A QoS Aware Power Save Protocol for Wireless Ad Hoc Networks

Laura Marie Feeney
Swedish Institute of Computer Science
Box 1263, SE 164 29 Kista, Sweden
Email: lmfeeney@sics.se

Abstract—**This paper describes a power save protocol for ad hoc networks. The protocol is largely independent of the details of the underlying MAC and friendly toward any overlying energy-aware ad hoc routing. A key advantage of the protocol is that it is fully asynchronous. Each station independently establishes a periodic sleep/wake cycle. 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 so as to maximize value with respect to some QoS function. A station can also adjust its phase relationships to avoid contention and increase effective bandwidth available to a flow, as well as reduce latency.**

I. Introduction

This paper presents ongoing work developing a new power save protocol for general purpose mobile multi-hop wireless networks. Each node maintains a fixed sleep/wake cycle independently of its neighbors, eliminating the need for global synchronization and providing for a balanced distribution of energy savings. The protocol is built around a phase discovery mechanism that allows two neighbors wishing to communicate to estimate the relative phase difference between their sleep/wake cycles. This allows a sender to determine the available transfer window for each receiver and appropriately schedule transmissions. The phase discovery mechanism supports easy phase adjustment, allowing nodes to adapt their transmission schedules to contention or to high priority flows. This kind of adaptivity is 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 sleep/wake cycle is defined such that certain guarantees can be made about the properties of transfer windows. Nevertheless, the effective capacity in a region is highly dependent on the phase distribution. A sender may be persistently unable to transmit all the traffic for a given receiver if the transfer window is too short, or if too many receivers have overlapping transfer windows, or if there is contention due to traffic on another link during the transfer window.

The phase discovery mechanism can be used to seamlessly adjust the phase of a node so as to increase the effective capacity of a link; for example, by finding a maximal transfer window for a high priority flow. For the current study, a simple randomized phase adjustment will be used. However, a potential strength of the phase adjustment approach is its extensibility and support for more complex adaptivity.

Section II presents the motivation and requirements driving this work. Section III presents related work. The protocol itself is described in section IV and some potential problems are discussed in section V. The paper concludes with discussion of ongoing work in developing the protocol and studying it in simulation.

II. Motivation and requirements

The network interface is a significant source of energy consumption in portable wireless devices [1]. At any given time, the energy consumed by an interface depends on its operating mode. A sleeping interface can neither transmit nor receive traffic and as a result, it consumes very little energy. To be able to transmit or receive, an interface must explicitly transition to the wake state, which requires both time and energy. An interface that is in the wake state can transmit or receive data at any time; if it is neither transmitting or receiving, is said to be idle. In all of these states, an interface consumes significantly more energy than it does in the sleep state, due to the number of circuit elements that must be powered. There has recently been interest in investigating the energy consumption of commercially available network interfaces.

Largely because of its ubiquity, IEEE 802.11b is particularly interesting. Measurements [2] [3] [4] [5] show that an *idle* network interface can consume 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 while sleeping (66mW - 130mW) [1]. A rough calculation based on data in [4] for Lucent devices suggests that an interface sending ten 128-byte broadcasts per second and receiving the same from each of five neighbors consumes only about 1% more energy than an idle interface.

In short, sending and receiving are not the dominant source of energy consumption: being awake and ready to send or receive traffic is. To reduce energy consumption, an interface must therefore spend as much time as possible in the sleep state. In this view, a power save protocol is a coordination mechanism to arrange that stations that want to exchange traffic are awake at the same time.

For the case of a multihop wireless ad hoc network, this is a challenging research problem. In a wireless infrastructure network, communication is mediated by a preconfigured access point, which is generally assumed to have no energy consumption constraints. The access point can therefore remain constantly awake and clients spend most of their time sleeping, waking up periodically to receive buffered traffic. This kind of coordination is difficult to emulate without this kind of infrastructure. The network topology may be highly dynamic and

[1] Measurements vary somewhat depending on manufacturer and model; the values cited are representative. (Some interfaces support more than one sleep mode.)

there is no resource rich, centralized element around which to construct a power save mechanism. The fact that nodes cooperatively form the routing infrastructure imposes a further requirement to maximize system lifetime by ensuring that energy consumption is balanced across the nodes in the network.

The goal of this work is to develop a practical power save protocol that operates effectively in an infrastructureless environment. The protocol is intended for use in general purpose ad hoc networks; it is specifically not directed to the special case of a sensor network.

To maximize the applicability of the work, the protocol should depend as little as possible on the details of a specific MAC protocol and should be able to work with any kind of collision avoidance mechanism. (Like most research in this area, the protocol is based on IEEE 802.11.) Similarly, the protocol should place as few requirements as possible on the overlying routing protocol. Of particular importance is the interaction with energy aware routing. Because the power save protocol operates at a low layer, it must take care to avoid introducing inappropriate feedback effects to higher layer energy management.

Most importantly, the protocol must provide support for QoS functionality. Because an ad hoc network has variable link quality, dynamic topology and complex interference across multiple links and between disjoint flows, the ability to adapt easily to the unpredictable environment is the essential feature. To this end, the coordination mechanism that is responsible for energy management must also be aware of network QoS constraints in its scheduling of the network interface.

In addition to reducing energy consumption, any solution must address the following performance goals:

- Minimize adverse impact on capacity, throughput, packet latency and route latency.
- Minimize overhead imposed by the energy management scheme.
- Maximize system lifetime by minimizing disparities in energy consumption.

The infrastructureless environment also poses operational constraints on protocols for ad hoc networks. Clearly, the protocol must be capable of localized operation. In this work, it is also suggested that an asynchronous solution is highly desirable. This is discussed in more detail in Section III-C below.

III. RELATED WORK

The first subsection gives a brief overview of energy management in ad hoc networks, focusing on elements above the MAC and hardware layers. The following subsections consider particularly relevant work – IEEE 802.11b[6], Span[2], and AFECA [7] – in more detail.

A. Overview

A wide range of power control techniques are used in wireless communication to reduce interference and conserve energy. This strategy has particularly interesting applications in a multihop network. Because the transmission power required to achieve a given SNR at a receiver increases exponentially with distance, the total energy required to transmit a packet via a relay node may be less than the energy required to transmit it directly. The formulation in [8] explicitly takes into account both the variable transmit power and the fixed cost of receiving.

This observation forms the basis for several research problems. The minimum energy broadcast problem is to find a set of relays and transmit powers such that a broadcast transmission from a given source node is rebroadcast to every other node in the network with minimum total energy cost [9]. The topology control problem is to find a set of transmit powers that obtain a minium power topology, while still maintaining network connectivity. This is addressed using heuristic [10], propagation modeling [8], proactive [11], and directional [12] methods. Power control also increases spatial reuse, but the presence of widely varying transmit powers in a multihop network greatly complicates collision avoidance. Low power transmissions cannot be sensed by distant nodes, which may then initiate transmissions using sufficient power to disrupt ongoing transmissions. This problem is addressed using adaptive power [13] and network-optimal power [14] [15].

A minimum power topology can be used to find a minimum energy route for a given packet, but this greedy approach does not address another requirement of ad hoc networks. Because nodes forward traffic on each other's behalf, the routing load imposed on a node may cause it to deplete its battery prematurely. Energy aware routing attempts to maximize the network lifetime by taking into account the battery reserves at each node when selecting routes. A number of metrics for evaluating power aware routing metrics are presented in [16]. A linear programming technique which combines energy aware routing with topology control is described in [17] [18]. An alternative approach [19], distributes the traffic load so as to take advantage of charge recovery effects in the battery.

More specialized energy management techniques have also been developed, particularly for sensor networks. Sensor networks are usually modeled as dense, low mobility networks comprised of largely interchangeable nodes, all of which are participating in a common data gathering activity. Power save protocols such as [7], discussed in detail below, leverage some of these assumptions. Other energy saving techniques for sensor networks are closely tied to the sensor data processing itself. One example is [20], in which data fusion is used to minimize the amount of sensor data that is forwarded.

B. 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 [6] standard is a common choice for use in ad hoc networking research. The standard includes power save mechanisms for use in both infrastructure (BSS) and infrastructureless (IBSS) operating modes.

IBSS power save is most relevant to ad hoc networking, although there are differences between a multi-hop wireless network and an IBSS, where each station explicitly discovers and synchronizes itself to a single, connected IBSS.

A synchronized beacon interval is established by the station that initiates the IBSS and is maintained in a distributed fashion. In addition to the 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.

At the beginning of the beacon interval, stations contend, using random backoff, to transmit the synchronization beacon. Once the synchronization beacon has been transmitted, 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 (DTIM) are sent to the appropriate broadcast or multicast address, but are not acknowledged. Only beacons, ATIM's/DTIM's 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 go back to sleep. All other stations remain awake throughout the remainder of the beacon interval. Using ordinary IEEE 802.11b DCF access, each station transmits first broadcast and multicast traffic, then any unicast traffic for which an ATIM acknowledgment was received. Traffic which is not transmitted (e.g. due to lack of time in the ATIM window or beacon interval) is announced in successive beacon intervals until it is eventually discarded.

Although power save is part of the IEEE 802.11 standard, there appear to be few published results regarding its effectiveness. A simulation study described in [21] examined the effectiveness of the power save protocol for a fully connected eight-node IBSS. The experiment measured throughput and sleep time for a variety of beacon intervals, ATIM window lengths and offered loads. The choice of beacon interval is important: short intervals give superior power savings, but at the cost of significantly reduced throughput. For a wide range of beacon intervals, throughput is maximized when the ATIM window occupies about 25% of the beacon interval. As a general observation, the authors suggest that "if we were to sacrifice about 10% in throughput, we could save up to 30% energy". However, such savings are obtained only at quite moderate loads; as offered load increases from 15% to 30%, the savings declines substantially.

Simulations described in [2] also study the performance of IEEE 802.11 power save, slightly modified for a multihop ad hoc environment. In this case, the IEEE 802.11 power save protocol had little impact on energy consumption, while significantly increasing packet latency. The authors suggest that this is partly due to the use of a geographic routing protocol that required periodic broadcasts, something that the IEEE 802.11 protocol handles poorly.

C. *Dominating techniques*

The work most closely related to the proposed protocol is Span [2].

Span is one of several [22], [23] ad hoc networking protocols based on the notion of a dominating set. In Span, "coordinators" — a group of nodes that form a connected dominating set over the network — do not sleep. Non-coordinator nodes follow a synchronized sleep/wake cycle, exchanging traffic using a algorithm based on the beaconing and traffic announcement methods of IEEE 802.11 IBSS power save. The routing protocol is integrated with the coordinator mechanism so that only coordinators forward packets, acting as a low latency routing backbone for network. Span is intended to maximize the amount of time nodes spend in the sleep state, while minimizing the impact of energy management on latency and capacity.

The set of coordinators is determined using a localized algorithm intended to approximate a minimal, capacity preserving set of coordinators. Nodes periodically wake up and exchange neighbor information, then schedule a coordinator announcement, using an adaptive backoff algorithm. Nodes with high connectivity and energy reserves announce themselves more quickly than less effective ones, which volunteer later and only if they are still needed to obtain the dominating set. Rotating the coordinator role in this way tends to balance nodes' energy reserves, even in the case of initially unequal reserves.

Simulation using ns-2 suggests that Span provides about 50% energy saving, with little impact on throughput, latency and packet loss. Rotation of the coordinator role equalizes energy consumption and the time to first node failure increases 50% and the network half-life doubles. The results also support the informal calculation in section II. Even when Span is used to limit idle energy consumption, sending and receiving traffic accounts for well under 10% of the total energy consumed.

The synchronized nature of the Span protocol reveals a major limitation of this approach. Both the beaconing in the underlying 802.11 power save protocol and the coordinator election require synchronization. Coordinator election is based on knowledge of the local topology, based on periodic broadcast neighbor discovery. This cannot be mediated by the coordinators because nodes have to be awake simultaneously to determine their connectivity.

Span's global heartbeat requires two kinds of synchronization. The first ensures that the stations' oscillators tick at the same rate. (The IEEE 802.11 standard specifies hardware tolerance of 10^{-4}.) This is fairly straightforward. The second ensures that the stations are synchronized in phase. This is a 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 beacon interval for the IBSS and "subsequent" stations explicitly associate themselves with exactly one IBSS and synchronize themselves to it. This kind of approach can lead to the "parking lot problem" — problematic race conditions that occur among a group of devices are turned on (more or less) simultaneously. Nevertheless, this method works well for scenarios in which a master station for the network can be conveniently designated and all the other stations can be configured to recognize that master. In effect, it requires that all nodes must be initialized together within the same well-connected cloud or that there is some mechanism for identifying the "right" cloud for a node to associate with.

This limitation is especially unfortunate because the ad hoc networking model is specifically 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[2].

[2] Assuming they belong to the *same* military, of course.

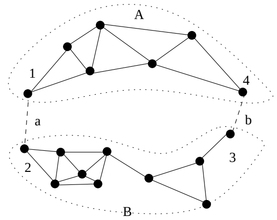

Fig. 1. This kind of connectivity might be caused by a building or hill in the middle of the "ring".

In order to merge two (or more) networks having different phases, some mechanism must be developed which allows the networks to discover and synchronize with each other. Though soluble, this kind of distributed consensus problem is non-trivial. As a *simple* illustration, consider the complexity involved when two connected clouds merge as in Figure 1. Suppose link *a* between nodes 1 and 2 is created and node 2 issues commands to change the phase of cloud *B* to that of cloud *A*, but before the message has propagated to node 3, the link *b* between nodes 3 and 4 is created and node 4 begins to issue commands to synchronize cloud *A* to the phase of cloud *B*. The amount of complexity and overhead required to address this issue is a strong argument for an asynchronous solution.

D. Adaptive techniques

Another interesting power save protocol is the adaptive fidelity energy conservation algorithm (AFECA)[7]. While the protocol can operate in conjunction with any ad hoc network, it is most suitable for a sensor network.

In the basic energy conservation algorithm (BECA), 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.

AFECA is designed to work in conjunction with an on-demand routing protocol and although the power save protocol operates asynchronously, it has strong timing dependencies on the underlying routing protocol. The fundamental interval is the listening interval, which is matched to the route discovery (RREQ) retry interval for the routing protocol. If the sleep interval is some integral multiple k of the listening interval, then it will take at most $k+1$ retries until any given neighbor receives a broadcast RREQ. Once the neighbor receives the RREQ, it transitions to the active state. If the timeout for the active state is greater than the retry interval, potential 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. While packets are being forwarded, nodes on the route will remain active and the other nodes will return to the sleep/listen cycle.

AFECA is an extension of BECA. In a dense, uniformly-distributed (i.e. sensor) network, many nodes are logically equivalent with respect to network reachability and sensor coverage. Nodes can therefore adapt by increasing their sleep interval in areas of higher network density, which can be estimated based on the number of neighbors overheard.

Like Span, AFECA has been studied in the ns-2 environment, using IEEE 802.11 as the MAC layer and AODV as the routing protocol. The strategy shows overall energy saving was on the order of 35% - 45%, across a range of traffic loads, with a sleep interval of 10 seconds. There was, however, 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.

While the techniques used in BECA/AFECA can be applied to any kind of ad hoc network, two limitations make this power save protocol most suitable for sensor networks. One limitation is the high overhead for broadcast. A RREQ may be rebroadcast many times, with an increasing level of redundancy each time. If route discovery and repair are rare operations, this is a minor drawback. If the network supports 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 for higher layer protocols. The authors suggest that proactive routing protocols based on periodic table exchange may be vulnerable.

The second limitation is the interaction with energy aware routing. Because nodes that have recently forwarded traffic remain awake, they are more likely to participate early in the route discovery process and are therefore more likely to be designated as forwarding node for additional routes. Depending on traffic patterns, this feedback behavior may lead to unequal distribution of routing load and poorly distributed energy consumption.

This problem is also suggested by simulation results for network lifetime. AFECA increases network lifetimes, especially in extremely dense networks, where the time to last node failure doubles and the network half-life increases by as much as 50%. This 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.

IV. Phase announcement power save mechanism

A. Overview

Having discussed the strengths and weaknesses of a number of power save protocols, this section presents the new protocol. Additional details of the protocol operation can be found in [24].

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

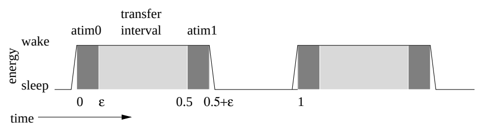

Fig. 2. Sleep/wake cycle.

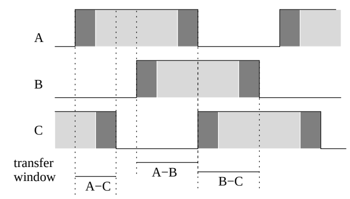

Fig. 3. Transfer windows. $ATIM$ (dark) and transfer interval (light) are shaded.

the traffic announcement is not used as a request to "stand by", but rather as a way for stations to discover their relative phase. Once two stations have determined their relative phase, each can predict when the other will be awake to receive traffic, which is sent using the basic access mechanism of the underlying MAC.

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, which ensures that all the nodes spend about the same proportion of their time in the sleep state. This approach is in distinct contrast with Span, where the coordinator election algorithm takes each node's energy reserves into account, but which also requires frequent re-election of the dominating set in order to rotate the coordinator functionality. While the proposed protocol is less responsive than Span to variation in the nodes' actual energy reserves, it does ensure that the power save protocol will not exacerbate any inequality. It is also better than AFECA, which problematically 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 recevies 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 links can help to isolate flows with QoS requirements from interference.

B. Overlap principle

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

Definition: Let the protocol define 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 $(0.5 - \epsilon)$. Independently scheduled sleep/wake intervals defined in this way have the following useful *overlap property*.

Statement: 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, either the sub-interval $[0..\epsilon]$ or the sub-interval $[0.5..0.5 + \epsilon]$, measured on T, will be completely contained in the awake interval of station R, regardless of the phase difference between them.

Proof: Let $0 \leq \phi < 1$ 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.

C. Phase discovery

Figure 2 shows the sleep/wake cycle followed by all stations. The awake interval of a station is divided into three subintervals: the $ATIM_0$ interval, $[0..\epsilon]$; the transfer interval, $[\epsilon..0.5]$; and $ATIM_1$ interval, $[0.5..0.5 + \epsilon]$. The station sleeps in the interval $[0.5 + \epsilon..1.0]$ [3]

According to the overlap principle, for each station T, at least one of its two $ATIM$ intervals will be completely contained in the awake interval of each of its neighbors. 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 transmits a broadcast traffic indication message, listing stations for which it has pending traffic and its current estimate (if any) of its phase difference with respect to each receiver. Receivers use the traffic indication message to make their own estimate of the phase difference between themselves and the sender. If the receiver's estimate differs from the sender's by more than some threshold, the receiver informs the sender with an $ATIM\_ACK$ message.

The sender uses the phase estimates to calculate the available transfer windows with respect to each receiver. Figure 3 shows the phase relationship among three stations with non-overlapping transfer windows. The sender only attempts to transmit data to receivers that are expected to be awake, using the basic access mechanism provided by the MAC layer. Transmissions are scheduled according to an "earliest deadline first" order, selecting the receiver whose transmission window ends soonest. This strategy is appropriate for the case where all data

[3] To simplify the discussion below, it is assumed that each network interface schedules state transitions with perfect accuracy and switches between states instantaneously. This can be compensated for in protocol implementation.

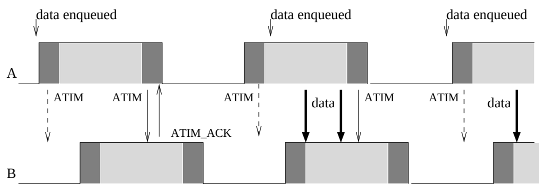

Fig. 4. Phase announcement ($ATIM_1$) and data transfer.

has equal priority. Given an appropriate QoS framework, however, the transmission schedule could be based on maximizing return for any kind of time-value function.

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 $ATIM$ intervals are just the (combined) transfer window for broadcast and multicast traffic. Once the initial phase estimates are obtained, the data delivery is largely decoupled from the $ATIM$ traffic.

Figure 4 shows an example. The first data packet is buffered until an $ATIM\_ACK$ is received in response to the second $ATIM$ message. If the phase estimation is completed too close to the end of the transfer window (as in this case), data transfer is delayed until the next window. Once the phase estimate is established, data transmission depends only on the transfer window. No further $ATIM/ACK$ exchange is required, although the $ATIM$ will continue to announce the current phase estimate, to detect any change in phase. Note that it is not necessary for B to respond to A during the $ATIM_1$ interval on A. Station B can estimate its phase difference with A based on the $ATIM$ message and send the $ATIM\_ACK$ during its own $ATIM_0$ interval instead.

Because stations operate asynchronously, an $ATIM$ interval for a station is not distinguished. The $ATIM$ and $ATIM\_ACK$ will face contention from both $ATIM$ and data traffic. This problem can be mitigated by giving $ATIM$ traffic priority as control traffic, if this is supported by the underlying MAC protocol, as in IEEE 802.11.

D. Phase adjustment

The sleep/wake cycle defined above guarantees that the total length of the transfer windows between any two stations is at least 2ϵ. (For two stations with opposite phase, it is exactly the two $ATIM$ intervals that overlap.) However, this does not translate into any guarantee about the available capacity of the link. In addition to the well-known [25], [26] capacity limitations in a wireless network, link capacity may suffer also if the transfer window is too short, or if the sender has too many receivers with overlapping transfer windows, or if there is too much contention during the transfer window.

Clearly the phase distribution among a group of stations significantly affects the available capacity of a link or region. As a trivial example, Figure 3 above shows a felicitous distribution of phases. Because the transfer windows of the three stations do not overlap, they cannot contend with each other.

One advantage of the phase discovery approach is that a station can seamlessly change its phase with respect to its neighbors. Phase adjustment can be used to increase the effective capacity of a region and reduce latency along a path. Examples of how phase adjustment can be used are presented below.

If a sender is unable to broadcast an $ATIM$ in a timely fashion, it should alter its phase with respect to competing traffic as quickly as possible. If the sender is persistently unable to send other broadcast traffic in its $ATIM$ intervals, it can also alter its phase with respect to competing traffic. In these cases, it is the sender that alters its phase, because there are multiple receviers. To do so, the sender remains awake though the union of the awake intervals of the old and new phases, transmitting a (possibly empty) $ATIM$ in each of its new $ATIM$ intervals.

It may also happen that a sender is persistently unable to transmit all of the traffic pending for some receiver. In this case, either the sender or the receiver can alter its phase to increase the length or timing of the transfer window. For the latter, the sender signals the overload condition to the receiver in the $ATIM$ and the receiver responds with an $ATIM_ACK$ providing a new phase. The receiver must remain awake through the union of the awake intervals of the old and new cycles. To ensure that all of the receiver's other correspondent nodes are informed of the change, the receiver must also announce its new phase via $ATIM$ announcement.

For the case of a delay sensitive flow, more complex phase adjustment may be desirable. A receiver should able to forward all the delay sensitive traffic in the interval in which it was received. That is, the upstream sender's phase should occupy the first half of the receiver's awake interval and the downstream receiver's awake interval should occupy the second half. While it is easy to make such rules in the isolated case, such specific strategies are not generally possible to acheive in network with multiple flows.

Ongoing work is currently based on the principle that when a phase change is needed, a random change in phase is both simple and likelty to be effective. So when a station changes its phase, it does so by an amount uniformly distributed on $[\epsilon, 1-\epsilon]$. This approach has the advantage not only of simplicity, but also of potentially avoiding problematic feedback effects. Based on current work, it remains to be seen whether this simple approach will be effective in practice.

E. Phase error and link failure

Any interface which provides link layer acknowledgements is capable of detecting link failure if a timely acknowledment is not received. Detection of link failure is important for ad hoc networks, which must adjust quickly to link failure.

The phase discovery mechanism has the additional complication that failure to receive timely acknowledgement may indicate an incorrectly scheduled transmission, rather than link failure.

it is important for a station that changes phase maintains both phases for an interval. Nevertheless it is still possible for a station to have an erroneous phase estimate. Because a failed transmission consumes so much time, it is important to reduce this possibility. A sender can include in its $ATIM$ any station with which it has recently exchanged traffic and for which it is maintaining a phase estimate.

It is worth noting that this limitation is largely theoretical, as there is no IEEE 802.11 standard mechanism for an interface to signal link failure, so this must be handled at a higher layer.

V. Problems and Limitations

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. somewhat less than 50%. But a protocol such as Span, which maintains a connected dominating set also has a fundamental (though less easily calculated) limits on the number of non-coordinator nodes and the proportion of time they spend in the sleep state. In practice, both Span and AFECA report best case energy savings of around 50%. Given the low overhead in the proposed protocol, it seems reasonable to expect that its energy savings will be competitive with these protocols.

The second concern is packet latency. In IEEE 802.11 power save, a packet may be forwarded only hop per beacon interval. Span and AFECA both reduce latency by ensuring that nodes along the forwarding path do not sleep. In the proposed protocol, an intermediate behavior may be expected: a packet's forward progress can be delayed by the periodic sleep/wake cycle, but the effect is mitgated by the longer transfer windows and overlapping awake intervals.

The third concern is system lifetime. The proposed protocol provides a common level of energy savings across the network, independent of node density. This may help to prevent unexpected interactions with higher layer protocols. However, unlike Span, the protocol cannot compensate for other sources of imbalance in energy reserves across the network. This must be done by higher layer energy aware routing or application level mechanisms.

VI. Conclusion and ongoing work

This paper presents ongoing work developing a QoS aware power save protocol for mobile multi-hop wireless networks. This work addresses a meaningful problem defined by a challenging set of requirements, which it has shown itself largely capable of meeting. The approach differs from existing solutions and addresses some perceived limitations in these solutions. It is argued that the proposed protocol can be expected to exhibit energy savings comparable to existing published work, as well as having qualitative advantages in simplicty, asynchronous operation and adaptivity to the complex channel contention found in multi-hop networks. The basic mechanisms developed in the protocol are highly extensible and have the potential to support advanced QoS functionality.

Due to this favorable expectation, work is ongoing to implement the protocol in the ns2 simulation environment, using IEEE 802.11 as the underlying MAC layer.

References

[1] M. Stemm, P. Gauthier, D. Harada, and R. H. Katz, "Reducing power consumption of network interfaces in hand-held devices," in *Proc. of 3rd International Workshop on Mobile Multimedia Communications (MoMuc-3)*, Sept. 1996.

[2] B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris, "Span: An energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks," in *Proc. of 7th Annual International Conference on Mobile Computing and Networking*, pp. 85–96, 2001.

[3] J.-P. Ebert, B. Burns, and A. Wolisz, "A trace-based approach for determining the energy consumption of a WLAN network interface," in *Proc. of European Wireless*, pp. 230–236, Feb. 2002.

[4] L. M. Feeney and M. Nilsson, "Investigating the energy consumption of a wireless network interface in an ad hoc networking environment," in *Proc. of IEEE Infocom*, Apr. 2001.

[5] R. Kravets and P. Krishnan, "Power management techniques for mobile communication," in *Proc. of 4th Annual International Conference on Mobile Computing and Networking (MobiCom'98)*, 1998.

[6] IEEE Computer Society LAN MAN Standards Committee, *IEEE 802.11 Standard: Wireless LAN Medium Access Control and Physical Layer Specifications*, Aug. 1999.

[7] Y. Xu, J. Heidemann, and D. Estrin, "Adaptive energy-conserving routing for multihop ad hoc networks," Tech. Rep. 527, USC/Information Sciences Institute, Oct. 2000.

[8] V. Rodoplu and T. H.-Y. Meng, "Minimum energy mobile wireless networks," *IEEE Journal on Selected Areas in Communications*, vol. 17, pp. 1333–1344, Aug. 1999.

[9] J. Wieselthier, G. Nguyen, and A. Ephremides, "On the construction of energy-efficient broadcast and multicast trees in wireless networks," in *Proc. of IEEE Infocom*, 2000.

[10] R. Ramanathan and R. Hain, "Topology control of multihop wireless networks using transmit power adjustment," in *Proc. of IEEE Infocom*, vol. 2, pp. 404–413, 2000.

[11] J. Gomez, A. T. Campbell, M. Naghshineh, and C. Bisdikian, "Conserving transmission power in wireless ad hoc networks," in *Proc. of IEEE Conference on Network Protocols (ICNP'01)*, Nov. 2001.

[12] R. Wattenhofer, L. Li, P. Bahl, and Y.-M. Wang, "Distributed topology control for wireless multihop ad-hoc networks," in *Proc. of IEEE Infocom*, pp. 1388–1397, Apr. 2001.

[13] S. Agarwal, S. Krishnamurthy, R. Katz, and S. Dao, "Distributed power control in ad hoc wireless networks," in *Personal and Indoor Mobile Radio Communication (PIMRC)*, 2001.

[14] S. Narayanaswamy, V. Kawadia, R. S. Sreenivas, and P. R. Kumar, "Power control in ad-hoc networks: Theory, architecture, algorithm and implementation of the COMPOW protocol," in *Proc. of European Wireless*, pp. 156–162, Feb. 2002.

[15] M. Sanchez, P. Manzoni, and Z. J. Haas, "Determination of critical transmission range in ad-hoc networks," in *Proc. of Multiaccess Mobility and Teletraffic for Wireless Communications Workshop*, Oct. 1999.

[16] S. Singh, M. Woo, and C. S. Raghavendra, "Power-aware routing in mobile ad hoc networks," in *Proc. of 4th Annual International Conference on Mobile Computing and Networking (MobiCom'98)*, pp. 181–190, 1998.

[17] J. H. Chang and L. Tassiulas, "Routing for maximum system lifetime in wireless ad-hoc networks," in *Proc. of Allerton Conference on Communication, Control, and Computing*, Sept. 1999.

[18] J.-H. Chang and L. Tassiulas, "Energy conserving routing in wireless ad-hoc networks," in *Proc. of IEEE Infocom*, vol. 1, pp. 22–31, 2000.

[19] C.-F. Chiasserini and R. R. Rao, "Routing protocols to maximize battery efficiency," in *Proc. of IEEE Milcom*, Oct. 2000.

[20] W. R. Heinzelman, J. Kulik, and H. Balakrishnan, "Adaptive protocols for information dissemination in wireless sensor networks," in *Proc. of 5th Annual International Conference on Mobile Computing and Networking (MobiCom'99)*, pp. 174–185, 1999.

[21] H. Woesner, J.-P. Ebert, M. Schlager, and A. Wolisz, "Power saving mechanisms in emerging standards for wireless LANs: The MAC level perspecitve," *IEEE Personal Communications*, vol. 5, pp. 40–48, June 1998.

[22] R. Sivakumar, P. Sinha, and V. Bharghavan, "CEDAR: A core-extraction distributed ad hoc routing algorithm," *IEEE Journal on Selected Areas in Communications, Special Issue on Ad Hoc Networks*, vol. 17, pp. 1454–65, Aug. 1999.

[23] J. Wu, M. Gao, and I. Stojmenovic, "On calculating power-aware connected dominating sets for efficient routing in ad hoc wireless networks," in *Proc. of IEEE International Conference on Parallel Processing*, pp. 346–353, Sept. 2001.

[24] L. M. Feeney, "An asysnchronous power save protocol for wireless ad hoc networks," Tech. Rep. T2002:9, Swedish Institute of Computer Science, July 2002.

[25] J. Li, C. Blake, D. S. J. D. Couto, H. I. Lee, and R. Morris, "Capacity of ad hoc wireless networks," in *Proc. of 7th Annual International Conference on Mobile Computing and Networking*, pp. 61–69, 2001.

[26] P. Gupta and P. R. Kumar, "The capacity of wireless networks," *IEEE Transactions on Information Theory*, vol. 46, pp. 388–404, Mar. 2000.