

An Asynchronous Power Save Protocol for Wireless Ad Hoc Networks (Rev 1.1)

Laura Marie Feeney
Swedish Institute of Computer Science
Computer and Network Architecture Lab
Box 1263, SE 164 29, Kista, Sweden
Web: <http://www.sics.se/cna>

Abstract

This report describes a power save protocol for ad hoc networks. The protocol is largely independent of the details of the underlying MAC and is friendly toward any overlying energy-aware ad hoc routing. A key advantage is that the protocol is fully asynchronous. Neighbors that wish to communicate estimate the relative phase difference between their sleep/wake cycles. A station uses this phase information to order its pending transmissions to use the available periods of overlap most efficiently. Stations can also adjust their phase relationships to avoid contention and reduce latency for delay sensitive flows.

The proposed protocol is defined in considerable detail and it is argued that the protocol is likely to exhibit good energy savings as well as acceptable network performance. The proposed protocol is also carefully distinguished from related work in this area. Based on these arguments, it is recommended that work continue to implement the proposed protocol in a fully featured simulation environment and more carefully study its effectiveness.

1 Introduction

This report proposes a new power save protocol for general purpose mobile multi-hop wireless networks. The proposed power save protocol operates at the link layer management level. While intended for use in connection with any CSMA/CA MAC layer, the protocol is here defined as a variant of IEEE 802.11 power save mechanisms. There is no particular requirement on the operation of the overlying ad hoc routing protocol, as the phase announcement protocol only considers communication with next-hop neighbors..

A key advantage of the protocol is that stations operate asynchronously, each following a fixed, independent sleep-wake cycle. The protocol is based on a phase announcement mechanism by which neighbors that wish to communicate determine the relative phase difference between their sleep-wake cycles. This allows stations to determine their common awake intervals and to schedule transmissions appropriately.

Stations can easily adjust their relative phase, allowing them to adapt their transmission schedules to contention or to high priority flows. This kind of adaptivity is expected to prove useful for providing QoS support in ad hoc networks, which are characterized by complex effects of interference across multiple links and between disjoint flows.

The report defines the basic phase announcement protocol in some detail and provides more speculative discussion on phase adjustment mechanisms to provide support for QoS traffic.

2 Motivation

The network interface is a significant source of energy consumption in portable wireless devices [22]. An interface that is sleeping can neither transmit nor receive any kind of control or data traffic. As a result, it consumes very little energy. In order to be able to transmit or receive, an interface must explicitly transition to the wake state. An interface that is awake can transmit or receive data at any time; if it is doing neither, it is said to be idle. In all of these states, an interface will consume significantly more energy than it does in the sleep state, due to the number of circuit elements that must be powered.

There has been some recent interest in investigating the energy consumption of commercially available network interfaces, e.g. [3], [7], [13], [22], [6]. This data is useful because it gives researchers a sense of relative costs of various operations and allows them to focus their attention on the key sources of energy drain.

In part because of its ubiquity, the case of IEEE 802.11b is particularly interesting. Measurements ¹ show that an *idle* network interface consumes

¹Measurements vary somewhat depending on manufacturer and model; the values cited

over 800mW. This is comparable to the energy consumed while receiving or transmitting (1000mW and 1300mW respectively), and an order of magnitude larger than the energy consumed by a sleeping node (66mW - 130mW). One can make the following “back of the envelope” calculation based on data presented in [7] for Lucent IEEE 802.11 hardware: An interface sending ten 128-byte broadcasts per second and receiving the same from each of five neighbors consumes about 1% more power than an idle interface.

In short, data traffic is not the primary source of energy consumption: To reduce energy consumption, an interface must spend as much time as possible in the sleep state. Some coordination mechanism is therefore required so that endpoints that wish to exchange traffic can arrange to be awake at the same time.

In an infrastructure wireless network, communication is mediated by a predetermined access point, which is generally assumed to have access to mainline power. The access point is continually awake and buffers traffic for its sleeping clients. Achieving this kind of coordination is difficult in an infrastructureless networking environment. The network topology is highly dynamic and there is no resource rich, centralized element around which to construct a power save mechanism. The fact that nodes in an ad hoc network cooperatively form the network infrastructure imposes a further requirement on energy management techniques: to maximize system lifetime by ensuring that energy consumption is appropriately balanced across the nodes in the network.

The goal of this work is to define a link layer management element which implements a power saving protocol suitable for multihop ad hoc networks. The work is intended to be applicable to a broad range of ad hoc networking scenarios. In particular, it is not directed to the special case of a sensor network. The protocol should depend as little as possible on the details of a specific media access control protocol. Nor should it impose limitations on an overlying ad hoc routing protocol; it must be friendly to energy aware routing. Finally, the protocol must be responsive to QoS requirements for supporting delay or jitter bounded traffic.

Any solution proposed for an ad hoc network must be fully distributed; it must rely only on local information. Due to the complexity of obtaining phase synchronization in a dynamic network that is very prone to link and node failure, an asynchronous solution is also highly desirable.

In addition to the obvious requirement for minimizing energy consumption, any solution must address the following performance requirements:

- Minimize adverse impact on capacity, throughput, latency and route discovery.

are fairly representative. Some interfaces support multiple sleep modes, with varying energy and transition costs.

- Minimize overhead imposed by the energy management scheme. Ideally, overhead diminishes to zero in the case of an idle network. This principle is not only aligned with principles of on-demand routing, it is also important for supporting scenarios involving long lived, intermittently trafficked networks.
- Maximize system lifetime by minimizing disparities in energy consumption. This issue is primarily addressed at the routing and application layers. Because the proposed power save protocol operates below these layers, it must take care to avoid introducing inappropriate feedback caused by the sleep/wake patterns.

The following sections provide a general overview of energy management techniques for ad networks, followed by a detailed description of three solutions based on link layer management.

3 Background

This section gives a brief overview of work in energy consumption, while the following sections consider particularly relevant work in more detail.

Media access control schemes based on TDMA can exhibit good energy consumption properties because each node can exchange traffic only during the slots which have been allocated to it by the TDMA scheduling algorithm. Spatial-reuse TDMA permits several stations to share a slot, as long as their spatial separation is such that their transmissions do not interfere with each other, making it appropriate for use in multi-hop wireless networks. The distributed computation and assignment of schedules for an STDMA [9] protocol for ad hoc networks is a topic of active research. A Bluetooth scatternet is another an instance of a TDMA-based multi-hop network which has received considerable attention [11].

Wireless media access protocols based on CSMA/CA do not have an assigned schedule to avoid conflicts. Instead they use a combination of carrier sensing and explicit control traffic to prevent conflict. A mechanism by which nodes use control traffic to determine when they cannot send or receive traffic, allowing them to sleep while nearby nodes are transmitting is described in [19].

Like TDMA protocols, CSMA/CA protocols can include power save mechanisms in their management subsystems. The IEEE 802.11 standard for MAC sublayer management describes power save mechanisms for use in both BSS and IBSS mode operation. An alternative to the IEEE 802.11 mechanism is described in [3]. Because of its relevance to the proposed protocol, work in this area is described in greater detail in following sections.

A wide range of power control techniques are also used in wireless communication to reduce interference and conserve energy. Power control has

particularly interesting applications in a multihop network. Because large scale path loss is exponential, the power needed to successfully transmit data to a receiver with a given BER is generally modeled as exponentially proportional to the distance to the receiver. The balance between transmit power and the probability of retransmission has been widely studied, e.g. [4].

Power control can also be used to control the effective node density of a multihop ad hoc network. Such topology control techniques for are used to minimize interference and improve energy efficiency by selecting transmit power levels that provide maximum network connectivity for minimum energy cost, e.g. [16], [17], [26], [23], [24].

Such techniques can also increase spatial reuse, but the presence of widely varying transmit powers in a multihop network greatly complicates the collision avoidance problem. Low power transmissions cannot be sensed by distant nodes, which may then initiate their own transmissions using sufficient power to disrupt the low power transmission. This problem is often poorly addressed by work in this area, but some possible solutions are found in e.g. [1], [8], [15], [18].

Another issue in energy management for ad hoc networks is usually addressed at the routing layer. Because nodes in an ad hoc network cooperatively forward traffic, a solution which minimizes hop-count, or even one which minimizes per-packet energy consumption, may negatively impact network lifetime. Energy-aware routing attempts to maximize the network lifetime by taking into account the battery reserves at each node in the network, as well as minimizing the energy consumed per unit of data delivered. A variety of metrics and protocols have been developed, some of which combine energy aware routing with topology control, e.g. [20], [2], [28], [17].

Higher layer information is also used to aid in conserving energy used in communication, e.g. [12], [22]. Such techniques are often limited to infrastructure based networks, in which the communication resources of each node are used only to support applications running at that node.

An important class of application specific ad hoc networks is the sensor network: a collection wireless enabled sensors deployed at random over a geographic area. The sensors monitor some aspect of their environment and the information they collect is forwarded to a gateway node, possibly for further processing or to notify a remote operator of some anomalous event.

Energy efficiency is particularly important in these networks because the energy consumed in communication can dominate total energy consumption of small, specialized devices. Many energy saving techniques for sensor networks are closely tied to the application scenario, e.g. [10], which uses data fusion techniques to minimize the amount of information that is forwarded in the network.

4 IEEE 802.11 Power Saving Mechanisms

Due to the widespread availability of inexpensive hardware and its relatively stable and complete protocol definition, the IEEE 802.11 standard is the default choice for use in ad hoc networking research. The IEEE 802.11 standard includes power saving mechanisms for use in both infrastructure (BSS) and infrastructureless (IBSS) operating modes. These mechanisms are outlined below.

4.1 Power saving in a BSS

For BSS operating mode, the access point is responsible for buffering and forwarding traffic to stations in its BSS. The access point is assumed to have no energy constraints and remains awake at all times. Each station explicitly associates itself with an access point, with which it negotiates power save state and parameters.

The access point establishes the beacon period to which BSS stations are synchronized. At each beacon interval, the access point generates a broadcast beacon which includes a synchronization timestamp and traffic indication map (TIM). The TIM identifies the stations for which the access point has buffered traffic. Stations in the BSS are synchronized to the access point; waking at the beginning of the beacon interval and remaining awake until the beacon has been received. (The beacon can be delayed due to media contention). Stations with no pending traffic return to the sleep state. If a station receives a beacon indicating pending traffic, it sends an acknowledgment message to the access point. The access point forwards pending traffic to the station, which remains awake until the access point indicates that there is no more traffic pending for the station.

In every n th beacon interval, the TIM includes an indication (DTIM) of whether there is pending broadcast or multicast traffic. If such traffic is pending, it is sent before any pending unicast traffic. All stations remain awake until the access point indicates that there is no further broadcast or multicast traffic, after which stations not also identified as unicast destinations return to the sleep state.

A station operating in “ultra low power” mode may choose not to wake up in every beacon interval. If the station elects not to wake up to receive a beacon containing a DTIM, the broadcast or unicast traffic is lost. Unicast traffic indications in the TIM are persistent, so traffic is lost only if the access point expires the pending traffic before the station receives and responds to the relevant TIM.

4.2 Power saving in an IBSS

The case of an IBSS is more relevant to the ad hoc networking, although it is worth noting that there are some differences between an IBSS and a general multihop wireless network. In particular, each station explicitly discovers and joins itself to a single IBSS, synchronizing itself to the beacon interval of that IBSS.

This common beacon interval is established by the station that initiated the IBSS and maintained in a distributed fashion. All of the stations wake up at the beginning of the beacon interval and each one calculates a random delay, after which it sends a beacon containing a synchronization timestamp. Each station synchronizes itself to the first beacon it receives and cancels its own pending beacon transmission.

In addition to a common beacon interval, the IBSS also defines a fixed length “ATIM window”, which occurs at the beginning of each beacon interval. All stations in the IBSS wake up at the beginning of the beacon interval and remain awake until the end of the ATIM window.

Once the beacon has been transmitted and received, each station sends an ad hoc traffic indication message (ATIM) to every other station for which it has pending unicast traffic. Each station that receives such an ATIM responds with an acknowledgment. Announcements of broadcast and multicast traffic are sent to the appropriate broadcast or multicast address, but are not acknowledged. ATIM transmission times are randomized using backoff and DCF access mechanisms. Only beacon, ATIM and ATIM acknowledgments are sent during the ATIM window.

At the end of the ATIM window, stations that have not sent or received ATIM announcements return to the sleep state. If a station sends or receives an ATIM, it remains awake throughout the remainder of the beacon interval. Each station transmits first broadcast and multicast traffic, then any unicast traffic for which an ATIM acknowledgment was received. Again, the IEEE 802.11 DCF access mechanisms are used. Because stations remain awake for the entire beacon interval following an ATIM window in which they send or receive an ATIM, a station may send additional traffic to another station that is known to be awake without further ATIM negotiation.

Traffic which was not successfully announced (e.g. due to lack of time in the ATIM window) or was announced, but not successfully delivered (e.g. due to lack of time in the beacon interval) is announced again in the following ATIM interval (or times out and is discarded).

4.3 Effectiveness of IEEE 802.11 Power Saving

The effectiveness of the power saving mechanism depends heavily on the values selected for the beacon and ATIM intervals, as well as the offered load. If the ATIM window is too short, not enough traffic can be

announced during the ATIM window, possibly not even enough traffic to utilize the entire beacon interval. If the ATIM window is too long, not only are stations required to spend more time awake, but more traffic may be announced than can be sent in the remainder of the beacon interval. If the beacon interval is too short, the overhead of the sleep-wake cycle, beaconing, and traffic announcements will be high. If the beacon interval is too long, many stations may attempt to announce traffic at each ATIM window, such that many destinations will need to remain awake after the ATIM window.

Although these power saving mechanisms are part of the IEEE 802.11 standard, there appear to be relatively few published research results regarding their effectiveness. Parameters that affect the overall effectiveness of IEEE 802.11 power saving mechanisms were studied in [3], [25] and unfortunately, the results are only moderately encouraging.

The key criteria is naturally the amount of power savings, but factors such as increased latency, decreased throughput and unequal distribution of power consumption must also be taken into account. It is important to note that the percentage of time spent in the sleep state is only an indication of the actual energy savings, which will be reduced by the costs of the wake-sleep transition, beaconing and ATIM traffic, all of which increase as the beacon interval decreases.

4.4 PTOLMEY-based simulation

A simulation study described in [25]², considered the effectiveness of the power save protocol for a fully connected eight-node IBSS. The experiment measured throughput and proportion of time spent in the sleep state for a variety of beacon intervals (35 - 205ms), ATIM window lengths (up to at least 40% of the beacon interval) and offered loads (15 - 60% of available bandwidth). The results show considerable complexity in the choice of beacon interval: short beacon intervals give superior energy saving, but at the cost of substantially reduced throughput. As a general observation, the authors suggest that “if we were to sacrifice about 10% in throughput, we could save up to 30% energy”.

For a moderately to heavily loaded network and a wide range of beacon intervals, throughput is maximized when the ATIM window occupies about 25% of the beacon interval. Because a smaller window results in lower energy consumption while still providing acceptable throughput for lightly loaded networks, the authors suggest a mechanism for maintaining an adaptive ATIM window.

Beacon intervals smaller than 95 milliseconds had noticeably lower throughput, but significantly higher energy savings. In the case of a 30% offered load, only intervals of 35 and 45 milliseconds showed significant (30-40%)

²Based on IEEE 802.11 Draft 4.0.

sleep time, while beacon intervals of 125ms and more resulted in negligible (0-10%) sleep time. In the case of a 15% offered load, the shortest beacon intervals resulted in sleep times of 65-70%, while the longer intervals exhibited sleep times in the 20-35% range. In part because the study considered only a fully connected IBSS, latency was not measured; in the usual case, latency is one beacon interval.

4.5 NS-based simulation

The simulations described in [3] consider the case of a IEEE 802.11 network supporting mobile stations and multihop paths. The authors compared the latency, packet loss and energy savings obtained using IEEE 802.11 both with and without IBSS power saving mode in the context of their proposed protocol, Span, which is discussed below.

In these experiments, a beacon interval of 200ms and an ATIM window of 40ms was “experimentally determined” to provide low packet loss rate. The 60Kbps simulated traffic consisted of twenty CBR streams of three 128 byte packets per second. The resulting load is substantially greater than 3% of 2Mbps would imply, not only because each packet traverses multiple hops, but also because the IEEE 802.11 CSMA/CA mechanism greatly reduces the capacity of a multihop network [14].

The simulation results show that in the mobile multihop case, the IEEE 802.11 power save mode reduced energy consumption very little compared to IEEE 802.11 without power saving. This results is at least partially due to the experimental context, which was to study the Span protocol. The geographic routing protocol used to forward traffic requires periodic broadcast traffic, which forces all stations to remain awake through the beacon interval.

The results also suggest that for a multihop network, the cumulative latencies become extremely high because each packet is forwarded at most one hop in each beacon interval. This issue becomes even more significant if these kinds of strategies are combined with energy and capacity saving techniques that replace a smaller number of high transmit power hops with larger number of low power short hops.

5 Related Work

5.1 Domination Techniques

The work most closely related to the proposed protocol is Span [3]. Span is intended to maximize the amount of time nodes spend in the sleep state, while minimizing the impact of power savings on latency and network capacity.

In Span, a local election is used to elect coordinators from among all the nodes in the network. Coordinators are selected such that they form a connected dominating set: Each node has at least one neighbor that is a coordinator and any two coordinators are connected by a path that includes only coordinators. The coordinators remain awake at all times and therefore form a low latency routing backbone for the network. The Span coordinator election algorithm is intended to approximate a minimal capacity-preserving set of coordinators.

Various techniques based on dominating sets have been proposed in [21] and [26]. The former defines a routing protocol based on gatewaying traffic among a collection of nodes approximating a minimum (non-connected) dominating set for the network. Energy efficiency is not specifically addressed in this work. The latter presents a synchronous algorithm for selecting a connected dominating set, preferring nodes which have a high “energy level”.

The Span mechanism is intended to operate in conjunction with any MAC/PHY layer that provides for sleeping and polling for buffered traffic, though the details of the protocol are defined in terms of IEEE 802.11 MAC. In this context, the authors propose some small optimizations to the basic IEEE 802.11 IBSS power saving mechanism.

The most important of these is the advertised traffic window. In addition to the ATIM window, Span also defines an advertised traffic window, which includes the interval immediately following the ATIM window. Only traffic advertised during the ATIM window, that is, traffic involving non-coordinator nodes, is sent during the advertised traffic window. The remainder of the beacon interval is used exclusively by the coordinator nodes to exchange traffic among themselves. Because the coordinators are always awake, advertisements are not needed. This has the advantages of reducing the amount of ATIM traffic and shortening the ATIM window, as well as ensuring that traffic for non-coordinator nodes faces as little contention as possible, to minimize the time these nodes remain awake. The reported experiments used a beacon period of 300ms, with an ATIM window of only 20ms and an advertised traffic window of 100ms.

Broadly speaking, simulation suggests that Span provides about 50% energy saving compared to IEEE 802.11, with little impact on throughput, latency and packet loss. Even when Span is used to limit idle node energy consumption, sending and receiving traffic accounts for well under 10% of the total energy consumed. However, Span has two limitations that can be addressed more effectively using an asynchronous approach.

The first limitation is that the coordination mechanism requires broadcast HELLO messages to maintain the coordinator backbone, even when there is no data traffic. If Span is used in conjunction with a MAC or routing protocol that also requires such messages, coordination may not introduce much additional overhead. This was true in the case of the Span

performance analysis reported in [3], which uses a simple geographic routing protocol. But if this traffic is not otherwise needed, the cost of coordination is pure overhead. In particular, on-demand ad hoc routing protocols were designed to avoid this problem and so may not be good candidates for use with Span.

The second, and more important, limitation is the synchronized nature of the Span protocol. In part, this synchronization is connected with the underlying beaconing in the IEEE 802.11 protocol. But aside from this, the coordination mechanism also requires that all nodes periodically exchange HELLO broadcasts containing neighbor information. This cannot be mediated by the coordinators: Non-coordinator nodes must be awake simultaneously to perform neighbor discovery

Maintaining the common beacon interval requires two different kinds of synchronization. The first ensures that the stations' oscillators tick at the same rate (within some tolerance; the IEEE 802.11 specification names 10^{-4} m, which can be done using well known techniques. The second ensures that the stations are synchronized in phase. This is a different kind of challenge because the choice of phase is completely arbitrary.

The IEEE 802.11 IBSS solves this problem in a centralized way. The "first" station initializes the ID and beacon interval for the IBSS and "subsequent" stations explicitly associate themselves with exactly one IBSS and synchronize themselves to it.

Nevertheless, this approach works well for some application scenarios, specifically those in which a master station for the network can be conveniently designated and all the other stations can be configured to recognize that master. This requires, in effect, that all nodes are initialized together within the same well connected cloud or that there is some mechanism for identifying the "right" cloud for a node to associate itself with.

This limitation is unfortunate because the ad hoc networking model is intended to support more flexible methods of creating a network. In particular, consider the case of two separate task groups, e.g. military units on patrol, each of which has formed an ad hoc network. When the two groups meet, their networks should merge seamlessly together.³

If the two networks have different phases, some mechanism must be developed which allows the two networks to synchronize with each other and exchange traffic. The problem is non-trivial and any solution will introduce additional complexity, as well as additional bandwidth and energy overhead on the system. The following example illustrates the potential complexity.

³Political layer compatibility is *not* my department.

5.2 Adaptive Techniques

The techniques proposed in [27] are fully asynchronous and exhibit good energy savings. The protocols are intended to be integrated with an on-demand ad hoc routing protocol; that is, a protocol which discovers routes by means of broadcast flooding. While the protocol can, in principle, operate in conjunction with any on-demand routed ad hoc network, its best behavior is likely to be found in sensor networks.

The idea behind the basic energy conservation algorithm (BECA), is outlined below. Each node independently transitions between the sleep state and one of two logical wake states: listening and active. In the absence of traffic, a node alternates between the sleep state and the listening state. If a node sends or receives traffic, it transitions to the active state. Nodes in the active state return to the sleep state only after they have been idle for some time.

Because the network is asynchronous, there is no guarantee that there is a live path to the destination, or even that the destination itself is awake, at the time a route request (RREQ) is generated. In order to ensure that a multi-hop request reaches its destination, there must be a careful relationship among the various parameters defined by both the energy conservation protocol and the route discovery protocol.

The fundamental unit is the listening interval, which is matched to the RREQ retry interval for the routing protocol. If the sleep interval is some integral multiple k of the listening/retry interval, then it will take at most $k + 1$ retries until any given neighbor receives the RREQ. Once the neighbor receives the RREQ, it transitions to the active state. As long as the timeout for the active interval is greater than the retry interval, the intermediate nodes will remain awake until the route discovery process has completed, taking at most $D(k + 1)$ retries, where D is the network diameter. Once the route has been established, only the nodes on the route will remain active. The other nodes will be idle and once the active interval has timed out, they will return to the low energy consumption sleep-listen cycle. Once traffic along the route ceases, the nodes on the route will also become idle and return to sleep-listen cycle.

The adaptive fidelity energy conservation algorithm (AFECA) is an extension of BECA. In a dense, uniformly-distributed ad hoc network, most nodes are logically equivalent with respect to network reachability. Nodes can therefore adapt their sleep interval depending on the estimated network density. In the simplest case, each node sets its sleep interval using a random value k uniformly distributed on $[1..N]$, where N is the number of observed neighbors. Because the estimate affects latency rather than correctness, it is not necessary to spend resources on precise estimation. Although the feedback effect ensures that nodes which have a low density estimate spend more time listening for neighbors, in an idle network, AFECA reduces BECA.

More complex forms of adaptation are possible, though the adaptive fidelity mechanism is not explicitly intended to be capacity preserving, as in the case of Span.

Like Span, AFECA has been studied using the ns2 simulation environment, using AODV as the ad hoc routing protocol. The strategy shows good energy savings, Overall energy saving was on the order of 35% - 45%, across a range of traffic loads, with a minimum sleep interval of 10 seconds. There was a significant increase in route latency, which averaged well under one second for unmodified AODV, but averaged between six and ten seconds using AFECA. AFECA also exhibited slightly higher loss rates than unmodified AODV.

Like Span, AFECA provides substantial energy savings. The protocol has a number of limitations, some of which make the protocol most suitable for use in sensor network scenarios.

One problem is that broadcast is expensive – in bandwidth, as well as energy – because it requires many rebroadcasts. As seen above, a RREQ must be broadcast several times, in order to be likely to reach many neighbors. As long as route discovery and repair is a rare operation, this is a minor drawback. If the network is to support other services and applications that also rely on broadcast, then the cost becomes more of an issue. Moreover, because “logical broadcast” requires several broadcasts spread over a relatively long interval, there is a risk of synchronization problems. In particular, the author suggest that proactive routing protocols based on periodic table exchange may be vulnerable.

Another problem is that in the current protocol, neither the routing nor the power save mechanism is aware of the energy level of a node. In fact, because nodes that have recently forwarded traffic remain awake, they are more likely to participate early in the route discovery process and so be designated as forwarding node for additional routes. Depending on traffic patterns, this may lead to unequal distribution of routing load and poorly distributed energy consumption. This is also suggested by simulation results.

AFECA provides significantly increased network lifetimes, especially for the case of an extremely dense networks, where the time to last node failure doubles. The network half-life increases by as much as 50%. This kind of metric is appropriate for sensor networks, where the duration of sensor coverage in an area is more important than the availability of any particular device. For application scenarios that are oriented toward personal communication, however, the loss of connectivity to any individual device is more significant. With AFECA, there is almost no increase in time to first node failure and only a small increase in the 90% node lifetime.

6 Phase Announcement Power Save Mechanism

6.1 Overview

Having discussed some of the strengths and weaknesses of a number of power save protocols, this section presents a new protocol. Like the protocols above, it is largely independent of the details of the underlying media access protocol, though it too is described in the context of the IEEE 802.11 MAC.

In the proposed protocol, each station independently alternates between sleep and wake states. The sleep-wake pattern is defined such that it possible to make certain guarantees about the overlapping of awake intervals for each pair of stations. In this respect, the protocol is similar to AFECA. Unlike AFECA, these overlap guarantees are used to support a traffic announcement mechanism, similar to those used in Span and IEEE 802.11 power saving.

The proposed protocol differs significantly from both of these protocols, however, in that the traffic announcement is not used as a request to “stand by”, but rather as a way for neighboring stations to discover the phase difference between their sleep-wake cycles. Once two stations have determined their relative phase, each can predict when the other will be awake to receive traffic. Stations attempt to send traffic, using whatever method is provided by the underlying MAC, only when they expect the receiver to be awake.

Structurally, this approach differs a little bit from others in that the communication adapts to the sleep-wake cycle, rather than the other way around. Each node maintains a fixed sleep-wake cycle, significantly reducing the possibility that the power save protocol will exacerbate any inequality in energy consumption. This is in distinct contrast with Span, which must continually negotiate a routing backbone of non-sleeping coordinator nodes, and with AFECA, which requires that nodes along an active path remain awake until they forward no more traffic.

Although it is possible to guarantee a minimum overlap between two stations, it is not possible to guarantee that the network is capacity preserving. For example, the distribution of sleep-wake cycles among nearby nodes may be such that the overlap intervals for several active links coincide, leading to high levels of interference. Or the distribution of sleep-wake cycles along a path may be such that each intermediate station begins its sleep phase just after it receives a packet for forwarding. Conversely, a friendly distribution of sleep-wake cycles can be most helpful. A convenient sequence of overlap intervals along a path will allow for latencies comparable to those provided by a routing backbone. Alternatively, minimizing the overlap with other active links can help to isolate flows with QoS requirements from interference.

The complexity and effectiveness of such mechanisms is the topic of ongoing research. The initial investigation focuses on the effectiveness of a random phase distribution.

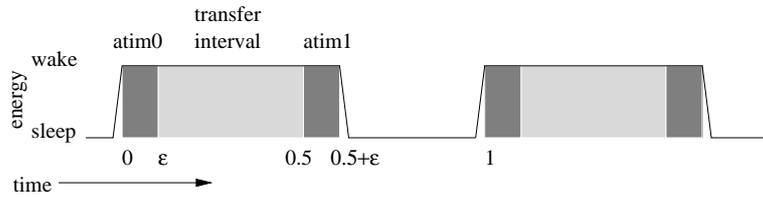


Figure 1: Sleep/wake cycle.

6.2 Overlap Principle

We begin with the following trivial observation: If all the stations are awake more than half of the time, then each pair of stations will have an overlapping awake period, regardless of phase.

More formally, the protocol defines two well-known constants: an interval I , whose length is normalized to 1, and a value ϵ , $0 < \epsilon \leq 0.25$. Let each station independently follow a schedule consisting of an awake interval of duration $(0.5 + \epsilon)$ followed by a sleep interval of duration $(0.5 - \epsilon)$. See Figure 1.

Independently scheduled sleep-wake intervals defined in this way have the following useful *overlap property*: For each pair of stations T and R , in each interval I , there exists at least one sub-interval of I in which both T and R are in the awake state. Moreover, for station T , either the sub-interval $[0, \epsilon]$ or the sub-interval $[0.5, 0.5 + \epsilon]$ will be completely contained in the awake interval of station R , regardless of the phase difference between them.

Proof: Let ϕ be the phase difference between T and R . Measured on T , the awake interval of T is $[0, 0.5 + \epsilon]$ and the awake interval of R is $[0 + \phi, 0.5 + \epsilon + \phi]$.

If $0 \leq \phi < 0.5$, the awake interval of R cannot begin after $t = 0.5$ and cannot end before $t = 0.5 + \epsilon$, as measured on T . Thus the interval $[0.5, 0.5 + \epsilon]$ measured on T is contained by an awake interval of R . If $0.5 \leq \phi < 1$, the awake interval of R cannot begin after $t = 1$ and cannot end before $t = 1 + \epsilon$, as measured on T . Thus the interval $[0, \epsilon]$ measured on T , is contained by an awake interval of R .

6.3 Phase Announcement Protocol Overview

The awake interval of a station is broken down into three subintervals: the ATIM-0 interval ($[0, \epsilon]$) at the beginning of the awake interval, the ATIM-1 interval ($[0.5, 0.5 + \epsilon]$) at the end of the awake interval, and the transfer interval ($[\epsilon, 0.5]$) in between. The station sleeps in the interval $[0.5 + \epsilon, 1.0]$. See Figure 1. To simplify the discussion below, let us assume that each network interface can schedule sleep-wake state transitions with perfect accuracy and

can transition between states instantaneously. (Building appropriate guard times into the protocol is fairly straightforward.)

According to the overlap principle, for each station T , one or both of its two ATIM intervals will be completely contained in the awake interval of each neighbor of T . Consequently, any broadcast or multicast data that T transmits in both of its ATIM intervals will (in the absence of error) be received by every neighbor.

During its ATIM intervals, each station therefore transmits a broadcast traffic announcement listing the stations for which it has pending traffic and its current estimate (if any) of the relative phase difference for those stations. Receivers use this traffic indication message to establish their own estimate of the phase difference between the stations. If the receiver's estimate differs from the sender's, the receiver informs the sender by responding with an ATIM-ACK. The sender then uses these phase estimates to calculate its transfer windows with respect to each receiver and order pending transmissions so as to maximize value with respect to some QoS metric.

It is worth emphasizing that the ATIM is an announcement of current phase estimates and a request for phase updates, not a request to stand by to receive traffic. The two ATIM intervals are just the (combined) transfer window for broadcast and multicast traffic.

Because stations operate asynchronously, there is no distinguished ATIM interval as in IEEE 802.11 power save. The ATIM and ATIM-ACK will face contention from both traffic announcement and data traffic. This problem can be mitigated by giving ATIM traffic priority as control traffic, if this is supported by the MAC protocol. If a sender is unable to transmit ATIM traffic in a timely fashion, it can select a new phase with respect to the competing ATIM interval or transfer window.

It may also happen that a sender is persistently unable to transmit all of the traffic pending for some receiver. The transfer window may be too short, or several receivers may have overlapping transfer windows, or there may be high levels of contention during the transfer window. The sender can signal the overload condition using the traffic indication message. In response, the problematic receiver can select a new phase with respect to the sender, announcing the updated phase in the ATIM-ACK ⁴.

One of the strengths of the phase adjustment approach is its ability to support a variety of algorithms for determining when and how to adjust the phase. This is the focus of ongoing work.

⁴The actual mechanism for adjusting the phase is simple. In the worst case, the station remains awake through the union of an old and a new sleep-wake cycle.

neighbor	phase estimate	status	
ID	—	<i>unknown</i>	no phase information
ID	ϕ	<i>idle[timeout=N]</i>	no traffic in N intervals
ID	ϕ	<i>pending</i>	pending traffic
ID	ϕ	<i>overload[size=N]</i>	pending traffic from previous intervals
ID	ϕ	<i>unreachable[retries=N]</i>	unreachable for N intervals

Table 1: Values in the neighbor state table

7 Protocol Operation

This section describes the operation of the phase announcement power save protocol in detail, with specific reference to potential implementation in the IEEE 802.11 context.

Transmit queue management is described as if it is synchronized to each station’s awake interval. However, given appropriate locking, enqueued traffic may be transmitted to any receiver (that is not in the *unreachable* state) for which there is a known phase estimate.

7.1 Neighbor State Table

Each station maintains a phase table containing the ID, estimated phase, and status of each station with which it has recently communicated. The values in the state table are defined in Table 1.

The descriptions below assume a logically separate FIFO packet queue for each neighbor to which the sender will transmit packets. Packets are placed on these queues by the routing subsystem, which determines the appropriate next hop neighbor for each packet.

For simplicity, transmit queue management is described as operating synchronously with the power save protocol. Given appropriate locking and transmit schedule management (see below), packets can be forwarded to any reachable neighbor for which the phase estimate suggests the likelihood of successful transmission.

As packets are appended to the transmit queues, the neighbor state table is updated according to the following rules.

- If a packet is queued for a neighbor for which there is no information, an entry with status *unknown* is added to the state table.
- If a packet is queued for a neighbor which is in the *idle* state, its status is changed to *pending*.
- If a packet is queued for a neighbor which is in any other state (*pending*, *overload*, *unreachable*), its status does not change.

At the beginning of its awake interval, a station updates the status of each neighbor in the state table. This allows *idle* and *unreachable* neighbors to be purged from the neighbor table.

- If a receiver is in the *pending* or *overload* state and has no packets on its transmit queue, its status is set to *idle*, initializing the *idle timeout* value.

In this case, the sender successfully transmitted all pending traffic in the previous interval and there is no new traffic for this neighbor. In order to efficiently handle flows with packet inter-arrival times that are larger than the power save interval, the sender continues to maintain a phase estimate for neighbors to which it has recently forwarded traffic. To avoid wasting bandwidth resources maintaining phase information for an idle link, this information eventually times out of the state table, as described below.

- The *timeout* value of each neighbor that is in the *idle* state is incremented and any neighbor whose timeout exceeds the idle timeout limit is deleted from the table. By definition, such a receiver has no packets in its transmit queue.
- The *retries* value of each neighbor that is in the *unreachable* state is also incremented. If the number of retries exceeds the retry limit, the neighbor is deleted from the table. Packets remaining its transmit queue may be treated as undeliverable. Depending on the routing protocol and its implementation, a route recovery procedure may be initiated. In this case, packets on the transmit queue may be “salvaged” by the routing subsystem and forwarded to an alternate next-hop neighbor.

7.2 ATIM-0 Window

The first part of each station’s awake interval is the ATIM-0 window. The ATIM-0 message is based on the contents of the neighbor state table. The station broadcasts an ATIM-0 message containing the receiverID, estimated phase difference and status of each station in its receiver table.

The ATIM-0 message is sent as close to the beginning of the station’s awake interval as possible. That is, the ATIM-0 message should be prioritized as control traffic at the MAC layer. In the case of IEEE 802.11, this assigns to ATIM traffic a minimum contention window.

In addition to the neighbor table data, the ATIM message also contains a timestamp. In keeping with the requirement for asynchronous operation, this is a relative, rather than an absolute, timestamp. It indicates the time, relative to the beginning of the station’s awake interval, at which the ATIM message was transmitted.

In the case of IEEE 802.11, a timestamp can be generated using mechanisms similar to those used in TSF beacon generation, which records the clock value at the time at which the frame is transferred to the physical (PHY) layer. The ATIM message can include both this clock value and the clock value at the beginning of the awake interval, allowing the receiver to compensate for channel acquisition delay.⁵

Each receiver of the ATIM-0 message records the time, relative to the beginning of its own awake interval, at which it received the frame. Again, for IEEE 802.11, mechanisms similar to those used for TSF beacon processing can be used to record the ATIM-0 message arrival time. The receiver uses these timestamps to observe the relative phase difference between itself and the sender.

If the ATIM-0 message indicates that the phase of the receiver is *unknown*, the receiver adds the ID of the sender to its own neighbor state table, recording the observed phase difference and setting the status to *idle*. In this case, the receiver also provide the sender with phase information, by transmitting an ATIM-ACK message to the sender. Otherwise, the ATIM-0 message contains the sender's current estimate of the phase difference between the sender and receiver. The receiver can compare this value with the observed value and with its own recorded estimate of the phase difference. If the actual phase difference has not changed, these values should all be in very close agreement and the receiver's phase estimate need not be altered. (Statistical methods can be used to filter small variations, if needed.)

If the phase difference has changed, the phase difference observed at the receiver will differ from the estimate given in the ATIM-0 message. In this case, the receiver updates the phase estimate recorded in its neighbor state table and informs the sender by transmitting an ATIM-ACK message to the sender.⁶ Like the ATIM-0, the ATIM-ACK is also sent as quickly as possible, taking priority over other data traffic waiting to be sent. Note that the the ATIM-ACK is transmitted only if incorrect information is detected in the senders neighbor state table, in order to minimize traffic being sent in this critical window. Some implications of this choice are discussed in the following section.

After sending the ATIM-0, the sender delays for interval WAIT-0, accumulating ATIM-ACK messages and recording new and (especially) updated phase estimates for each receiver in its neighbor state table. The optimal value of WAIT-0 is not yet clear, though $\frac{n}{n+1}\epsilon$ seems reasonable. There needs to be enough time to accumulate updated phase information before sending data traffic, because retransmissions to unreachable nodes wastes

⁵If necessary, a timestamp of zero could be used. This would require some statistical processing at the receiver to compensate for variable channel acquisition delays when establishing an estimate.

⁶In addition to signaling deliberate change of phase, this also provides a primitive mechanism for correcting differences in clock rate between stations.

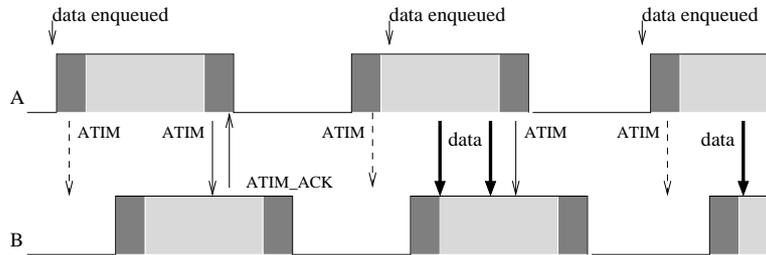


Figure 2: Phase announcement and data transfer.

time and bandwidth. But the waiting time also reduces the time available to send data traffic.

Any pending broadcast traffic must also be transmitted during both the ATIM-0 and ATIM-1 intervals, so that it is received by all neighbors.

7.3 Transfer Interval

Based on the phase information obtained from the ATIM-ACK responses, the sender can calculate the available transfer window for each neighbor. After the WAIT-0 interval, the station selects the order in which it will transmit packets to neighbors for which it has a phase estimate. The most straightforward ordering is “earliest deadline first”. That is, receivers are sorted in increasing order of the time remaining in their transfer windows⁷. The data traffic is transmitted using whatever basic mechanism is provided by the MAC layer; this is DCF in the case of IEEE 802.11. Protocol operation is shown in Figure 2. There is scope for significant optimization here, based on QoS techniques for admission control and maximization of various time-value functions that are widely used in the real time systems community.

The transmit order computed above is not a “feasible schedule” in the TDMA or real time sense. It may happen that it is not possible to send all the pending traffic to one or more receivers. Before sending a packet, the station checks whether the intended receiver is still awake, based on its estimated phase. If the receiver is no longer awake, the sender sets its status to *overload* and leaves the packet in the transmit queue. The sender then continues with the next receiver in the calculated transmit order.

If packet transmission fails, the sender sets the receiver status in its neighbor table to *unreachable*, initializing the retry counter. In the case of IEEE 802.11, transmit failure means that the transmitter does not receive a timely CTS or ACK after the retransmission interval defined in the protocol.

⁷Some special case is needed to handle neighbors which have two disjoint transfer windows.

The sender leaves the packet in the receiver's transmit queue and proceeds to the next receiver in the transmit order.

There are three reasons such a transmit failure could occur:

1. Neither the phase nor the connectivity has changed and the transmission was lost due to transient failure, such as interference. In this case, it is probably as well to move on to another receiver during this interval and try again with the problematic receiver during the next interval.
2. The transmitting and receiving stations still have connectivity, but their relative phase has changed such that the receiver is no longer awake in the expected interval. There are three reasons why the transmitting station failed to discover this in the ATIM-0 interval.
 - The receiving station was not or is longer awake during the transmitting station's ATIM-0 interval and so did not receive the ATIM-0 message containing the outdated estimate. In either case, the receiving station will, by the overlap property, be awake and receive an ATIM-1 (see below) message sent during the ATIM-1 interval and the phase information will be updated at that time.
 - It is possible that the receiving station was awake during the transmitting station's ATIM-0 interval, but either the ATIM-0 message containing the outdated phase information or the ATIM-ACK containing the updated value was lost due to collision, congestion or transient failure. In this case, the phase information will not be updated in the ATIM-1 interval, but will (probably) be corrected in the next ATIM-0 interval.
 - A much less likely possibility is that there is a small error in the phase estimate and the transmission was attempted at the very end of the transfer window. Such race conditions are unlikely to persist.
3. The receiving station is no longer in range of the transmitting station. Subsequent attempts to retransmit the packet will also fail and when the retry count is exceeded, appropriate recovery measures will be taken. A disadvantage of this scheme is that a sender cannot declare a neighbor unreachable until it has at least attempted to obtain up-to-date phase information in both the ATIM-0 and ATIM-1 intervals. The problem of detecting and responding to connectivity changes is an important one in ad hoc networking. There is considerable variation in proposed solutions, as there is no standardization in this cross-layer signaling.

Note that it is not possible to detect the case in which the phase has changed such that the data traffic is successfully sent, despite the out-of-date phase estimate, but the receiver is not awake during the ATIM-0 interval and has not seen the incorrect phase estimate. The phase estimate will be corrected in the ATIM-1 interval.

7.4 ATIM-1 Window

The ATIM-1 interval begins for the last ϵ portion of the station's awake interval. At this time, the station stops transmitting data messages, setting the status of each receiver for which it still has pending traffic to *overload*.

The ATIM-1 message is defined and processed in the same way as the ATIM-0 message. The sender accumulates ATIM-ACK messages over a WAIT-1 interval, analogous to the WAIT-0 interval discussed above. Other non-ATIM broadcast traffic is also be retransmitted in the ATIM-1 interval.

Because the WAIT-1 interval covers most of the ATIM-1 interval, the sender cannot transmit much traffic based on new phase information obtained from ATIM-ACK messages received during the WAIT-1. This means that traffic destined for neighbors for which new phase information was just obtained will probably not be transmitted until sender's next awake interval.

This additional latency is incurred only when the phase information is unknown or has changed. Assuming uniform phase distribution and stable phases, the average size of the transfer window is not affected. Once a phase estimate has been established, traffic may be sent at any point in the transfer window. Nevertheless, traffic to neighbors whose awake interval overlaps the ATIM-1 interval is more prone to disruption than to neighbors whose awake interval coincides with the sender's ATIM-0 window. The ATIM traffic must therefore be prioritized, in so far as possible, at the MAC layer. Contention problems in the ATIM windows is discussed in more detail below.

8 Phase Adjustment

It is clear that the phase distribution among the stations in a region or along a forwarding path will have a significant impact on the effective bandwidth and latency experienced by a flow. By appropriately managing phase distribution, it is possible to mitigate the effects of power save on network performance.

Given the highly flexible routing available in ad hoc networks, the effects of phase distribution can be taken into account in the route selection process. In addition, stations can easily change their phase, making it possible for the power save protocol to adapt to QoS sensitive flows.

8.1 Phase-aware Route Selection

If phase information is included at each hop in the route discovery process, the power save-related delay of the route can be evaluated by the destination and used to select a preferred route. For example, for a delay-sensitive flow, a preferred route is one where for each intermediate node, the transfer window of the downstream link follows the transfer window of the upstream link. However, this metric cannot take contention from nearby links into account, nor is it clear how it might be integrated with other metrics, like path length or residual energy. Moreover, such a route is not guaranteed to exist, particularly in a sparse network.

8.2 “Overloaded” Receivers

If a node cannot send all of its pending traffic to a receiver, the receiver’s status is set to *overload* and the value of *size* is incremented. One obvious cause for overload is an infelicitous phase relationship between the station and the receiver. There may be insufficient transfer interval between the awake interval of the sender and receiver or there may be a high level of contention during the overlap.

One advantage of this asynchronous scheme is that it is relatively inexpensive for nodes to change their phase. The node simply stays awake for the union of its old and new awake intervals for one interval, then maintains only the new phase. The node can update the phase estimate information it includes in its ATIM message, so that its receiving nodes will only need to update their own state tables. Other nodes traffic must discover the phase change via the ATIM/ATIM-ACK mechanism.

If a link is persistently in the *overload* state, that is, such that the *size* of the overload on the affected link continues to increase, it may become necessary to adjust the phase of the sender and/or the receiver. Despite the low bandwidth overhead associated with phase adjustment, the node that changes phase must spend extra time awake, to help ensure that no correspondent attempts to send traffic based on stale phase information (see the ATIM-1 problem above). The energy overhead of the phase change is potentially significant and argues against too frequent phase adjustment.

8.3 Phase Adjustment Strategies

Phase adjustment mechanisms are still the least well understood part of the proposed power save protocol; three possible approaches are presented below. Distributed computation of an optimal phase distribution, in addition to being difficult, may not be particularly rewarding in a highly dynamic network⁸. A key requirement for any strategy is that it avoid too frequent

⁸It is almost certainly possible to create a topology and collection of flows for which there is no good solution.

adjustment. It is particularly important to avoid feedback loops, where for example, overload on link A-B results in a phase adjustment that leads to overload on link B-C (or even C-D), which in turn leads to a phase adjustment resulting in overload on link A-B.⁹ The focus will therefore be on heuristic approaches which can mitigate the most significant bottlenecks.

There are two components to phase adjustment. Both the sender and receiver on an overloaded link¹⁰ must first determine whether they should perform adjustment, and then determine the amount by which their phases should change. If a node has more than one active link, either as a sender or a receiver, phase adjustment requirements of the multiple links must be reconciled.

The simplest approach is probabilistic. When a sender indicates an a receiver in the *overload* state, both the sender and receiver probabilistically decide whether to adjust their phase. To avoid feedback effects, the probability of adjustment should remain low following a phase adjustment, then slowly return to a default level. The amount by which each node changes its phase can also be set randomly, although any adjustment should be large enough to be meaningful. In this case, it is simple to reconcile phase adjustment on multiple links: a probabilistic decision is made if any link is overloaded, but the size of the change is independent of which link(s) led to the adjustment. The effect of conditioning phase adjustment on other factors, such as node density or the number of overloaded links, can be more carefully explored in simulation.

An alternative approach is to attempt to equi-distribute the awake intervals of active links based on observed contention in each node's neighborhood. Non-active (i.e. non forwarding) nodes note the overlap intervals of their active neighbors and the times of highest contention. These nodes each adjust their own phase so that their awake interval is likely to experience as little contention as possible. When an non-active node becomes active, it is therefore less likely to cause or experience overload.

Another, more complex alternative is for the sender and receiver to attempt determine an optimal, rather than a random, adjustment. In this case, it is useful to examine whether the overload is more likely due to a short transfer interval or to high levels of contention during the overlap interval. That is, if the transfer interval between the sender and receiver is longer than some threshold (i.e. half of the awake interval), then the overload is likely to be due to contention and the sender and receiver should both adjust their phase by about one-half to avoid contention. In this case,

⁹To some extent, there is no point in having a phase adjustment strategy that "tries too hard". Experience [14], [5] highlights capacity issues in multihop IEEE 802.11 networks — these networks are likely to suffer overload in any case.

¹⁰Consider the case where the sender and receiver have substantially overlapping awake intervals, but there is significant contention caused by nearby receivers. In this case, not only the sender, but also some receiver must adjust in order to reduce contention.

the resulting transfer interval should be reduced to below the threshold, to avoid feedback loops. If the transfer interval is short, the sender can adjust its phase to have greater overlap with the sender. This strategy becomes much more difficult to define clearly in the case of multiple flows, however.

9 Problems and Limitations

This section presents four limitations of the proposed protocol. These issues must be carefully considered in simulation work, but the report also suggests that the protocol has a good chance of comparing favorably with existing proposals.

- The most obvious limitation of the proposed scheme is that the proportion of time that a station spends in the sleep state is fundamentally limited to $(.5 - \epsilon)$, i.e. less than 50%, even in the case of an idle network. Unlike protocols such as Span, the proportion of time spent in the sleep state is independent of the network density.

The proposed protocol has every chance of comparing favorably to a protocol such as Span, where some proportion of nodes must also remain awake to act as coordinators and all the nodes must remain awake though the ATIM interval. Moreover, the proposed protocol has no overhead associated with maintaining the coordinator backbone and does not have to take care to rotate the coordinator functionality.

In the case of an idle network, applying a similar sleep-wake pattern – that is, $N + e$ intervals running the power-save protocol, $N - e$ intervals sleeping – might also be possible. Such a scheme has a very high latency (seconds) when traffic is initiated, but may be appropriate for some application environments.

- The proposed protocol also enforces a bandwidth limitation, such that only a factor $0.5 + \epsilon$ of the available bandwidth is available on a given link. This problem is of particular concern given the well known [14] limitations of IEEE 802.11 in this respect. However, the extended contention seen in CSMA/CA based multihop networks argues for adapting a station's sleep wake schedule to find "minimum-contention" transfer windows. Nevertheless, this may prove a substantial limitation, especially in the case of a network supporting a single flow.
- In order to maximize energy savings, the length of the ATIM window (ϵ) must be minimized. There are four protocol elements which must be accommodated in the ATIM window, all of which will tend to increase its length.

1. The ATIM traffic is transmitted during the ATIM window. Due to the asynchronous nature of the power save protocol, a station's ATIM windows may overlap with both ATIM and data transfer windows of nearby stations. ATIM traffic must therefore have priority over data traffic. Nevertheless, the ATIM window must be long enough to account for the possibility of deferring the ATIM message for at least the length of a single data message, in addition to the possibility of contention between ATIM traffic.
2. The WAIT interval, during which a station accumulates ATIM-ACK responses. Even if a station already has phase information for a neighbor, the time and retransmission cost of a failed data transmission makes it important to obtain updated phase information before attempting to transmit. While it is possible to require an ATIM-ACK from each neighbor with pending traffic, the repeated ACK implosion, serving only to confirm the correctness of an existing estimate, is unlikely to be efficient.
3. Broadcast data must also be transmitted in both ATIM windows. In addition to the increase in bandwidth required, this increases the amount of traffic which must be transmitted during the ATIM window.
4. Some data traffic may need to be transmitted in the ATIM window. In the case of two stations which are precisely out of phase, the transfer window will be equivalent to the ATIM windows. This traffic cannot be transmitted until after the WAIT interval, which must therefore occupy less than the full ATIM window. Such poor phase distribution can be avoided using phase adjustment.

If contention in the ATIM window proves problematic, there are number of possible ways to reduce the amount of ATIM traffic. For example, the ATIM message may not be transmitted in every interval. Alternatively, nodes can obtain phase estimates for neighboring nodes by eavesdropping. Nevertheless, it seems likely that phase adjustment will be important in distributing ATIM and data transfer windows of nearby links.

In general, the proposed mechanism is most appropriate for networks which are primarily intended to support persistent data flows and relatively little broadcast traffic. This not only minimizes the requirement for ATIM traffic, but also enhances effectiveness of phase adjustment in response to congestion.

- Power save mechanisms tend to introduce large latencies into the system. Simulations indicate that 802.11 power save is particularly problematic in this respect, in part because the network is synchronized

to the beacon interval. Span reduces packet latency significantly by maintaining a routing backbone over which the packet can be forwarded with minimal delay. In the proposed protocol, an intermediate behavior might be expected: a packet's forward progress can be delayed by the periodic nature of the protocol, but the combination of longer transmit windows and overlapping awake intervals will mitigate the effect.

The problem can also be addressed by incorporating latency information into the route discovery process to discover forwarding paths with favorable overlap patterns for delay-sensitive flows. The more general QoS mechanism allows for stations to adjust their relative phase to reduce latency. Nevertheless, it certainly seems possible to create a topology and collection of flows for which such optimization is impossible.

- Any protocol which provides link-layer acknowledgments is capable of declaring link failure if a timely acknowledgment is not received. Detection of link failure is particularly important for ad hoc routing protocols, which must act quickly to repair route failures.

In principle, the proposed protocol has the additional complication that failure to receive acknowledgment may indicate a change of relative phase, rather than link failure. This means that both ATIM-0 and ATIM-1 messages must fail to receive an ATIM-ACK response before a failure can be suspected.

There is a wide range of behaviors found in existing protocols and simulations and route (i.e. link) failure is defined with varying levels of aggressiveness. In many routing protocol implementations, link failure detection is performed at the network layer (e.g. active or passive acknowledgment or hello messages). Moreover, the problem of link layer failure detection and notification to higher layers is not specific to the work discussed here. There is no 802.11 standard mechanism (much less a link-independent one), for a network interface to report link failure detection to the upper layer.

- Because the station is allowed to transmit data only in certain periods, the proposed protocol will obviously be less responsive to very bursty traffic than unmodified IEEE 802.11. Again, this is a common problem with power saving protocols in which traffic announcements are limited to a specific interval. It is not yet clear how the proposed protocol will perform in this respect, in part because there seem to be no published results on this aspect of power save protocols.

10 Conclusion

This report presents a proposed asynchronous power save protocol for mobile multi-hop wireless networks. This report establishes the following points:

- The proposed solution addresses a significant problem. An effective solution will improve the usability of IEEE 802.11 based interfaces in an ad hoc environment.
- The proposed solution is significantly different from that studied in related work, having the specific advantages of asynchronous operation and density independent performance.
- The proposed protocol may be expected to exhibit energy saving performance which is at least comparable to the best published results, as well as some advantage in simplicity and extensibility.
- The proposed protocol may prove suitable for providing adaptive QoS support to minimize the effects of congestion and reduce latency for delay-sensitive flows.
- The proposed protocol has been described in sufficient detail to permit its implementation in an appropriate simulation environment.
- The report suggests that the proposed work presents a number of avenues for further investigation, dependent on promising results of the initial work.

Due to this favorable expectation, it is proposed to strongly pursue developing this protocol in the ns2 simulation environment, using IEEE 802.11 as the underlying MAC layer. Further development of the phase adjustment mechanism will be necessary to obtain good performance.

It should also be noted that the general problem of determining an optimal phase assignment given flow information seems to be closely related to the bin packing problem (which is computationally hard) and to various problems in distributed schedule assignment.

Notes and acknowledgments

Rev 1.1 updates the definition of the ATIM-0 and ATIM-1 messages, which now have the same format and semantics. This eliminates a great deal of complexity formerly associated with using ATIM-1 as an explicit error detection mechanism.

Thanks to Kyle Jamieson of MIT LCS for helpful discussion.

References

- [1] AGARWAL, S., KRISHNAMURTHY, S. V., KATZ, R. H., AND DAO, S. K. Distributed power control in ad hoc wireless networks. In *Personal and Indoor Mobile Radio Communication (PIMRC)* (2001).
- [2] CHANG, J.-H., AND TASSIULAS, L. Energy conserving routing in wireless ad-hoc networks. In *Proc. of IEEE Infocom* (2000), vol. 1, pp. 22–31.
- [3] CHEN, B., JAMIESON, K., BALAKRISHNAN, H., AND MORRIS, R. Span: An energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks. *ACM Wireless Networks Journal* 8, 5 (Sept. 2002), 481–494.
- [4] CHLAMTAC, I., PETRIOLI, C., AND REDI, J. Energy-conserving go-back-N ARQ protocols for wireless data networks. In *Proc. of IEEE International Conference on Universal Personal Communication (ICUPC'98)* (Oct. 1998).
- [5] Personal communication. Christian Tschudin (University of Basel), Oct. 2002.
- [6] EBERT, J.-P., BURNS, B., AND WOLISZ, A. A trace-based approach for determining the energy consumption of a WLAN network interface. In *Proc. of European Wireless* (Feb. 2002), pp. 230–236.
- [7] FEENEY, L. M., AND NILSSON, M. Investigating the energy consumption of a wireless network interface in an ad hoc networking environment. In *Proc. of IEEE Infocom* (Apr. 2001).
- [8] GOMEZ, J., CAMPBELL, A. T., NAGHSHINEH, M., AND BISDIKIAN, C. Conserving transmission power in wireless ad hoc networks. In *Proc. of IEEE Conference on Network Protocols (ICNP'01)* (Nov. 2001).
- [9] GRONKVIST, J. Assignment methods for spatial reuse TDMA. In *Proc. of Workshop on Mobile and Ad Hoc Networking and Computing (MobiHoc'00)* (2000).
- [10] HEINZELMAN, W. R., KULIK, J., AND BALAKRISHNAN, H. Adaptive protocols for information dissemination in wireless sensor networks. In *Proc. of 5th Annual International Conference on Mobile Computing and Networking (MobiCom'99)* (1999), pp. 174–185.
- [11] JOHANSSON, P., LEE, Y. Z., GERLA, M., AND KAZANTZIDIS, M. Bluetooth: an enabler of personal area networking. *IEEE Network, Special Issue in Personal Area Networks* (Sept. 2001).

- [12] KRAVETS, R., CALVERT, K., AND SCHWAN, K. Power-aware communication for mobile computers. In *Proc. of Sixth International Workshop on Mobile Multimedia Communications (MoMuc-6)* (1999).
- [13] KRAVETS, R., AND KRISHNAN, P. Power management techniques for mobile communication. In *Proc. of 4th Annual International Conference on Mobile Computing and Networking (MobiCom'98)* (1998).
- [14] LI, J., BLAKE, C., COUTO, D. S. J. D., LEE, H. I., AND MORRIS, R. Capacity of ad hoc wireless networks. In *Proc. of 7th Annual International Conference on Mobile Computing and Networking* (2001), pp. 61–69.
- [15] NARAYANASWAMY, S., KAWADIA, V., SREENIVAS, R. S., AND KUMAR, P. R. Power control in ad-hoc networks: Theory, architecture, algorithm and implementation of the COMPOW protocol. In *Proc. of European Wireless* (Feb. 2002), pp. 156–162.
- [16] RAMANATHAN, R., AND HAIN, R. Topology control of multihop wireless networks using transmit power adjustment. In *Proc. of IEEE Infocom* (2000), vol. 2, pp. 404–413.
- [17] RODOPLU, V., AND MENG, T. H.-Y. Minimum energy mobile wireless networks. *IEEE Journal on Selected Areas in Communications* 17, 8 (Aug. 1999), 1333–1344.
- [18] SANCHEZ, M., MANZONI, P., AND HAAS, Z. J. Determination of critical transmission range in ad-hoc networks. In *Proc. of Multiaccess Mobility and Teletraffic for Wireless Communications Workshop* (Oct. 1999).
- [19] SINGH, S., AND RAGHAVENDRA, C. PAMAS - power aware multi-access protocol with signalling for ad hoc networks. *ACM Computer Communication Review* (July 1998).
- [20] SINGH, S., WOO, M., AND RAGHAVENDRA, C. S. Power-aware routing in mobile ad hoc networks. In *Proc. of 4th Annual International Conference on Mobile Computing and Networking (MobiCom'98)* (1998), pp. 181–190.
- [21] SINHA, P., SIVAKUMAR, R., AND BHARGHAVAN, V. CEDAR: a core-extraction distributed ad hoc routing algorithm. In *Proc. of IEEE Infocom* (1999), vol. 1, pp. 202–209.
- [22] STEMM, M., GAUTHIER, P., HARADA, D., AND KATZ, R. H. Reducing power consumption of network interfaces in hand-held devices. In *Proc. of 3rd International Workshop on Mobile Multimedia Communications (MoMuc-3)* (Sept. 1996).

-
- [23] WATTENHOFER, R., LI, L., BAHL, P., AND WANG, Y.-M. Distributed topology control for wireless multihop ad-hoc networks. In *Proc. of IEEE Infocom* (Apr. 2001), pp. 1388–1397.
 - [24] WIESELTHIER, J., NGUYEN, G., AND EPHREMIDES, A. On the construction of energy-efficient broadcast and multicast trees in wireless networks. In *Proc. of IEEE Infocom* (2000).
 - [25] WOESNER, H., EBERT, J.-P., SCHLAGER, M., AND WOLISZ, A. Power saving mechanisms in emerging standards for wireless LANs: The MAC level perspective. *IEEE Personal Communications* 5, 3 (June 1998), 40–48.
 - [26] WU, J., DAI, F., GAO, M., AND STOJMENOVIC, I. On calculating power-aware connected dominating sets for efficient routing in ad hoc wireless networks. *IEEE/KICS Journal of Communication Networks* 4, 1 (Mar. 2002), 59–70.
 - [27] XU, Y., HEIDEMANN, J., AND ESTRIN, D. Adaptive energy-conserving routing for multihop ad hoc networks. Tech. Rep. 527, USC/Information Sciences Institute, Oct. 2000.
 - [28] XU, Y., HEIDEMANN, J., AND ESTRIN, D. Geography-informed energy conservation for ad hoc routing. In *Proc. of 7th Annual International Conference on Mobile Computing and Networking* (July 2001), pp. 70–84.