao, V. Ailawadhi, and G. Pottie. Protocols for Self-Organization of a Wireless Sensor Network. *IEEE Personal Communications, Special issue on "Networking the physical world"*, 7(5):16–27, October 2000.

[20] R. Szewczyk, J. Polastre, A. Mainwaring, and D. Culler. Lessons from a Sensor Network Expedition. In *Proceedings of the First European Workshop on Wireless Sensor Networks (EWSN '04)*, January 2004.

[21] Y. Tseng, C. Hsu, and T. Hsieh. Power-Saving Protocols for IEEE 802.11-Based Multi-Hop Ad Hoc Networks. In *Proceedings of IEEE INFOCOM*, June 2002.

[22] Tijs van Dam and Koen Langendoen. An Adaptive Energy-Efficient MAC Protocol for Wireless Sensor Networks. In *Proceedings of the first Int'l Conference on Embedded Networked Sensor Systems (SenSys)*, pages 171–180. ACM Press, 2003.

[23] Alec Woo and David Culler. A Transmission Control Scheme for Media Access in Sensor Networks. In *Proceedings of 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom '01), Rome, Italy*, 2001.

[24] Wei Ye and John Heidemann. Medium Access Control in Wireless Sensor Networks. Technical Report ISI-TR-580, USC Information Sciences Institute, October 2003.

[25] Wei Ye, John Heidemann, and Deborah Estrin. Medium Access Control with Coordinated, Adaptive Sleeping for Wireless Sensor Networks. *IEEE/ACM Transactions on Networking*, June 2004.