Available online at www.sciencedirect.com
SCIENCE DIRECT®
Ad Hoc Networks 3 (2005) 51–68
Ad Hoc Networks
www.elsevier.com/locate/adhoc

On-demand power management for ad hoc networks ☆

Rong Zheng *, Robin Kravets

Department of Compute Science, University of Illinois at Urbana-Champaign, Urbana, IL 61801, USA

Received 17 August 2003; accepted 6 September 2003
Available online 26 November 2003

Abstract

Battery power is an important resource in ad hoc networks. It has been observed that in ad hoc networks, energy consumption does not reflect the communication activities in the network. Many existing energy conservation protocols based on electing a routing backbone for global connectivity are oblivious to traffic characteristics. In this paper, we propose an extensible on-demand power management framework for ad hoc networks that adapts to traffic load. Nodes maintain soft-state timers that determine power management transitions. By monitoring routing control messages and data transmission, these timers are set and refreshed on-demand. Nodes that are not involved in data delivery may go to sleep as supported by the MAC protocol. This soft state is aggregated across multiple flows and its maintenance requires no additional out-of-band messages. We implement a prototype of our framework in the ns-2 simulator that uses the IEEE 802.11 MAC protocol. Simulation studies using our scheme with the Dynamic Source Routing protocol show a reduction in energy consumption near 50% when compared to a network without power management under both long-lived CBR traffic and on–off traffic loads, with comparable throughput and latency. Preliminary results also show that it outperforms existing routing backbone election approaches.

Keywords: Ad hoc networks; Energy conservation; Power management; Soft state; On demand

1. Introduction

With the proliferation of portable computing platforms and small wireless devices, ad hoc wireless networks have received more and more attention as a means for providing data communications among devices regardless of their physical locations. Wireless communication has the advantage of allowing untethered communication, which implies reliance on portable power sources such as batteries. However, due to the slow advancement in battery technology, battery power continues to be a constrained resource.

It has been observed that in ad hoc networks, energy consumption does not always reflect active communication in the network [1]. Experimental results reveal that the energy consumption of wireless devices in an idle state is only slightly smaller than that in a transmitting or receiving state. Therefore, it is in general desirable to turn the radio off when it is not in use, termed as power

☆ The work is sponsored in part by MURI/AFOSR under contract number F49620-00-1-0330 and National Science Foundation grant number ANI-0081308.

* Corresponding author.
E-mail addresses: zheng4@cs.uiuc.edu (R. Zheng), rhk@cs.uiuc.edu (R. Kravets).

1570-8705/$ - see front matter © 2003 Elsevier B.V. All rights reserved.
doi:10.1016/j.adhoc.2003.09.008

management. Motivated by these observations, several energy conservation protocols [2,3] have been proposed to take advantage of the route redundancy in dense ad hoc networks by turning off devices that are not required for global network connectivity. However, in these protocols, the decision about which set of nodes to leave on is only based on geographical/topological information, thus is oblivious to the actual traffic load in the network. Since many applications of ad hoc networks are data-centric, maintenance of global connectivity is costly and unnecessary when no traffic or only localized traffic is present in the network.

Various techniques, both in hardware and software, have been proposed to reduce energy consumption for mobile computing devices in wireless LANs [4,5]. In contrast, power management in ad hoc networks is a more difficult problem for two reasons. First, in ad hoc networks, a node can be both a data source/sink and a router that forwards data for other nodes and participates in high-level routing and control protocols. Additionally, the roles of a particular node may change over time. Second, there is no centralized entity such as an access point to control and maintain the power management mode of each node in the network, buffer data and wake up sleeping nodes. Therefore, power management in ad hoc networks must be done in a distributed and cooperative fashion. A major challenge to the design of a power management framework for ad hoc networks is that energy conservation usually comes at the cost of degraded performance such as lower throughput or longer delay. A naive solution that only considers power savings at individual nodes may turn out to be detrimental to the operation of the whole network.

In this paper, we propose an on-demand power management framework targeting generic ad hoc networks. To achieve reduced energy consumption while maintaining effective communication, our framework integrates routing information from on-demand ad hoc routing protocols and power management capabilities from the MAC layer. Energy conservation is achieved by judiciously turning on and off the radios of specific nodes in the network. The novelty of our framework is that such power management decisions are driven by active communications in the network. For the purpose of energy conservation, connectivity is only maintained between pairs of senders and receivers and along the route of data communication.

Transitions between power management modes for each node are associated with a soft-state timer that is established and refreshed by data and control messages in the network. Once the soft state is established, subsequent data delivery can be expedited without incurring additional delays from waking up sleeping nodes along the route. The length of the soft-state timer reflects the adaptiveness of the power management framework to variations in traffic load. Since the operations of transmitting to a sleeping node and an active node are different, we present mechanisms to discover a neighbor's power management mode. In this context, neighbor discovery is challenging because a node in power-save mode cannot monitor the channel consistently. Therefore, any neighbor information may be ambiguous. This situation is even worse if nodes are mobile.

Our framework is not limited to any specific routing or MAC protocols. This extensibility is a key benefit of our design since it enables the use of our framework in various scenarios and allows the integration of new protocols as they become available. To verify our framework, we present a prototype using the IEEE 802.11 MAC protocol and evaluate it using Dynamic Source Routing (DSR) [6] and greedy geographical forwarding protocol in the ns-2 [7] simulator. Under a wide range of traffic patterns and load, our prototype achieves 40–60% savings in power consumption as compared to a network without power management. In addition, our prototype minimally increases latency during the initial setup stage, but achieves an average latency comparable to a network without power management.

The rest of the paper is organized as follows. We first layout the design space for power management protocols in ad hoc wireless networks and give a brief overview of existing approaches in Section 2. Then we discuss how each approach fits into the design space. In Section 3, we present the building blocks and technical details of our on-demand power management framework. Section 4 describes a prototype based on IEEE 802.11 MAC. Extensive simulation results are presented in Section 5. Finally, we conclude the paper and discuss future extensions in Section 6.

2. Design space

Power management in ad hoc networks spans all layers of the communication protocol stack. Each layer has access to different types of information about the communication in the network, and thus uses different mechanisms for power management. The MAC layer does power management using local information while the network layer can take a more global approach based on topology or traffic characteristics. In this paper, we consider power management approaches that save energy by turning off the radios of nodes in the network. Other energy conservation mechanisms such as topology control and power controlled MAC protocols [8–10] are considered orthogonal and the benefits can be combined.

Similar to ad hoc routing protocols, power management schemes range from proactive to reactive. The extreme of proactive can be defined as *always-on* (i.e. all nodes are active all the time) and the extreme of reactive can be defined as *always-off* (i.e. all nodes are in power saving mode by default) (see Fig. 1). Given the dynamic nature of ad hoc networks, there needs to be a balance between proactiveness, which generally provides more efficient communication, and reactiveness, which generally provides better power saving.

In this section, we outline the design space of power management in ad hoc networks and describe where existing approaches fit into this design space based on their adaptability to network traffic.

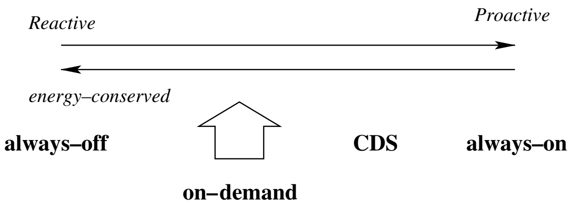


Fig. 1. Design space of power management schemes.

2.1. MAC layer approaches

At the MAC layer, power management decisions are made based on local information. The time scale for power management can be per-packet or a short time interval. Such approaches are limited by the lack of access to information about the topology and traffic in the network.

The PAMAS power-saving medium access protocol [11] turns off a node's radio when it overhears a packet not addressed to it. The effectiveness of PAMAS is limited to reducing the power consumption of processing unnecessary packets. Note that PAMAS alone can be considered a proactive approach to power management, however it may be combined with most high level power management schemes that aim to reduce idle time energy consumption.

The IEEE 802.11 MAC provides low-level support for power management such as buffering data for sleeping nodes and synchronizing nodes to wake up for data delivery. The network interface has five physical states: transmitting, receiving, idle, sleeping and completely power-off. Energy consumption in the sleeping state is significantly less than in the transmitting/receiving/idle state. In the IEEE 802.11 specification, a node can be in one of two power management modes, active mode (AM) or power-save mode (PS). In active mode, a node is awake and may receive data at any time. In power-save mode, a node wakes up periodically to check for incoming traffic. The transition between power management modes is left to higher-level power management protocols and is unspecified in the documentation.

STEM [12] proposes a similar approach to the IEEE 802.11 power management, but uses an independent control channel to avoid the clock synchronization needed by IEEE 802.11. STEM uses asynchronous beacon packets in a second control channel to wakeup intended receivers. After transmissions have ended (e.g. after a time-out, etc.), the node turns its radio off in the data channel. Similar to IEEE 802.11, sleeping nodes with traffic destined for them are woken up on demand, but decisions about when a node should go back to sleep are based on local information. STEM does not provide mechanisms for indicating the power management state of a node. Instead, the power management state is only maintained on a per-link basis between nodes with active communication. Therefore, it is possible that an initiator node will experience the delay from waking up a receiver node, even if the receiver is already awake due to recent communication with a third node.

In S-MAC [13], the authors propose a mechanism called message passing that modifies a network allocation vector (NAV) for virtual channel reservation in IEEE 802.11 MAC type of protocols. The length of NAV is determined by the duration of a burst of messages. The virtual reservation serves two purposes: (1) it mandates the receiver to remain on throughout the transmission of the burst, and (2) it prevents other nodes from transmitting during this interval. Though message passing may be desirable for certain types of applications for sensor networks, it can be inefficient for more generic ad hoc networks. Additionally, the reservations may cause some nodes to be starved.

A pure MAC layer approach as specified by the IEEE 802.11 MAC (i.e. nodes are always in power-save mode) can be considered as the most reactive approach to power management in our design space. In Section 5, we demonstrate that a network that relies solely on the IEEE 802.11 MAC for power management can be highly inefficient even though some communication is still possible. As future research, we will investigate the interaction between intelligent MAC layer approaches, such as STEM and S-MAC, with our on-demand framework.

2.2. Connected dominating set approaches

At the network layer, power management schemes can take advantage of topological information. The connected dominating set approaches use neighborhood or global information to decide the set of nodes that form a connected dominating set (CDS) for the network, where all nodes are either a member of the CDS or a direct neighbor of one of the members. Nodes in the CDS serve as the "routing backbone" and remain on all the time to maintain global connectivity. All other nodes can choose to sleep if necessary.

CDS approaches such as GAF [3] and SPAN [2] conserve energy by reducing routing redundancy in dense networks. Selection and maintenance of the CDS requires local broadcast messages that may consume a significant amount of energy [1]. In addition, regardless of whether or not traffic is present in the network, all backbone nodes must be on all the time. Therefore, CDS approaches can be categorized as proactive.

Based on these observations about MAC layer and CDS approaches, we propose an on-demand power management framework to explore the design space between proactive and reactive by adapting to the traffic characteristics inside the network.

3. On-demand power management

The goal of on-demand power management is to base power management decisions on traffic patterns in the network. By reacting to changes in these patterns, nodes that do not carry any traffic can be dispensed from consuming a significant amount of energy. Varying the adaptiveness to network load in our protocol balances the tradeoff between latency, throughput and energy consumption.

The key idea of our on-demand power management framework is that transitions from power-save mode to active mode are triggered by communication events such as routing control messages or data packets and transitions from active mode to power save mode are determined by a soft-state timer. The soft-state timer is refreshed by the same communication events that trigger a transition to active mode. A node keeps track of its neighbors' power management mode either by HELLO messages or by snooping transmissions over the air. For direct unicast messages, if the next hop is in active mode, the message is delivered immediately as allowed by the queuing discipline.

3.1. A cross-layer design for power management

Power management in ad hoc networks can benefit from a cross-layer design that leverage both network layer and MAC layer information. Knowledge about route setup and packet forwarding can provide hints about when power management should be performed. Since the route discovery phase of on-demand routing protocols determines the path subsequent packets will follow, nodes along this route should be as responsive as possible. On the other hand, any effective power management protocol requires a mechanism to awaken a sleeping receiver when packet delivery is imminent. This is usually handled by low-level mechanisms at the MAC or physical layers. Higher-level power management techniques can benefit from information about and access to the mechanisms used to provide such services.

Our power management framework leverages the capability of modern MAC protocols, such as the IEEE 802.11 MAC, to switch power management states of nodes and buffer data if necessary for sleeping nodes. It also uses routing information to decide when to turn nodes on and off, which ties energy consumption with active communication in the network. The interaction of the power management module with other communication layers is illustrated in Fig. 2.

3.2. Power management mode and state transition

In our framework, a node can be in one of two power management modes: active mode (AM) and power-save mode (PS). In active mode, a node is awake and may receive data at any time. In power-save mode, a node is sleeping most of the time and wakes up periodically to check for pending messages. Packets destined to a node in power-save mode will experience delay on the order of the length of the sleeping cycle.

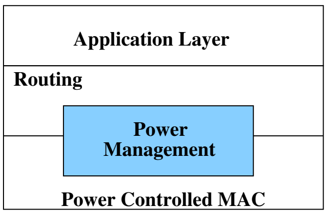


Fig. 2. Location in the protocol stack of our power management framework.

Transitions from power-save mode to active mode are triggered by communication events in the network. Transitions from active mode to power-save mode are determined by a soft-state keep-alive timer. Initially, all nodes are in power-save mode. Upon reception of packets, a node starts the keep-alive timer and switches to active mode. Timer values depend on the type of packet received. Upon expiration of the keep-alive timer, a node switches from active mode to power-save mode.

If all packets trigger a node to stay awake with a keep-alive timer on the order of the network lifetime, our scheme degenerates to an always-on network without power management. On the other hand, if the keep-alive timer is always set to zero, our framework degenerates to the most reactive MAC layer approach discussed in Section 2. Therefore, the choice of different keep-alive timer values varies the reactiveness of the protocol and strikes different trade-offs between energy consumption and data delivery efficiency.

In an ad hoc network, if a path is going to be used, the nodes along that path should be awake as to not cause unnecessary delay for data transmission. If a path is not going to be used, the nodes should be allowed to sleep. During the lifetime of the network, different messages will indicate different levels of "commitment" to using a path. Knowledge of the semantics of such messages can help make better power management decisions, which is a missing piece in most MAC layer power management approaches.

On one end, most control messages (e.g. link state in table-driven ad hoc routing protocols, location updates in geographical routing, route request messages in on-demand routing protocols etc.) are flooded throughout the network and provide poor hints for the routing of data transmissions. Such control messages should not trigger a node to stay in active mode. On the other end, data transmissions are usually bound to a path on relatively large time scales. Therefore, data transmissions are a good hint for guiding power management decisions. For data packets, the keep-alive timer should be set on the order of the packet inter-arrival time to ensure that nodes along the path do not go to sleep during active communication. There are also some control messages, such as route reply messages in on-demand routing protocols and query messages in sensor networks, that provide a strong indication that subsequent packets will follow this route. Therefore, such messages should trigger a node to switch to active mode. The time scale of the keep-alive timer for such a transition should be on the scale of the end-to-end delay from source to destination so the node does not transition back to power save mode before the first data packet arrives.

One important feature of the keep-alive timer is that it is refreshed on demand. Whenever a node receives a routing message or a data packet, it sets the timer with the maximum of what is left for the current keep-alive timer and the value associated with the received message. Therefore, only per-node, instead of per-flow, information is needed for power management. Ideally, the keep-alive values for data packets should be larger than the inter-arrival time of data packets. In reality, since a node can be both data source/sink and forwarding router simultaneously, its keep-alive timer is an aggregation of various timer values, i.e., old timer will be extended when new communication events arrive. Therefore, the performance is quite insensitive to the choice of these timer values.

3.3. Obtaining neighbors' power management mode

Since communication with a neighbor is only possible if the neighbor is in active mode, it is necessary for nodes to track power management modes of neighbors. In our framework, each node maintains a neighbor list that caches a neighbor's power management mode and a time-stamp of the most recent update from this neighbor.

A neighbor's power management mode can be discovered in two ways. The first way is through explicit local HELLO message exchanges with piggybacked information about the power management mode of a node. HELLO messages should be transmitted at fixed intervals regardless of the power management mode of a node. Link failure is assumed if no HELLO messages have been received during successive intervals, since the loss of only one HELLO message may have been caused by a broadcast collision.

The second way to discover a neighbor's power management mode is via passive inference. Depending on the capability of the hardware and the MAC protocol, a node may be able to operate in promiscuous mode and passively snoop messages in the air. With MAC layer support, a node's power management mode can be piggybacked in the control header of MAC layer data units. There are two challenges to using passive inference. First, nodes in power-save mode cannot hear messages from their neighbors and so do not have a good basis for determining the power management mode of their neighbors. Second, nodes in power-save mode may not be transmitting and so their neighbors will have difficulty differentiating nodes that are in power-save mode from nodes that are away or dead. Therefore, special care must be taken to distinguish between nodes that move away from ones that are in power-save mode.

Since the use of HELLO messages is expensive, our framework uses two types of indicators for such passive inference. The first indicator is a lack of communication during a time interval. When no communications have been observed from a node that was in active mode, the neighbor is assumed to be in power-save mode. The value of the this interval should be based on the keep-alive timer since the length of keep-alive timer indicates the maximum amount of time a node commits to be in active mode when no messages are received. If a node does not hear from its neighbor during the keep-alive period, it is very likely that either that neighbor moved away or it has switched to power-save mode.

The other indicator is packet delivery failure to the neighbor (e.g. indicated by an RTS retry time out in IEEE 802.11). Based on the observed power management mode of the neighbor, a packet delivery failure is treated in two stages. First, if the neighbor was originally in active mode, it is considered to have switched to power-save mode. Second, if the neighbor was originally in power-save mode, it is now considered unreachable. Data for this node is discarded at the MAC layer. The rationale for this two-stage process is that transition to an intermediate stage provides a second chance to salvage data for a neighbor that has switched to power-save mode since the last update.

Compared to using HELLO messages, passive inference does not rely on additional control messages, which is more desirable from an energy conservation perspective. However, the ambiguity of link failure and the power management mode of a neighbor can result in delayed data transmission, but as will be shown in Section 5, this approach in general works well.

4. A prototype based on the IEEE 802.11 MAC

In this section, we present a prototype of our framework based on the IEEE 802.11 MAC. First, we give a brief overview of IEEE 802.11 power management functions and then we discuss the implementation details of our prototype. Note that our on-demand framework can be easily implemented over other MAC protocols including those using asynchronous wakeup mechanisms [14].

4.1. Overview of IEEE 802.11 power management in ad hoc networks

In the IEEE 802.11 specification, all nodes in the network are synchronized to wake up periodically. Broadcast/multicast messages or unicast messages to a power-saving node are first announced during the period when all nodes are awake. The announcement is done via an ad hoc traffic indication message (ATIM) inside a small interval at the beginning of the beacon interval called the ATIM window.

During the ATIM window, nodes that have buffered data for sleeping nodes transmit an ATIM management frame that contains the identity of the intended receivers. If a node receives a directed ATIM frame during the ATIM window (i.e. it is the designated receiver), it sends an acknowledgment and stays awake for the entire beacon interval waiting for the data to be transmitted. Broadcast/multicast messages announced in the ATIM need not be acknowledged. Immediately after the ATIM window, nodes can transmit buffered broadcast/multicast frames, data packets and management frames addressed to nodes that have acknowledged a previously transmitted ATIM frame. Following the transmission of all buffered data, nodes transmit data destined to nodes that are known to be in the active state for the current beacon interval. In IEEE 802.11, a node's power management status is indicated in the frame control field of the MAC header for each packet.

4.2. Our prototype

We experiment with on-demand routing protocols as DSR [6] or AODV [15] as well as stateless routing protocols such as greedy geographical routing protocols. Unless otherwise specified, the results are reported with using DSR protocol.

The complete state transition diagram is shown in Fig. 3. Transitions between power-save and active mode are triggered by packet arrivals and expiration of the keep-alive timer. Sub-state transitions inside power-save or active mode indicate the physical state of the node and are controlled by the IEEE 802.11 MAC power management functions.

To maintain the neighbor list, the prototype uses passive inference to update neighbors' power management modes and link states. Nodes snoop transmissions in the air when they are awake and update their neighbor lists based on the control field of the MAC header of packets. Entries for unreachable neighbors are purged periodically. Due to the use of beacon messages, changes in link availability can be detected proactively as follows.

Let the beacon interval be I and let the degree of a node be bounded by d. The interval N (measured in units of beacon intervals) of two successive beacon messages sent by node n follows the geometric distribution $N \sim (1/d)(1-1/d)^{N-1}$ with mean d. Therefore, if a node has not heard any beacon messages from a particular neighbor for more than $c \cdot d$ beacon intervals, where c is a protocol-specific constant, the node is likely away or "dead". The degree of a node is obtained from the neighbor list and a node can use its own degree to approximate its neighbor's degree. This method

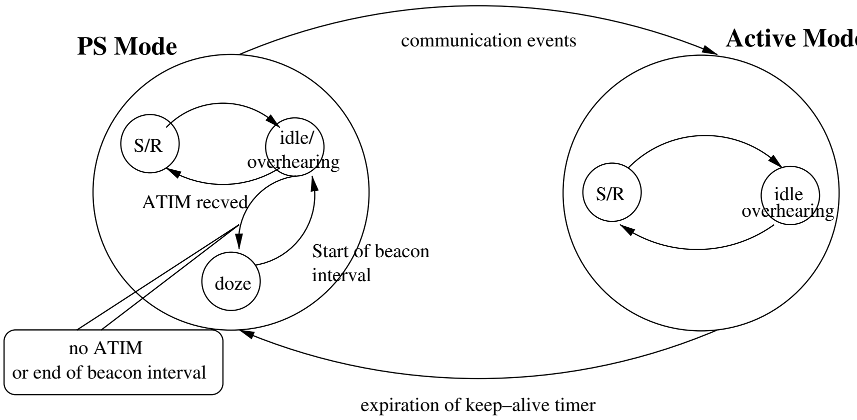


Fig. 3. State transition diagram for power management modes enhanced with IEEE 802.11 physical states.

can be combined with events of packet delivery failure to better infer the availability of a link between neighbors. The major benefit of inferring a neighbor's state by snooping is that it does not incur out-of-band control messages and therefore scales to large networks.

Table 1 lists various timers and values for different messages. To determine the values of various keep-alive timers, consider a k-hop route from source n_0 to sink n_k. Suppose the one-way delay from node n_0 to node n_i on this route is d_i. The inter-arrival time of data packets at the source is $1/\lambda$. Let the beacon interval be I. Therefore, the time it takes for the route request message to reach node n_i from n_0 is $i * I + d_i$ on average. The time it takes node n_i to receive the route reply message is $(2k + 1 - i) * I + d_{k+1} + d_i$ under the assumption of symmetric routes. Assuming the data source will immediately transmit the data upon reception of the route reply message, the time between the reception of the route reply message and the reception of the first data packet at node n_i is $i * I + 2 * d_i$. Finally, assuming no additional queuing at intermediate nodes, the packet inter-arrival time at node i is $1/\lambda$. Therefore, on the pessimistic side, the length of the keep-alive timers for different message types should be chosen to be larger than these estimates to ensure low latency for data delivery. If available, information about the network dimensions and traffic patterns can be used to select these values based the above discussion. In our performance evaluation in Section 5, we do not assume availability of such knowledge. In our implementation, we set RTRQ_KEEPALIVE to 0, RTRL_KEEPALIVE to 5 s, DATA_KEEPALIVE, SRC_KEEPALIVE and DST_KEEPALIVE to 2 s. The REFRESH_INTERVAL is set to 5 s. Since keep-alive timer values can be aggregated, the performance is quite insensitive to the choice of these values. As part of our future work, we will investigate techniques to adapt keep-alive timer values based on measurements in the network.

Table 1
Timers and messages

Message type	Value
Route request	RTRQ_KEEPALIVE
Route reply	RTRL_KEEPALIVE
Data at intermediate node	DATA_KEEPALIVE
Data at source	SRC_KEEPALIVE
Data at sink	DST_KEEPALIVE

5. Performance evaluation

We implemented our prototype in the ns-2 [7] network simulator using the CMU wireless extension [16]. To evaluate the effectiveness of our proposed scheme, we conducted several simulations using different traffic models in both static and mobile networks.

The effectiveness of power management schemes can be evaluated by (1) longevity: the network should remain operational for as long as possible, and (2) efficiency: data transmissions should experience low loss and low latency. Longevity is normally characterized by the lifetime of the network, which is application-specific and is tightly coupled with how the network is being used. In this paper, we only focus on the power consumption per unit data delivery defined as follows:

$$\text{energy\_goodput} = \frac{\text{total\_bit\_transmitted}}{\text{total\_energy\_consumed}}, \qquad (1)$$

where the total bits transmitted are calculated for *application-layer data packets* only. The unit of energy goodput is bit/J, which in essence captures the energy utilization of the network with all control overhead considered. Efficiency of data delivery is characterized by the end-to-end latency and the packet delivery ratio defined as the total amount of data received divided by the total amount of data transmitted.

For comparison, we use the most reactive and proactive schemes as baselines. The first is the pure IEEE 802.11 MAC layer power management, termed as *always-off* (i.e. every node is always in power-save mode). The second is without any power management, termed as *always-on* (i.e. all nodes are active throughout the simulation). Unless otherwise stated, there are 50 nodes randomly placed in a 1500 m × 300 m rectangular plane. All nodes communicate with half-duplex wireless radios that conform to IEEE 802.11-based WaveLAN wireless radios with a bandwidth of 2 Mbps and a nominal transmission radius of 250 m. In all simulation scenarios, the network is never partitioned and there are no error-induced losses. DSR is used for routing. Similar experiments were performed with AODV with similar results. We use the same energy model as in [2], which is shown in Table 2. The energy consumption for switching between awake and sleeping states is negligible and thus not considered here. All data packets are of length 128 bytes. Different data packet sizes will affect the throughput of all schemes in a similar fashion. The beacon interval and ATIM window are set to 0.4 and 0.02 s respectively.

Table 2
Power consumption model

Transmit	Receive	Idle	Sleep
1400 mW	1000 mW	830 mW	130 mW

5.1. Effectiveness of the IEEE 802.11 power management functions

The transmission of beacon messages consumes power. Furthermore, regardless whether there is an announcement or acknowledge frame, a node must be awake during the ATIM window. Therefore, it can be expected that the smaller the beacon interval, the more power will be consumed. Similar arguments apply to the length of the ATIM window.

Fig. 4 plots the energy consumption of beacon messages against the beacon interval and ATIM window size for a static 1000 m × 1000 m network with 75 nodes. This measurement indicates how much "raw" energy is consumed in a network implementing IEEE 802.11 power management without any packet delivery. From Fig. 4, we can roughly tell that once the ratio of the ATIM window size and the beacon interval is fixed, the power consumption is almost constant. This is because in the power model we use, the difference between the power consumption of transmitting, receiving and idle states is not very significant. This implies that to reduce the beacon interval

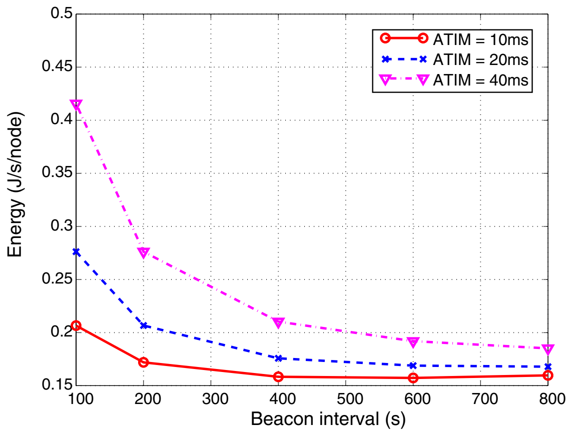


Fig. 4. Energy consumption of beacon messages.

while keeping a roughly constant power consumption for beacon messages, the length of ATIM window should be reduced by the same ratio.

5.2. Study of keep-alive timers

To understand how the choice of keep-alive timers affects the performance of on-demand power management protocol, we compare different combinations of keep-alive timers settings.

5.2.1. Impact of cross-layer information

In the on-demand power management framework, exploiting cross-layer information involves associating different keep-alive timer with different message types (in particular, control and data messages). In contrast, MAC layer alone cannot differentiate among the semantics of higher layer messages. To demonstrate the usefulness of cross-layer information, we first consider a pure MAC layer approach that is agnostic to message types (termed *pure MAC*). Every message (both control and data messages) triggers the setup or refreshing of the soft-state timer. Next, we compare the results with a scheme that triggers the keep-alive timer by data messages only (termed *data only*). Table 3 summarizes the setting of different schemes. The shaded column corresponds to the scheme proposed in Section 4 (termed *mixed*). Fig. 5 shows the loss rate and energy consumption of 10 long-lived CBR connections in a static network transmitting at different rate.

As expected, with the help of cross-layer information such as route replies, the loss rate can be reduced for schemes that setup keep-alive timers for control messages. Data packets do not need to

Table 3
Setup of keep-alive timers

Timers	Value		
	Data only	Mixed	Pure MAC
RTRQ_KEEPALIVE	0	0	2 s
RTRL_KEEPALIVE	0	5 s	2 s
DATA_KEEPALIVE	2 s	2 s	2 s
SRC(DSR)_KEEPALIVE	0 s	2 s	0

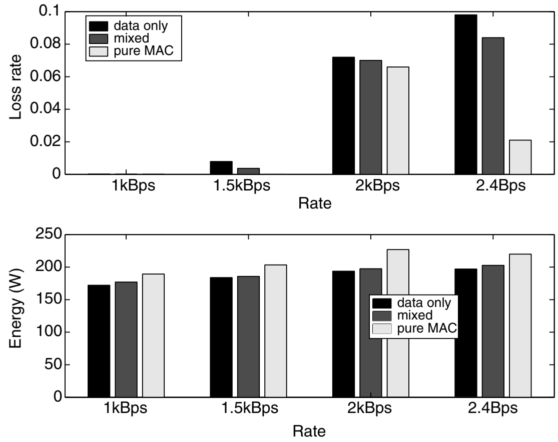


Fig. 5. Comparison of loss rate and energy consumption with and without cross-layer information.

be buffered at the forwarding path and thus, the chance of packet loss due to buffer overflow is reduced. The pure MAC achieves the lowest loss rate at the expense of higher energy consumption. Route discovery messages unnecessarily setup keep-alive timers at nodes that may not involve in data forwarding. We expect the increase in energy consumption to be more pronounced for *pure MAC* scheme with high mobility scenario.

In summary, the mixed scheme that uses both data messages and route reply messages as future traffic indicators can achieve a good balance between energy conservation and packet delivery efficiency. for the rest of the paper, we only experiment with the mixed scheme.

5.2.2. Effect of DATA_KEEPALIVE timer

It is also interesting to consider the sensitivity of the performance to the value of the DATA_ KEEP_ALIVE. As shown in Fig. 6(a), when the DATA_KEEPALIVE is on the same order as the beacon interval (= 400 ms in the simulation), the packet delivery ratio is low for light load. This is due to premature timeout of the KEEPALIVE states in between packet arrivals. When the packet inter-arrival time gets smaller with increasing load, the DATA_KEEP_ALIVE timer can sustain the subsequent packet delivery without premature timeout. For DATA_KEEPALIVE greater than 2 s, the packet delivery ratio remains similar for all

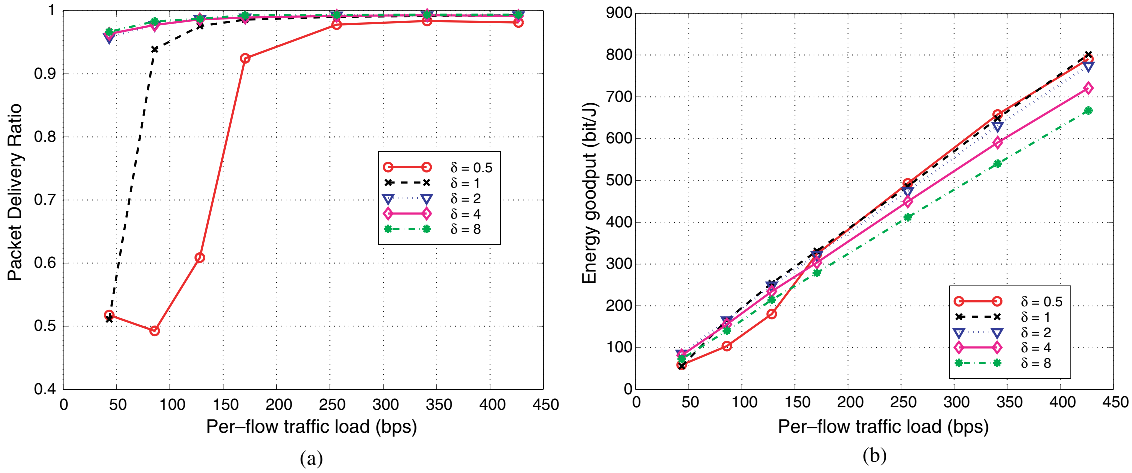


Fig. 6. Impact of different DATA_KEEPALIVE timer, 30 on–off connections, on = 10 s, off = 100 s, 50 nodes, 1500 × 300 static network: (a) packet delivery ratio and (b) energy throughput.

traffic load. Fig. 6(b) also indicates that to maximize the energy goodput, the DATA_KEEPALIVE should be adjusted based on the actual traffic load.

5.3. Experiments in static networks

In this section, we study the performance of our prototype in a static network.

5.3.1. Long-live CBR traffic

In this set of simulations, we simulate long-lived CBR connections at different transmission rates. We compare loss rate, latency and energy consumption for an *always-on* scheme, an *always-off* scheme and our on-demand prototype. There are 10 randomly chosen sender–receiver pairs started randomly between 0 and 100 s. The simulation results presented here are averages over four different scenarios.

Fig. 7 shows the packet delivery ratio and energy goodput as traffic load changes. The *always-off* scheme does not work well under high traffic load. The packet delivery ratio decreases drastically as the load increases. When the traffic load is high, it is possible that there is not enough time to announce all buffered data packets (and get acknowledgments back) due to the limit of the ATIM window size. Depending on the implementation of the MAC and routing protocols, packets get dropped due to delayed transmission and incur further re-transmissions or route discovery. This "chain effect" will result in a pathological network.

As shown in Fig. 7, our prototype achieves similar packet delivery ratios to the *always-on* scheme, while the energy throughput is nearly doubled. The reduced packet delivery ratio at higher traffic load is due to the fact that data delivery is only possible outside the ATIM window since data arriving during the ATIM window need to be buffered temporarily. This slightly lowers the capacity of the network. At low traffic load, the *always-off* scheme can achieve higher energy goodput than the *always-on* scheme, but as the traffic load increases, the energy goodput will eventually suffer due to high loss. In comparison, our on-demand prototype works consistently better than both schemes. The linear area of the energy consumption curve corresponds to the region in which the energy consumption is constant (independent of traffic load). When the traffic load is low to medium, no matter how fast sources transmit, the percentage of time a node stays awake in both our prototype and the *always-on* scheme is roughly constant since there is always data to deliver. Our prototype gains by reducing the number of waking nodes. As the traffic load

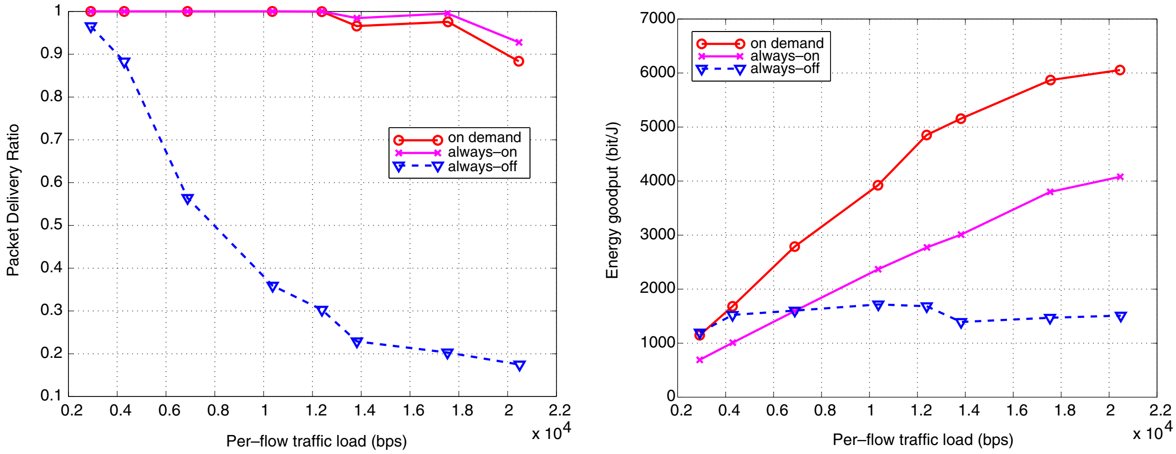


Fig. 7. Packet delivery ratio and energy goodput vs. pause time, 10 long-lived CBR connections, 50 nodes, 1500 × 300 static network.

increases, the number of collision gets higher and the growth of energy goodput slows down for all schemes.

Fig. 8 shows the end-to-end packet delay for a single 3-hop connection during the simulation. With our scheme, apart from the initial setup stage, data packets experience similar latency as those in the *always-on* scheme. This is because after the initial setup stage, nodes in active mode have been established along the route. A node can deliver unicast data directly to its next hop neighbor without the need to make an announcement in the ATIM window.

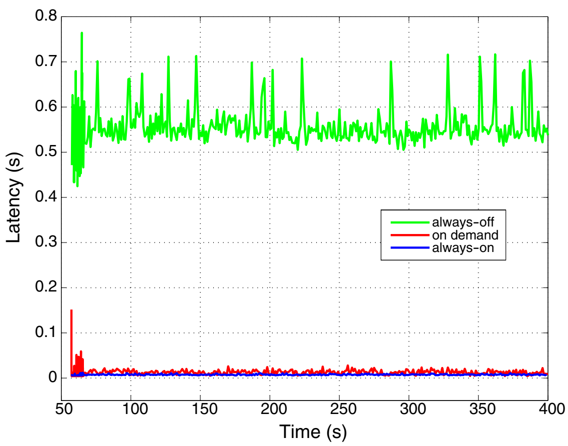


Fig. 8. End-to-end delay of one 3-hop connection, rate = 1 kbps.

5.3.2. On–off traffic

To understand how well our proposed protocol works under more realistic traffic patterns, we simulate on–off traffic with 30 sender–receiver pairs. Both busy and idle intervals follow exponential distribution with means of 10 and 100 s respectively. The simulations run for 900 s. Note that with 30 sender–receiver pairs, most of the 50 nodes in the network are either involved in data forwarding, sending or reception at some time in the simulation.

Similar to the previous set of simulations, we compare the packet delivery ratio and energy goodput vs. traffic load for different schemes. Again, we observe high packet delivery ratio (or low loss rate) and high energy goodput for our prototype (see Fig. 9). The end-to-end packet delay of a 3-hop connection is shown in Fig. 10. Since the idle period of the connection is very long, the keep-alive timers at intermediate nodes will eventually time out. Nodes must be woken up at the beginning of the next busy period. This explains the spikes at the beginning of each busy interval.

One thing noticeable about the energy goodput curves in Fig. 9 is that under very light traffic load, the *always-off* scheme achieves slightly higher energy goodput than our prototype. This is because, in the IEEE 802.11 MAC, a node will go to sleep at the end of ATIM window if no ATIM

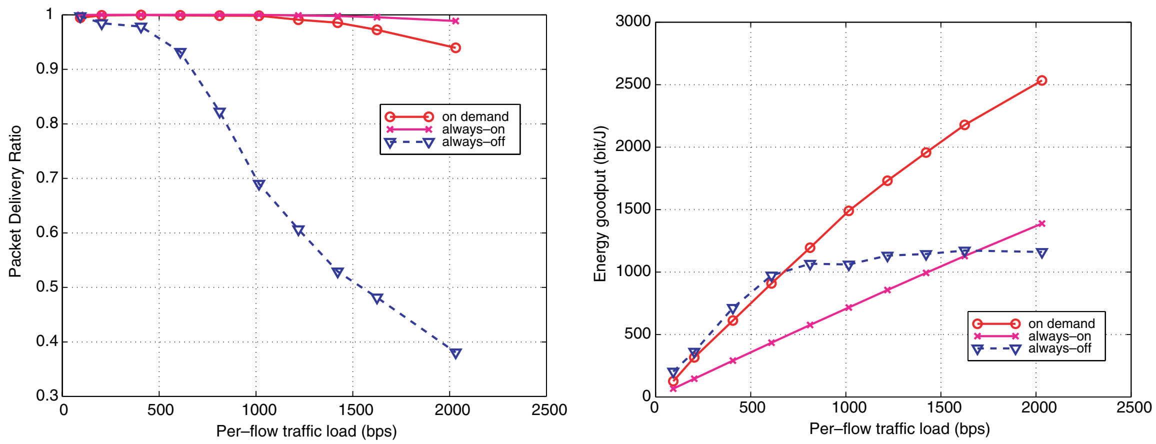


Fig. 9. Packet delivery ratio and energy goodput vs. traffic load. 30 on–off connections, on = 10 s, off = 100 s, 50 nodes, 1500 × 300 static network.

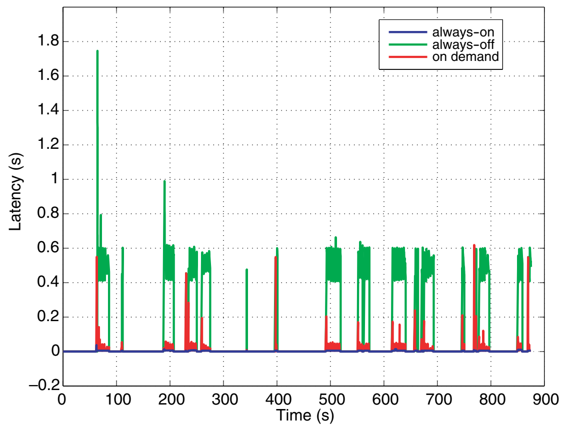


Fig. 10. End-to-end delay of one connection under on–off traffic, rate = 1 kbps.

announcement is received. At light traffic loads, for example, when the sending rate in busy periods is 1 kbps, the inter-arrival time of data packets is roughly 1 s, which spans several beacon intervals. The *always-off* scheme allows further energy savings by turning off nodes on a time scale as small as a beacon interval. Our prototype only switches power management modes on the time scale of the keep-alive timers. To better illustrate this point, we compare the time for actual packet delivery to the time that the wireless interface is turned on in the simulation. For clarity of presentation, we filter out the jumps due to beacon messages since they are very short and frequent.

Fig. 11 shows the duty cycle of a data source/sink node and a forwarding node. We separate routing messages from data packets. A "1" in the top three plots corresponds to an arrival or departure event of the corresponding type of packet. The bottom plot shows the time intervals of the node's active mode with a "1" corresponding to active. This plot consists of an envelope of the union of the top three plots. The values of DATA_KEEPALIVE and RTRL_KEEPALIVE determine the tail of the envelop. Since the start time of each sender–receiver pair varies, there are routing discovery/reply messages up to 200 s into the simulation. Since there is no mobility in this network, all packet losses are caused by collisions. Recall in Section 3, we described the two-stage process to determine link availability to a neighbor. In the simulation, no additional route discovery is incurred after 200 s. This indicates that the two-stage process works well in distinguishing a unreachable node from a neighbor in power-save mode.

5.4. Impact of mobility

In this section, we study how mobility affects the performance of our on-demand prototype. With mobility, nodes on inactive routes may still remain

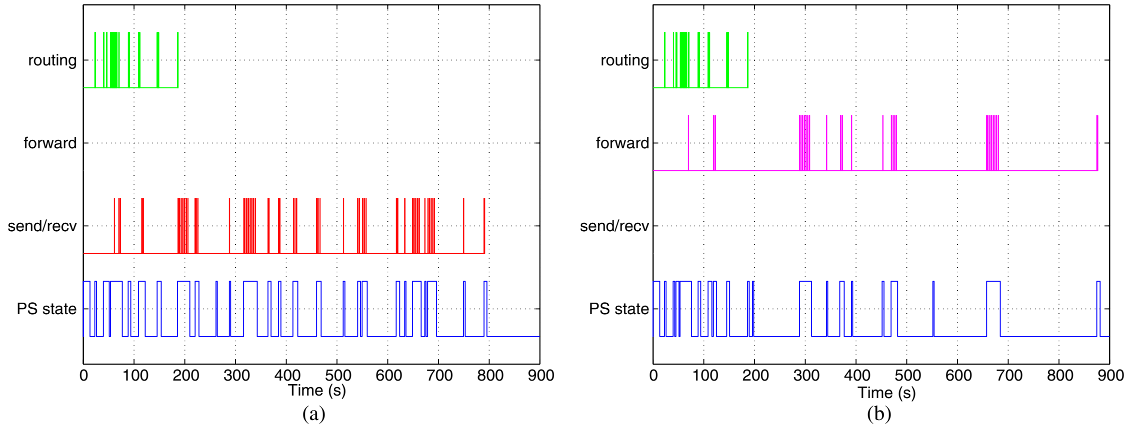


Fig. 11. Actual duty cycles of a node: (a) a data source and (b) a router node.

on for the rest of their keep-alive timer. As a result, we would expect that the gain in energy goodput will be reduced in the case of high mobility.

The simulation setup is as follows. The maximum speed of each node is 20 m/s and the pause time varies from 15 to 75 s. The results presented are an average of four different scenarios of 400 s runs.

As shown in Fig. 12, the energy savings of our proposed scheme is not as significant as in the static scenarios. However, it still conserves a significant amount of energy compared with the *always-on* scheme. Also it performs much better than the *always-off* scheme in terms of both energy goodput and packet delivery ratio. An *always-off* network is no longer functional in high mobility with loss rates as high as 50–60%. An interesting observation is the *always-off* scheme does not conserve any energy in the case of high mobility. The reason is that frequent route discovery messages flooded in the network cause a node to stay awake most of the time.

Fig. 13 shows the end-to-end delay of a 3-hop connection. When a link fails due to mobility, route discovery messages will be sent out. Some nodes on the new routes might be in power-save

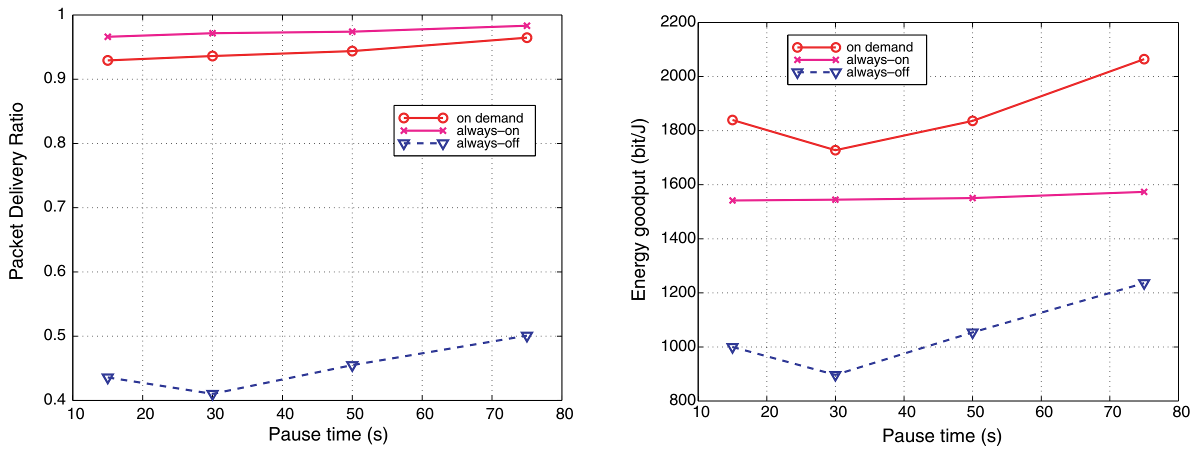


Fig. 12. Packet delivery ratio and energy goodput vs. traffic load with mobility, 10 CBR CNN, 50 nodes, 1500 m × 300 m region, speed = 20 ms.

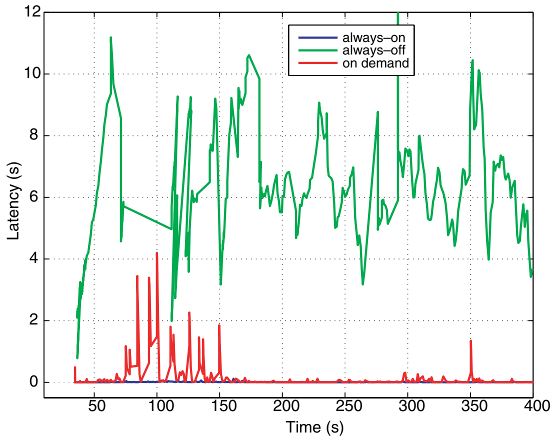


Fig. 13. End-to-end delay for one connection, speed = 20 m/s, pause = 50 s, rate = 1 kbps.

mode and need to be triggered into active mode, causing the spikes in the on-demand curve.

5.5. Experiments with geographical routing protocols

With geographical routing, no route establishment stage is needed before actual data transmission. Therefore, unlike in on-demand routing protocols such as DSR and AODV, where route reply is a good indication for future communication, only data transmission can provide such hint. Upon arrival of each data packet, a node establishes or refreshes its keep-alive soft-state. When a node is awake, upon reception of a data packet, it extends its keep-alive timer by the same length. In this set of simulations, we experiment with greedy geographical routing forwards packet to a neighbor that is closest to the destination at each hop. We assume that location information can be obtained through GPS or other location service. Overhead of obtaining these location information is not simulated.

There are altogether 10 long-lived CBR connections in a 50-node 1500 m × 300 m network. As shown in Fig. 14, the energy throughput of on-demand protocols is larger than both the always-on and always-off network, while the loss rate is comparable to that of always-on network. Another interesting observation in Fig. 14 is the energy throughput curve is roughly linear. This is because the energy dissipation is proportional to the number of active nodes in the network.

5.6. Comparison with GAF

It is very difficult if not impossible to make a fair comparison between our framework and existing CDS approaches like Span and GAF, since the different schemes are based on different assumptions. We highlight the major difference between our framework, GAF and Span in Table 4.

Both Span and GAF assume that data sinks and sources are separated from pure forwarding nodes in their evaluation. In the case of mixed source/

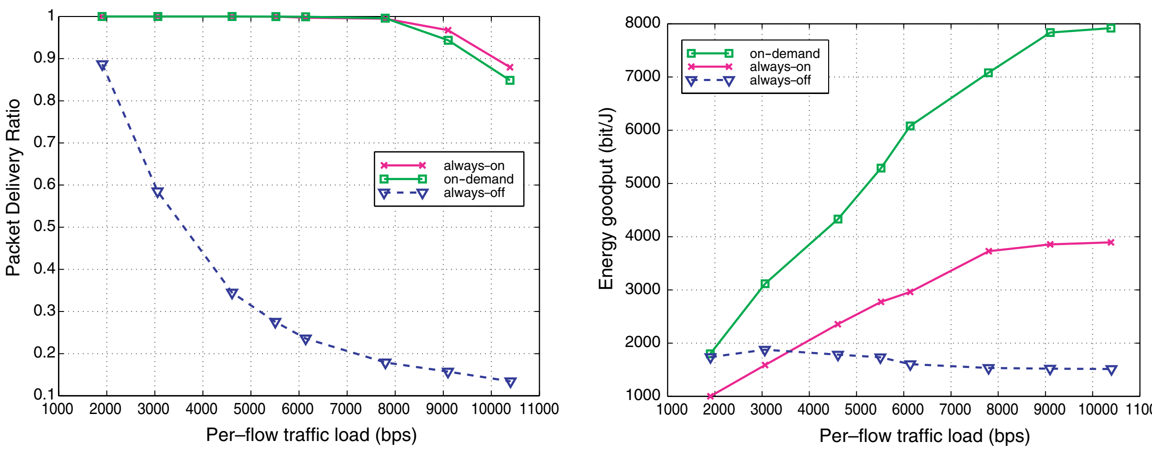


Fig. 14. Packet delivery ratio and energy goodput vs. traffic load. 10 long-lived CBR connections 50 nodes, 1500 × 300 static network.

Table 4
Qualitative comparison of various power management protocols

	MAC support	GPS	Routing	Separation of data sink/source and routers	Local broadcast	Reactive
GAF	No[a]	Yes	Any	Yes	Yes	No
Span	Yes	No	Any[b]	Yes	Yes	No
On demand	Yes	No	*Any*	No	No	Yes

[a] Conceptually, if GAF is applied to networks with mixed data sink/source and routing nodes, signaling mechanisms are needed to wake-up data sinks upon the arrival of data packets. One possible solution is the use of the IEEE 802.11 MAC.
[b] The current implementation of Span in ns-2 is coupled with geographical routing protocols.

sink/forwarding nodes scenarios as used in the previous simulations, the specification of both protocols is incomplete. GAF has no mechanism for signaling the data sink for incoming data. In Span, it is unclear whether the election of coordinators should consider the fact that some nodes may be required to be turned on as data sources or sinks.

In this section, we compare our prototype with GAF since it can be readily used with any routing protocol. To avoid the need to signal data sinks for incoming packets in GAF, we only simulate scenarios where data sources/sinks are at the periphery of the network and internal nodes are dedicated for routing. We simulate a 1000 m $\times$ 1000 m plane with 85 nodes. Connections are between nodes n_{2i} and n_{2i+1} where $i = 0, 1, \ldots, 4$. The locations of nodes n_{2i} and n_{2i+1} are $(0, i * 250)$ and $(1200, i * 250)$ respectively. Data sources/sinks do not participate in packet forwarding. In GAF, the data sources/sinks are on all the time. We simulate on–off CBR traffic with a fixed busy interval of 10 s and an idle interval varying from 0 to 200 s. During a busy interval, each source is transmitting at 1 kbps. We use the same power consumption model for both schemes as listed in Table 2. There is no mobility in this network. For comparison, we also simulate the *always-on* scheme.

Fig. 15 shows the energy consumption of routing nodes under the different schemes. The energy throughput metric does not apply to this set of simulations since the energy consumption of senders and receivers are not evaluated in GAF. The energy consumption of both GAF and the *always-on* scheme are roughly constant with respect to the idle interval. Regardless of whether

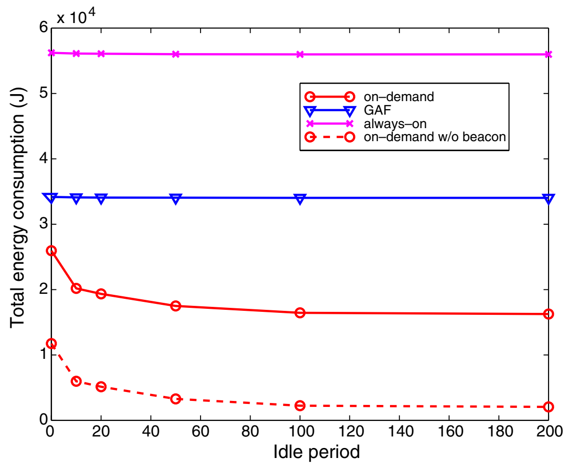


Fig. 15. Comparison with GAF, five on–off connections, on = 10 s, 85 nodes, 1000 m $\times$ 1000 m.

there are data transmissions in the network, either the routing backbone or all the nodes need to be on all the time. By comparison, our prototype achieves further energy conservation by adjusting a node's duty cycle based on its traffic load. The average packet delivery ratio for GAF, on-demand and the *always-on* scheme are 84%, 99.4% and 100% respectively. The reason that the packet loss rate is higher in GAF is that the rotation of grid leaders will induce packet losses.

One interesting data point in Fig. 15 is energy consumption when the idle period is 0. This corresponds to the case of injecting long-lived CBR traffic into the network. Our scheme consumes less energy than GAF even in this scenario for two reasons: (1) grid leader election consumes a significant amount of energy, and (2) there still exists some routing redundancy in GAF due to the limitation on grid size that any two nodes in neighboring grids should be able to communicate directly.

In our scheme, a routing backbone backbone is not explicitly elected. Instead, the route discovery phase "automatically" selects the nodes to be turned on. Given a snapshot of the network at any particular time, the number of nodes that are active in our scheme is less.

Fig. 15 also shows the "net" energy consumption of our prototype after eliminating that of beacon messages. We observe that a significant amount of energy is consumed by beacon messages and so claim that it is necessary to devise more light-weight synchronization mechanisms to wake up nodes.

In summary, our prototype can achieve more energy conservation when compared to GAF by (1) adapting to traffic load and adjusting a node's duty cycle, and (2) invoking fewer intermediate nodes for data delivery. It should be noted that this comparison is biased against our prototype since in GAF, data sources/sinks need to stay awake all the time, while in our prototype, the data sources/sinks can go to sleep during idle periods.

6. Conclusions and future directions

In this paper, we present an on-demand power management framework that reduces energy consumption in ad hoc networks while maintaining effective throughput. It explores the design space between proactive and reactive power management approaches by adapting to the traffic load in ad hoc networks. In our framework, transitions between power management modes of a node are triggered by packet arrivals and the expiration of a soft state timer called the keep-alive timer. Various messages can serve to set up and refresh the keep-alive timer on demand.

We implement a prototype of our framework based on the IEEE 802.11 MAC in the ns-2 simulator. Simulation studies show that our framework consumes significantly less energy than a network without power management under a wide range of traffic patterns and mobility scenarios, while maintaining a good balance between energy conservation and communication efficiency. In addition, the performance of our scheme degrades gracefully in the presence of high mobility. Comparisons with GAF show that our scheme can achieve better energy savings by adjusting a node's duty cycle based on its traffic load and invoking fewer intermediate nodes for data delivery.

Based on our current design and the results of the simulations presented in this paper, we are under-way to investigate the extension of our framework in the following directions.

6.1. Better handling of mobility

When mobility is high, intermediate nodes may remain in active mode longer than necessary, which will result in reduced energy savings. In addition, frequent broadcast messages for route discovery may hurt overall performance since broadcast messages cause all nodes to stay awake during a beacon interval. Techniques such as mobility prediction or proactive handoffs may be used to reduce the number of unnecessarily nodes that remain on despite route changes [17]. For on-demand routing protocols such as DSR that maintain routing caches, we are investigating the integration of power management with the caching strategy to better handle mobility.

6.2. Load balancing

To improve the longevity of the network, load balancing needs to be considered. In both Span and GAF, load balancing is done by rotating the role of coordinators and grid leaders among neighboring nodes in a periodic fashion. Load balancing of this kind only addresses part of the problem since members of the CDS not involved in data transmission still need to remain on. So far, we have not considered load balancing issues in our framework. The correct policy for load balancing is dependent on the communication goals of the network.

Acknowledgements

The authors would like to thank Prof. Jennifer Hou, Casey Carter, Seung Yi, Albert F. Harris III and anonymous reviewers for quality feedbacks.

References

[1] L.M. Feeney, M. Nilsson, Investigating the energy consumption of a wireless network interface in an ad hoc networking environment, in: Proceedings of IEEE INFOCOM, April 2001.

[2] B. Chen, K. Jamieson, H. Balakrishnan, R. Morris, Span: an energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks, in: Proceedings of ACM/IEEE 7th International Conf. on Mobile Computing and Networking (MobiCom 2001), July 2001.

[3] Y. Xu, J. Heidemann, D. Estrin, Geography-informed energy conservation for ad hoc routing, in: Proceedings of ACM/IEEE 7th International Conf. on Mobile Computing and Networking (MobiCom 2001), July 2001.

[4] R. Kravets, P. Krishnan, Application-driven power management for mobile communication, Wireless Networks 6 (4) (2000) 263–277.

[5] M. Stemm, R. Katz, Reducing power consumption of network interfaces in hand-held devices, in: Proceedings of the 3rd International Workshop on Mobile Multimedia Communications, 1996.

[6] D.B. Johnson, D.A. Maltz, Dynamic source routing in ad hoc wireless networks, draft-ietf-manet-dsr-04.txt, 2001.

[7] UCB/LBNL/VINT, Network simulator—ns (version 2), January 2000. Available from <http://www-mash.cs.berkeley.edu/ns/>.

[8] R. Wattenhofer, L. Li, P. Bahl, Y.-M. Wang, Distributed topology control for power efficient operation in multihop wireless ad hoc networks, in: Proceedings of IEEE INFOCOM, 2001.

[9] S. Narayanaswamy, V. Kawadia, R.S. Sreenivas, P.R. Kumar, Power control in ad hoc networks: theory, architecture, algorithm and implementation of the COMPOW protocol, in: European Wireless Conference, 2002.

[10] J.P. Monks, V. Bharghavan, W. Hwu, A power controlled multiple access protocol for wireless packet networks, in: Proceedings of IEEE INFOCOM, 2001.

[11] S. Singh, M. Woo, C.S. Raghavendra, Pamas: power aware multi-access protocol with signalling for ad hoc network, ACM Computer Communication Review 28 (3) (1998) 5–26.

[12] C. Schurgers, V. Tsiatsis, S. Ganeriwal, M. Srivastava, Topology management for sensor networks: exploiting latency and density, in: Proceedings of the Third ACM International Symposium on Mobile Ad Hoc Networking and Computing, 2002.

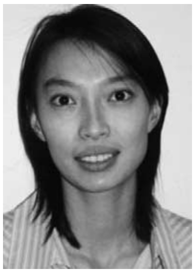
[13] W. Ye, J. Heidemann, D. Estrin, An energy-efficient MAC protocol for wireless sensor networks, in: Proceedings of the 21st Annual Joint Conference of the IEEE Computer and Communications Societies, 2002.

[14] Y.-C. Tseng, C.-S. Hsu, T.-Y. Hsieh, Power-saving protocols for IEEE 802.11-based multi-hop ad hoc networks, in: Proceedings of the 21st Annual Joint Conference of the IEEE Computer and Communications Societies, 2002.

[15] C.E. Perkins, E.M. Royer, S. Das, Ad hoc on demand distance vector (AODV) routing, draft-ietf-manet-aodv-07.txt, 2000.

[16] CMU monarch extensions to ns. Available from <http://www.monarch.cs.cmu.edu/>.

[17] T. Goff, N. Abu-Ghazaleh, D. Phatak, R. Kahvecioglu, The broadcast storm problem in a mobile ad hoc network, in: Proceedings of the 7th International ACM/IEEE Conference on Mobile Computing and Networking, 2001.



Rong Zheng received her BS and MS in Electrical Engineering from Tsinghua University, Beijing, China, in 1996 and 1998 respectively. She is now working towards her Ph.D. degree at the Department of Computer Science, University of Illinois, at Urbana-Champaign since 2001. Her current research interests include power management in multihop wireless networks, design and analysis of sensor network applications. She also conducted research in the areas of flexible mobility support in Mobile IP, Diff-Serv architecture for the Internet, active queue management and MPLS routing algorithms.

Robin Kravets is currently an assistant professor at the Computer Science Department at the University of Illinois, Urbana-Champaign. She received her Ph.D. from the College of Computing, Georgia Institute of Technology in 1999. She is the head of the Mobius group at UIUC, which researches communication issues in mobile and ad hoc networking, including connectivity management, transport protocols, power management, admission control, location management, routing and security. She actively participates in the mobile networking and computing community, both through organizing conferences and being on technical program committees. She is currently an Associate Editor of MC2R: Mobile Computing and Communications Review, a publication of ACM SIGMOBILE.