

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, The State University of NJ

*Rutgers Department of Computer Science Technical Report
DCS-TR-552*

April 2004

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 [17]. 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, the cost of replacing the battery of the sensor nodes is an expensive operation, exceeding the cost of the node itself. This typically happens when sensor networks 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 [9]. 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. 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 [16] 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 [2, 7, 8], and have been successfully used in commercial wireless communication systems such as GSM [13]. 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 [5].

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.

Recently, a number of distributed protocols have been proposed in literature [4, 10, 12, 14, 15, 17] that attempt to schedule communication among sensor nodes of a multi-hop network. Etiquette protocol differs from these protocols in the following important ways:

- In Etiquette protocol, the onus of communication lies with the sender. As we explain later, this feature is critical to operation in dense networks.
- Duty cycle of nodes in Etiquette protocol does not limit the maximum packet load they can handle.
- Etiquette protocol supports many-to-many communication, including ability to perform local broadcast.

We elaborate on these differences in section 4. Our simulation results show that Etiquette protocol significantly reduces the average energy consumed in delivering each bit compared to S-MAC [17] protocol.

Etiquette protocol is well suited for sensor networks deployed for data gathering and event tracking applications.

In the next section we provide details of the Etiquette protocol. In section 3, we present simulation results showing performance of Etiquette protocol. In section 4, we discuss related work and, finally in section 5, we conclude the paper.

2 Etiquette protocol

2.1 Analogy

We explain the intuition behind Etiquette protocol and its working with the help of an analogy. Consider a lazy teaching assistant (TA) 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.

2.2 Basic idea

Our key observation is that *bandwidth is “free” in sensor network*. The “expensive” commodity in this network is the node’s energy. This observation holds for our target class of applications where packet load is small compared to the channel bandwidth. This simple observation suggests that a strategy that is able to save the energy of sensor nodes, even at the expense of consuming extra bandwidth, is a winner. Etiquette

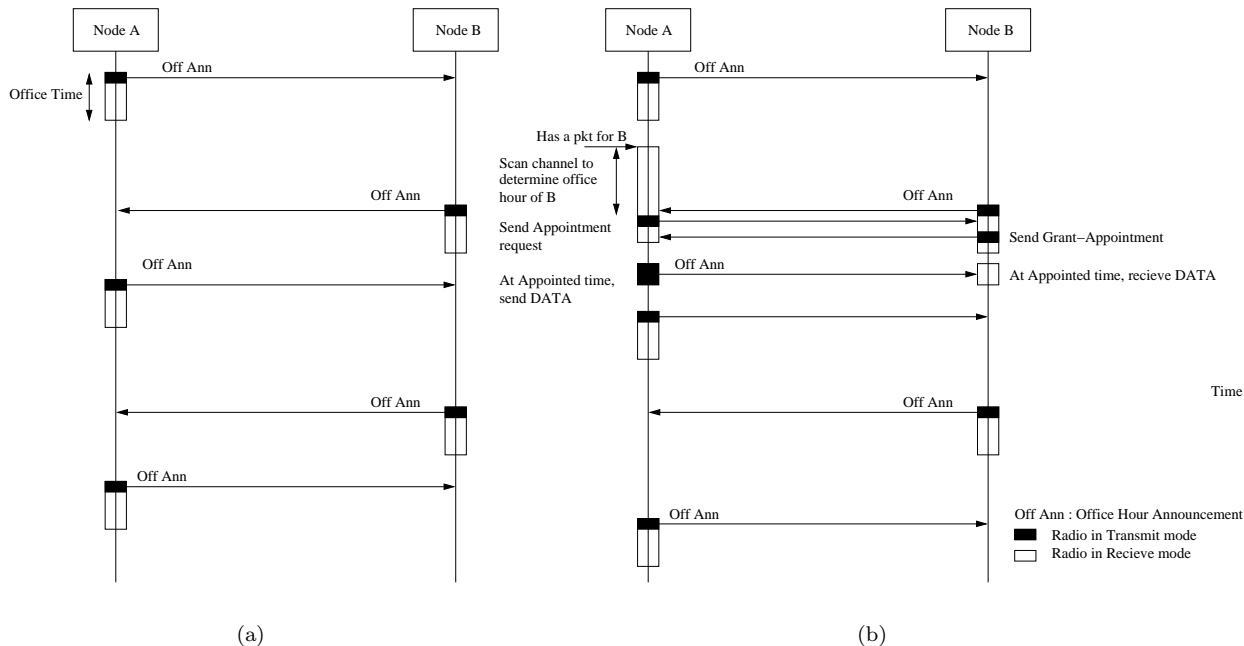


Figure 1: Schematic diagram illustrating the basic protocol. (a) Nodes hold office hours at regular intervals. (b) Steps performed by node A in communicating a data packet to node B.

protocol is based on this observation. Our key idea is to allow dynamic and flexible scheduling of inter-node communication.

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 (figure 1(a)). An appointment request packet specifies the desired duration of appointment (based on size of the payload). The node has the option of *granting* the appointment request or *denying* it. *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 1(b) illustrates this interaction. With this style of interaction, a node knows the precise interval of time at which its neighbor is going to communicate with it. At all other times it can turn its radio OFF without loss of communication.

Nodes are allowed to 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.

The procedure for determining the office hours of a neighboring node (say, B) is very simple. A node simply turns its radio ON to start scanning 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. A node caches all the received office-hours announcement messages 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 designate any time-slice of its choice as its office hours, independent of the office hours of its neighbors. It also allows the receiver to change its office hours without the need for informing any of its neighbors. This greatly simplifies the operation of the node. 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. Now a node may be idle-listening only during its office hours. We discuss strategies for further reducing idle-listening in section 2.4.2. Schemes that put onus of communication with the receiver (e.g., [4]) do not scale with the density of the network. This is because a node has to regularly poll all its neighbors to check if anyone is interested in communicating with it.
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. Note that although the communication is scheduled, there is still a chance of collision. This is because a node selects the appointment slots based on locally available knowledge of when the channel is free. It is possible for neighboring nodes to assign overlapping appointment slots. As explained in section 2.3.3, we address this problem by simply using larger appointment slots than required for communicating the payload, and letting MAC layer handle the overlaps in appointment slots.
3. Etiquette protocol is “*slot-free*” — it does not divide the time in slots. This *avoids* any need for nodes to synchronize on slot boundaries.
4. 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. They turn their radio ON at the start of the guard band thereby preventing loss of packets because of timing errors. We discuss this in more detail in section 2.3.4.

These aspects are key to the simplicity of the protocol.

2.3 Details

In order for node A to communicate with its neighbor B, it needs to negotiate an appointment with B. This involves three steps: determine B’s office-hours, request an appointment during B’s office-hours, and finally, perform the communication at the appointed time. In this section, we describe each of these basic steps in detail.

2.3.1 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.

Assuming that the office hours of node B has a period of T_{off}^B , the expected scanning time for node A is $E[T_{scan}^A] = \frac{T_{off}^B}{2}$. In applications where the latency requirement is relaxed, T_{off}^B 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 ($E[T_{scan}^A]$) by half. Since a blurb is a short message, the energy expended by node B in sending it is very small. Thus, sending blurb saves much more energy than it costs. If node B were to send two equally spaced blurb messages, it would reduce $E[T_{scan}^A]$ 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 $\frac{T_{off}^B}{2(n+1)}$.

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 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 is given by:

$$O_{off-hr-disc} = 3D J_{blurb} + K \frac{T_{off}^B}{2(n+1)} J_{recv} \quad (1)$$

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

$$n^* = 3D \sqrt{\frac{KT_{off}^B J_{recv}}{2J_{blurb}}} - 1 \quad (2)$$

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 appointment request if it obtained the node's current office hours by scanning. Assuming that recent past is a good predictor for future, node can use this estimate 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 to the total overhead in the network:

$$O_{off-hr-disc}^* = 3D \sqrt{2KT_{off}^B J_{recv} J_{blurb}} - J_{blurb} \quad (3)$$

Next, we derive the expression for a node's total contribution to overall energy overhead in the network.

Holding office hours depletes the energy reserves at a node. This energy overhead has a constant and a variable component. The constant component, O_{const} , is contributed by the energy spent in turning the radio ON at the start of office hours, sending the office hour announcement message, and turning the radio OFF at the end of office hours. Any idle time during the office hours contributes to the variable component. The magnitude of variable component is dependent on the strategy for dynamically adjusting the duration of office hours (we discuss this in more detail in section 2.4.2). For simplicity, let us assume that the chosen strategy is able to make this component negligible. Thus,

$$O_{\text{holding-off-hr}} = 3D_{\text{const}} \quad (4)$$

When a node holds office hours, it contributes to the total energy overhead in the network. This contribution per office hour period is a sum of three factors: (a) Energy overhead in holding office hours (equation 4), (b) Energy overhead in sending blurbs, and (c) Energy overhead of neighboring nodes in scanning the channel to determine the node's office hours. Assuming that a node is always able to choose the optimal number of blurbs to send during each office hour period, the sum of last two factors is represented by equation 3. For a node with office hour period T_{off} , the total energy overhead contributed by the node over a long time interval T is given by:

$$O_{\text{overhead}/\text{node}}(T) = 3D_{\text{const}} \frac{T}{T_{\text{off}}} (O_{\text{holding-off-hr}} + O_{\text{off-hr-disc}}^*) \quad (5)$$

From equations 3, 4, and 5, we get:

$$\begin{aligned} O_{\text{overhead}/\text{node}}(T) &= 3D_{\text{const}} \frac{T}{T_{\text{off}}} (O_{\text{const}} + \sqrt{2KT_{\text{off}}J_{\text{recv}}J_{\text{blurb}} - J_{\text{blurb}}}) \\ &= 3D_{\text{const}} \frac{T}{T_{\text{off}}} (O_{\text{const}} - J_{\text{blurb}}) + T \sqrt{\frac{2KJ_{\text{recv}}J_{\text{blurb}}}{T_{\text{off}}}} \end{aligned} \quad (6)$$

Equation 6 captures an interesting aspect of the system. Equation 3 shows that a node's contribution to the overhead increases as the period of office hours increases. However, as the office hour period increases, the node holds fewer office hours. Equation 6 shows that the combination of these two factors results in an overall decrease in the energy overhead contributed by a node. 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 place 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 any operating point that best fits the needs of the applications.

2.3.2 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 $c * P_{\text{max}}$ units of time in order to gather the 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. It uses the gathered information in selecting a time-slice for its office hours. It *tries* to select a time-slice that has minimum 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*.

2.3.3 Requesting appointment

An appointment request specifies four parameters:

1. *Appointment length*: This is a function of the size of the payload that node A 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. *Desired Time Interval for appointment*: 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*: This 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.

When deciding on the appointment time, node B looks for an appropriate-size slot when the channel is expected to be free. For this purpose it uses only locally available information. This includes its own set of appointments, schedule of its current and next office hours, and its knowledge of appointments of its neighbors (which is very likely to be incomplete). Node B needs to select a slot that does not overlap with these existing “reservations”. The slot should also lie in the time-interval specified in the request, and it should end before the starting of the B’s next office hours. If node B finds more than one slots that matches the constraints, it chooses the one that starts sooner.

Node B attempts to avoid overlap between the appointment slot and its office hours. However, an overlap is allowed if no other slot fits the constraints. Note that an overlap is likely to be more energy-efficient for node B, as it effectively reduces the duration of its office hours. However, this may prevent some neighbors from seeking an appointment with node B.

If requesting node asks for more than one appointment, node B attempts to find appointment slots that are back-to-back — separated by a small inter-frame space. This has the advantage that appointments can be conveyed in a compact form. It is possible for node B to grant less than the requested number of appointment slots. Note that requesting multiple appointments reduces the overhead associated with communicating a data packet.

If node B is able to find an appointment slot it sends a *grant-appointment* message to node A containing the start time of the appointment. Additionally, it includes all the fields of the appointment request except for the “desired interval for appointment”. On the other hand, if node B is unable to find a slot that satisfies all the constraints, it sends a *deny-appointment* message to node A.

Note that it is quite possible that the appointment slot chosen by node B overlaps with that of its neighbor(s). The existing MAC layer handles channel contention in such a case. In most cases, we expect MAC layer to be able to recover from such a condition as the channel load is expected to be smaller than the channel bandwidth. In order to allow MAC layer a chance to recover from this condition, the requesting node always requests an appointment slot that is m times the size actually needed for communication, where m is a small number (e.g., 2). Requesting an appointment slot greater than the actual need may sound inefficient at first. However, its biggest advantage is that it allows a node to operate based on its local knowledge. This avoids the need for collecting schedule of neighbors thereby making the protocol scalable to dense networks.

Periodic appointments share the same period as that of the office hours of node B. As a result, when node B grants an appointment to node A, it repeats at the same relative position with respect to the start of the office hours of node B.

If node A receives grant-appointment message, it records the appointment time, turns OFF its radio, and waits for appointment slot. On the other hand, if node A receives deny-appointment message, it selects another “desired-interval” and requests again. If all possible intervals have been exhausted, it postpones its request until the next office hours of node B. It discards the packet if the total number of attempts exceed a certain *retry-threshold*.

It is possible for node A to not hear any response from node B. Node A waits for a certain pre-defined *request-timeout* interval before sending the request again. If the number of attempts exceed the *retry-threshold*, it assumes that node B is no longer in the neighborhood, or is dead. It discards all the packets destined for node B.

2.3.4 Communicating at the appointed time: Handling Timing issues

Both nodes A and B turn ON their radio at the appointed time and perform the communication. Note that in reality it is possible for node A’s knowledge of appointment time to differ from node B’s knowledge of appointment time. This is because when node B sends out grant-appointment message, it mentions the appointment time as an offset from its current time. However, this packet does not reach node A instantly. It

incurs delay which is a combination of MAC layer’s delay in accessing the channel and the transmission time for the packet. Node A can only compensate for the transmission time for the packet; it cannot determine the random delay at the MAC layer in accessing the channel. In order to take care of this problem, nodes introduce a “*guard band*” around the appointment slot. Although the actual communication happens at the appointment time, both the nodes turn ON their radio at the start of the guard band. This prevents loss of communication. Note that the size of guard band is dictated by the maximum uncertainty. This uncertainty can be substantially reduced by using a MAC layer that timestamps the packet just before transmitting it [6].

2.3.5 Support for local broadcast/multicast

In order to support broadcast, all nodes agree on a common broadcast slot. This slot is of small fixed size and it repeats periodically at a pre-defined fixed rate. All nodes keep their radio ON during this slot. A node has two options for broadcasting a packet — it can directly send the packet during the broadcast slot, or it can send a small control message (broadcast appointment message) during the broadcast slot indicating the time that the node would send a broadcast. In the latter option, all neighbors tune in at the appointed time to receive the broadcast. This option is useful for broadcasting packets whose size exceeds a certain pre-defined threshold. This allows a small broadcast slot to accommodate many packets.

Nodes agree on a common broadcast slot in a manner similar to the way in which nodes synchronize their ACTIVE state in S-MAC. Instead of sending SYNC messages as in S-MAC, nodes include their chosen common broadcast slot in their office hour announcement message. As in S-MAC, it is possible for a node to have neighbors that have different notion of common broadcast slots. 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. Thus, having common broadcast slot whose period can be set independently of the office hour period of nodes is advantageous.

2.4 Optimizations

2.4.1 Short-circuiting appointments

If a node wants to communicate a small packet to its neighbor, instead of requesting an appointment for it, 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 [11], where a node does not send RTS/CTS messages for packets whose size is below the `RTSThreshold`.

2.4.2 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. Later on, it adapts the duration based on its load of *appointment-requests*. 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. For sensor networks monitoring events, where occurrence of an event can trigger a burst of data communication, an aggressive strategy should be used for changing office hours. On the other hand, for sensor networks sensing habitats and environments, where sensors report periodically, a conservative strategy is more appropriate. If the

response of the chosen strategy does not match the application requirement, either the office hour duration would be smaller than needed or it will be larger than needed. The first case will force certain nodes to try requesting for appointment in the subsequent office hours thereby increasing latency. In the latter case, the node would be idle for certain fraction of its office hours, thereby wasting energy. Thus the choice represents a tradeoff between energy consumed and latency of operation.

2.4.3 Caching information about office hours of neighbors

Once a node gets to know the office hours of its neighbor (either by accidental overhearing or by actively scanning for it), it caches the information. Channel scanning is an expensive operation, and caching allows a node to avoid repeating it during every round. It is important to note that the office hours of a neighbor may change without notice. However such changes happen rarely. Thus, in most cases caching is effective. This cached information is aged unless renewed.

3 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. We simulated IDEAL-802.11 protocol by setting the energy consumed in idle-listening to 0.

For now, our comparison of these three protocols is based on simulations. We have used ns-2 [1] simulator for this purpose.

3.1 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 a packet spends 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. For broadcast packets, a packet is considered received if all the neighbors have received the packet.

3.2 Simulation setup and parameters

We have chosen a 6 node topology shown in figure 2 in our simulations. In this topology, nodes are arranged in a grid where each cell is a square. The communication range is set such that a node located at one vertex of the cell can only communicate as far as the node on the diagonally opposite corner of the cell. As a result, the pair of nodes in the middle are in communication range of all the nodes, whereas pair of nodes at the end can only communicate with each another and with the pair in the middle. This creates not only multiple nodes contending for the channel but also creates hidden terminals. Though this topology is not representative of topologies in real world, it simplifies our task of creating a scenario with not only multiple nodes contending for the medium but also hidden terminals. We believe that our simulations results hold qualitatively for general topologies.

In our simulations, each node generates packets based on exponential distribution. We varied mean of the distribution. Each time node has a packet to transmit, it randomly selects one of its one-hop neighbors as the destination. This scenario is based on the “stress-test” used in [12].

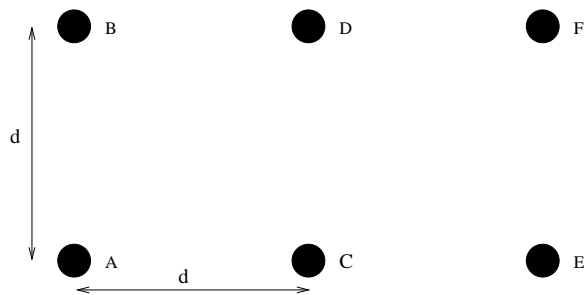


Figure 2: Simulation Topology

Data Packet Size	55 bytes	Transmission Power	24.75 mW
Channel Bandwidth	20 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

Table 1: Parameters for the simulations

We measure the performance of the protocols for two different duty cycles: 10% and 20%. The parameters for the simulation are shown in table 1. The parameters for transmit and receive power are based on those used in [17]. We assume that sensor nodes use Manchester encoding [17], which encodes each bit as 2 bits.

It is important to point out two features of the Etiquette protocol used in the simulation:

- As explained in section 2.4.2, nodes adapt the duration of their office hours. In this 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 by half when the fraction exceeds τ_{high} .
- In the context of Etiquette protocol, we define duty cycle as the ratio of initial office hour duration ($d_{off}^{initial}$) to the office hour period. For simulating duty cycles of 10% and 20%, we use the same $d_{off}^{initial}$ (2 secs), and but set the office hour period to 20 secs and 10 secs, respectively. If instead we had used the same office hour period and chosen two different values of the initial office hour duration for simulating duty cycles of 10% and 20%, we would not see any significant difference in performance. This is because nodes quickly adapt the duration of their office hours to match the load.

3.3 Results

3.3.1 Effect of sending blurbs in Etiquette Protocol

In order to show the effect of sending blurbs in Etiquette protocol, we do not allow nodes to automatically adjust the number of blurbs sent. Instead we set a number that is used by all the nodes in the network. This number is a parameter in our simulations. Figure 3 show its effect on energy expended per bit. Compared to the performance when no blurbs are sent, sending two blurbs per office hour period significantly reduces the energy expended per bit, improving the energy efficiency of the protocol. However, as the number of blurbs sent per period are increased further, the average energy cost of delivering each bit increases. Thus, for this setup and scenario, the optimum network-wide setting for the number of blurbs is 2.

It is important to note that, in general, allowing nodes to automatically adjust the number of blurbs sent is more efficient that a network-wide setting for the number of blurbs.

For the specific range of number of blurbs considered here, we have not seen any adverse effect on the packet delivery ratio in the network. This is to be expected as blurbs are very small messages.

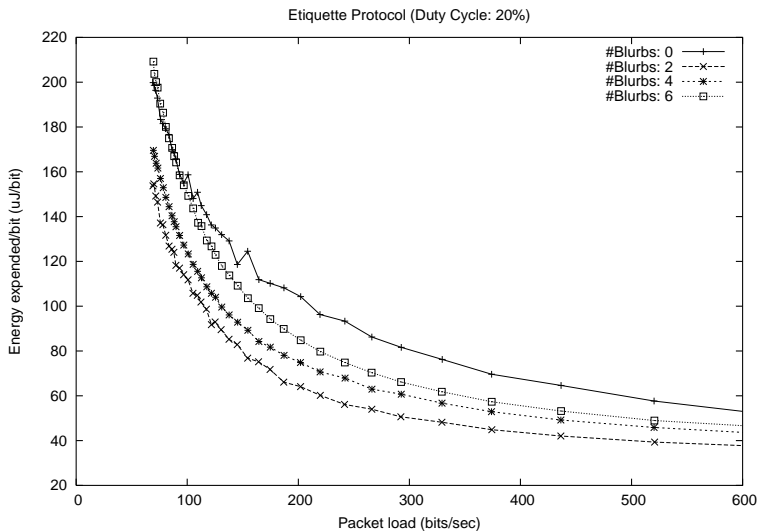


Figure 3: Simulations results showing the effect of sending blurbs on the average energy expended in delivering each bit for Etiquette protocol

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

We first compare the performance of Etiquette, S-MAC, and IDEAL-802.11 with respect to the packet delivery ratio. For both S-MAC and Etiquette protocol, we consider two duty cycles: 10% and 20%. Figure 4 shows that as the packet load increases beyond a certain threshold, the percentage of packets delivered in S-MAC drops sharply. This is because a certain duty cycle limits the maximum packet load that can be handled by the nodes. On the other hand, the percentage of packets delivered in Etiquette protocol degrades only slightly at higher packet load and remains close to 100%. Notice that the performance for two different duty cycles is almost the same. This illustrates that duty cycle in Etiquette protocol does not constrain the maximum packet load that can be handled by a node. As mentioned earlier, this is because with a single request packet, a node can request multiple appointment packets. As long as the duration of office hours is long enough to accommodate a request/response packet from every neighbor that is interested in communicating with the node, the percentage of packet delivered is only limited by the channel bandwidth.

Figure 5 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. Notice that for both the protocols, the energy efficiency improves with the packet load. This is to be expected as the overhead of the protocol starts to get amortized over the number of packets delivered in the network. Note that 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. This is to be expected as in the region of low packet load, the effectiveness with which different protocols cut down the energy used in the network is tested. Notice the steep increase in the energy cost of delivering each bit in this region as the packet load decreases. The larger the duty cycle the steeper the increase. This is because there is a certain minimum overhead associated with each protocol, independent of the packet load in the network.

In the region of low packet load, Etiquette performs significantly better than S-MAC. The exact performance is dependent on the duty cycle of Etiquette protocol. For the same duty cycle, the average cost of delivering each bit in Etiquette protocol is almost half as much as that in S-MAC. This illustrates the effectiveness of the strategy for adjusting the duration of office hours based on the percentage idle time during its office hours in the recent past.

For smaller duty cycle, the energy cost of delivering each bit in Etiquette protocol is lower. This supports

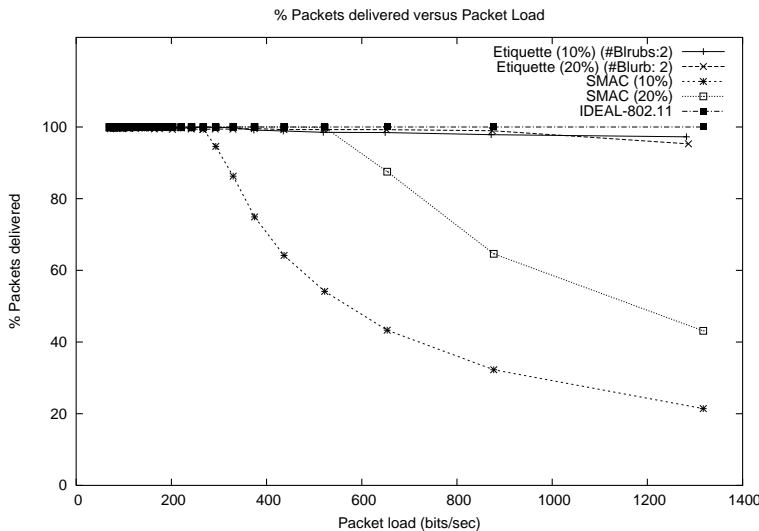


Figure 4: Simulations results comparing the percentage packets delivered for S-MAC, Etiquette, and IDEAL-802.11 as the packet load in the network is increased

our analysis in section 2.3.1, and is to be expected because nodes hold office hours less frequently.

At higher packet load, nodes need to have their radio ON for a significant fraction of the cycle and the specific protocol used would not matter. In fact the average cost of delivering each bit for “always-ON” protocols like the unmodified 802.11 will also converge to the same point as packet load increases. In the limit, the average energy consumed per bit for Etiquette protocol will be slightly worse than IDEAL-802.11 and S-MAC with 100% duty cycle (always-on mode), because of the extra overhead involved in holding office hours, and requesting appointments before actual communication.

For S-MAC, we do not extend the graph beyond the point when the packet delivery ratio has degraded below 90%. Notice that in this region, the average energy consumed per bit flattens off because the node’s duty cycle limits the amount of energy it can spend in delivering the packets. This also limits the number of packets delivered in the network. Notice that at around 200 bps, the performance of S-MAC with 10% duty cycle performs almost the same as with 20% duty cycle. This is because at this point, the packet delivery performance of S-MAC with 10% duty cycle has dropped sharply (figure 4) and the average energy cost per bit has flattened off.

Figure 5 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. In S-MAC, the latency for smaller duty cycle is slightly higher than that for larger duty cycle. This is because the expected queuing delay is half of the length of the interval in a cycle when the node is not in ACTIVE state. For smaller duty cycle, this interval is larger. 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 4). 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 duty cycle does not change with the increase in packet load. This is to be expected as Etiquette protocol is able to handle the range of packet load used in the simulation, without any significant loss in percentage of packets delivered (figure 4).

In summary, in Etiquette protocol, lowering the duty cycle (i.e., increasing the office hour period) reduces

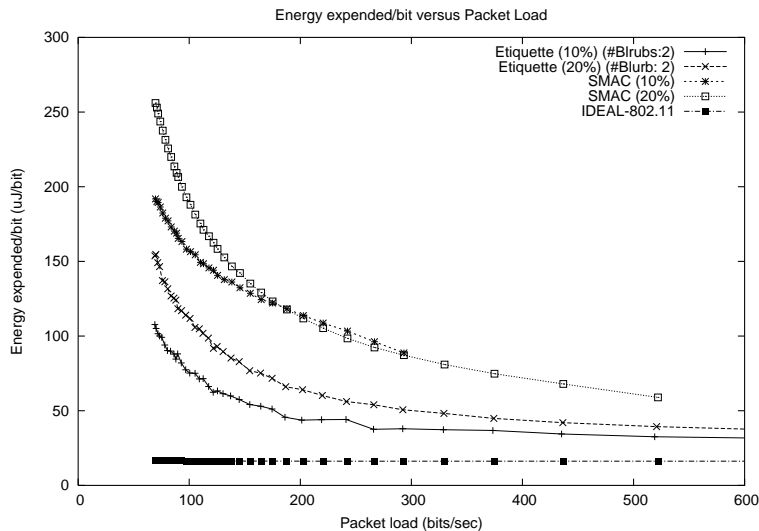


Figure 5: Simulations results comparing the average energy expended/bit for S-MAC, Etiquette, and IDEAL-802.11 as the packet load in the network is increased

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.

4 Related Work

Energy-efficient communication is a hotly pursued topic of research in sensor networks. The work on MAC layer for sensor networks is perhaps most closely related to this work. Recently a number of MAC layer protocols have been proposed for multi-hop sensor networks. In this section, we briefly review the protocols that are most relevant to our work. Interested readers are referred to [16] for an excellent survey of MAC protocols for network of energy-constrained nodes. In S-MAC [17], all nodes cycle through ACTIVE and SLEEP state. Node try to synchronize with each other so that they are all in ACTIVE state at the same time and in SLEEP state at the same time. While in ACTIVE state, nodes use CSMA/CA for communication. All nodes operate at the same pre-set duty cycle which dictates the amount of time spent in the two state. In S-MAC, since a certain duty cycle translates into a particular maximum throughput, network designer would have to set the duty cycle according to the worst-case estimate of the throughput required. This is 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. This is clearly not energy-efficient for sensors networks. In Etiquette protocol duty cycle does not affect the maximum throughput achievable.

T-MAC [15] protocol attempts to improve S-MAC by trying to identify intervals during the ACTIVE state of a node when the radio can be turned OFF. As in S-MAC, a certain duty cycle in T-MAC translates into a particular maximum throughput, and as a result, suffers from the same problem mentioned above.

In ReSync [4], all nodes periodically send out “intent” messages indicating the address of the destination nodes that they would like to communicate with and the time at which they would communicate. Even if a node does not have any message to transmit, it still needs to send an intent message. ReSync is fundamentally different from Etiquette protocol in that the onus of communication lies with the receiver. This requires all

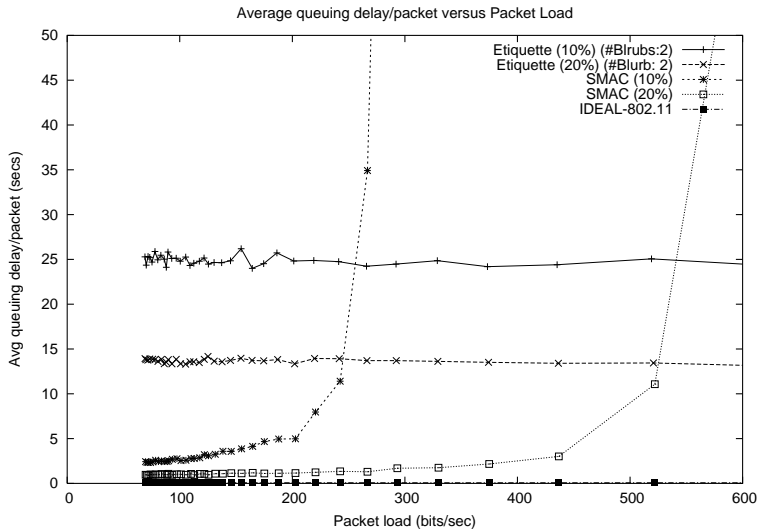


Figure 6: Simulations results comparing the average queuing delay in S-MAC, Etiquette, and IDEAL-802.11 as the packet load in the network is increased

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. In Etiquette protocol, the onus of communication is with the sender. If none of the neighbors is communicating with a node, it sleeps for a longer time interval.

IEEE 802.15.4 [3] standard defines the PHY and MAC specification for battery constrained nodes in low rate wireless personal area networks. Although the MAC layer specification allows both star and peer-to-peer topologies, the details for how nodes can address the idle-listening problem in peer-to-peer communication mode is not addressed. Distributed Mediation Device (MD) protocol [14] tries to fill this gap. In this protocol, a mediation device (MD) allows devices around it to communicate with each other. It functions similar to a telephone answering machine where a device can leave “intent-to-communicate” for other devices. When a device discovers (via the mediation device) that its neighbor intends to communicate with it, it synchronizes with that neighbor enabling communication. In a multi-hop network, each node takes “turn” for a small period of time to become MD. Nodes enter MD mode periodically with a period, P_m , independent of one another. As a result, it is possible to have multiple MDs in certain periods and no MDs in certain others. In the worst case, a node is guaranteed to have an MD in its neighborhood within P_m units of time. The paradigm of “leaving a message” forces nodes to regularly poll any MD device in the neighborhood for any “intent to leave” messages for it. This is wasteful for a node when no neighbors intends to communicate with it. Also, there is no support in the protocol for performing local broadcast, a fundamental operation in sensor networks. In Etiquette protocol, the receivers do not take any proactive action in communication; the onus of communicating with them lies with the sender.

Flexible power scheduling [10] 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 supports does not. We believe that the ability to support many-to-many communication is fundamental to many sensor network applications.

T-RAMA [12] is a collision-free protocol where a node starts with random schedule but adapts it based

on the schedules of its neighbors. T-RAMA requires a node to maintain consistent view of two-hop topology information, and also the schedules of all its neighbors. As a result, each node maintains state that is proportional to the density of nodes in the network. This affects the scalability of the protocol. Also, collecting two-hop topology information and keeping it consistent may be costly, especially in dense networks.

5 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. 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.

6 Acknowledgments

We thank Prof. Roy Yates for helpful discussions that improved the quality of this work. We thank K.V. Vihari for his insightful comments on the earlier drafts of this paper.

References

- [1] The network simulator - ns-2. <http://mash.cs.berkeley.edu/ns>.
- [2] Specification of the bluetooth system: Core, 2001. <http://www.bluetooth.org>.
- [3] *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.
- [4] 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.
- [5] Arikan E. Some complexity results about packet radio networks. *IEEE Transactions on Information Theory*, 30(4):681–685, July 1984.
- [6] Saurabh Ganeriwal, Ram Kumar, and Mani B. Srivastava. Timing-sync protocol for sensor networks. In *Proceedings of the first international conference on Embedded networked sensor systems*, pages 138–149. ACM Press, 2003.
- [7] J.C. Haartsen. The bluetooth radio system. *IEEE Personal Communications*, 7:28–36, February 2000.
- [8] 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.
- [9] 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://tinysos.millennium.berkeley.edu/papers/tos.pdf>.

- [10] 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.
- [11] 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.
- [12] 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 international conference on Embedded networked sensor systems*, pages 181–192. ACM Press, 2003.
- [13] Theodore S. Rappaport. *Wireless Communications: Principles and Practice*. Prentice Hall, 2001.
- [14] 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.
- [15] Tijds van Dam and Koen Langendoen. An adaptive energy-efficient mac protocol for wireless sensor networks. In *Proceedings of the first international conference on Embedded networked sensor systems*, pages 171–180. ACM Press, 2003.
- [16] Wei Ye and John Heidemann. Medium Access Control in Wireless Sensor Networks. Technical Report ISI-TR-580, USC Information Sciences Institute, October 2003.
- [17] 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.