

The Pulse Protocol: Energy Efficient Infrastructure Access

Baruch Awerbuch, David Holmer, and Herbert Rubens

Department of Computer Science

Johns Hopkins University

Baltimore, MD

{baruch, dholmer, herb}@cs.jhu.edu

Abstract—We present the Pulse protocol which is designed for multi-hop wireless infrastructure access. While similar to the more traditional access point model, it is extended to operate across multiple hops. This is particularly useful for conference, airport, or large corporate deployments. In these types of environments where users are highly mobile, energy efficiency becomes of great importance. The Pulse protocol utilizes a periodic flood initiated at the network gateways which provides both routing and synchronization to the network. This synchronization is used to allow idle nodes to power off their radios for a large percentage of the time when they are not needed for packet forwarding. This results in substantial energy savings. Through simulation we validate the performance of the routing protocol with respect to both packet delivery and energy savings.

I. INTRODUCTION

Wireless networking today is predominantly used to provide mobile users with untethered access to fixed infrastructure. This allows users to move freely throughout the office or warehouse while remaining continuously connected with the office network and the Internet. In these types of environments a majority of the traffic is moving between the mobile nodes and the fixed infrastructure, as opposed to between the mobile nodes themselves such as in ad hoc networks. While traditional access point devices currently provide this capability, they have a limited coverage range and thus many access points are required to provide coverage of a given area. One solution to this problem is to use a routing protocol that allows the users to traverse multiple hops to the nearest access point. This greatly expands the coverage range of each access point while simultaneously reducing costs and simplifying deployment. Although a number of routing protocols have been proposed by the wireless networking community, they have been primarily designed for peer-to-peer ad hoc networks and not specifically optimized for fixed infrastructure access.

Multi-hop fixed infrastructure access networks typically contain up to a large number of mobile users with no readily available power resources. While these networks may contain a large number of users, generally only a small subset of them would be communicating at one time. This necessitates a protocol that scales to high node densities, handles topological changes due to mobility, and is highly energy efficient.

Several methods have been proposed for energy conservation. For example, the 802.11 standard provides power saving functionality, but it only operates in a single hop environment.

A number of power saving protocols have been designed for ad hoc networks, but none of them have focused specifically on this type of infrastructure access application. Since this infrastructure access model is a more specific case of the general ad hoc model, it may be possible to design a protocol that extracts additional performance and power saving.

Our Contribution. We present the Pulse protocol that utilizes a periodic flood, which we refer to as a *pulse*, initiated at the network gateways to provide both routing and synchronization to the network. This periodic pulse forms a spanning tree rooted at the network gateways. By tracking its current parent in the tree, each node has a continuously updated route towards the nearest network gateway. This allows nodes to maintain connectivity with fixed infrastructure across multiple wireless hops; thereby increasing the coverage area of a traditional access point based system. Nodes are able to synchronize with the pulse, which allows idle nodes to power off their radios a majority of the time, except when they are required for packet forwarding. This results in substantial energy savings. Through simulation we validate the performance of the routing protocol with respect to both packet delivery and energy savings.

This paper is organized as follows: In Section II we present our infrastructure access model and power model. We discuss existing strategies for power conservation in Section III. In Section IV we describe in detail the Pulse protocol and provide simulations in Section V.

II. PROBLEM DEFINITION AND MODEL

A. Infrastructure Access Model

While the utility of wireless networks extends to a wide range of applications, we would like to consider specifically the application of multi-hop infrastructure access. Currently, a majority of wireless network deployments involve the use of access points which utilize the IEEE 802.11 Point Coordination Function (PCF) to control access to the wireless medium through centralized coordination. These access points provide access to fixed infrastructure to all nodes within a single hop. Multi-hop operation is not currently specified as part of the IEEE standard. This limitation complicates wireless network deployment by requiring every access point to be wired into the fixed infrastructure and requiring a large number of access points to provide adequate coverage of a given area. By extending the limited access point model to a multi-hop

model where nodes can hop across multiple hops to reach the nearest access point, greater flexibility is provided. This model is similar to the multi-hop cellular model [1] but with an emphasis on data networks. Multi-hop operation can be accomplished by using the Distributed Coordination Function (DCF) instead of the PCF and running an additional routing protocol in order to allow communication across hops. This is similar to the way standard ad hoc routing protocols function.

Existing access point deployments are currently utilized for conferences, airports, or for business networks. In these types of environments wired access is infeasible due to the temporary nature of the participants. In addition, these environments would be likely to contain an extremely large number of participants, resulting in high network density, and variable mobility. Nodes in the network could be completely stationary for long periods of time at conferences, but continuously in motion at trade shows. While high density and high mobility make the routing problem difficult, the actual traffic duty cycle would most likely be low consisting primarily of email traffic and web surfing. In these environments power management is extremely important since there are a large number of devices which are not actively being used. Also, the devices are untethered and not necessarily near any power sources.

B. Power Consumption Model

In order to analyze the power efficiency of routing protocols, it is important to first understand exactly how power is consumed by wireless interfaces. In this work we will specifically be referring to 802.11 wireless adapters. The wireless interface is capable of being in four possible operational states, each of which consumes power at a specific rate. The least power consuming state is the *sleep state*. While in the sleep state the wireless card itself is still consuming a small amount of power, but the radio (which typically consumes the most power) is turned off. While in this state, the card is unable to send or receive packets and has no knowledge of activities taking place on the medium. Since only the radio is powered off, the card can switch the radio off and on quickly. If the card is completely powered off (not just the radio) the reactivation time is much longer.

The wireless card can also be in an *idle state*, meaning its radio is powered on, but it is not currently sending or receiving data. On-demand routing protocols typically spend a great deal of time in this state, since they need to be continuously ready to receive route requests. While in the idle state the card is continuously monitoring the medium sensing for a carrier signal which would cause it to enter the receiving state. The card is in the *transmit* or *receive* state when it is actively sending or receiving.

According to the power consumption measurements for commonly available 802.11b cards [2] (Table I), the power consumption in the sending or receiving state is not much more than the power consumption in the idle state, while the sleep state consumes significantly less power. The idle state consumes only 36% less power than continuously transmitting. The sleep state however consumes 95% less power than

TABLE I
802.11B CARD POWER CONSUMPTION

Transmit	Receive	Idle	Sleep
1327.20 mW	966.96 mW	843.72 mW	66.36 mW

continuously transmitting. As a result, in order to achieve maximum power savings a protocol must utilize the sleep state as frequently as possible.

III. ENERGY CONSERVATION STRATEGIES

There has been a great deal of research conducted with regard to energy efficiency in wireless ad hoc networks as well as in sensor networks where it could be considered even more important due to more limited resources. In general, this work seems to fall into two main categories. The first technique attempts to control the amount of power used to transmit a packet such that only the power required to get the packet to a specific destination is used. The second category involves the design of distributed protocols which allow the nodes of the network to be placed in a sleep mode. The sleep mode category is further divided into three types of approaches: connected active subset, asynchronous wake up, and synchronous wake up. Each of these strategies has advantages and disadvantages when applied to the stated infrastructure access and power model.

A. Power Control

Topology control protocols and least energy path routing protocols [3][4][5] both attempt to provide energy savings by controlling transmission power. The fundamental concept that drives these protocols is that long range transmissions require greater power than short range transmissions. As a result, sending a packet using several short range hops can consume less total transmission power than sending the packet directly to the destination.

The main disadvantage of power control protocols is that transmission power consumption usually represents a small fraction of total consumed system power in typical 802.11 radios. This is due to both the high idle energy consumption, and the low transmission duty cycle of a typical node in multi-hop shared medium wireless networks. In these networks nodes must take turns transmitting, so any particular node will only have the opportunity to transmit a fraction of the time. As a result power control strategies are fundamentally limited to reducing the overall power consumption by a fraction of 36%. This type of power saving strategy is much more useful when using much higher power radios where transmission power begins to dominate the total power consumption, and in CDMA systems where the transmission duty cycle is much higher.

B. Connected Active Subset

The intuition behind a connected active subset protocol, such as SPAN [6] or GAF [7], is that when there are many nodes close together in a multi-hop wireless network, only a subset of these nodes need to be active in order to maintain

network connectivity. These protocols strive to keep only a small subset of nodes awake in the network to provide network connectivity, and then place the rest of the nodes in a sleep state for the vast majority of the time. Often, the members of the active subset are rotated in order to distribute the energy consumption more evenly between different network nodes and to accommodate network topology changes due to mobility.

The main advantage of the connected active subset strategy is that there is little impact on communication. Packets primarily travel through nodes that are always on, and thus experience low delay. Similarly, since the subset is effectively all the non-leaf nodes of a network wide spanning tree, it is still possible to use broadcast traffic.

One main disadvantage of the active subset strategy is that it is inherently dependent on node density for energy savings [8]. The basic premise is that there are enough nodes that only a small number of them are needed at any one time. In low density networks, almost no power can be saved using this strategy because almost every node must stay active.

Another main disadvantage of this strategy is the overhead required to maintain an effective subset. Since nodes are mobile, the subset must be continually updated in order to provide complete coverage. Even if nodes were not mobile, the subset must be rotated in order to avoid completely draining the resources of a few nodes. Since coordination is required every time the subset changes, this can cause significant amounts of communication traffic which both limits scalability and reduces good-put by cutting into available medium time.

C. Asynchronous Wake-up

The idea behind the asynchronous wake-up strategy [9] is that by using a carefully designed wake-up schedule, every node in the network should be able to sleep for some fraction of the time. Furthermore, due to the schedule, the node will be guaranteed to be awake at the same time as any particular neighboring node in the network within a bounded amount of time, without requiring any type of network clock synchronization.

The main advantage of this strategy is that little coordination is required between nodes. Also since every node uses the same wake-up schedule, the network is inherently balanced in terms of equal power use by different nodes. In addition, the energy savings are independent of node density allowing efficient operation in low density networks.

However, while the asynchronous strategy has low protocol overhead and good energy efficiency, these come at the price of reduced communication quality and capabilities. The asynchronous strategy only guarantees that any two nodes will be on at the same time within a bounded time period; that guarantee does not hold for any number of nodes beyond two. In other words, all the nodes a packet must traverse along a path will not all be on at the same time, so the packet may be delayed by up to the bounded time for every hop it traverses. Similarly all of a nodes neighbors will not be on at the same time, thus traditional broadcast is also impossible. Instead

“broadcast” messages must be individually unicast to each neighbor. Since the vast majority of wireless routing protocols depend on broadcast for efficient operation, this is a major drawback of the asynchronous strategy and greatly decreases its real world practicality. In addition, the asynchronous wake-up protocol makes heavy use of beacon packets (several per second) in order to detect when neighbors are awake. Since every node must send these beacons, the scalability of this strategy can be compromised in high density networks.

D. Synchronized Wake-up

Synchronized wake-up approaches operate by obtaining and maintaining network wide clock synchronization and allowing decisions in the network to be made at specific time intervals. This type of approach is able to save the greatest amount of power, especially in idle networks, since all of the nodes in the network can turn off their radios for extended periods of time. This is able to occur regardless of network properties such as density. The other major advantage of this type of approach is that since nodes are always active at the same time, network broadcasts are still possible. This allows traditional ad hoc routing protocols to function, which depend on broadcast for efficiency. Most power saving protocols typically do not take this approach due to the difficulty in establishing network-wide synchronization.

The most well known synchronized power saving strategy is the 802.11 Power Save Mode (PSM). This protocol only works within a single hop, making it not applicable to the model we are considering. Zheng et. al. [10] provide a protocol which extends the 802.11 PSM to operate across multiple hops. Their strategy provides path activation, minimizing per packet delay. However their synchronization strategy requires MAC layer implementation in order to achieve the sub-millisecond accuracy required by the 802.11 PSM, and does not handle the case of partitions and merges which can occur in an ad hoc environment.

The Pulse protocol is also a synchronized wake-up approach. Therefore it allows broadcast and allows all the nodes in the network to power off their radios when the network is idle. Our protocol also uses path activation to eliminate per hop delay, but differs from the existing synchronized protocols in that the time scale is much larger, and that a proactive routing service is provided simultaneously to the power saving functionality. The larger time scale of the Pulse protocol allows it to operate with much coarser time synchronization (on the order of 10 milliseconds) which can be implemented without MAC layer integration.

IV. PULSE PROTOCOL

A. Overview

The protocol design is centered around a flood we refer to as a *pulse*, which is periodically sent at a fixed *pulse interval*. This pulse flood originates from infrastructure access nodes (*pulse sources*) and propagates through the entire ad hoc component of the network. This rhythmic pulse serves two functions simultaneously. It serves as the primary routing

mechanism by periodically updating each node in the networks route to the nearest pulse source. Each node tracks the best route to the pulse source by remembering only the node from which it received a flood packet with the lowest metric. The propagation of the flood forms a loop free routing tree rooted at the pulse source. In addition, it is used to provide network-wide time synchronization.

If a node needs to send and receive packets, it responds to the flood with a reservation packet. This reservation packet is sent up the tree to the pulse source. The reservation packet contains the address of the node making the reservation, and is used to setup reverse routes at all nodes on the path between the pulse source and the sending node. This reservation mechanism operates similarly to the route response mechanism used in AODV [11]. Note that it is unnecessary for a node to send a reservation packet in response to the flood, unless it has packets to transfer. A node that is actively communicating must send a reservation packet for every pulse it receives to keep the reverse route fresh. When a node has not sent or received packets for at least a complete pulse interval, it no longer sends a reservation packet in response to the pulse.

The Pulse protocol uses the time synchronization provided by the flood to create a fixed period of time during which all nodes in the network are active. During this *pulse period*, the pulse flood propagates, and nodes can reply with reservation packets. Since a node that does not send or forward a reservation packet will have no packet forwarding responsibilities until the next pulse occurs, it may place its radio in sleep mode until the next pulse period begins. This node deactivation is what allows the Pulse protocol to conserve power.

The ratio between the pulse period and the pulse interval determines the duty cycle of the protocol. This duty cycle is the primary factor that determines the idle power consumption of every node in the network. Therefore, reducing the pulse period results in increased energy efficiency. However, the pulse period must be long enough so that the pulse flood and reservation packets can be delivered. In order to minimize this time, data traffic is halted, eliminating contention between data packets and the flood.

Fig 1 shows the Pulse protocol in an example network. Every node in the example network has a route towards the pulse source as indicated by the grey arrows. Both nodes A and B are actively communicating, and have each sent a reservation packet up the tree to the pulse source. The reservation packets have setup reverse routes as indicated by the black bi-directional arrows. Nodes that have forwarded a reservation stay on to forward data and are colored black. The rest of the nodes in the network may turn off until the next pulse.

The Pulse protocol exhibits several features of both proactive and on-demand protocols. While the Pulse flood proactively maintains a route from all nodes in the network to the pulse source, reverse routes are established on-demand, but maintained proactively. Since idle nodes in the network power off their radios, a node attempting to initiate a connection must wait until the following pulse to reserve a route. This results in

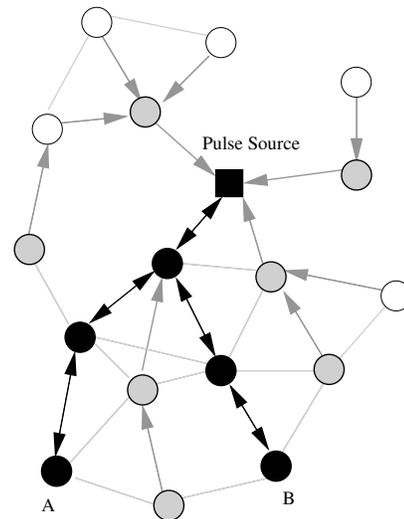


Fig. 1. Pulse protocol example

an average route acquisition delay of half a pulse interval. This concept of path acquisition latency is similar to that exhibited by on-demand protocols.

B. Design Methodology

The goal of the Pulse protocol is to provide multi-hop infrastructure access to mobile users. The traffic pattern in the proposed model consists primarily of communication between mobile users and fixed infrastructure. The intuition behind our protocol design is that performance can be gained by exploiting the fact that almost all communication in the network shares a common end-point.

The periodic pulse flood exploits the communication concentration at the pulse source by providing every node in the network with a continuously updated route. Infrequently, nodes in the network may need to establish peer to peer connections, which are relayed through the pulse source. This results in all of the routes in the network leading to the pulse source and eliminates the need for any additional routing overhead.

One unique quality of the Pulse protocol is its inherent scalability according to many metrics. The protocol scales to large networks with regard to coverage area by allowing the simultaneous operation of multiple pulse sources. Additionally, the multi-hop nature of the protocol allows each pulse source to cover a much greater area than the traditional access point model. Also, since all other routing traffic aside from the periodic pulse is unicast, the route acquisition process creates only local traffic on the network. In contrast, traditional on-demand protocols must flood and re-flood the network for each active connection in order to establish and maintain routes.

Scalability to high levels of mobility is provided by the proactive pulse flood. As the mobility level increases, many route failures begin to occur throughout the network. In the Pulse protocol, all broken routes are repaired simultaneously within one pulse interval using one flood and one unicast for every active node. In contrast, an on-demand protocol may

initiate one flood and one unicast for every broken active route, a proactive link-state protocol may generate one flood per link failure. As the number of failures increases, this results in congestion due to the additional routing overhead, limiting the scalability of these protocols to high levels of mobility. In addition, if a hello protocol is used instead of link layer feedback, a link failure is typically detected when two consecutive hello packets have been missed. The default pulse interval used in our simulations is 2 seconds, which allows the fault to be repaired before a typical hello protocol would even detect it.

The Pulse protocol requires that nodes are always powered on during the pulse period and that no data packets are sent during this time interval. The pulse interval used for simulations was 2 seconds, of which 112 milliseconds were required for the pulse period. This ratio results in the protocol consuming 5.6% of the available network resources. A number of factors come as a result of this decision. The total bandwidth available to nodes in the network is limited to 94.4% of the actual bandwidth as a result of this fixed overhead. These timings determine the duty cycle of idle nodes in the network. Nodes which are not communicating or forwarding packets are required to be active 5.6% of the time to participate in the protocol, but can place their radios in a sleep mode for the remaining 94.4% of the time. While the overhead of many routing protocols, particularly those which function on-demand, increases as a result of increased node mobility, route failures, high node density, or a sudden increase in the number of traffic sources, the pulse protocol's overhead remains fixed. The effectiveness of this technique is best seen through our simulation results in Section V.

C. Timing and Phases

The Pulse protocol continuously cycles through four distinct phases. Fig 2 indicates these phases and visually depicts the duty cycle of the two second pulse interval used in the simulation section. Nodes must power on before the anticipated pulse arrival time to ensure that it is not missed due to a synchronization error, this period is labelled as *Early Power On* in the diagram. An initial upper bound on this period would be a full network diameter, which we define as the amount of time for a flooded packet from the pulse source to reach every node in the network, since every node in the network would be synchronized with at least that precision. A more accurate mechanism, described below, allows this time to be significantly smaller in practice. The next phase is referred to as *Forward Pulse Flood*. During this time interval the pulse is flooded to all nodes in the network. This requires a full network diameter to reach all of the nodes. The protocol then enters the *Forward Reservations* which allows enough time for any reservation packets to be forwarded back to the pulse source. This period of time has to be long enough such that the last node in the network that receives the pulse flood is able to return a reservation packet to the source before the nodes in the network enter the next phase. Again, this requires a full network diameter worth of time. The next period, labelled *Data Transfer / Sleep* in the diagram, is where nodes which

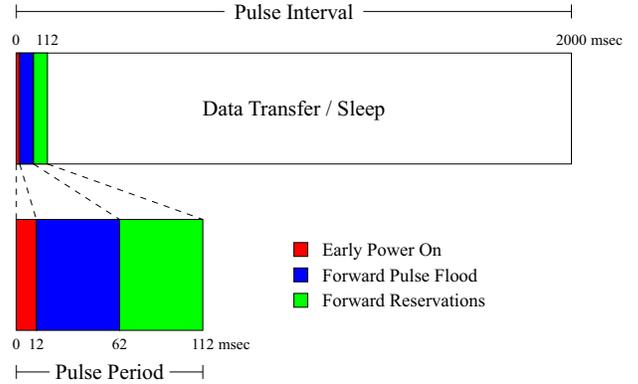


Fig. 2. Pulse Protocol Timing Diagram

did not send or forward reservation packets power off their radios until they need to wake up just before the next pulse. Nodes which have been reserved remain on and take part in actively transferring data during this period of time.

D. Flood Propagation

The pulse flood originates at the pulse source, and is sent at a fixed time interval. Several parameters are used to tune the flood for fast propagation, high node coverage, and good path selection. The flood provides both routing and synchronization, so it must be tuned to serve both needs simultaneously.

A pulse packet contains only a few fields: a sequence number, a cost metric used for route selection, and an accumulated delay timer used to increase the time synchronization accuracy. This keeps the size of the packet to a minimum, increasing the number that can be transmitted in a small amount of time.

Two timing parameters govern the flood propagation: jitter and delay. Upon receiving the first pulse packet, a node sets a timer for retransmission of the pulse packet. A uniform random number between delay and delay + jitter is selected for this timer. When the timer expires, the pulse packet is retransmitted with an incremented cost field, and the retransmission delay added to the accumulated delay field. The random retransmission jitter is a well known technique used by many flooding protocols to help prevent collisions between nodes that received the same broadcast. The fixed delay is a mechanism used by the pulse protocol to enhance the initial accuracy of the routing metric. Adding a fixed delay dramatically increases the chance that the first pulse packet heard will have the lowest cost metric. This is a desirable feature for the pulse protocol, because a node must reserve a route almost immediately upon hearing the pulse flood in order to meet the tight timing requirements needed for low power operation. A node is committed to a path once it is reserved, even if knowledge of a better path becomes available. The fixed delay maximizes the chance that the best path will be known before the path is reserved. As an optimization, fixed delay is not used by nodes within two hops of the pulse source, since these nodes always hear the best path first.

E. Time Synchronization

Nodes in the network must acquire and maintain accurate synchronization with the pulse source in order to function effectively. Acquisition is accomplished by remaining in a listening state until a pulse flood is received. Each flood packet contains a relative time offset which represents the amount of time elapsed since the pulse flood was initiated. Using the received time, the offset, and its own local oscillator, a node can predict when the next pulse flood will be sent by the source.

Since the offset in the flood packet does not include all sources of delay the flood packet may have experienced (such as MAC contention delay), and since the local oscillator is not perfect, the time sync is only partially accurate. In order to compensate for this, each node keeps track of the earliest pulse start time received over all recently received pulses. In addition, every node wakes up a sync interval early in order to avoid missing the pulse flood due to an imperfect sync. In the event that a node misses the pulse flood, it will remain in a listening state until it can re-acquire synchronization on the next flood.

F. Path Reservation

Reservation packets serve two purposes. They activate intermediate nodes so that they remain awake in order to forward the data packets and they create reverse route entries to the set of active nodes. When a node in the network receives a pulse update packet and has data to transfer, it creates a reservation packet and forwards it immediately (once it has committed on a route) up the tree to the pulse source. A reservation packet contains a list of node IDs, initialized to the ID of the node which created the reservation and a cost field initialized to the node's distance from the pulse source. When an intermediate node forwards a reservation packet it creates reverse routes to the nodes indicated by the list of IDs, sets the cost field equal to its own distance to the pulse source, and appends its node ID if it has not already forwarded or sent a reservation. After all of the reservation packets have been received by the pulse source, it has reverse routes to all of the active nodes in the network.

All other nodes in the network that do not send or forward reservation packets turn their radios off at the end of the pulse period. Nodes that are sending data for long periods of time need to resend a reservation packet in response to every pulse flood. This is required since the topology could be continuously changing. Since every time the pulse update is flooded through the network a new tree could be formed, nodes with active connections need to repeatedly send a reservation packet in response to each pulse, so that the current nodes that form its path remain on. Nodes that are not in the process of transferring data do not have to ever initiate a reservation packet.

The Pulse protocol requires a mechanism which allows it to overhear reservation packets sent by neighboring nodes. This can be accomplished by either enabling promiscuous operation or broadcasting reservation packets. Broadcasting

reservation packets would require an additional reservation acknowledgement to increase their reliability.

Nodes track the least cost reservation that they overhear. If a node has overheard a reservation packet, but is not part of an active path and thus goes to sleep, the node can perform a *fast activation* if an application data transfer is initiated. This means that the node can turn on its radio immediately and send through the node it overheard the best reservation from. Therefore, any node adjacent to an active path avoids the delay incurred waiting for the next pulse period. The nodes that are capable of fast activation are colored grey in Fig 1.

In addition, when a node overhears a reservation packet it creates reverse route entries through the node which it heard the reservation from if the entries do not already exist. This mechanism allows a node to have reverse routes to all active nodes in both its sub-tree and the sub-tree of its neighbors. This can allow peer-to-peer packets to go directly to the destination without first passing through the pulse source.

These reverse routes also help prevent packet loss caused by synchronous route switching. This type of loss can occur when a packet is halted by an intermediate node on its way down the tree due to the start of a pulse period. Since the route is refreshed during the pulse period, the intermediate node might not be on the new path. However, if the intermediate node is adjacent to the new path, it overhears the destination's reservation packet, installs a route, and can deliver the packet. Otherwise, the intermediate node would not have a route and would drop the packet.

G. Paging

In the event that packets arrive at the pulse source destined for a node that does not have a currently active path, the pulse source will page the node on the next pulse flood. Paging simply involves placing the node's id in the pulse flood packet. When a node receives a flood packet containing its id, it responds with a path reservation packet. This activates the path and sets up the route from the pulse source to the node. Thus data packets can be delivered to nodes that are not currently active. This can occur when data has not been sent for a while on an open connection, or when a new connection is being initiated to an ad hoc node (from either the infrastructure network or another ad hoc node).

H. Multiple Pulse Source Integration

One advantage of the Pulse protocol is that it can be operated using several infrastructure attached pulse sources. This is useful in the case where high performance and wide coverage area are desirable. In order for several pulse sources to operate together, they must all be reachable via the infrastructure network. All the pulse sources must use the same pulse interval, and must all be synchronized with each other (i.e. the pulse should start at the same time from every pulse source). This can be accomplished using a traditional network time sync protocol such as NTP over the infrastructure network. The pulse flood then originates from several points in the ad hoc network and propagates until reaching the edge

of the network or the flood from another pulse source. Each node tracks the nearest pulse source. Thus each pulse source becomes the center of a multi-hop cell. Nodes can move through the network roaming seamlessly between different pulse sources. Pulse sources must also coordinate to make sure packets from the infrastructure network are routed to the appropriate pulse source on their way to the final destination node, however the details of this coordination are not the specific focus of this paper.

V. SIMULATION

The simulations in this work are run in version 2.1b9a of NS2[12], and are designed to emulate the proposed multi-hop infrastructure access model. All communication occurs with a single stationary node that is placed in the center of the network. When using the Pulse protocol, this node serves as the pulse source. 802.11 radios with a bandwidth of 2 Mbps and a nominal range of 250 meters are used. The RTS threshold is set to 128 bytes to allow small unicast routing packets to be sent without using virtual carrier sense. NS2 default settings are used for all protocols.

A. Timing Parameter Selection

An implementation of the Pulse protocol was created in version 2.1b9a of the NS2[12] network simulator. An initial set of experiments were conducted in order to find appropriate values for the protocol timing parameters. The purpose of these experiments is to show the relationship between network scenarios and the timing values required for good protocol operation. In order to accomplish this, we use a set input variables to produce a wide range of scenarios and measure the performance of various aspects of the protocol under these scenarios.

The input variables consisted of the: physical network size, node density, flood repeat delay, and flood repeat jitter. Using these input variables, many random static networks are generated, and the Pulse protocol is run for several pulse periods in each. During these simulations, data was gathered on the synchronization error, delay in receiving the pulse, and path length optimality. Ninety-ninth percentile summary statistics are computed from this data in order to represent a worst case metric. Each combination of physical network sizes (square side length) of 1, 2, and 4 kilometers, node densities of 50, 100, and 200 nodes per square kilometer, flood delays and jitters from one to ten milliseconds were all simulated. The results of these simulations indicate that the parameters listed in the first part of Table II should provide reasonable performance in networks up to 2km by 2km with all simulated node densities.

The worst case path optimality metric confirms that high quality paths are selected using these flooding parameters. The multiplicative path length increase is used to judge path optimality. The multiplicative path length increase is computed by dividing the chosen path length by the best possible path length. This metric more heavily penalizes path length increases on short paths than the traditional additive path length

TABLE II
PULSE PROTOCOL PARAMETERS

Flood Retransmission Delay	4 msec
Flood Retransmission Jitter	1 msec
Power On Before Pulse	12 msec
Flood Propagation	50 msec
Reservation (estimated)	50 msec
Pulse Interval	2 sec

increase metric. This is appropriate because an additional hop causes a greater performance degradation for short paths than it does for long paths. The worst case metric results show a path length increase of only 2%, 5%, and 11% for the 50, 100, and 200 node densities of the 2km by 2km network. This near linear relationship with density is caused by the increased likelihood of collisions due to the greater number of senders in range of each other.

The remaining timings in the second part of Table II were not directly calculated by the simulations. The reservation time is estimated as being no greater than the flood propagation time; both are approximately one network diameter, and the reservation packets are not artificially delayed. The pulse interval must be chosen to provide a good compromise between energy savings and activation delay. We have selected a value of 2 seconds in order to provide high power savings while keeping the worst case activation delay low.

These parameters are used in every simulation in this section, regardless of actual network size or node density. While this results in less energy savings for small sized networks where the timings could be tightened, having one set of parameters that functions in a range of networks results in greater deployment flexibility.

B. Simulation Setup

The traffic pattern is different than what has been commonly studied. In addition to all nodes communicating with a single end point, we use a random exponentially distributed on/off traffic generator. The use of this generator allows every node in the network to be a traffic source, as opposed to a small number of nodes sending fixed rate (CBR) flows. Each node stays off for an exponentially distributed length of time with a specified average, then comes on and sends at a fixed rate (10 kbps using 512 byte packets) for an exponentially distributed amount of time with an average of ten seconds, then repeats the process. This traffic model has a number of properties. By adjusting the average off time, any average offered load can be achieved. In addition, since the load is composed of fixed rate flows, setting the offered load simultaneously determines the average number of active flows (e.g. setting an offered load of 0.2 Mbps results in an average of 20 flows active at a time). Finally, this on/off scheme continuously changes the set of active flows. The average on time and average number of active flows determines the rate of change (e.g. an offered load of 0.1 Mbps and an average on time of 10 seconds results in an average of 10 active flows with one flow changing per second).

A modified random way-point mobility model is used in the simulations. The modifications are designed to address the concerns raised in [13] about the validity of the standard random way-point model. In order to achieve more steady mobility characteristics, nodes select a speed uniformly between 10% and 90% of the given “max” speed. This helps ensure that the average speed does not drop drastically over the course of the simulation. In addition, 300 virtual seconds of mobility are generated before the start of the simulation. When the simulation starts, nodes are already in motion. This allows the average speed and node distribution to stabilize before the simulation starts. In our simulations, pause time is always set to zero, and the level of mobility is controlled by changing the maximum speed parameter. Unless otherwise stated, 300 seconds are simulated.

C. Routing Evaluation

In order to evaluate the effectiveness of the Pulse protocol, we must examine not only the amount of energy savings, but also its ability to function as a routing protocol in a mobile multi-hop wireless network. A protocol that seriously compromises network performance would not be useful in the proposed model no matter how much power it saved.

In this experiment our goal is to evaluate the network performance of the Pulse protocol by comparing it with both AODV [11] and DSR [14], two on-demand ad hoc wireless network routing protocols. Neither protocol is specifically designed to save power, however the on-demand approach attempts to minimize routing overhead. It should be reiterated that neither AODV or DSR were originally designed for the single destination infrastructure access environment we are simulating in this paper. They were both primarily designed to support the peer to peer traffic patterns found in ad hoc networks. However, infrastructure access is one of the primary potential uses of a multi-hop wireless network. Therefore, it is logical to evaluate the performance of these protocols in this type of model.

Fig 3 shows several dimensions of information regarding the performance of the three tested routing protocols. The page x-axis shows three network sizes. The page y-axis shows four levels of mobility. For each combination of network size and mobility, a sub-graph is shown. Each sub-graph x-axis shows the average offered load produced by the on/off traffic generators, and each sub-graph y-axis shows the resulting average delivery ratio. This figure is setup so that the degree of difficulty increases as the scenario is located further up and more to the right on the page.

The most striking feature apparent in these results is the performance of the Pulse protocol under high mobility (top of the page). These results illustrate the effectiveness of the Pulse protocol design. Its proactive route maintenance and low fixed routing overhead, even under a large number of simultaneous faults, yields delivery ratios that are only minimally reduced even at the highest simulated levels of mobility (20 m/s max speed). The delivery ratios of the on-demand protocols drop significantly as mobility is increased.

The two smaller network sizes simulated are actually networks of the same physical size (1km by 1km) but different node densities (50 vs. 100 nodes per square kilometer). Little difference in the delivery ratios is seen between these two densities. Although the largest simulated network contains 200 nodes and significantly different delivery ratios, it has a much larger physical size of 2km by 2km and thus has a node density of only 50 nodes per square kilometer. The lower delivery ratios in this larger network are due to the fact that the average number of hops a packet must traverse has been greatly increased, this results in the network reaching saturation at a much lower offered load than in the 1km by 1km networks. In order to specifically isolate node density, we conducted an additional set of experiments. Using a 1km by 1km - 5 m/s max - 0.2 Mbps offered load scenario, we varied the node density from 50 to 700 nodes per square kilometer (greater node densities were not possible due to logistical constraints). The pulse protocol was able to achieve average delivery ratios of greater than 98.7% in all simulated densities.

It is interesting to note the wide gap between the performance of the AODV and DSR protocols. In these simulations, the DSR protocol significantly out performs AODV in almost all scenarios. We believe that this difference can be attributed to DSR’s aggressive route caching using promiscuous listening, and shorter default timings for route request propagation and retry. In general, the route caching strategy used by on-demand protocols is not well tuned for infrastructure access networks. While the entire network is updated with a route to a mobile node during the route request flood, the much more useful fresh route to the gateway node is only provided to nodes along the reply path. However, since DSR promiscuously listens to packets on the medium, any node adjacent to the discovered path overhears the route response, and can add that information to its route cache. This aggressive caching is particularly effective in infrastructure access networks since all of the traffic is destined for the same node. This greatly increases the cache hit rate when compared with traditional peer-to-peer traffic patterns.

In order to evaluate the delay characteristics of the routing protocol, the 1km x 1km - 100 node - 5 m/s max scenario was simulated (Fig 4). The graph displays the average per packet end-to-end delay of both the Pulse and DSR protocols. AODV is omitted because its delivery ratio is not comparable in this scenario. These results indicate that in practice the additional delay incurred by the power saving aspects of the Pulse protocol (path activation delay and data halting during the pulse period) have little effect on the overall average delay. One of the main factors that makes this true is the *fast activation* technique described above. This technique allows any node that is adjacent to an active path to activate immediately without waiting for the next pulse period, and drastically reduces the impact of activation delay.

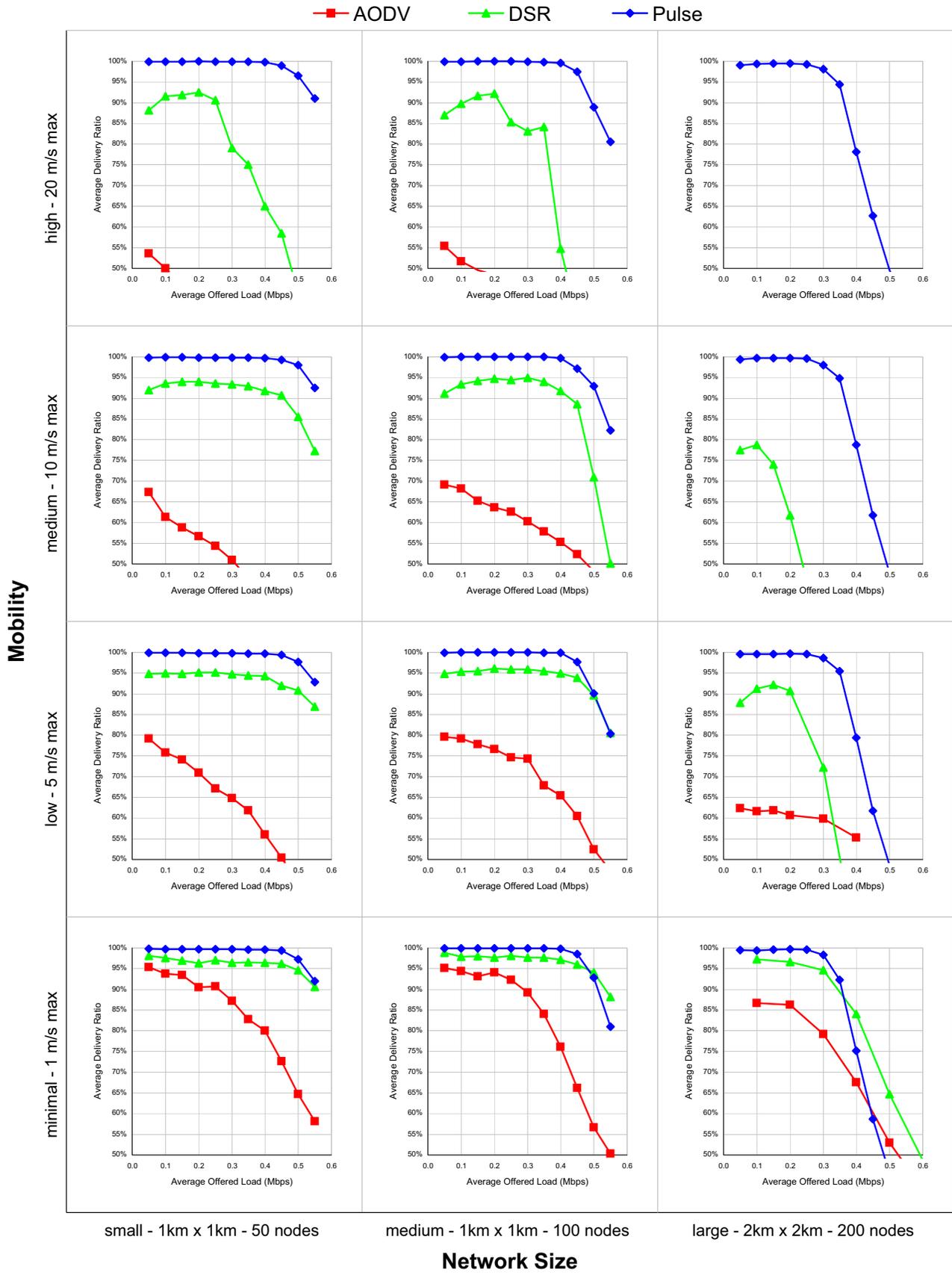


Fig. 3. Routing evaluation results using random way-point mobility and exponential on/off traffic

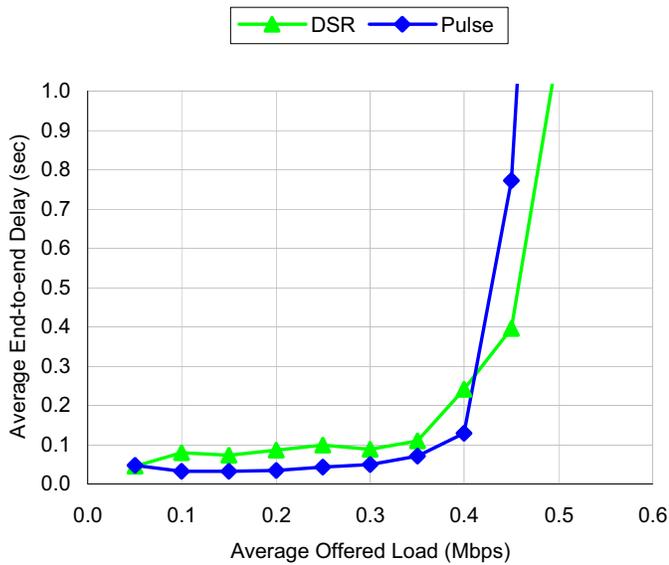


Fig. 4. End-to-end delay in the 1km x 1km - 100 node - 5 m/s max scenario

D. Energy Conservation Evaluation

Fig 5 shows the average per node power consumption versus the average offered load in the 1km x 1km - 100 node - 5 m/s max scenario. This particular case was selected since it seems to be representative of a typical infrastructure access environment.

As expected from protocols that were never originally designed with power saving in mind, AODV and DSR both burn energy at an almost equal rate. The average power consumption for these protocols is completely dominated by idle energy consumption. The additional energy used for the transmission and reception of packets results in a relatively small increase in the average power consumption.

In contrast, the average power used by a node running the Pulse protocol is substantially less. We see a savings over the DSR protocol of between 23% and 78% depending on offered load. The strong linear relationship between offered load and energy consumption is a direct result of the path activation feature of the Pulse protocol. This feature causes all nodes that are sending, receiving, or forwarding traffic to enter a full power on state in order to maximize network performance. As a result, the average power usage is directly related to the fraction of nodes that are activated. There is also a direct relationship between the offered load and the number of simultaneously sending nodes when using our exponential on/off traffic generator. As the network load increases, the number of senders increases, which determines the fraction of active nodes in the network. The fraction of active nodes determines the final average power consumption. If the load is increased to the point where every node in the network was transferring packets, the Pulse protocol would use virtually the same amount of power as an on-demand protocol. At the opposite extreme, when there is no load on the network, the power reduction capabilities of the Pulse protocol

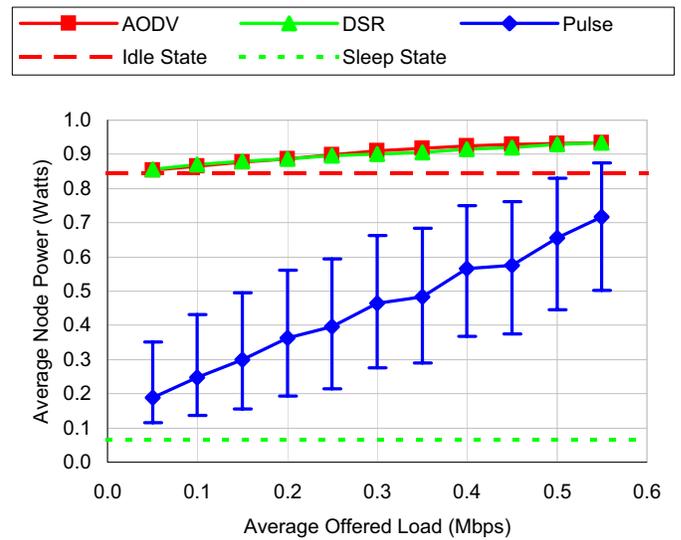


Fig. 5. Energy consumption in the 1km x 1km - 100 node - 5 m/s max scenario

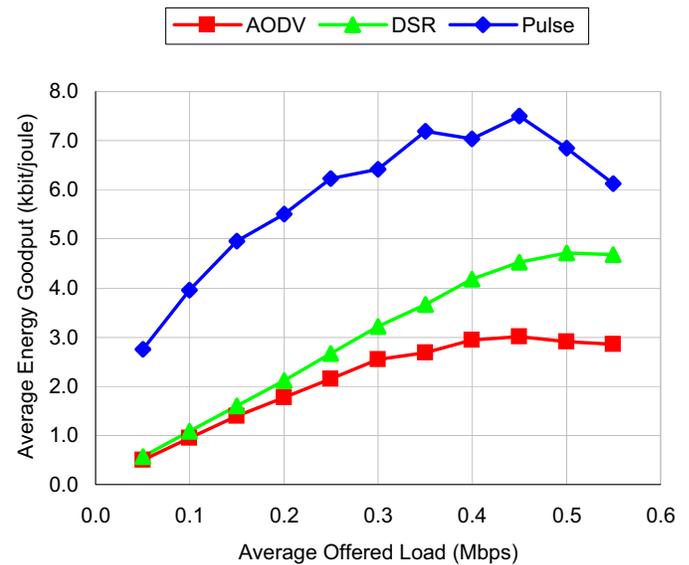


Fig. 6. Energy goodput in 1km x 1km - 100 node - 5 m/s max case

have the maximum effect. This is appropriate for the target infrastructure access model where the majority of nodes are expected to be idle at any particular time.

Fig 6 plots energy goodput (kilobytes delivered per joule of energy consumed) versus the offered load. This shows that even though the average power usage increases with higher offered loads, the energy efficiency also increases. In other words, the higher energy consumption rate is offset by the higher throughput obtained, increasing the overall efficiency. We see that the efficiency continues to increase until the network reaches saturation. At this point, congestion prevents further throughput increases. Since DSR and AODV consume energy at an almost the constant rate regardless of load, their energy efficiency is directly related to the throughput they

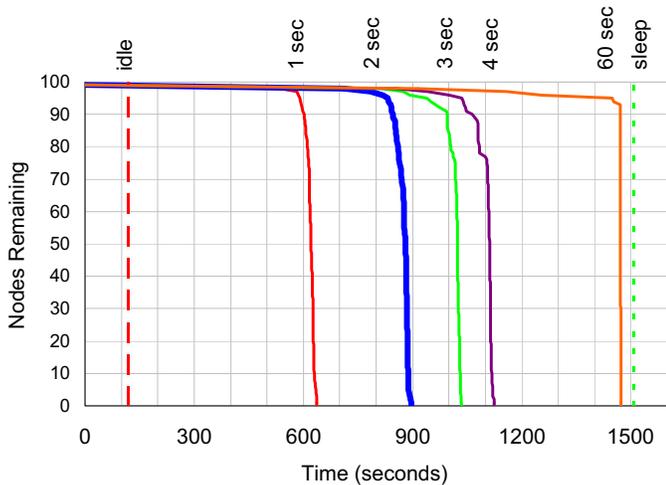


Fig. 7. Idle network lifetime in 1km x 1km - 100 node - 5 m/s max scenario

obtain. Thus each protocol shows a linear increase in efficiency with offered load until the protocol reaches saturation. The higher efficiency of DSR is due to its higher delivery ratio in this scenario. The Pulse protocol achieves a 1.3 to 4.8 times increase in energy efficiency over the DSR protocol in the simulated scenarios.

E. Idle Network Lifetime

A set of experiments were conducted to investigate the idle network lifetime as a function of the pulse interval. These experiments were conducted in the 1km x 1km - 100 node - 5 m/s max scenario. Each mobile node in the network is given a battery that provides 100 joules of energy, and the simulation is run until all nodes have exhausted their energy supply. A series of trials were conducted where the pulse interval was set to 1, 2, 3, 4, and 60 seconds. The 2 second interval used in the above experiments is hi-lighted for reference. Increasing the pulse interval increases the route acquisition latency, but also results in a lower duty cycle which corresponds to additional power savings. In these experiments, no traffic was generated except for the periodic pulse floods. This simulates a network where most of the devices are on but not being used (as would usually be the case with a cell phone or PDA).

The number of remaining nodes as a function of time for each of the simulations is shown in Fig 7. Also shown is the lifetime of a node that is always in the idle state, and the lifetime of a node that is always in the sleep state. A network of nodes running a pure on-demand protocol would always be in the idle state with no traffic flow, and in this setup all nodes would expire at 119 seconds. Even at the fastest pulse interval setting of 1 second, the lifetime of the network is increased over five times, despite the overhead of providing proactive routes to every node in the network. In the 2 second pulse interval case used in the simulations above, the network lifetime is increased approximately seven and a half times. This 2 second interval lifetime extension is significantly greater than the published results for protocols using the

connected active subset scheme (GAF [7] and SPAN [6]), and is comparable to the extension provided by the synchronous on-demand protocol in [10] and by the most aggressive power saving variant of the asynchronous protocol in [9].

The sleep state represents an upper bound on the performance of any power saving protocol operating under the given power model (see Table I), as it is not possible to do better than a network of nodes that *never* power on their radios. We can see a clear relationship between the length of pulse interval and the resulting network lifetime. As the pulse interval increases, the network lifetime begins to asymptotically approach the upper bound. This shows that there is a clear tradeoff between path activation latency and energy savings. The 60 second interval lifetime shows that it is possible to tune the pulse protocol to achieve near ideal levels of energy saving in low performance networks where route acquisition latency is not a major concern. This option may be particularly useful for sensor networks where energy saving is the primary concern.

The idle network lifetime results show that the Pulse protocol does an excellent job of conserving power for all nodes in the network simultaneously. This is indicated by the relatively sharp transition from all nodes being alive to all nodes being dead. In contrast, the connected active subset schemes usually have a much more gradual transition since critical nodes in low density portions of the network are often selected as members of the subset and receive virtually no lifetime extension. This behavior can be seen in the published results for both GAF and SPAN.

VI. CONCLUSION

We have presented the Pulse protocol, an energy efficient protocol for ad hoc infrastructure access. An extensive set of simulations have demonstrated that this protocol is effective at both routing and conserving energy. Compared with existing on-demand routing protocols, the Pulse protocol was able to match or exceed their delivery ratios under a wide range of network sizes, mobilities, node densities, and traffic loads. In addition, the protocol was shown to extend the idle network lifetime by over 7.5 times. These results indicate that the Pulse protocol is appropriate for multi-hop infrastructure access, particularly when high performance, scalability, and energy efficiency are simultaneously desired.

REFERENCES

- [1] Ying-Dar Lin and Yu-Ching Hsu, "Multihop cellular: A new architecture for wireless communications," in *IEEE INFOCOM*, 2000.
- [2] Laura Marie Feeney and Martin Nilsson, "Investigating the energy consumption of a wireless network interface in an ad hoc networking environment," in *IEEE INFOCOM*, 2001.
- [3] V. Rodoplu and T. Meng, "Minimum energy mobile wireless networks," in *EEE International Conference on Communications, ICC'98*, June, vol. 3, pp. 1633–1639.
- [4] J. Chang and L. Tassiulas, "Routing for maximum system lifetime in wireless ad-hoc networks," in *37-th Annual Allerton Conference on Communication, Control, and Computing*, September 1999.
- [5] Jae-Hwan Chang and Leandros Tassiulas, "Energy conserving routing in wireless ad-hoc networks," in *INFOCOM*, 2000, pp. 22–31.
- [6] Benjie Chen, Kyle Jamieson, Hari Balakrishnan, and Robert Morris, "Span: An energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks," *ACM Wireless Networks*, vol. 8, no. 5, September 2002.
- [7] Ya Xu, John Heidemann, and Deborah Estrin, "Geography-informed energy conservation for ad hoc routing," in *Proceedings of the seventh annual International Conference on Mobile Computing and Networking MOBICOM*, 2001.
- [8] Douglas M. Blough and Paolo Santi, "Investigating upper bounds on network lifetime extension for cell-based energy conservation techniques in stationary ad hoc networks," in *Proceedings of the eighth annual international conference on Mobile computing and networking*. 2002, pp. 183–192, ACM Press.
- [9] Rong Zeng, Jennifer C. Hou, and Lui Sha, "Asynchronous wakeup for ad hoc networks," in *The Fourth Annual Symposium on Mobile Ad Hoc Networking and Computing*, 2003.
- [10] Rong Zeng and Robin Kravets, "On-demand power management for ad hoc networks," in *IEEE INFOCOM*, 2003.
- [11] Charles E. Perkins and Elizabeth M. Royer, *Ad hoc Networking*, chapter Ad hoc On-Demand Distance Vector Routing, Addison-Wesley, 2000.
- [12] "The network simulator - ns2," <http://www.isi.edu/nsnam/ns/>.
- [13] Jungkeun Yoon, Mingyan Liu, and Brian Noble, "Random waypoint considered harmful," in *IEEE INFOCOM*, 2003.
- [14] David B. Johnson, David A. Maltz, and Josh Broch, *DSR: The Dynamic Source Routing Protocol for Multi-Hop Wireless Ad Hoc Networks*. in *Ad Hoc Networking*, chapter 5, pp. 139–172, Addison-Wesley, 2001.