Wireless Networks 8, 681–700, 2002

Improving Energy Efficiency of Centrally Controlled Wireless Data Networks

JOHN A. STINE * and GUSTAVO DE VECIANA
The University of Texas at Austin, Austin, TX 78712, USA

Abstract. Wireless network access protocols can assist nodes to conserve energy by identifying when they can enter low energy states. The goal is to put all nodes not involved in a transmission into the doze state. However, in doing so, one must tradeoff the energy and other costs associated with the overhead of coordinating dozing with the energy savings of putting nodes to sleep. In this paper, we define three alternative directory protocols that may be used by a central node to coordinate the transmission of data and the dozing of nodes. We attempt to optimize their performance by using scheduling and protocol parameter tuning. In addition, we consider the impact of errors and error recovery methods on energy consumption. Although one can argue that carefully scheduling transmissions will improve performance, ultimately, appropriately tuning protocols reduces scheduling's significance. In most cases, scheduling transmissions between the same nodes contiguously and ordering such transmissions shortest processing time first results in good performance. The most critical feature that contributes to an access protocol's effectiveness is its ability to minimize the time it takes to inform nodes that they may doze. However, the ability of our protocols to conserve energy is highly dependent on (1) network size, (2) traffic type (e.g., down/uplink, and peer-to-peer) and (3) channel bit error rate. In particular, we show that when protocols are faced with packet errors, more elaborate schemes to coordinate the dozing of nodes can pay-off. We conclude by recommending an energy conserving implementation of the IEEE 802.11 Point Coordination Function.

Keywords: energy conserving protocol, wireless medium access control (MAC), wireless network, 802.11 MAC, point coordination function (PCF), power saving

1. Introduction

It is now well accepted that medium access control (MAC) protocols can be designed to support energy conservation. Energy consumption in a transceiver is hardware dependent. Generally, three energy consumption states are defined: transmitting, receiving, and dozing. Table 1 presents the energy consumption rates in these three states for some commercial and experimental transceivers. There are three critical observations that affect protocol design. The first is that there is a relatively high energy consumption rate while in the receive state. It ranges from half to more than three quarters of the energy used in the transmit state. This high rate of energy consumption is associated with the processing of incoming signals. Second, the default state is the receive state despite whether a node is receiving a packet.[1] Sophisticated receivers must continuously attempt to synchronize to incoming signals in order to detect their presence. Third, a node is unaware of the activity of a network when it is in a low energy state. If it is in a low energy state it can neither detect when another node is trying to send it traffic or detect when it has the opportunity to transmit. Therefore, the goal of energy conserving MAC protocol design is to provide structure that allows nodes to predict when they will not be participating in packet exchanges. This goal is most challenging for MAC protocols that attempt to statistically multiplex the channel because of the randomness of packet arrivals.

* Corresponding author.
[1] This observation is empirically confirmed in [6,15,16].

In this paper, we consider the simplest environment for a MAC protocol to manage the use or low energy states, a centrally controlled network where a point coordinator (PC) is fully aware of all upcoming data exchanges. Wireless networks that rely on central control generally alternate between contention (CP) and contention free periods (CFP). The PC learns of the traffic that needs to be exchanged during the CP and then manage its exchange during the CFP. The most accepted protocol that operates in this manner is the Point Coordination Function (PCF) of the IEEE 802.11 MAC protocol [1]. At the start of the CFP, the PC has already selected a set of packets that will be transmitted. The only remaining issues are how to schedule and how to direct the packet transmissions. Our objective is to answer these questions from an energy perspective. We seek to determine how a central node managing packet transmission can reduce energy consumption.

This paper attempts to present a comprehensive study of this problem considering multiple contributing factors, namely, traffic control mechanisms and scheduling algorithms used by the PC, the primary direction of the traffic (to be explained in section 3.2), and the effects of packet errors. We consider three different types of protocols and in each we evaluate the full range of their configurable parameters to identify where the protocol performs best. Although our work is comprehensive, its greatest utility in the short term is that it provides insight in how to best use the energy conservation mechanisms that are part of the 802.11 PCF.

Table 1
Digital radio power states.

Radio	Transmit	Receive	Stand-by/doze
Lucent WaveLAN/IEEE Turbo 11 Mb Card [9]	285 mA	185 mA	9 mA
RoamAbout 915 MHz DS/ISA [8]	600 mA	300 mA	36 mA
RoamAbout 2.4 GHz DS/ISA [8]	365 mA	315 mA	30 mA
RoamAbout 2.4 GHz FH/ISA [8]	325 mA	185 mA	5 mA
Nokia C020/C021 Wireless LAN Card [10]	1.7 W	1.3 W	0.2/0.1 W
Aironet PC4800B In-Building Client Adapter [7]	350 mA	250 mA	<10 mA

This table uses the units that the manufacturer's data sheets use. Power is normally listed in watts but only the Nokia specification specified energy consumption in watts. Since the purpose of this table is not to compare transceivers but to illustrate the relative rates of energy consumption in different states we have kept the manufacturer's chosen units.

This paper is organized as follows. Section 2 reviews recent literature on energy conservation protocols. Section 3 provides background on the environment in which we expect our protocols to work. It provides definitions that are used throughout the paper and describes the physical properties of the network that constrain our designs. Sections 4, 5, and 6 develop the details of how our suggested protocol mechanisms work as they support different directions of traffic. We also integrate our discussion of scheduling into these sections. In section 7 we present the results of our simulations. We start by making protocol choices that favor throughput. We then use the results of these simulations to justify alternative strategies for each protocol mechanism that allow relaxing throughput in favor of least energy consumption. In section 8, we recommend how to apply our results to an implementation of the IEEE 802.11 Point Coordination Function. Finally, section 9 concludes the paper.

2. Literature review

There is a substantial amount of literature on the role protocols play in conserving energy. Protocols may use four sets of mechanisms to reduce energy consumption:

1. Help nodes enter low energy states.
2. Choose routes that consume the least energy.
3. Selectively use nodes based on their energy status.
4. Reduce overhead.

As a result, there is a gamut of topics this literature covers, but in close examination, very few papers address the role access protocols play in managing low energy states even though this promises to provide the best opportunities to conserve energy.

Both the the ETSI HIPERLAN [5] and IEEE 802.11 MAC [1] standards include mechanisms for managing low energy states. HIPERLAN's mechanism requires nodes desiring to doze to specifically coordinate a dozing cycle with another supporting node that agrees to act as a surrogate destination for the dozing node's traffic while it is dozing. Although the supporting node could be a PC, all scheduling decisions are done *a priori* to packet arrival and, so, HIPERLAN is not relevant to our research. Additionally, we are aware of no papers that study HIPERLAN's energy conservation mechanism. The 802.11 MAC provides two mechanisms, one designed for use in an ad hoc environment and one that is used together with the PCF. The PCF mechanism is applicable to our problem. The PC manages dozing by transmitting a traffic indication map (TIM) at the beginning of each CFP and periodically within. We describe this mechanism in detail later but in summary, the TIM provides sufficient information for all nodes in the network to learn if they will be exchanging traffic before the next TIM. If a node is listed in the TIM, it remains awake, otherwise it dozes until the next scheduled TIM. We considered this mechanism in our study of the applicability of the PCF for tactical military communications [18] and learned that the choice of TIM period can dramatically affect energy consumption. This observation motivated this work. Other authors have also explored the energy conservation characteristics of the 802.11 protocol but have focused on the ad hoc mechanism. 802.11's ad hoc mechanism supports periodic dozing where nodes may doze on their own initiative so long as they wake up for specified periods to determine if traffic to them is pending. The mechanism that informs nodes of pending traffic is based on contention and so its effectiveness is contingent on network load. In [3], the authors study the effect of load and the parameters of 802.11's ad hoc energy conservation mechanism. In [20], the authors compare the 802.11 protocol, along with others, with their Energy Conserving MAC (EC-MAC) protocol. Although the EC-MAC protocol supports a centrally controlled network, the comparison considered 802.11's ad hoc mechanism. 802.11's ad hoc mechanism does not support centralized traffic management. The results are especially misleading since the EC-MAC was designed to manage the transmission of small 53 byte ATM packets, which makes the signaling overhead of 802.11 very costly.

The two most notable works on the use of a central controller to manage the use of lower energy states are EC-MAC covered in [12–14,20] and protocols for identification networks covered in [4]. EC-MAC provides a list mechanism to schedule data transmissions between the base station and the surrounding mobile nodes. After the schedule is transmitted, the nodes in the network are awake only during the periods that they exchange traffic. This mechanism performs most efficiently when packets to or from the same destination are sent in contiguous slots because of the time it takes for nodes to transition between states. The paper discusses other scheduling issues associated with achieving quality of service but does not consider any scheduling approaches to further enhance the conservation of energy.[2] The work in [4] analyzes several protocols for centrally controlled access where the communications occur between a base station and a very large number of mobile nodes. In the first protocol, group tagged TDMA, groups of nodes awaken in a TDMA manner to listen for traffic from a base station. In the second, the directory protocol, the base station announces who will receive traffic in a directory and nodes listed in the directory remain awake. In the third, pseudorandom access, the base station and the mobiles know when each other are awake based on a random number generator. Of these, only the directory protocol uses the base station's knowledge of pending traffic. The deficiency of this work for our objective is that it only considers a single direction of traffic, downlink from the base station. Additionally, it does not explore the effect of scheduling the transmissions listed in the directory. Nevertheless, it provides an elegant mathematical model, which inspired the model developed in our appendix A.

A couple of other papers cover energy conserving access protocols for ad hoc networks. Power Aware Multiple Access (PAMAS) [11] promotes a method where nodes enter a low energy state when the channel is not available for them to use, i.e. another pair of nodes are exchanging data. Based on the critical observations listed in the introduction, this protocol has the paradoxical result that the network can only conserve energy if it is being used. Low use networks will consume more energy than high use networks since nodes only doze when they know other nodes are exchanging data. Since completing the research reported in this paper, we have also developed an ad hoc access protocol, Synchronous Collision Resolution,[3] that conserves energy [17,19]. This protocol was designed to apply the primary lesson of this research. The most effective energy conserving mechanisms minimize the time it takes for an awakening node to monitor the network to determine when it can return to the doze state.

3. Background

3.1. Centrally controlled data transmission

Centrally controlled data transmission uses a point coordinator (PC) to manage traffic transmission. The PC first learns what traffic needs to be transmitted in a contention period (CP) and then directs its transmission in a contention free period (CFP). In this paper, we focus on the CFP only. We assume that the PC has learned of k transmissions in the CP. The purpose of our investigation is to determine the control, scheduling and error handling schemes that offer the best energy conservation characteristics for the delivery of these k packets.

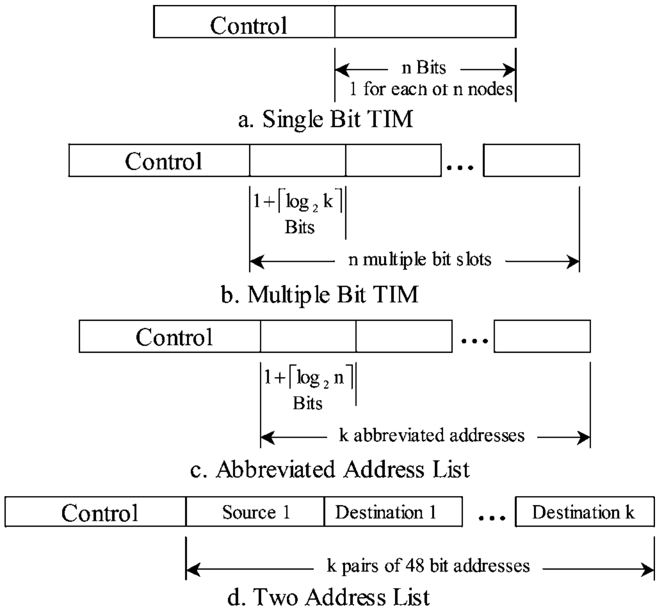


Figure 1. Directories for k packet exchanges in a network with n nodes.

In this study, we assume that the CFP may change in size to accommodate retransmission of failed packets. The transition from the CFP to the CP is directed once all pending packets are sent. As a result, the time it takes to transmit a specific quantity of traffic is inversely proportional to throughput.

The PC manages transmissions by broadcasting a directory. The purpose of these directories is two-fold, (1) manage who has access to the channel, and (2) help nodes not involved in the transmissions to doze. The content of the directory may vary. At the very least, it identifies all transmitters that will be active during the subsequent CFP. At most, the directory will list the source and destination of all transmissions, their size, and the order in which they will take place. With this information, nodes can make informed decisions on when they may doze. Figure 1 illustrates the directory structures that will be considered in this paper.

There are two broad classes of directories, traffic indication maps (TIM) and traffic lists. The TIM is a bitmap. The PC assigns every node a position in the bitmap when the node first associates itself with the network. Depending on the number of bits used in each position of the TIM, a TIM may indicate which nodes will participate in a data exchange, whether they will transmit or receive, or even how many exchanges they will participate in. The TIM does not normally indicate the order of transmissions or with whom the data will be exchanged.[4] This lack of information requires the PC to direct each transmission during the CFP using a poll. A poll is a packet that identifies the nodes that will participate in the next data exchange. Traffic lists, on the other hand, provide all information about the CFP. Each transmission has an entry

[2] Scheduling has no effect on the performance of protocols that use list directories when nodes only awaken for their exchanges.

[3] Patent pending.

[4] In this paper we provide an exception when we use an implied transmission schedule with the multiple bit TIM protocol. See section 5.2.

in the list. These entries specify the source and the destination of each transmission and its duration.

The tradeoffs between TIMs and traffic lists are the length of the directory itself, the overhead to control transmissions, the amount of information that is available to help nodes to doze, and the ability of the protocol to adapt to transmission errors. In this paper, we explore these options and attempt to use all methods to improve performance.

3.2. Traffic models

The effectiveness of an energy saving protocol depends on the predominant direction of traffic in the network. We consider four basic types of wireless traffic patterns and refer to them as Type I, II, III, and IV traffic. Type I traffic corresponds to nodes attempting peer-to-peer communication. Such networks are often referred to as ad hoc networks. Energy conserving MAC protocols supporting this traffic must synchronize each source's transmission with a period that the destination is awake. In Type II traffic, data is only transmitted to and from a base station. The base station may provide access to a larger wired network, it may be the central server for a group of mobile clients, or it may be the intermediary for intracellular traffic. There is no synchronization requirement between source and destination as the base station is always awake. In Type III traffic, the communications consist of downlink packet transmissions only as in a paging system. These are the simplest protocols to analyze as there is no requirement to manage the individual access of the mobile nodes, only the transmission of downlink messages and the scheduling of dozing periods. By contrast, Type IV traffic consists of a central node collecting data from mobile nodes such as in a telemetry application. In this paper, we separately develop and compare protocols for each of these traffic patterns.

3.3. Physical layer

Our protocols are designed for a *single* channel network on which traffic is half duplex. Destinations must respond to all successfully received packets with an acknowledgement so sources can confirm their reception.

Wireless networks have three timing considerations. First, time must be allowed for nodes to transition between transceiver states. Second, because of propagation and transition delays, silent periods necessarily occur between transmissions. Third, the network uses silent periods as an indication that the channel is idle. To simplify our analysis we use a single period, a slot time, to account for both the time it takes to transition among transceiver states and the time between transmissions. We use the slot time as the basic time unit in our analysis. We let S denote a slot time which for this work will be the time it takes to transmit 48 bits.[5]

[5] Typical transition times for 802.11 transceivers is on the order of 5 μs for transitioning into the doze state and upwards of 750 μs out of the doze state. Meanwhile, the 802.11 interframe spaces would be less than 30 μs. The 48 bit time unit is conservative for interframe spaces but optimistic for transitions out of the doze state.

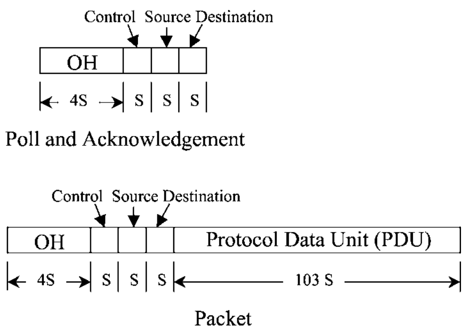


Figure 2. Poll, acknowledgement and packet structures.

In order to receive a transmission, a receiver must be synchronized with the transmitter. Therefore, the physical layer precedes each packet transmission with a series of bits to support synchronization and equalization. We assume that a receiver will not receive a transmission unless it is awake and receives these overhead bits first. In other words, a receiver cannot wake up in the middle of a transmission and then receive data. We label this overhead period as OH and define it as the time to transmit 192 bits,[6] 4 slot times.

3.4. Packet sizes and error rates

The packet sizes used in a network can significantly affect the rate energy is consumed because of the cost of overhead. Overhead is not normally affected by packet size. Since overhead is associated with each packet transmission, networks that use smaller packets will consume more energy for the same amount of data transferred. Networks that strive to support 53 byte ATM cells are likely to be less efficient that networks designed to support the 512 byte minimum size Ethernet packets. The packet size we used is based on the error rates we considered.

Our error model uses independent bit errors with a constant error rate. The objective is to correlate packet error probability with packet size. We chose a packet size of 103 slots, i.e. 618 bytes. This size is just below the threshold where a bit error rate of 10^{-5} makes it beneficial to split the packet and send it in two transmissions.[7] The second error rate used in our analysis, 10^{-4}, yields significantly more packet errors. The protocols are thus compared in low and in high *packet* error environments.

Figure 2 illustrates the structures of polls, packets, and acknowledgments. As can be seen the physical layer overhead bits precede each transmission. The content of the polls and acknowledgements is the same as the packet overhead, consisting of one slot for each address and one slot for any control information.

[6] The physical overhead of the IEEE 802.11 protocol takes 192 μs. Four slots is exactly 192 μs when the data rate is 1 Mbps.

[7] Large packets can be split in two and even with additional overhead be transmitted in less time, on average, than as a single packet due to reduced retransmission overheads.

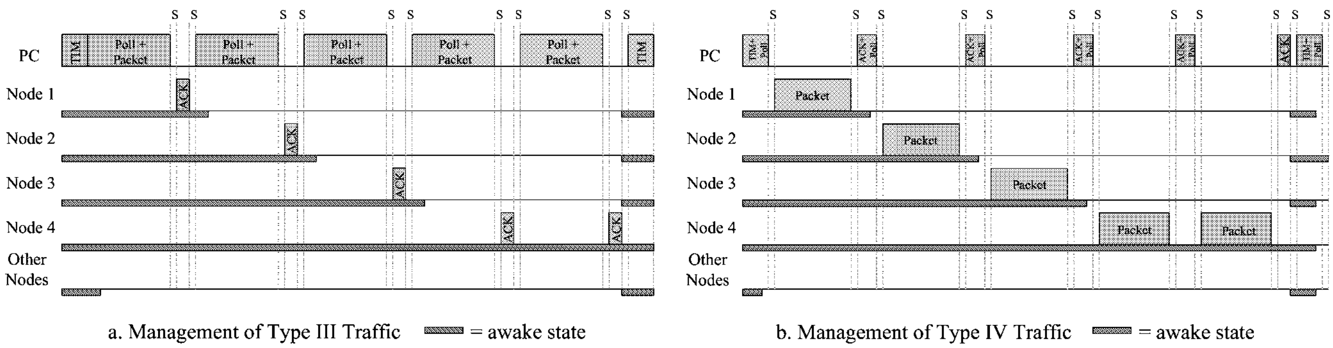


Figure 3. Type III and IV traffic management using single bit TIMs.

3.5. Error recovery

There are three alternative error recovery policies. In the first, which we call immediate retransmission, nodes attempt to retransmit a failed transmission immediately. Protocols provide a mechanism for the transmitting node to retain control of the medium after its transmission fails. In the second recovery method, which we call delayed retransmission, packet failures are rescheduled by the PC. This requires the PC to be omniscient of the status of each transmission. In the last recovery method, which we call recontention, nodes that are unsuccessful in sending their packets contend again starting in the next CP. Recontending contributes to congestion and wastes energy so it is not considered any further in our research.

Immediate and delayed retransmission support energy conservation since both allow retransmission without contention. Immediate retransmission has a throughput advantage over delayed retransmission since the protocol does not allow silent time. On the other hand, the delayed retransmission protocol has a timing advantage that makes it easier to keep idle nodes in the doze state. These issues are explored for each protocol.

3.6. Distribution of packet exchanges

The effectiveness of energy conserving protocols is dependent on which pairwise combinations of nodes exchange packets. The most optimum is if all packets are exchanged between just two nodes since then all but two nodes would doze during the CFP. Similarly, any tendency to cluster packets between common pairs of nodes will make energy conservation protocols more effective since fewer nodes would have to stay awake to participate in data exchanges. To preclude any skewing of results, we use a uniform distribution to select which pairs of nodes exchange data. Thus, every node is equally likely to participate in packet exchanges. This is the most challenging of scenarios for energy conserving protocols of the type we are studying.

3.7. Comparison measures

The objective of our analysis of different protocols is to compare their performance. We assume that at the beginning of a period of centrally controlled data transmissions that the protocols that are being compared have identified the same k packets to send. Guarantees of quality of service and access fairness are managed at a higher level. The measures of performance for our comparison are total service time and network energy consumption. Service time is the total time required to deliver the k packets. Network energy consumption is the total time all nodes in the network are awake until the k packets are delivered. We do not count the energy consumed by the PC. In properly functioning protocols only one node transmits at a time and the number and length of transmissions are the same, so we make no distinction between the time spent in the receive versus transmit state. A third measure of performance is the energy consumed by a node per packet exchanged during the CFP. In this case, nodes are categorized based on the number of packets they exchange during the CFP, so protocols can be evaluated in terms of where energy is wasted, e.g., in idle nodes, in nodes that send only one packet in the CFP, etc.

4. Protocol and scheduling options for Types III and IV traffic

4.1. Single bit TIMs

4.1.1. Description

Figure 3 illustrates the sequence of events that occur in the single bit TIM method of servicing Type III and IV traffic. At the beginning of the CFP, the PC broadcasts a TIM that indicates which nodes should remain awake. The PC then uses polls to direct the subsequent packet exchanges. Note that the PC piggybacks polls with transmitted packets and acknowledgements in an effort to reduce the physical layer overhead.

This protocol is very similar to the 802.11 PCF. The difference is that in 802.11 a packet source identifies when a packet is the source's last as part of the packet overhead whereas in our version this information is implied by whom the PC polls. This difference allows the PC to retain control of dozing for Type IV traffic. In 802.11, a mobile node participating in Type IV exchanges would remain awake as long as it has packets pending service even thought the PC has no in-

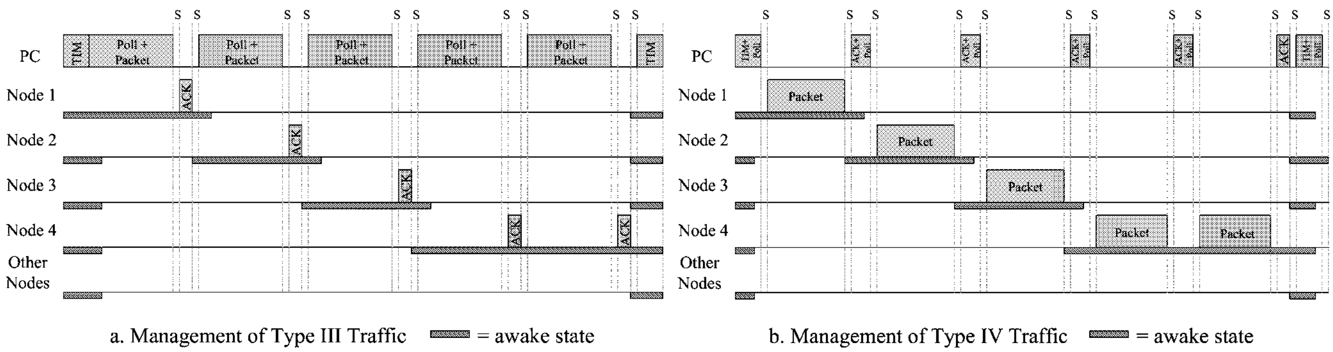


Figure 4. Type III and IV traffic management using multiple bit TIMs.

tent of polling it. In our implementation, nodes may enter the doze state after they have participated in a packet exchange and the PC polls another node. Thus, we require multiple exchanges between common pairs of nodes be sent in contiguous transmission slots. As is illustrated in figure 3, nodes remain awake from the transmission of the TIM until their last exchange. Scheduling conserves energy by minimizing the average time spent awake. This problem is identical to that of minimizing the average delay of jobs that need to be serviced by a common resource. It is known that the optimal schedule in this case is to serve the shortest job first [2]. In this context, job size corresponds to the number of packets to or from a given node.

At the cost of a small loss of throughput some energy efficiency may be gained by dividing the k transmissions into multiple TIM periods each announced with a separate TIM. This division allows a greater number of nodes to doze while waiting to exchange data. The tradeoff is that all nodes must awaken to receive the additional TIMs. We consider different sized TIM periods, i.e. packets exchanged per TIM, in our analysis.

Equations are derived in appendix A for the transmission time and for the expected energy consumption for a network using single bit TIMs with the scheduling methods described above in an environment with no errors.

4.1.2. Error recovery

The PC manages the recovery from a failed transmission. If the PC does not sense the transmission of a packet after a poll or the transmission of an ACK after a packet, it identifies an error condition. The PC then seizes control of the channel and directs the next data exchange. Since all nodes with pending traffic in the TIM period are awake, the PC can send the next poll immediately. If the delayed retransmission option is used the packet is not retransmitted until a subsequent TIM period otherwise the same node is polled. In both cases, however, data exchanges are not allowed to interfere with the transmission of the subsequent TIM. The penalty of having all nodes in the network awake waiting for the delayed transmission of a TIM is considered too great. Packets that fail to be exchanged during a TIM period are rescheduled in a subsequent TIM period.

4.2. Multiple bit TIMs

4.2.1. Description

Figure 4 illustrates the sequence of events that occur in the multiple bit TIM method of servicing Type III and IV traffic. As with the single bit TIM method, the PC broadcasts a TIM at the beginning of the TIM period and then directs the transmission of packets using polls. The distinction between the two methods is seen in how nodes doze. In the multiple bit TIM approach, nodes may also doze between the TIM and their exchanges. Multiple bit TIMs consist of multiple bits in each node's map position. These multiple bits indicate the number of packet exchanges that will occur with that node. Nodes can predict when they will exchange data if there is an implicit order in which nodes identified in the TIM may transmit data. We use the policy that all packets of the same node are exchanged contiguously and that these contiguous segments are scheduled first by quantity, fewest packets first, and then in bitmap order. This schedule is the most energy efficient approach when the immediate retransmission error recovery method is used. An explanation is provided in the next section.

Equations are derived in appendix A for the transmission time and for expected energy consumption of this protocol for a network with no errors.

4.2.2. Error recovery

The PC manages error recovery. The PC identifies errors if it does not sense the transmission of a packet after a poll or the transmission of an ACK after a packet. The PC may either reattempt transmission immediately or delay transmission until a subsequent TIM period. Figure 5 illustrates the tradeoff between the two recovery options after a failed poll. When the protocol uses immediate retransmission, all subsequent nodes scheduled to transmit or receive packets in the current TIM period will wake-up and have to wait for their exchange. We assume the control segment of polls include the packet number in the schedule and the target time for the transmission of the next TIM. Therefore, nodes that wake-up after a retransmission and that receive a poll can estimate when they would be rescheduled and can doze until that time. When the protocol delays retransmission until the next TIM period, the PC must transmit the next poll according to the transmission schedule so that it is transmitted when the intended recipients

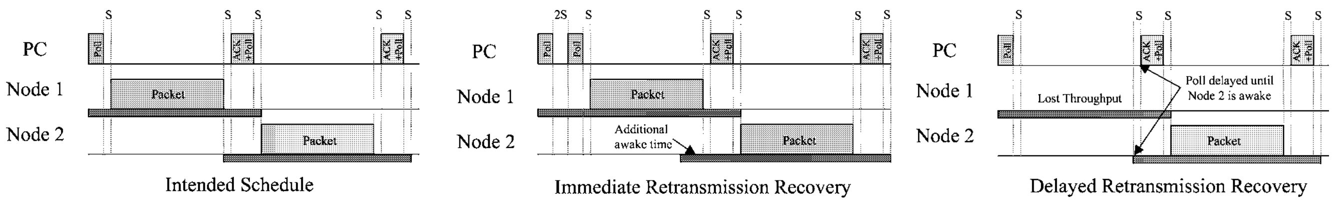


Figure 5. Multiple bit TIM protocol recovery from a failed poll for Type IV traffic.

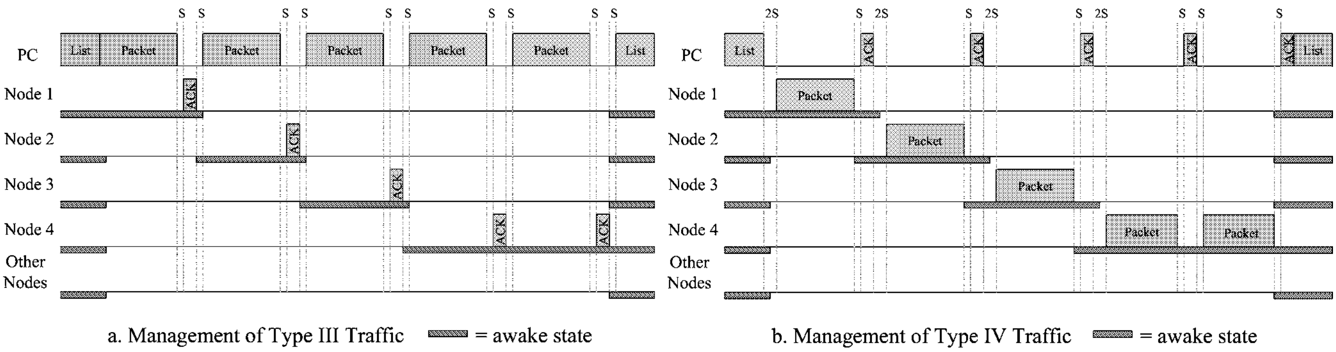


Figure 6. Type III and IV traffic management using single address lists.

are awake. If there is no response to a poll in Type IV traffic, the PC must still wait the duration of the planned packet transmission before it can poll the next node. This results in a loss of throughput.

Immediate retransmission penalizes nodes scheduled to transmit at the end of a TIM period. These nodes will wake-up and have to wait for the delays caused by the retransmissions that occur earlier. Scheduling the transmissions of contiguous packets fewest first minimizes the number of nodes that are awake.

4.3. Single address list

4.3.1. Description

Figure 6 illustrates the sequence of events that occur in the single address list protocol. It is very similar to that of the multiple bit TIM protocol. The differences are that the list is longer than the TIM and that the transmission of data is not preceded by a poll. Note that there is a difference in the awake time per packet between Type III and IV traffic. Since with Type III traffic the data transmission originates at the PC, there is a greater level of control resulting in less awake time. Indeed, mobile nodes need only wake up prior to the scheduled data transmission. With Type IV traffic, the mobile nodes are the sources. To be sure they are transmitting appropriately, they must wake-up early enough to monitor the preceding ACK to verify that it is their turn to transmit. We assume each packet transmission and ACK includes the packet number thus allowing subsequent nodes to determine the progress that has been made on the current transmission schedule. A node knows it can transmit when it monitors an ACK with the appropriate packet number.

Transmissions are scheduled in the same manner as done in the previous protocols, packets to nodes are sent contiguously in the order of fewest first. Sending packets contiguously reduces the number of transitions that nodes execute and in the case of Type IV traffic, mobile nodes only need to wake-up to listen to the ACK of another node's transmission once. Since immediate retransmission has the same effect as in the multiple bit TIM protocol of penalizing nodes scheduled to transmit late in the CFP, a fewest packets first schedule reduces the number of nodes that wake up early.

Equations are derived in appendix A for the transmission time and for expected energy consumption for the list protocol for a network with no errors.

4.3.2. Error recovery

This protocol can support both recovery options. With immediate retransmission each node reacts if the expected follow-on transmission does not occur. A source will retransmit a packet if it fails to sense an ACK. Similarly, a destination will retransmit an ACK if it fails to sense a follow-on packet. As with the multiple bit TIM protocol, nodes may wake up early being unaware of packet transmissions. Again we include the packet number in the control segments of all packet and ACK transmissions. Once nodes learn the packet number of the current exchange, they estimate a new wake-up time and return to the doze state. With delayed retransmission the PC is the critical player. With Type III traffic, the PC simply transmits packets at their scheduled times rescheduling those packets that are not acknowledged. With Type IV traffic, the PC transmits two types of ACKs, the standard ACK after it receives data and a negative ACK when it does not. Sending the ACK is critical since reception of the ACK is used by the mobile nodes as the signal when they may transmit. The PC reschedules packets that it does not receive.

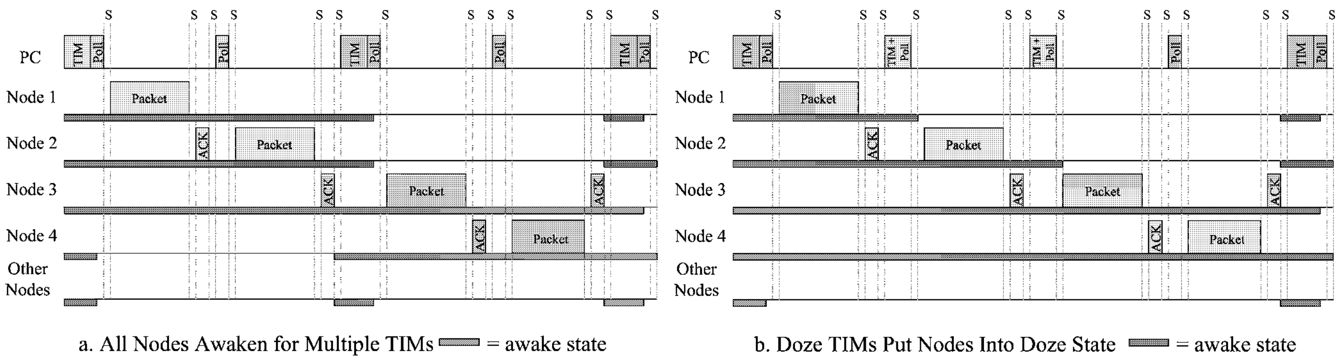


Figure 7. Alternative methods of single bit TIM control of Type I data transmission.

5. Type II traffic

The performance of protocols supporting Type II traffic is identical to that of Types III and IV traffic when the uplink and downlink are executed in separate cycles and can be extrapolated from these results when the traffic is integrated. Therefore, we do not provide any additional discussion of Type II traffic.

6. Protocol and scheduling options for Type I traffic

Two characteristics distinguish managing Type I traffic. The first is data exchanges occur between pairs of mobile nodes. Scheduling approaches must consider the fact that to exhaust one node's exchanges multiple other nodes must be awakened. A node can no longer assume that it is finished participating in data exchanges when the PC stops polling it. The second is that the PC no longer participates in the data exchange so it cannot be certain whether traffic was successfully exchanged between two nodes. Retransmissions must be initiated by the source when it identifies an error. If nodes are not provided a means to immediately correct a failed transmission, then the source will be required to contend again to send the traffic.

We also assume that all nodes in the network are within range of each other, which is a stronger requirement than all nodes being within range of the PC. If the PC can determine that nodes are not within range of each other, it can redirect transmissions perhaps relaying the traffic between the nodes. These corrective actions are not discussed in this paper but are essential, especially when lists are being used to manage traffic.

6.1. Single bit TIMs

6.1.1. Description

Figure 7 illustrates two traffic management methods for control of Type I traffic. Note that the exchanges are amongst the nodes and not with the PC. Since it is no longer possible to consolidate all communications of each node into contiguous slots, nodes can no longer assume that they can doze based on whom the PC polls. Nodes must be explicitly told that they may doze. So the single bit TIM method of servicing Type I traffic requires all nodes identified in the TIM to remain awake until a subsequent TIM puts them into the doze state.

Energy can be conserved in one of two manners. In the first, the CFP is divided into smaller TIM periods so that only a subset of the nodes that are participating in the CFP are awake each TIM period. In the second, in addition to using multiple TIM periods, additional TIMs which we will refer to as "doze TIMs" are used within the TIM period for the express purpose of putting nodes into the doze state.

Schedules can be based on two intuitive observations. First, the optimum schedule, when nodes remain awake throughout TIM periods, is a schedule that minimizes the average number of nodes that are awake each TIM period. Second, the optimum schedule when doze TIMs are used is one that allows the doze TIMs to put the most nodes to sleep soonest. Note that small TIM periods and errors reduce the significance of achieving these goals.

Minimizing the average number of nodes awake during each TIM period is a function of the number of packets scheduled per CFP and the size of each TIM period. In most cases, an exhaustive solution is far too complex. There are $k!/(j!\lceil k/j \rceil !)$ different combinations of packet exchanges when k packets are transmitted in j TIM periods. We attempted two heuristic algorithms to explore the significance of scheduling. The first, algorithm A, selects active nodes for the TIM period in a greedy fashion. It starts by selecting the two nodes with the most exchanges and then adds nodes to the active set based on how many exchanges their addition provides giving preference to the node that adds the most unless the TIM period can be filled with fewer. The second, algorithm B, seeks the minimum number of nodes that have just enough exchanges to fill the TIM period. It does this search by trying all combinations of m of the n nodes participating in the CFP incrementing m until enough exchanges can be found to fill the TIM period. Algorithm B is too complex for practical implementation but is still interesting for our purpose of evaluating the energy conservation potential of scheduling. Appendix B provides details of algorithms A and B.

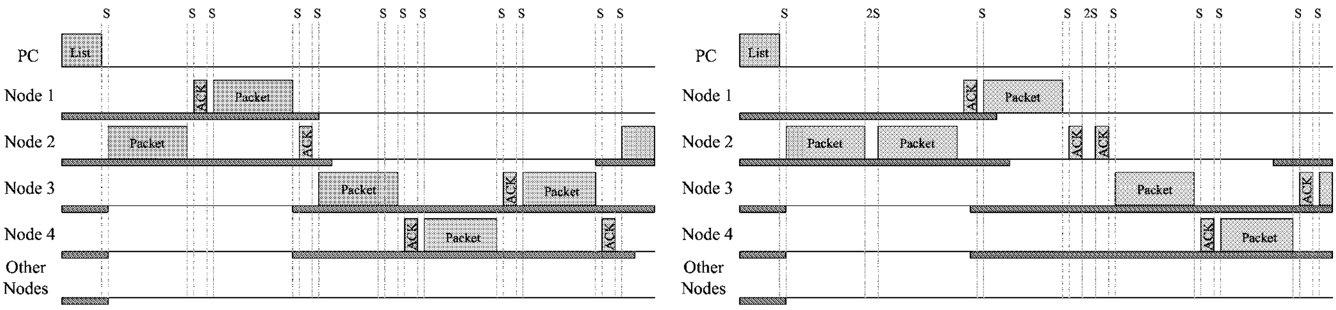
NEXT NODE TO SLEEP SOONEST

SOURCE	DESTINATION	NODES AWAKE
1	2	5
2	1	5
1	5	4
1	5	4
1	5	4
3	4	2
3	4	2
3	4	2
	TOTAL AWAKE:	28

MOST NODES TO SLEEP SOONEST

SOURCE	DESTINATION	NODES AWAKE
3	4	5
3	4	5
3	4	5
1	2	3
2	1	3
1	5	2
1	5	2
1	5	2
	TOTAL AWAKE:	27

Figure 8. Comparison of next node to sleep soonest versus most nodes to sleep soonest scheduling.



a. Traffic Management Without Errors = awake state

b. Traffic Management With Errors = awake state

Figure 9. Type I traffic management using a two address list.

Since algorithms that create schedules to put the most nodes to sleep soonest are very complex, we first considered an alternative scheduling approach that puts the next node to sleep soonest. Figure 8 illustrates the difference. In this scenario, 8 exchanges need to be made amongst 5 nodes. Each exchange has a source node and a destination node. The 5 nodes cannot doze until all their exchanges for the TIM period have been completed. In the next node to sleep soonest schedule, the exchanges of node 2 are scheduled first since node 2 can enter the doze state soonest. In the most nodes to sleep soonest schedule, the exchanges between nodes 3 and 4 are scheduled first since both nodes can enter the doze state at the conclusion and since the final tally of node-awake times is reduced from 28 to 27. Although not optimum, putting the next node to sleep the soonest is a very simple algorithm. Our algorithm C (see appendix B) seeks this schedule. It looks at nodes individually and schedules the transmissions of the node with the fewest exchanges first.

In an attempt to come closer to the most nodes to sleep soonest schedule, we attempted to combine algorithms A and B with C. The objective was to allow algorithm A or B to select a reduced number of nodes to participate in a TIM period and then to use algorithm C to schedule these exchanges. Reducing the number of nodes in a TIM period increases the likelihood that a next node to sleep soonest sort approaches a most nodes to sleep soonest schedule for that TIM period. We call these two hybrid algorithms E and F.

6.1.2. Error recovery

Portions of error recovery may be managed either by the PC or by each node transmitting a packet. The PC is the only entity that can manage recovery when a poll is not received. When a packet is not received either the PC or the node transmitting the packet may manage the recovery. Both learn of the failure when an ACK is not received. If the PC manages the error recovery it does so by polling the source a second time. If the transmitting node manages the error recovery then it attempts to retransmit the packet immediately. The advantage of letting the transmitting node manage retransmission is that the transmission of the poll is avoided. The disadvantage of node managed recovery is that it does not support delayed retransmission. We use PC managed recovery in our simulations.

6.2. Two address list

6.2.1. Description

The directory using a two address list consists of two addresses for each packet transmission. Abbreviated addresses are not used on account of the overhead and complexity of disseminating those addresses to all nodes. The addresses used are standard 48 bit MAC addresses that we assume are used for wireless nodes. Figure 9(a) illustrates the transmission of data using these lists. Pairs of nodes awaken for each transmission. To simplify identifying when each node has its turn to transmit and to minimize the time nodes spend transitioning, all exchanges between common pairs of nodes

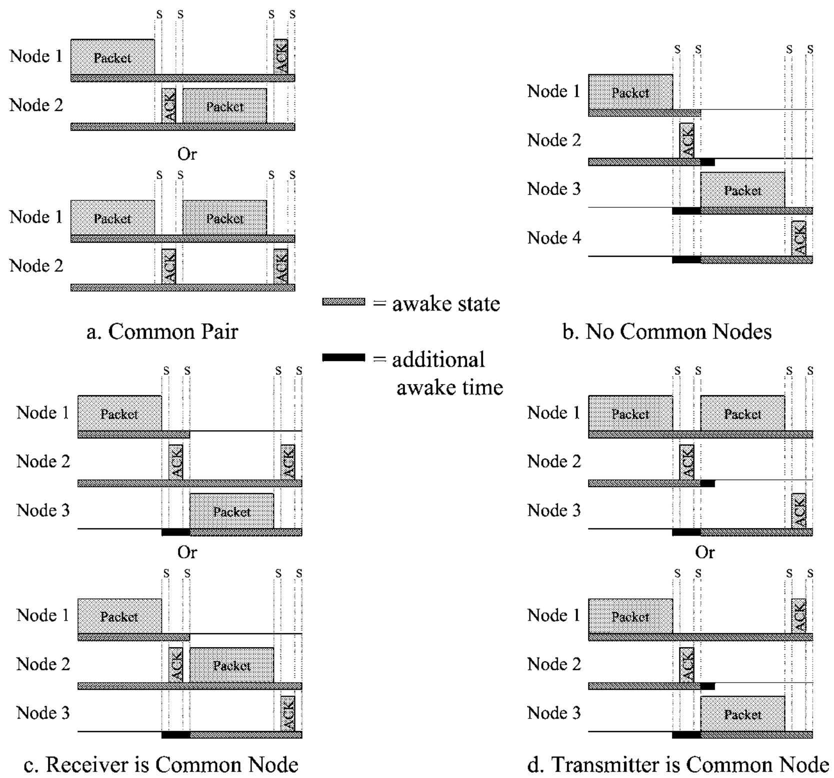


Figure 10. Energy consumption at pair transitions for Type I traffic using the list protocol.

are executed in contiguous slots. Each packet transmission and ACK includes the packet exchange number for the CFP. A pair of nodes knows it is their turn to transmit when their packet exchange number immediately follows the number announced in the last ACK. For this reason, all nodes except the first to transmit in the CFP will monitor an ACK before they transmit a packet.

Scheduling can improve energy performance by reducing the energy consumed at the transitions between packet transmissions. Figure 10 illustrates the possible transitions emphasizing the different amounts of energy consumed. As seen in this picture the best energy consumption occurs when the same nodes participate in the two packet exchanges followed by the transitions where at least one node participates in both exchanges. Algorithm D is a heuristic approach that attempts to create a schedule that optimizes according to this observation. Appendix B provides the details of algorithm D. A second scheduling objective is to minimize the effects of failures, i.e. minimize the number of nodes that wake up prematurely. This is identical to the objective of putting the most nodes to sleep soonest. So again, we use algorithm C to generate an energy-conserving schedule. Algorithm C also achieves some of the preferred transitions since it groups transmissions between common nodes together.

6.2.2. Error recovery

On account of the requirement for a node to monitor an ACK before transmitting, only immediate retransmission error recovery is attempted. If a destination does not receive a packet correctly, it will not send an ACK. If the source does not react by retransmitting the packet then the chain of transmissions identified in the list will be interrupted. It may be possible for the PC to detect the error and for either the PC or the transmitting node to send a pseudo ACK to prompt progression in the transmission list but the timing would still be compromised. Similarly, if the next node to transmit does not respond to an ACK then the sending node must continue to resend the ACK until it does. This activity also compromises timing. Therefore, we chose to only support immediate recovery. Figure 9(b) illustrates the transmission of data when errors occur. If the timing gets too bad and a node wakes up early and monitors either an ACK or the beginning of a packet transmission, the packet exchange number in these transmissions allows the node to return to the doze state since it can use this number to estimate when its next exchange is due.

7. Model and simulation results

We developed a simulation that modeled the protocols and scheduling algorithms described above. The study methodology considered the performance of the protocols and algorithms for different size networks, different quantities of exchanges, different error conditions, and different TIM periods (when TIM protocols were used). The scenarios used in the simulations were the same so the differences in the results reflect the effects of the protocols, scheduling algorithms, and error recovery methods as opposed to random events such as the occurrence of errors or clustering of exchanges between

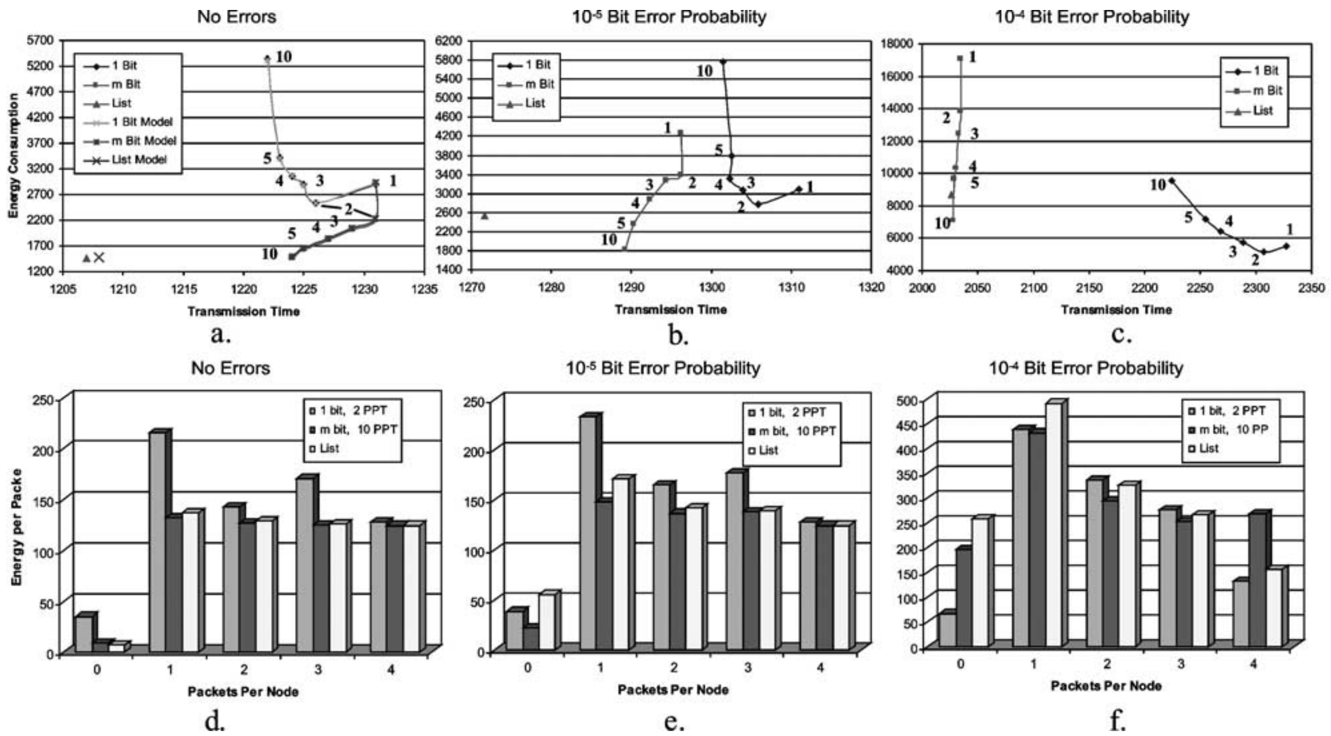


Figure 11. Performance of Type III traffic, 25 nodes, 10 packets.

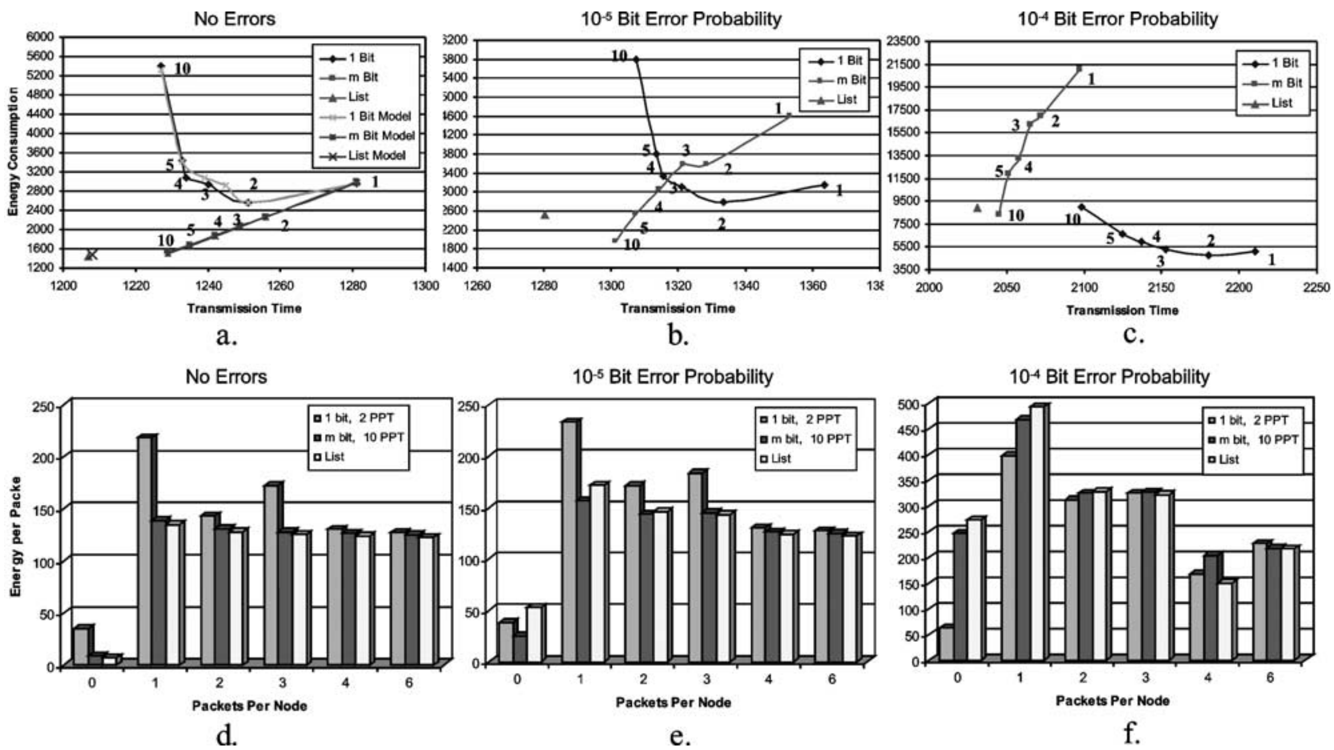


Figure 12. Performance of Type IV traffic, 25 nodes, 10 packets.

common pairs of nodes. Our first simulations used the immediate retransmission error recovery option. Total service time and total energy consumption (i.e. total time nodes are awake) was obtained for each simulation. Statistics of average energy consumption per node as a function of packets exchanged were obtained for each protocol with the best performing set of parameters (i.e. best TIM period measured in packets per TIM period (PPT)). Samples of the results are shown in figures 11, 12, and 13. Panels (a), (b), and (c) exhibit total energy consumption versus total transmission time,

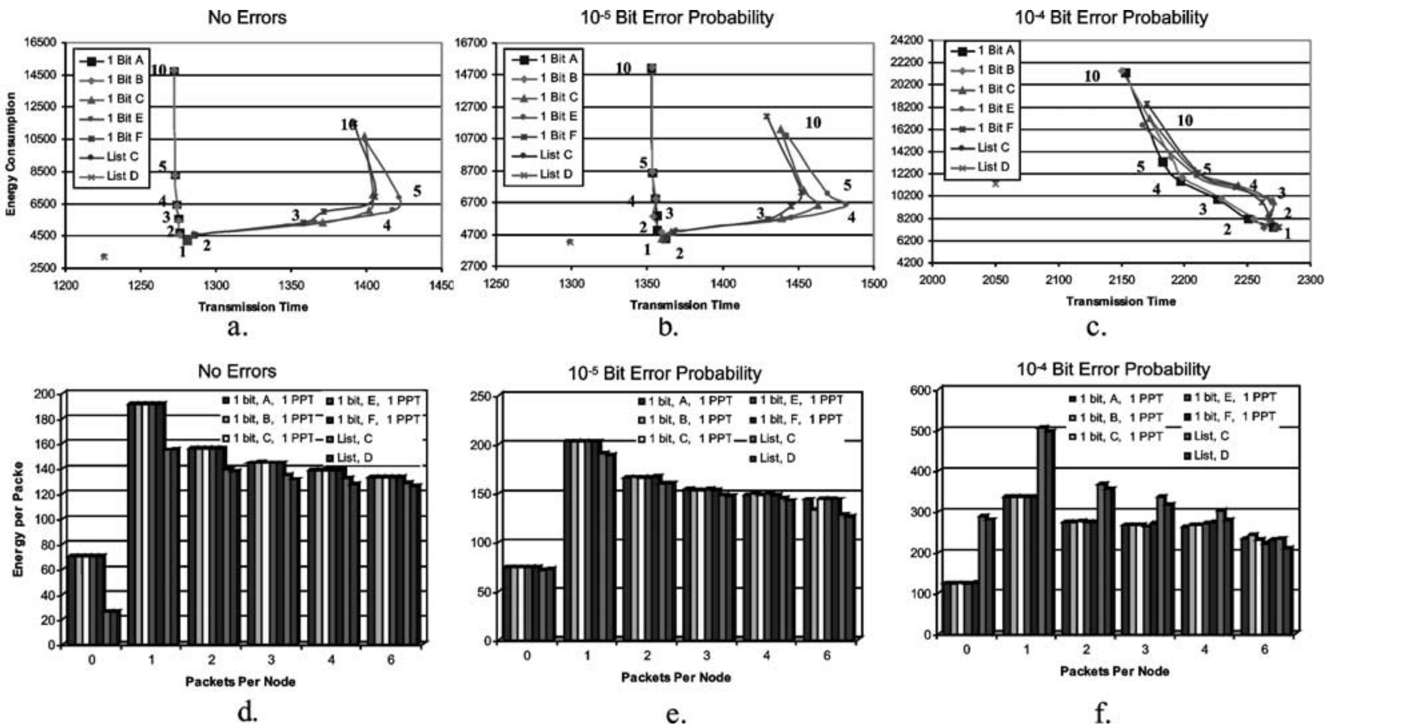


Figure 13. Performance of Type I traffic, 25 nodes, 10 packets.

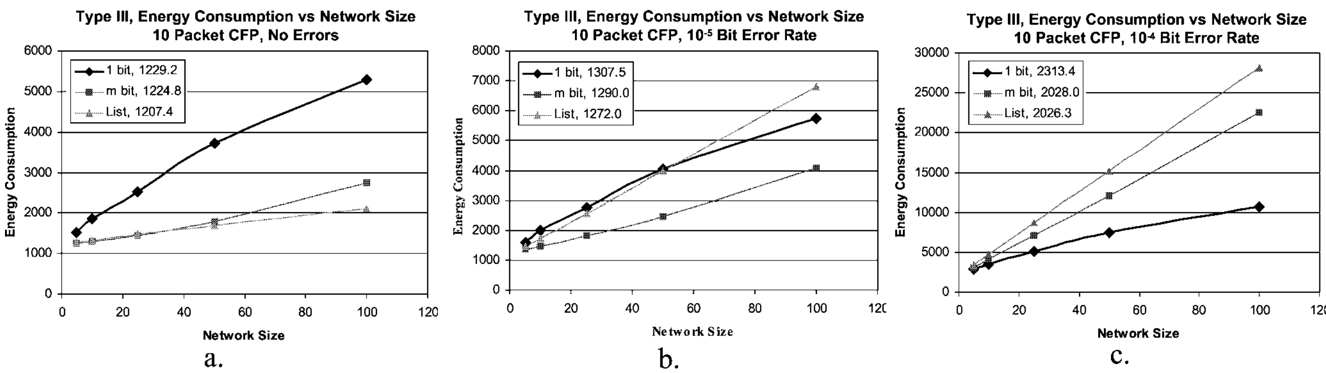


Figure 14. Protocol performance at optimum for Type III traffic with 10 packet CFPs. The legend provides the average transmission time for the protocols. Our results demonstrated that the network size had only a minor effect on the time it takes to transmit *k* packets despite the fact it affects the size of TIMs and lists.

each measured in slot times. The different points plotted for the TIM protocols correspond to different TIM periods and the numbers correspond to the number of packets per TIM. Panels (d), (e), and (f) show the energy consumed per packet transmitted parameterized by the total number of packets sent by each node in the CFP. Note the average energy consumed by nodes that sent no packets is identified by 0, the average energy per packet consumed by nodes sending just one packet is identified by 1, the average energy per packet consumed by nodes sending 2 packets each is identified by the number 2, etc.

In the case of Types III and IV data we validated our simulation model by comparing the simulation results to the model predictions include in appendix A. As illustrated in figures 11(a) and 12(a), the results match.

We simulated five different sized networks, 5, 10, 25, 50, and 100 nodes, under three different bit error rates, no errors, 10^{-5}, and 10^{-4} errors per bit, and for three traffic types, I, III, and IV. We ran 200 simulations for each set of conditions. Figures 11, 12, and 13 are samples of our data. Panels (a), (b), and (c) illustrate the performance of the protocols for different error conditions and for different TIM periods for a network size of 25 nodes and a CFP of 10 packets. Panels (d), (e), and (f) illustrate the energy consumed per node based on the number of packets the node sent in the cycle using the optimum TIM periods (i.e. least energy consuming) for the same networks. Figure 14 illustrates how the relative performance of the protocols changes with network size. Of interest is that both the error condition and the type of traffic affect which protocol performs best.

7.1. Observations and design implications

The results of the simulations bring out the following three key design concepts.

1. *Synchronize the waking up of dozing nodes with the broadcast of the status of the transmission schedule.*

Networks with errors consume large amounts of energy because of nodes waking early and having to wait for directory information before they can return to the doze state. Synchronizing the waking up of nodes to the transmission of the status of the transmission schedule minimizes this energy loss. The relative improvement in the performance of the 1 bit TIM protocol is attributed to this factor. Observe panels (d), (e), and (f) of figures 11, 12, and 13. Note that the energy consumed per node for those nodes not transmitting or transmitting just one packet in a cycle increases rapidly with errors for the m bit TIM and list protocols but remains nearly constant for the 1 bit TIM protocol. The most significant difference between the 1 bit protocols and the other protocols is that the 1 bit protocol provides this synchronization. The 1 bit protocol requires packet transmissions to be rescheduled into a subsequent TIM period if they would interfere with the broadcast of a TIM. As a result, dozing nodes are assured of receiving a TIM when they wake-up. In the m bit and list protocols, errors cause nodes to wake prior to expected exchanges and prior to the conclusion of the CFP. Although these two protocols provide a means for nodes waking early to re-estimate the time of the events they should be awake for thus allowing them to return to the doze state, they may have to wait a long time to get it, e.g., the length of a packet transmission. This penalty adds up especially as more nodes wake up early. It is for this reason that we see in figure 14(c) that the 1 bit protocol's relative performance increases with the size of the network.

2. *The optimum period for TIM protocols is independent of error conditions.*

We found that the optimum TIM period for 1 bit protocols is most affected by network size, i.e. the number of nodes in the network. The trend is that as the network gets larger a longer TIM period performs better. When networks are larger so too are the TIMs. Additionally, more nodes must listen to the TIMs. The penalty of more nodes listening to larger TIMs exceeds the penalty of a few nodes having to wait in the awake state for their turn to participate in a data exchange. Note that the TIM period is slightly lower for Type I traffic (see figure 13). Since in Type I traffic two nodes participate in each data exchange, more nodes on average remain awake in a TIM period. Fortunately, since the optimum TIM period is not affected by error rate, it can be selected using the models found in appendix A. Note, however, that the assumption in these models is that the traffic is uniformly distributed. Any clustering of traffic between a few nodes will tend to make larger TIM periods more attractive.

The optimum TIM period for the m bit TIM protocol is the transmission cycle size. There is no benefit to using multiple TIM periods per cycle. Sufficient information is provided in the polls to allow the nodes to correct their dozing periods.

3. *Scheduling algorithm C provides excellent performance for all types of traffic and for all types of protocols.*

Algorithm C is recommended for three reasons. It is the easiest algorithm to implement, it is the least complex of the algorithms listed in this paper, and it provides either optimum or near optimum schedules. When used with Type II, III, or IV traffic it is equivalent to the shortest processing time first scheduling policy and is therefore optimum. As demonstrated in the Type I traffic simulations all of the scheduling algorithms perform about the same for the 1-bit TIM protocols when the optimum TIM period is selected. The performance of the alternative scheduling algorithms used with the list protocols supporting Type I traffic were also nearly identical for all network sizes, CFP sizes, and error rates.

7.2. Improvements

The total energy consumption of each of the protocols described in this paper can be improved by focusing on two characteristics, synchronization and better dozing periods. The protocols that can benefit most from improvements in synchronization are the list and m bit TIM protocols. The 1 bit TIM protocol benefits from better dozing periods.

For both the list and m bit protocols we considered two possible synchronization methods. In the first, the base station monitors and announces the perceived error rate. Nodes that are not participating in data exchanges then use this error rate to estimate when the conclusion of the CFP will be and thus avoid waking up early. The risk with this procedure is that the base station cannot be assured that all nodes in the network are awake at the conclusion of the packet exchanges and must wait the worst case estimated time before directing the start of the CP. The second synchronization method attempted was to delay retransmission of failed packets until after a subsequent list or TIM. Nodes that wake up to receive a directory are assured to receive one. Also, nodes that wake up to participate in an exchange are assured of waking up on time.

The above recommendations were tested on both the multiple bit TIM and list protocols for Type III and IV traffic. In the case of the estimation versions the actual error rates (i.e. the ideal case) were used to estimate wake up times. Figure 15 illustrates the results for list and m bit protocols for Type III traffic and 10 packet CFPs. The results for Type IV traffic were similar. The estimation versions of the protocols are labeled with the letter E and the delayed retransmission versions are labeled with the letter D. The average service time is shown in the legend. As expected, the advantage of these protocol variations increases with both the error rate and the network size. The whole purpose of these variations is to react to errors and the benefit increases as there are more nodes that can benefit. The performance of the estimation versions also improves as the number of packets exchanged in the CFP increases. This is not the case for protocols using delayed transmission. The statistical nature of

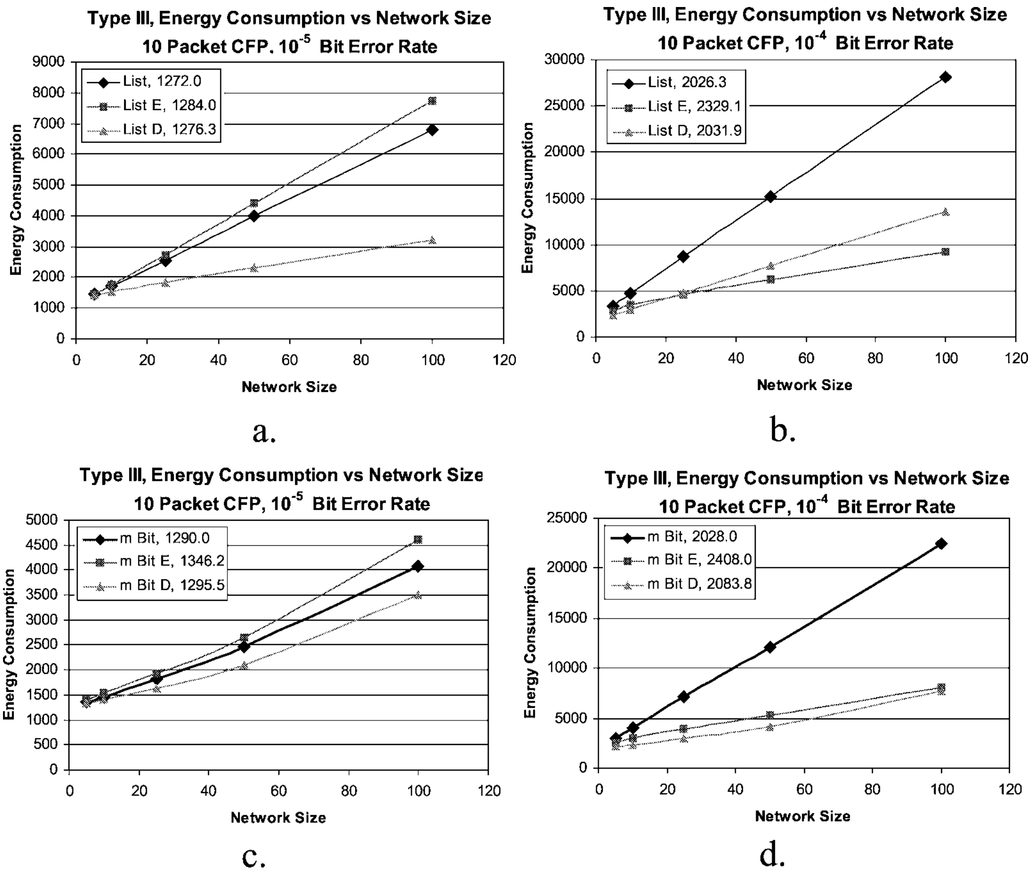


Figure 15. Comparison of protocols modified for improved performance with Type III traffic.

the estimation approach allows it to improve with larger numbers. The delayed retransmission, however, is likely to have more transmissions of lists or TIMs when there are larger CFPs resulting in more energy consumption by all nodes in the network. The delayed retransmission protocols cause an insignificant increase in transmission time, e.g., less than 1%. The estimation versions, however, result in longer transmission times. They were as much as 20% longer in our simulations.

Three improvement techniques were simultaneously attempted with the 1 bit TIM protocol: layered TIMs, local optimization of the TIM period, and delayed retransmission. Together, these techniques decrease the energy consumption of the 1 bit protocol despite the error conditions or the type of traffic.

The objective of the layered TIMs is to reduce the energy that is consumed by idle nodes listening to TIMs. The transmission cycle is layered into sequentially smaller TIM periods. A larger TIM period in the outer layer is subsequently divided into smaller TIM periods in the inner layer. The TIM for the outer layer reduces the set of nodes that listen to the TIMs of the inner layer. Therefore, at the beginning of the outer TIM period, two TIMs are transmitted. The first TIM specifies a large TIM period and puts nodes to sleep that will not be participating in any data exchanges for that larger period. The second TIM then manages the reduced set of awake nodes.

Local optimization of the inner TIM periods is motivated by our observation that the TIM periods are affected by the distribution of traffic and not by error rates. Each inner TIM period is selected based on the next transmissions scheduled. It is at least as long as the number of transmissions of the next pair of nodes. A longer TIM period is selected if the penalty of having the active nodes of the inner TIM period listen to an additional TIM is more than the penalty of having the next pair of scheduled nodes listen to the preceding packet transmissions.

Finally, delaying retransmissions eliminates one of the penalties of immediate retransmission. Immediate retransmission error recovery results in every node waiting to transmit in a TIM period staying awake for each retransmission. Delayed retransmission only penalizes the nodes involved in failures. Retransmissions are delayed until the next inner TIM period.

The techniques above were attempted for all types of traffic. Figures 16 and 17 compare the performance of the standard 1 bit TIM protocol to that of the optimized 1 bit TIM protocol for Type I traffic. The optimized 1 bit TIM protocol always consumes the least energy. Figure 16 shows that the relative size of the improvement increases with network size but decreases as the CFP increases. These results illustrate that the most energy is conserved by the first improvement technique, i.e. layered TIMs. The objective of the outer TIM is to reduce the energy consumed by idle nodes.

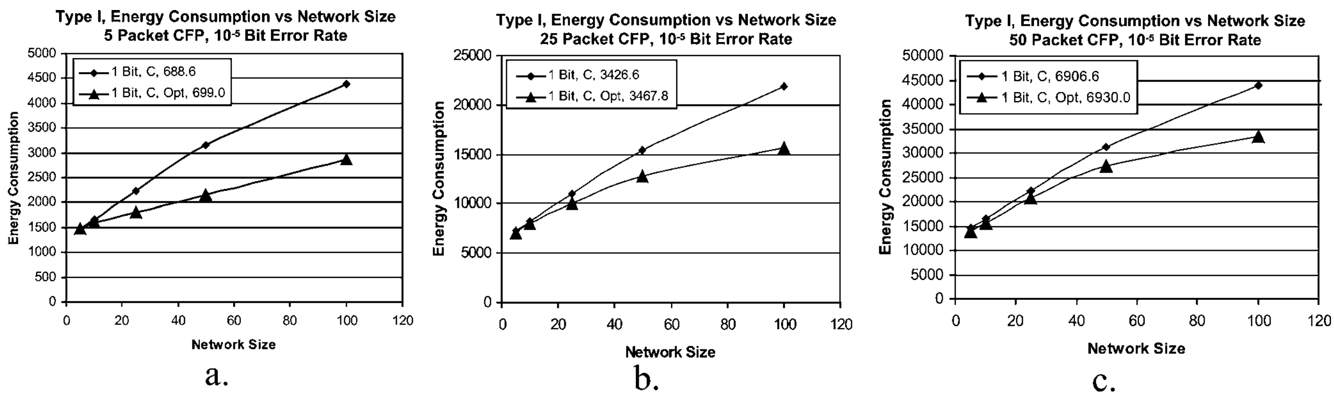


Figure 16. Comparison of the optimized and standard 1 bit TIM protocols with Type I traffic.

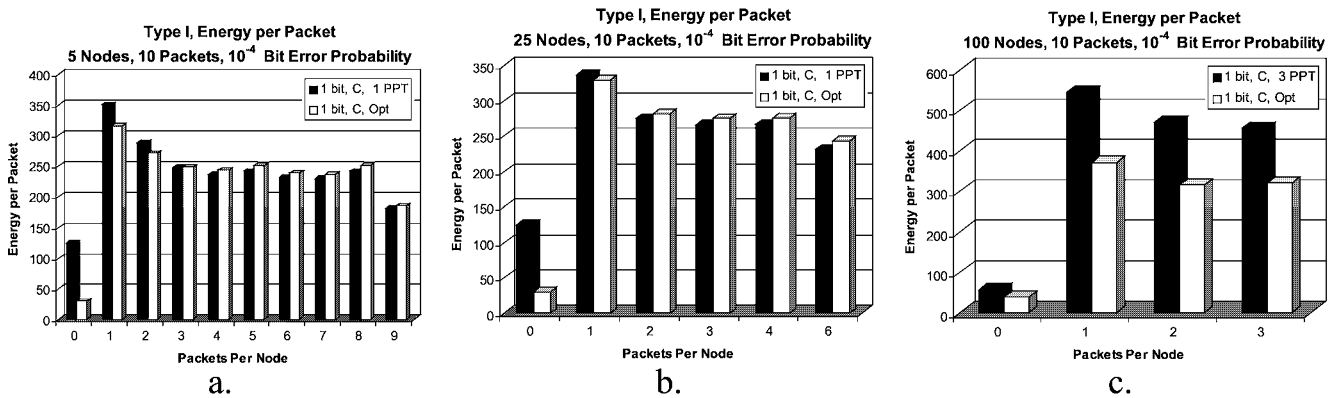


Figure 17. Comparison of the optimized and standard 1 bit TIM protocols with Type I traffic.

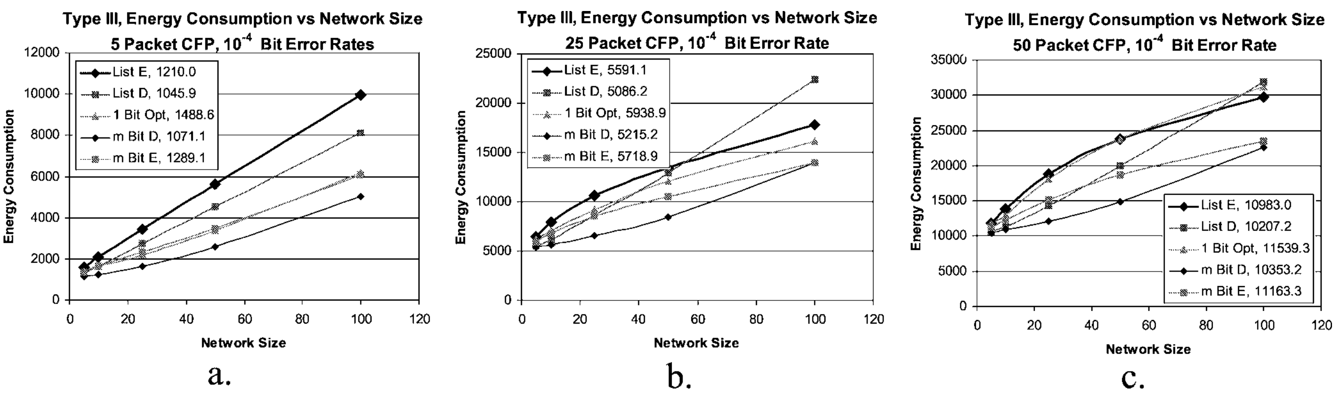


Figure 18. Comparison of improved protocols for Type III traffic.

The more idle nodes there are the more effective it is. The number of idle nodes increases as the network size increase, and decreases as the CFP increases. Figure 17 not only illustrates that the idle nodes benefit the most from the optimized protocol but that the benefits for nodes that transmit traffic increases with network size. The dependence on the network size is caused by the change in the optimum TIM period of the standard 1 bit TIM protocol. The larger TIMs increase the optimum TIM period for the standard 1 bit TIM protocol reducing the penalty to the idle nodes but increasing it for nodes that transmit traffic. The optimized 1 bit TIM protocol has consistent performance for all network sizes.

7.3. Choosing the best traffic management protocol

Our results show that protocol performance needs to be compared in the high error environments as some protocols that perform very well when there are no errors quickly degrade when efforts are made to resend failed transmissions. In figure 18 we compare the energy conservation of all the improvement techniques described in section 7.2 for Type III

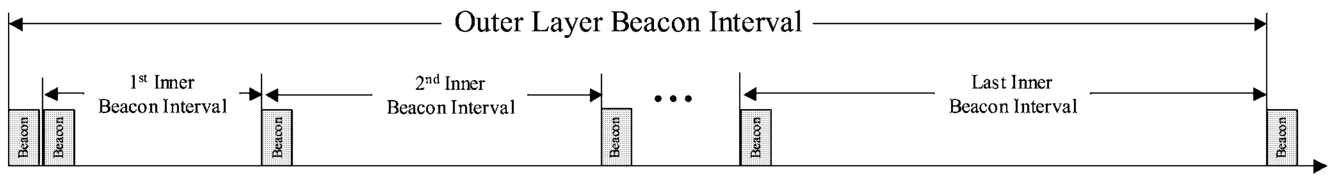


Figure 19. Use of aperiodic beacons in implementing the 1 bit TIM protocol in 802.11.

traffic in a high error environment. These graphs demonstrate that the performances of all the protocols are fairly close to each other with the relative difference depending on network size. The consistent result for both Type III and IV traffic, all network sizes, and all transmission cycle sizes is that the delayed retransmission version of the m bit TIM protocol is the best at conserving energy. Moreover, the delayed retransmission version of the m bit protocol was very competitive in throughput with only the delayed transmission version of the list protocol performing better.

The limitations of the m bit protocols are that they are only suitable for Type II, III, and IV traffic and that they require the use of fixed sized packets. The list protocols can be adapted for use with variable sized packets by providing additional fields in the directories for each packet length. The 1 bit protocols, however, require no modification, but the fixed TIM periods may result in more complexity in scheduling and a loss of capacity as packet sizes may not support filling the TIM periods completely with exchanges. If the protocol supports variable length TIM periods as used in the optimized 1 bit protocol, this is not a problem. Additionally, our results demonstrate that the optimized 1 bit TIM protocol provides the best energy conserving results for Type I traffic and competitive results for the other types of traffic. The optimized 1 bit TIM protocol achieves this energy conservation with a small sacrifice in throughput. The most attractive feature of the optimized 1 bit TIM protocol is that it can be applied in implementations of the point coordination function of the 802.11 MAC protocol.

8. Implementation of the optimized 1 bit TIM protocol in the 802.11 PCF

In 802.11 networks implementing the PCF, the PC is the lone node that transmits beacons. Included in these beacons are TIMs and all the timing information for when the next beacon will be transmitted. There is a hierarchical arrangement of timing information. There is a time unit (TU) defined as 1024 μs,[8] there is a beacon interval that defines the number of time units between successive beacons, there is a Delivery TIM (DTIM) period that defines the number of beacon intervals between successive DTIMs, and finally, there is a CFP period that specifies the number of DTIM intervals between the start of CFPs. (DTIMs are special beacons that all nodes are expected to awaken for since broadcast and multicast traffic follow them.) Since all timing information is included in each beacon and each node should adjust its timing information and dozing activity based on the receipt of these beacons, it is not too difficult to implement a layered TIM approach. All beacons become DTIMs and the timing information of each beacon is adjusted for the upcoming exchanges. The outer layer TIM would specify a long beacon interval consistent for the next outer layer TIM and then the first and every subsequent inner layer TIM would direct a shorter beacon interval consistent with the time required to exchange the selected packets. Figure 19 illustrates this use of aperiodic beacons. Undeniably, this approach circumvents the intent of these timing parameters but its use should not confound the network's performance provided the PC can keep track of what it is doing and nodes can adjust their activity with the receipt of each new DTIM.

[8] This quantization of the beacon interval limits the precision to exactly select optimum TIM periods but it only affects the throughput not the energy conservation characteristics of the protocol.

9. Conclusion

This paper described various methods for a central controller to manage the transmission of fixed sized packets. The results show that protocols can vary greatly and that their performance depends on the traffic type, the channel characteristics in which they operate, as well as the parameters, i.e. TIM period and CFP size, chosen for the protocol. These results also show that the scheduling policy is an important choice in designing the protocol but that in most cases a single algorithm, servicing the nodes with fewest transmissions first, should be used. It provides optimum performance for Type II, III, and IV traffic and excellent performance for Type I traffic. Fortunately, this scheduling policy is one of the easiest algorithms to implement.

The most significant conclusion of this research is that the focus of energy conservation should be to reduce the set of nodes in the receive state. Two goals in protocol design are shown to be critical. First, the protocol should attempt to put as many nodes to sleep as possible as early as possible and for as long as possible. Second, the protocol should coordinate transmissions such that nodes wake up to hear a transmission that enables them to return to the doze state immediately if it is appropriate. These objectives are more significant for larger networks where a greater percentage of the nodes can use low energy states.

Appendix A. Models for delay and energy consumption for protocols managing Types III and IV traffic

We model the transmission time and energy consumption of the 1 bit TIM, multiple bit TIM, and list protocols for Type III and IV traffic when there are no errors. The transmission times are easily modeled since they are independent of any scheduling or traffic distribution factors. They are dependent on the size of the network, n, the number of packets being transmitted in the contention free period (CFP), k, and in the case of the TIM protocols, the number of TIM periods that are used, j. For these models we define five timing variables, the time to transmit a packet including its overhead, τ_{Pkt}, the time to transmit a poll including its overhead, τ_{Poll}, the time to transmit an ACK including its overhead, τ_{ACK}, the time for the transmission overhead, τ_{OH}, and finally the duration of an interframe space, τ_{S}. The distinction between the equations for the three different types of traffic are the duration of the directories, the use of polls, the number of interframe spaces used, and some subtle end conditions.

We define $t_{1\,\mathrm{bit}}(k, j)$ as the time to transmit k packets if j TIM periods are used with the 1 bit TIM protocol. This is given by

$$t_{1\,\mathrm{bit}}(k, j) = j\left\lceil \frac{n}{|S|} \right\rceil + k(2\tau_{\mathrm{S}} + \tau_{\mathrm{Poll}} + \tau_{\mathrm{Pkt}} + \tau_{\mathrm{ACK}} - \tau_{\mathrm{OH}}) + c_1. \tag{A.1}$$

The first term of the equation (A.1) accounts for the transmission of the TIM. The size of the TIM is dependent on the number of nodes in the network adjusted to fit evenly into an integer number of time slots. Overhead is not included since the TIM is combined with the very next Poll thus overhead is included in the second term. We use $|S|$ to refer to the number of bits that can be transmitted in an interframe slot, in our case 48. The second term of (A.1) accounts for the time to transmit packets. Each packet transmission includes a poll, a packet, and an ACK. Since the polls are combined with either a packet or an ACK, in Type III or IV traffic respectively, one overhead transmission can be avoided. The interframe spaces occur between packet exchanges and between the packets and ACKs of each exchange. The third term accounts for the special end condition for Type IV traffic when the final ACK of a TIM period is not combined with a poll since a TIM is transmitted next. The constant c_1 has a value of 0 for Type III traffic and a value of $j \cdot (\tau_{\mathrm{OH}} + \tau_{\mathrm{S}})$ for Type IV traffic.

We define $t_{m\,\mathrm{bit}}(k, j)$ as the time to transmit k packets if j TIM periods are used with the multiple bit TIM protocol. It is given by

$$t_{m\,\mathrm{bit}}(k, j) = j\left(\tau_{\mathrm{OH}} + \left\lceil \frac{k(1 + \lfloor \log_2 \lceil k/j \rceil \rfloor)}{|S|} \right\rceil\right) + k(2\tau_{\mathrm{S}} + \tau_{\mathrm{Poll}} + \tau_{\mathrm{Pkt}} + \tau_{\mathrm{ACK}} - \tau_{\mathrm{OH}}) + c_1. \tag{A.2}$$

It only differs from (A.1) in the first term since the TIMs are a different size. Each TIM includes n multiple bit positions. Each position use the number of bits required to specify the number of packets that can be transmitted in a TIM period.

We define $t_{\mathrm{list}}(k)$ as the time to transmit k packets using the list protocol, which is given by

$$t_{\mathrm{list}}(k) = \left(\tau_{\mathrm{OH}} + \left\lceil \frac{k(1 + \lfloor \log_2 n \rfloor)}{|S|} \right\rceil\right) + k(m \cdot \tau_{\mathrm{S}} + \tau_{\mathrm{Pkt}} + \tau_{\mathrm{ACK}}). \tag{A.3}$$

The first term of (A.3) accounts for the time to transmit a list with abbreviated addresses. The size of the abbreviated addresses depends on the size of the network and the number of addresses in the list depends on the number of packets the list directs to be transmitted. The second term accounts for the time to transmit a packet. Each exchange includes the time to transmit a packet and an ACK. The number of interframe spaces, m, required depends on the traffic type. The PC requires precedence, i.e. priority, during the CFP for control purposes. To give this precedence to the PC we require that all mobile nodes wait two interframe spaces before attempting to transmit a new packet. The PC only needs to wait one and thus would have priority. Therefore, m is 3 for Type IV traffic and only 2 for Type III traffic.

Next we compute the average energy consumed in transmitting k packets. The energy consumed by a given collection of k packets depends on how these k packets are distributed among the network's n nodes and how they are scheduled to be transmitted. We determine the distribution of the packets by first conditioning on the number of nodes i spanned by the k packets to be transmitted. Let $p_{k,n}(i)$ denote the probability that the k packets are sent to/from i nodes in the network:

$$p_{k,n}(i) = \frac{\binom{n}{i} \sum_{j=0}^{i} (-1)^j \binom{i}{j} (i-j)^k}{n^k}. \tag{A.4}$$

Since our scheduling policy orders transmissions based on the number of packets exchanged with each mobile node, irrespective of which mobile node, our interest is in determining the likelihood of a given *type* of partition for k packets among i mobile nodes, where each mobile node participates in at *least* one exchange. Let $P_{i,k}$ denote one such partition and let $\mathcal{P}_{i,k}$ denote the set of all such partitions. We define a type of a partition $P_{i,k}$ as a vector $q(P_{i,k}) = (q_1, \ldots, q_i) \in \mathbb{N}^i$ with non-decreasing coordinates, where the jth coordinate corresponds to the number of packets sent to a destination receiving the jth smallest number of packets among the i nodes, thus $\sum_{j=1}^{i} q_j = k$. We shall let $Q_{i,k} \subset \mathbb{N}^i$ denote the set of vectors corresponding to all possible partition types for k packets among i destinations. Finally, for any $t \in Q_{i,k}$ we define a set of partitions $\mathcal{P}^t$ that have type t, i.e.

$$\mathcal{P}^t = \left\{ P_{i,k} \in \mathcal{P}_{i,k} \,\middle|\, q(P_{i,k}) = t \right\},$$

and $|\mathcal{P}^t|$ as the number of partitions of type t.

We shall let $p_{k,n}(t|i)$ denote the probability that a partition of Type $t \in Q_{i,k}$ is obtained given the k exchanges are among i of the network's mobile nodes. Using a counting argument one can show that

$$p_{k,n}(t|i) = \frac{|\mathcal{P}^t|}{\sum_{r\in Q_{i,k}} |\mathcal{P}^r|},$$
$$\text{where } |\mathcal{P}^t| = \begin{pmatrix} & k & \\ t_1 & t_2 & \cdots & t_i \end{pmatrix}, \quad \text{(A.5)}$$

since the traffic is assumed to be uniformly distributed and so partitions are equally likely.

For example, suppose there are 10 mobile nodes in the network labeled $d_1, d_2, \ldots, d_{10}$, and that three of these nodes are participating in five packet exchanges. The probability of this event is written $p_{5,10}(3)$ and is 0.18 by equation (A.4). Say the exchanges are with the following nodes: d_2, d_3, d_3, d_3, and d_4. Then the partition would be written as $q(P_{3,5}) = (1, 1, 3) = t$ where $t_1 = 1, t_2 = 1$, and $t_3 = 3$. The only other possible partition with three mobile nodes and 5 exchanges would be $q(P_{3,5}) = (1, 2, 2)$ and we find that

$$p_{5,10}(t|3) = \frac{\begin{pmatrix} & 5 & \\ 1 & 1 & 3 \end{pmatrix}}{\begin{pmatrix} & 5 & \\ 1 & 1 & 3 \end{pmatrix} + \begin{pmatrix} & 5 & \\ 1 & 2 & 2 \end{pmatrix}} = 0.4,$$

and the probability of this schedule over all possible schedules with $k = 5$ and $n = 10$ is $p_{5,10}(3)p_{5,10}(t|3) = 0.072$.

Now suppose the k packets to be transmitted correspond to a partition of type $t = (t_1, \ldots, t_i)$. Recall that $t_1 \leqslant t_2 \leqslant \cdots \leqslant t_i$ since jobs are scheduled shortest processing time first. Jobs are defined as the set of exchanges to a single node and its size is the number of exchanges in the set. Further suppose that these packets are equally distributed among j consecutive TIMs. We shall let $n_t(l)$ denote the number of nodes participating in exchanges during the lth TIM period and $m_t(r, l)$ the number of packets sent by the rth node participating in the lth TIM period in the partition of type t. Clearly $n_t(l)$ and $m_t(r, l)$, $r = 1, \ldots, n_t(l)$, are directly determined from t. So in the above example when we use TIM periods of size 2, then there are three TIM periods required to send the five packets and $n_t(1) = 2$, $n_t(2) = 1$, and $n_t(3) = 1$ and $m_t(1, 1) = 1$, $m_t(2, 1) = 1$, $m_t(1, 2) = 2$, and $m_t(1, 3) = 1$.

We now have the basic definitions to build our models. The models for each protocol consider four different energy components of the CFP, energy consumed in transmitting directories, adjustments for end condition of directory transmissions, energy consumed in transmitting packets, and, finally, adjustments for end conditions of packet transmissions, TIM periods, and/or the CFP. The first component is the energy consumed in receiving directories. All nodes are assumed to be awake to listen to these directories. For each directory that is transmitted, nodes must awaken and then return to the doze state if not identified as needing to stay awake. In our models we assume that all nodes awaken and return to the doze state in the first component and then adjust in the second component for those nodes that actually remain awake after the directory transmissions. In the third component we account for the energy consumed in transmitting packets. This accounts for the number of nodes awake and the duration of the transmission. Finally, we make adjustments for the end conditions such as the time for a node in the 1 bit and multiple bit TIM protocols to identify that it can enter the doze state and the time in the multiple bit TIM and list protocols a node must be awake before the first exchange.

We start with the model for the 1 bit TIM protocol. The amount of energy consumed in sending k packets is dependent on the partition t. Our approach to determining the expected energy consumption is to consider the energy consumed by each possible partition weighted by the probability of the partition's occurrence when traffic is distributed uniformly. We define the energy consumed by a particular partition of type t to be

$$\begin{aligned} e_{1\,\text{bit}_{k,j}}(t) = {}& j \cdot n \cdot \left(2\tau_S + \left(\tau_{OH} + \left\lceil \frac{n}{|S|} \right\rceil\right)\right) \\ & - \sum_{l=1}^{j} n_t(l)(\tau_S + \tau_{OH}) \\ & + \sum_{l=1}^{j} \sum_{r=1}^{n_t(l)} \big(n_t(l) + 1 - r\big) m_t(r, l) \\ & \times (2\tau_S + \tau_{Poll} + \tau_{Pkt} + \tau_{ACK} - \tau_{OH}) \\ & + c_1 + \sum_{l=1}^{j} \big(n_t(l) - 1\big)(\tau_S + \tau_{Poll}), \end{aligned} \quad \text{(A.6)}$$

where c_1 is the constant defined earlier to account for differences in Type III and IV traffic transmission. To determine the expected energy consumption we consider all possible partitions, i.e.

$$E\big[e_{1\,\text{bit}_{k,j}}(T)\big] = \sum_{i=1}^{\min(n,k)} p_{k,n}(i) \sum_{t\in Q_{i,k}} p_{k,n}(t|i) e_{1\,\text{bit}_{k,j}}(t). \quad \text{(A.7)}$$

To evaluate (A.7) we use a recursive algorithm to determine all $t \in Q_{i,k}$.

The expected energy consumption model for the multiple bit TIM protocol is formulated in much the same way. Again, energy consumption is dependent on the partition as follows:

$$\begin{aligned} & e_{m\,\text{bit}_{k,j}}(t) \\ & = j \cdot n \cdot \left(2\tau_S + \left(\tau_{OH} + \left\lceil \frac{n(1 + \lfloor \log_2 \lceil k/j \rceil \rfloor)}{|S|} \right\rceil\right)\right) \\ & \quad - j \cdot \tau_S \\ & \quad + k \cdot (2\tau_S + \tau_{Poll} + \tau_{Pkt} + \tau_{ACK} - \tau_{OH}) \\ & \quad + c_1 + \sum_{l=1}^{j} \big(n_t(l) - 1\big) c_2. \end{aligned} \quad \text{(A.8)}$$

The constant c_1 is as defined before. The constant c_2 accounts for the differences between Type III and Type IV traffic in the time a node must be awake prior to the first packet exchange. It is $2\tau_S$ for Type III traffic and $(\tau_{ACK} - \tau_{OH} + \tau_S)$ for

Type IV traffic. And finally, to determine the expected energy consumption we average over all possible partitions:

$$\mathrm{E}\big[e_{m\,\mathrm{bit}_{k,j}}(T)\big] = \sum_{i=1}^{\min(n,k)} p_{k,n}(i) \sum_{t\in\mathcal{Q}_{i,k}} p_{k,n}(t|i) e_{m\,\mathrm{bit}_{k,j}}(t). \tag{A.9}$$

The expected energy consumption model for the list protocol is much simpler since it only depends on the number of nodes participating in the CFP rather than the partition type and the number of TIMs. The energy consumed when i nodes participate in the CFP is

$$\begin{aligned} e_{\mathrm{list}_k}(i) = n \cdot \left(2\tau_\mathrm{S} + \left(\tau_\mathrm{OH} + \left\lceil \frac{k(1+\lfloor \log_2 n \rfloor)}{|S|} \right\rceil\right)\right) \\ - \tau_\mathrm{S} \\ + k\cdot(3\tau_\mathrm{S} + \tau_\mathrm{Poll} + \tau_\mathrm{Pkt} + \tau_\mathrm{ACK}) \\ + (i-1)(3\tau_\mathrm{S} + \tau_\mathrm{ACK}). \end{aligned} \tag{A.10}$$

The expected energy consumption is

$$E[e_{\mathrm{list}_k}] = \sum_{i=1}^{\min(n,k)} p_{k,n}(i) e_{\mathrm{list}_k}(i). \tag{A.11}$$

Appendix B. Descriptions of the scheduling algorithms used in our study of the transmission of Type I traffic

Algorithm A.

1. Select the pair of addresses that occur most and use these transmissions to fill the first slots of the transmission period.
2. If there are more slots in the TIM period, add any transmissions of pairs of nodes that are already awake that are not already part of the transmission schedule. Add the pairs that occur least frequently first.[9]
3. If there are more slots in the TIM period, identify the pair with one address already in the transmission set that occurs most frequently. Use these transmissions to fill the remaining slots of the TIM period.
4. Repeat steps 2 and 3 until all slots of the TIM period are filled or until no more pairs can be found to meet the criteria of these steps. If the latter occurs go to step 1.
5. Repeat all steps for subsequent TIM periods until all traffic has been transmitted.

Algorithm B.

1. Set $m = 2$. Let $q =$ the number of possible transmissions in a TIM period.
2. Consider all nodes remaining with traffic. Count the occurrences of transmissions between all combinations of nodes taken m at a time.
3. If there are no combinations of m nodes with a quantity of transmissions greater than or equal to q, increase m by 1 and repeat step 2.
4. Choose the combination of m nodes that meet the criteria of step 3 that has a quantity of transmissions closest to q. Schedule these in the next transmission slot.
5. Repeat starting at step 2 for the next transmission cycle.

[9] There is a benefit to exhausting as many of these pairs as possible since there is no penalty for participating in the given TIM period but there may be in a subsequent TIM period.

Algorithm C.

1. Count the frequency that each address appears in the set of transmissions waiting to be scheduled.
2. Schedule those transmissions involving the node that has the lowest frequency.
3. Repeat steps 1 and 2 until all traffic is scheduled.

Algorithm D.

1. Count the frequency that each node participates in data exchanges and select the node with the most exchanges.
2. Place all data exchanges involving the node identified by step 1 into a subgroup for transmission.
3. Repeat steps 1 and 2 for the transmissions that have not been added to a subgroup. Advance to step 4 if all exchanges have been added to a subgroup.
4. Group all exchanges within each subgroup between common pairs of nodes. If the identifying node of the subgroup is both transmitting and receiving in the set of a common pair, schedule the receptions last.
5. Go to the transition of the first two subgroups.
6. Reschedule the transmissions of the two adjacent subgroups such that at the transition there is a common node. Give preference to a node that receives in the forward subgroup. If necessary swap the order of the transmissions and receptions of the last pair in the forward subgroup.
7. If step 6 is successful and there are more transitions, advance to the next transition and repeat step 6. Otherwise, stop.
8. If common nodes cannot be found in the adjacent subgroups move the latter subgroup to the end of the transmission schedule and repeat step 6 for the new adjacent pair.

References

[1] ANSI/IEEE, Standard 802.11, Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications (IEEE Press, 1999).
[2] E. Buffa and R. Saring, *Modern Production/Operations Management*, 8th ed. (Wiley, New York, 1987) p. 290.
[3] J.-C. Chen, K.M. Sivalingam, P. Agrawal and S. Kishore, A comparison of MAC protocols for wireless local networks based on battery power consumption, in: *INFOCOM 98* (March 1998) pp. 150–157.

[4] I. Chlamtac, C. Petrioli and J. Redi, Energy-conserving access protocols for identification networks, IEEE/ACM Transactions on Networking 7(1) (February 1999) 51–59.

[5] ETSI, EN300 652 V 1.2.1, Broadband Radio Access Networks (BRAN); High Performance Radio Local Area Network (HIPERLAN) Type 1, Functional specification (July 1998).

[6] L.M. Feeney and M. Nilsson, Investigating the energy consumption of a wireless network interface in an ad hoc networking environment, in: *INFOCOM 2001* (April 2001).

[7] `http://www.aironet.com/products/in_building/PC4800B.asp`

[8] `http://www.networks.digital.com/dr/npg/deina-ch.html` (February 1999).

[9] Lucent Technologies, WaveLAN high-speed wireless network interface card IEEE Turbo 11 Mb specification (1999).

[10] Nokia, C020/C021 wireless LAN card technical specifications (1998).

[11] S. Singh and C. Raghavendra, Power efficient MAC protocol for multihop radio networks, in: *Personal, Indoor and Mobile Radio Communications Conference* (1998) pp. 153–157.

[12] K.M. Sivalingam, J.-C. Chen and P. Agrawal, Design and analysis of low-power access protocols for wireless and mobile ATM networks, Mobile Networks and Applications (MONET) (June 1998).

[13] K.M. Sivalingam, M. Srivastava and P. Agrawal, Low power link and access protocols for wireless multimedia networks, in: *IEEE Vehicular Technology Conference*, Phoenix, AZ (May 1997) pp. 1331–1335.

[14] K.M. Sivalingam, M. Srivastava, P. Agrawal and J.-C. Chen, Low-power access protocols based on scheduling for wireless and mobile ATM networks, in: *IEEE International Conference on Universal Personal Communications (ICUPC)*, San Diego, CA (October 1997).

[15] M. Stemm, P. Gauthier, D. Harada and R. Katz, Reducing Power Consumption of Network Interfaces in Hand-Held Devices, presented at the 3rd International Workshop on Mobile Multimedia Communications but not in Proceedings, `http://www.cs.berkley.edu/~stemm/pubs.html`

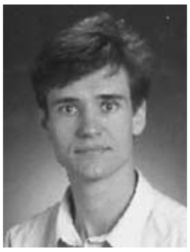
[16] M. Stemm and R. Katz, Measuring and reducing energy consumption of network interfaces in hand-held devices, IEICE Transactions on Communications E80-B(8) (August 1997) 1125–1131.

[17] J.A. Stine, Energy conserving protocols for wireless data networks, Ph.D. Dissertation, University of Texas at Austin (2001).

[18] J. Stine and G. de Veciana, Tactical communications using the 802.11 MAC protocol, in: *MILCOM 98* (October 1998).

[19] J.A. Stine and G. de Veciana, A comprehensive energy conservation solution for mobile ad hoc networks, in: *IEEE International Communication Conference* (2002).

[20] H. Woesner, J. Ebert, M. Schlager and A. Wolisz, Power-saving mechanisms in emerging standards for wireless LANs: The MAC level perspective, IEEE Personal Communications (June 1998) 40–48.



Gustavo de Veciana received his B.S., M.S., and Ph.D. in electrical engineering from the University of California at Berkeley in 1987, 1990, and 1993 respectively. In 1993, he joined the Department of Electrical and Computer Engineering at the University of Texas at Austin where he is currently an Associate Professor. His research focuses on issues in the analysis and design of telecommunication networks. Dr. de Veciana has been an editor for the IEEE/ACM Transactions on Networking. He is the recipient of a General Motors Foundation Centennial Fellowship in Electrical Engineering and a 1996 National Science Foundation CAREER Award, and co-recipient of the IEEE Bill McCalla Best ICCAD Paper Award for 2000.
E-mail: gustavo@ece.utexas.edu

John A. Stine received his BS degree from the United States Military Academy at West Point in 1981 and then his MS degrees in electrical engineering and manufacturing systems engineering in 1990 and his PhD degree in electrical engineering in 2001, all from the University of Texas at Austin. He served 20 years in the United States Army which included assignments as an Assistant Professor of Electrical Engineering at West Point and later as the coordinating analyst for the Army's Task Force XXI experiment, the US Army's first attempt to wirelessly network a brigade. Currently, he is employed by the Mitre Corporation where he does research in ad hoc networking and in the modeling of communications and information systems and their effects in combat simulation environments.