The Pulse Protocol:
Mobile Ad hoc Network Performance Evaluation

Baruch Awerbuch, David Holmer, and Herbert Rubens
Department of Computer Science
Johns Hopkins University
Baltimore, MD
{baruch, dholmer, herb}@cs.jhu.edu
Technical Report 12/10/2003

***Abstract*— We present a performance evaluation of the Pulse protocol operating in a peer-to-peer mobile ad hoc network environment. The Pulse protocol utilizes a periodic flood (the *pulse*) initiated by a single node (the *pulse source*) to provide both routing and synchronization to the network. This periodic pulse forms a pro-actively updated spanning tree rooted at the pulse source. Nodes communicate by forwarding packets through through this tree. In addition, nodes are able to synchronize with the periodic pulse, allowing idle nodes to power off their radios a large percentage of the time when they are not required for packet forwarding. This results in substantial energy savings. Through simulation we explore the performance of the protocol with respect to packet delivery ratio, delay, and energy efficiency.**

I. INTRODUCTION

The Pulse protocol [1] was originally presented as an energy efficient multi-hop infrastructure access routing protocol. In that work, nodes in the network tracked the least cost path to the nearest gateway and peer-to-peer traffic was assumed to be minimal. Under the infrastructure model, the protocol exhibited excellent performance with regard to mobility,scalability, and energy efficiency. This work evaluates the performance of the Pulse protocol under the more general peer-to-peer mobile ad hoc network model in order to determine if these desirable properties are preserved.

Typically, on-demand protocols such as DSR [2] and AODV [3], or proactive protocols such as OLSR [4] and TBRPF [5] have been preferred for peer-to-peer mobile ad hoc networks. Each of these protocols is an example of the *direct routing* strategy, where the route from source to destination is constructed by finding the shortest path using a given metric.

The Pulse protocol breaks from this tradition by employing a *tree routing* strategy. A spanning tree is constructed by a flood initiated by an elected node (the *pulse source*). Packets are then routed using a tree traversal. In other words, a packet originating from a source will be sent up the tree (towards the pulse source) until it reaches a node that is a parent of both the source and destination, then the packet is sent down the tree to the destination. Routing trees have been commonly used for multi-cast and broadcast traffic, but using trees for unicast traffic is not common. While this strategy results in paths that are longer than the direct routing strategy, it has inherent scalability benefits.

For example, the proactive pulse flood provides scalability to high levels of mobility. As the mobility level increases, many failures begin to occur throughout the network. In the Pulse protocol, all broken routes are repaired simultaneously within one pulse interval using one flood. In contrast, an on-demand protocol may initiate one flood for every broken active route, and 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, the Pulse protocol offers integrated energy saving functionality. The periodic pulse also serves as a network wide synchronization protocol. When nodes are not required for packet forwarding, they may power off their radios for the time between pulse floods. This results in substantial energy savings and drastically increases the lifetime of the network.

Our Contribution. We present a performance evaluation of the Pulse protocol under a general peer-to-peer mobile ad hoc network model. In this model, all nodes are mobile and no assumption about the location of the pulse source is made. In addition the power consumption of the pulse source must be included in the analysis.

Through simulation we explore the performance of the Pulse protocol by comparing it with DSR, an established on-demand mobile ad hoc networking protocol. We evaluate both protocols under a wide range of conditions by varying mobility, network density, network load, and number of flows. The delivery ratio, end-to-end delay, and energy consumption are evaluated.

II. PULSE PROTOCOL

In this section we provide a review of the Pulse protocol features and operation. In addition, practical considerations for deployment in mobile ad hoc networks are discussed.

A. Overview

The protocol design is centered around a flood (the *pulse*) which is periodically sent at a fixed *pulse interval*. This pulse flood originates from the *pulse source* and propagates through the entire ad hoc 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 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 [3]. Note that it is not necessary for a node to send a reservation packet in response to the flood, unless it is actively transferring packets. 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.

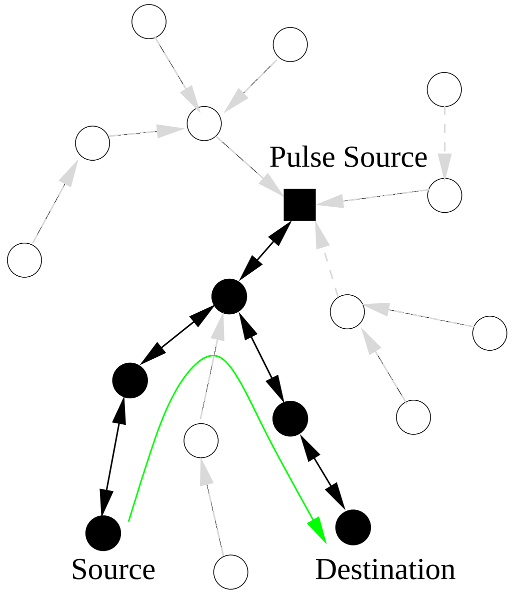

Fig. 1. Basic Pulse Protocol Example

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.

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).

Figure 1 shows the basic operation of the Pulse protocol in an example network. Every node in the example network has a route towards the pulse source as indicated by the dashed arrows. Since there is an active flow from the source to the destination, each sends a reservation packet up the tree to the pulse source in response to the pulse flood. The reservation packets have setup reverse routes as indicated by the solid 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 actual path of the traffic flow, as it traverses the tree from the source to destination, is also indicated in the figure.

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. Similarly to other on-demand protocols, the establishment of a route can result in activation delay. In the Pulse protocol, the worst case activation delay to establish a peer-to-peer connection is two pulse intervals (2 seconds with a one second pulse interval). The worst case occurs when a node receives a packet from the application layer immediately after the current pulse period ends. The node would need need to wait a full interval before it could send a reservation during the next pulse period. After the pulse period it can begin transmitting data. However, when the pulse source receives the data, it may have to wait an additional pulse interval until the next pulse period in order to page the destination.

During the reservation period nodes promiscuously listen to the reservation packets. If they overhear a reservation packet then they know that they are in range of a node which will be active during the next period. Thus any node that neighbors an active path can perform a *fast activation* if they have a data packet to send, meaning they turn on in the middle of the data transfer period and forward data to their active neighbor. Fast activation eliminates the need for nodes to wait until the next pulse interval, resulting in reduced activation delay.

In addition, when a node overhears a reservation packet it creates reverse route entries through the node which it heard the reservation from. This mechanism allows a node to have 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 take a *shortcut* across the routing tree, reaching the destination in fewer hops.

Figure 2 shows a representation of the full routing information maintained by the Pulse protocol. Since there is an active flow from the node labeled source to the node labeled destination, each node sends a reservation packet up the tree to the pulse source in response to the pulse flood. Nodes that have forwarded a reservation stay on and are colored black. The rest of the nodes in the network may turn off until the next pulse. Nodes that overheard a reservation (or are adjacent to the pulse source) may perform fast activation and are colored grey. Node A can perform fast activation since it was able to overhear B forward the source's reservation packet. Node D is an example of a node that did not overhear any reservation packets and would need to wait for the next pulse period to activate.

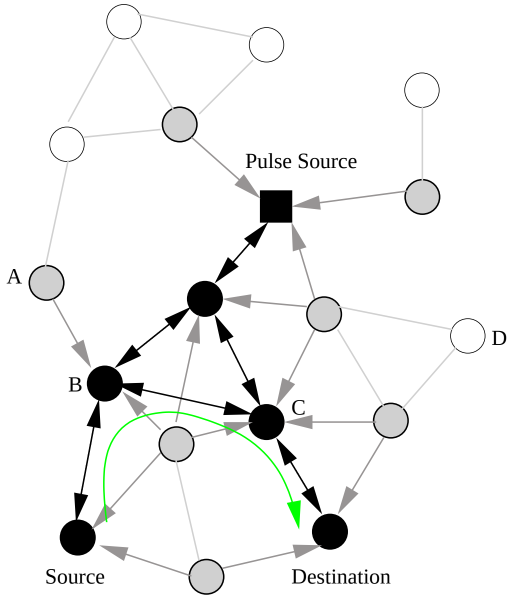

Fig. 2. Pulse Protocol Full Routing State

The reservation packets are used to setup up reverse routes on the active tree and shortcut routes. This figure shows all the underlying links in the network as lines connecting nodes. Arrows are used to indicate that a node has one or more routes through another node. For example, the bi-directional arrow between nodes B and C represents the shortcuts formed when B overheard C forward the destination's reservation packet and C overheard B forward the sources reservation packet. As a result B's routing table contains an entry for both C and the destination (as well as for the source and the pulse source). When the source sends a packet to the destination, it has no direct route, so it forwards the packet to B which is its default route to the pulse source. Since B has a shortcut to the destination, it sends the packet through C rather than up the tree.

One unique quality of the Pulse protocol is its inherent scalability according to many metrics. 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 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 feed back, a link failure is typically detected when two consecutive hello packets have been missed. The pulse interval used in our simulations is one second, 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 1 seconds, of which 152 milliseconds were required for the pulse period. This ratio results in the protocol consuming 15% 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 85% 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 15% of the time to participate in the protocol, but can place their radios in a sleep mode for the remaining 85% 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 III.

B. Pulse Source Selection

The Pulse protocol requires a node to serve as the designated pulse source. In infrastructure based ad hoc networks the gateway node serves as the pulse source. In ad hoc peer-to-peer networks, where there is no designated node, the distributed selection of the pulse source is required. The pulse source may be any arbitrary node in the network, no assumptions about its location or mobility are made in this work. There should exist a single pulse source for every connected component of the network. In order to accomplish this a leader election algorithm is required. A number of leader election algorithms exist which would be appropriate for selecting a pulse source from the set of nodes [6], [7]. Nodes should execute the leader election algorithm when connectivity to the pulse source is lost (e.g. due to mobility or failure). Since the pulse flood is periodic and used as the sleep synchronizer, all nodes in the component will become active in its absence and should all detect that the pulse source is no longer present in a similar time frame.

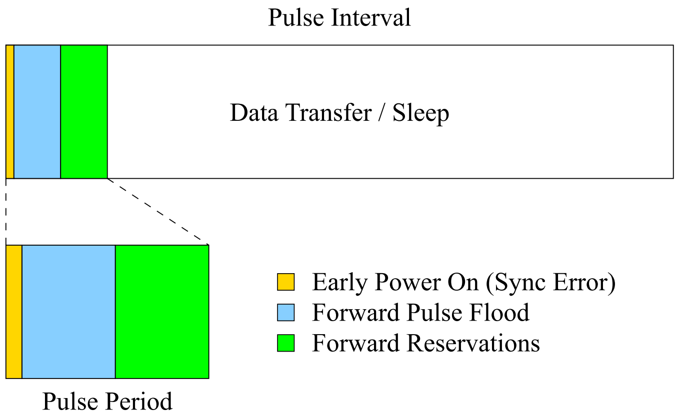

Fig. 3. Pulse Protocol Timing Diagram

When energy efficiency is a primary concern, it is desirable to rotate the task of the pulse source among all the nodes in the network. The node acting as the pulse source will be on if there are any active flows in the network, and will thus be burdened with a higher energy consumption rate then the average node. By periodically rotating the pulse source, this additional energy consumption can be spread evenly between all nodes in the network.

For completeness, a method for merging two separate components that come in range of each other should also be employed. This task is complicated by the power saving aspect of the Pulse protocol. While all the nodes in each component are synchronized, there is no guarantee that the two components will be synchronized with each other. As a result, due to the low duty cycle of the pulse period, the two components may operate in the same space without noticing each other (one component is asleep while the other is sending the pulse flood and vice versa). A strategy based on asynchronous wake-up [8] could be employed in to guarantee that the two components notice each other and merge within a bounded amount of time.

The implementation details for a leader election, rotation, and merging protocol are left for future work. The primary purpose of this work is to explore the performance characteristics of tree based routing in mobile ad hoc networks.

III. SIMULATION

A. Simulation Setup

This work uses the Pulse protocol implementation from [1] which runs in version 2.1b9a of NS2[9]. The simulation setup used in this work is intended to model a peer-to-peer mobile ad hoc network. Parameters including number of nodes, node mobility, and traffic load were all varied to explore the performance of the protocols under a variety of different network conditions. All simulations use a network size of 1 km by 1 km and are run for 300 virtual seconds. 802.11 radios with a bandwidth of 2 Mbps and a nominal range of 250 meters are used. Approximately 4,000 simulations were conducted to examine the delivery ratios, packet delay, and energy consumption.

DSR is configured with the NS2 default options. Timing parameters for the Pulse protocol (see Table I) are based on the experiments conducted in [1], but have been modified for the peer-to-peer network model. The pulse interval has been decreased to one second. The worst case activation delay in the peer to peer scenario is two pulse intervals, one to get to the pulse source, and a second to page the destination. In the fixed infrastructure scenario nodes only needed to get to the pulse source resulting in a worst case activation delay of one pulse interval. In order to maintain the same worst case activation latency of two seconds, a one second pulse interval has been used. In addition, the flood propagation and reservation times have been increased from 50 ms to 70 ms. This was changed in order to accommodate a mobile pulse source which could be anywhere in the network (not necessarily in the center). Also, since flows are peer to peer there are approximately twice as many reservation packets as in the infrastructure access scenario.

The traffic pattern is different than what has been commonly studied in mobile ad hoc network. A random exponentially distributed on/off traffic model is used which allows every node in the network to be a potential traffic source and destination, as opposed to a small fixed set of nodes. This exponential on/off model functions as follows: each flow 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. Flows between all pairs of nodes are created. 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 and average of 10 active flows with one flow changing per second).

TABLE I

PULSE PROTOCOL PARAMETERS

Flood Retransmission Delay	4 msec
Flood Retransmission Jitter	1 msec
Power On Before Pulse	12 msec
Flood Propagation	70 msec
Reservation (estimated)	70 msec
Pulse Interval	1 sec

A modified random way-point mobility model is used in the simulations. The modifications are designed to address the concerns raised in [10] 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.

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

TABLE II

802.11B CARD POWER CONSUMPTION

Transmit	Receive	Idle	Sleep
1.3272 W	.96696 W	.84372 W	.06636 W

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 [11] (Table II), 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 then continuously transmitting. As a result, in order to achieve maximum power savings a protocol must utilize the sleep state as frequently as possible.

C. Delivery Ratio Evaluation

In this experiment our goal is to evaluate the delivery ratio of the Pulse protocol by comparing it with DSR. Figure 4 shows several dimensions of information regarding the performance of the tested routing protocols. The page x-axis shows three network densities. The page y-axis shows four levels of mobility. For each combination of network density 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).

As node density increases the overhead of a flood increases. This affects both the Pulse protocol and DSR. The impact on DSR is greater since it floods more then the Pulse protocol in most cases. The Pulse protocol operates with tight timings for energy efficiency. As a result, when the flood propagation time increases, the amount of time remaining for reservation packets decreases. This results in a reduction in the maximum number of connections which can be simultaneously supported. This is evident in the decrease in the delivery ratio under high load at high node density.

D. Delay Evaluation

In order to evaluate the delay characteristics of the routing protocol simulations were performed with 50, 100, 200 nodes in the 5 m/s max speed scenario. The results are indicated in Figure 5. The graphs display the average per packet end-to-end delay of both the Pulse and DSR protocols. In addition, the Pulse protocol's end-to-end delay is broken down into two components. The route activation delay experienced by a packet is the sum of the time spent waiting for the next pulse period at an inactive sending node, and the time spent waiting at the pulse source while the destination is being paged. The network forwarding delay experienced by a packet is the remainder of the end-to-end delay not included in the activation delay, this includes time spent in interface queues as well as being held during the pulse period. The average over all packets of each of these delays is displayed in the figure.

A number of important observations can be made based on the delay results. When using a connection oriented protocol, such as TCP, only the initial connection establishment packet will experience the activation delay, while all the data packets will only experience the network forwarding delay. In addition, the activation delay is predominantly a result of the power saving

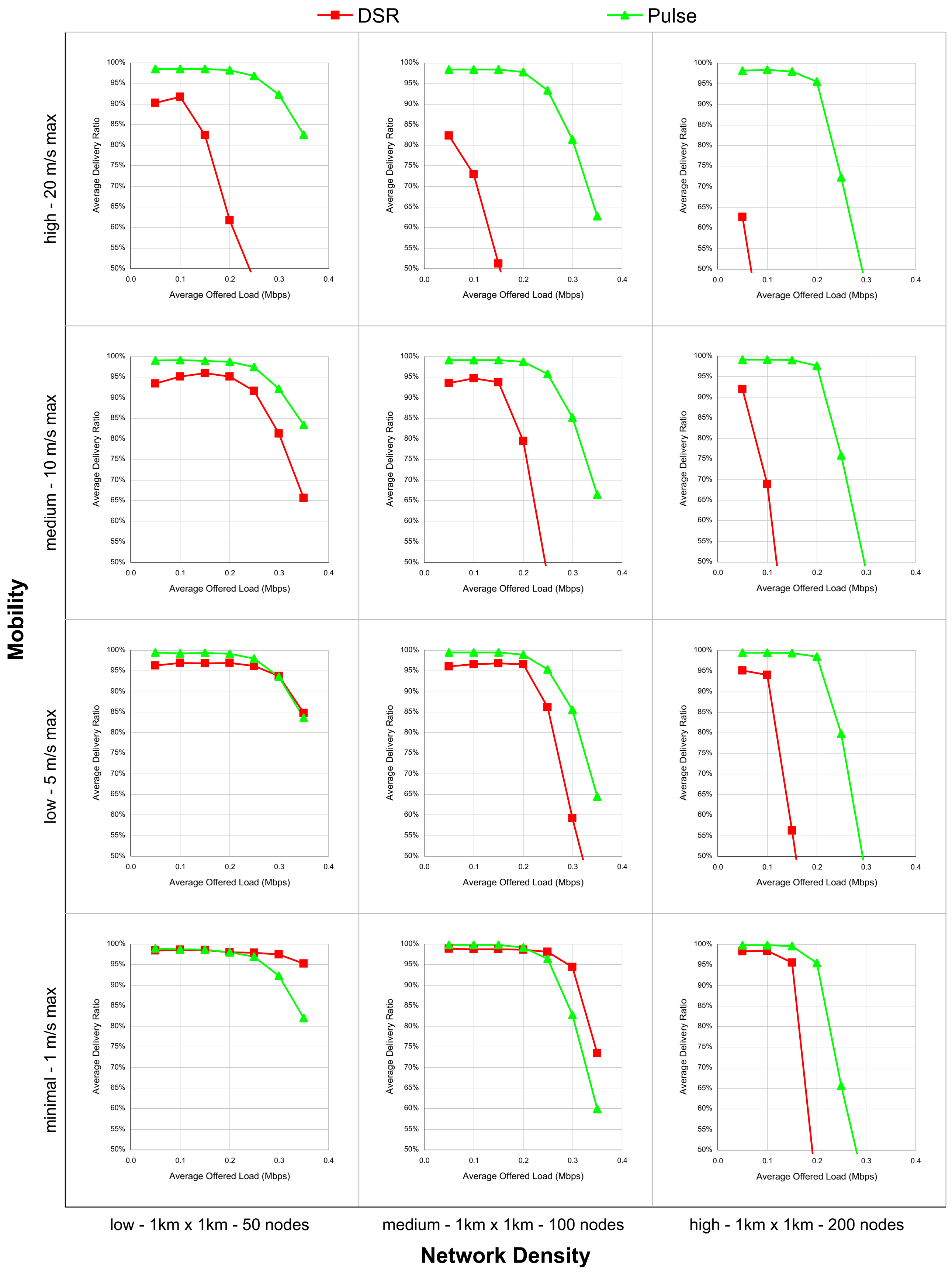

Fig. 4. Delivery ratio results using random way-point mobility and exponential on/off traffic

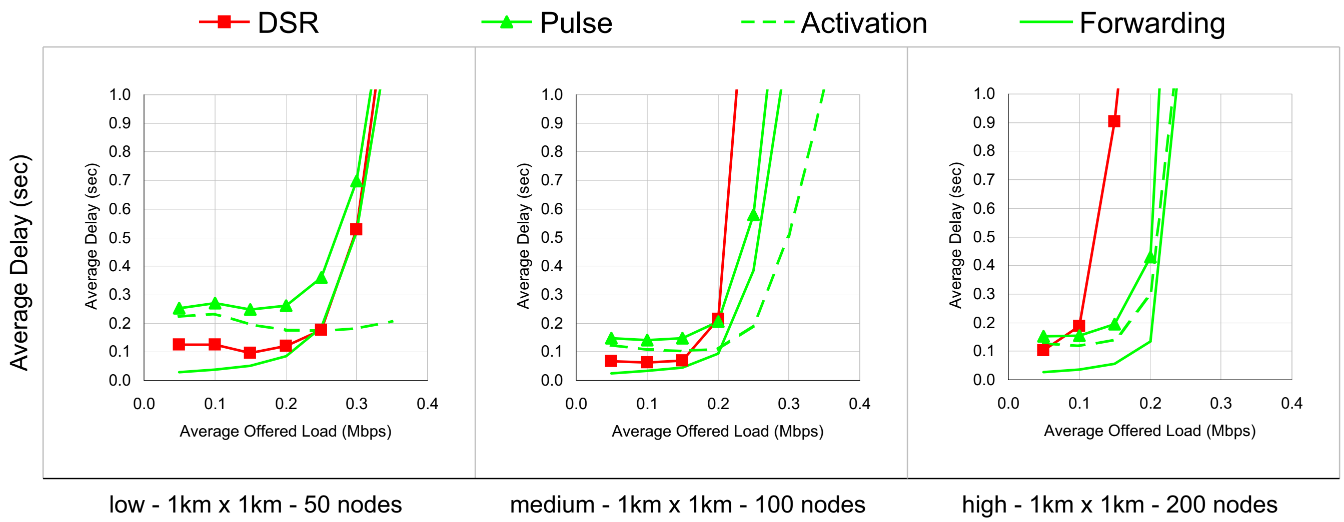

Fig. 5. Delay in the 1km x 1km - 5 m/s max scenario

capabilities of the protocol, so if power saving was not employed, the average per packet end-to-end delay would closely resemble the network forwarding delay. The effects of network density are also evident in the delay results. As the overhead of the flood increases, and the time remaining for reservation packets decreases, activation delay increases as some of the reservation packets fail to reach the pulse source before the end of the pulse period.

Recall that the worst case activation delay when establishing a peer-to-peer connection with the Pulse protocol is two pulse intervals (2 seconds with a one second pulse interval). In practice the activation delay can be lower then the worst case. There are a number of reasons why this is true. The fast activation feature of the protocol enables a large fraction of the network (neighbors of any active path) to avoid the delay incurred by waiting for the next pulse. In addition, paging is unnecessary if the destination node is already active sending, receiving, or forwarding data. If both cases are true, activation delay is completely eliminated.

E. Mobility Scalability Evaluation

In the previous simulations the Pulse protocol showed almost no decrease in performance as mobility increased. In order to further evaluate the effects of mobility a number of additional experiments were conducted. In the 50 node case maximum speeds of up to 100 m/s were simulated with loads of 0.15, 0.20 and 0.25 Mbps. The results are displayed in Figure 6 and show that the delivery ratio and delay scale approximately linearly with the maximum speed.

These results indicate that the Pulse protocol is able to perform well under a wide range of node mobility. The Pulse protocol maintains routes to the pulse source proactively, updating them every pulse interval. The proactive route maintenance helps prevent route failures from occurring and fixes them quickly when they do occur. Typically, routing protocols attempt to detect route failures and take action after they occur. For example, if a route failure is detected in an on-demand protocol it might attempt to perform a local repair, or flood the network with a route request. As the node mobility increases, the overhead of on-demand protocols increases since they need to maintain routes which are breaking often. Since the Pulse protocol is proactively maintaining routes and not re-acting to route failures, it is able to perform well even at high levels of mobility.

F. Density Scalability Evaluation

In order to specifically isolate node density, an additional set of experiments were conducted. Using a 1km by 1km - 5 m/s max scenario with offered loads of 0.10, 0.15, and 0.20, the node density was varied from 50 to 500 nodes per square kilometer. The results are indicated in Figure 7. The main obstacle to scaling in high density networks is operations which require every node to transmit, such as a flood or a hello protocol. It is the increased overhead of flooding which limits DSR's scalability in these scenarios. In order for the Pulse protocol to function correctly, it needs to be able to operate within its tight timings. This is strictly a requirement of the power saving aspect of the protocol. As the number of nodes increases, the overhead of the pulse

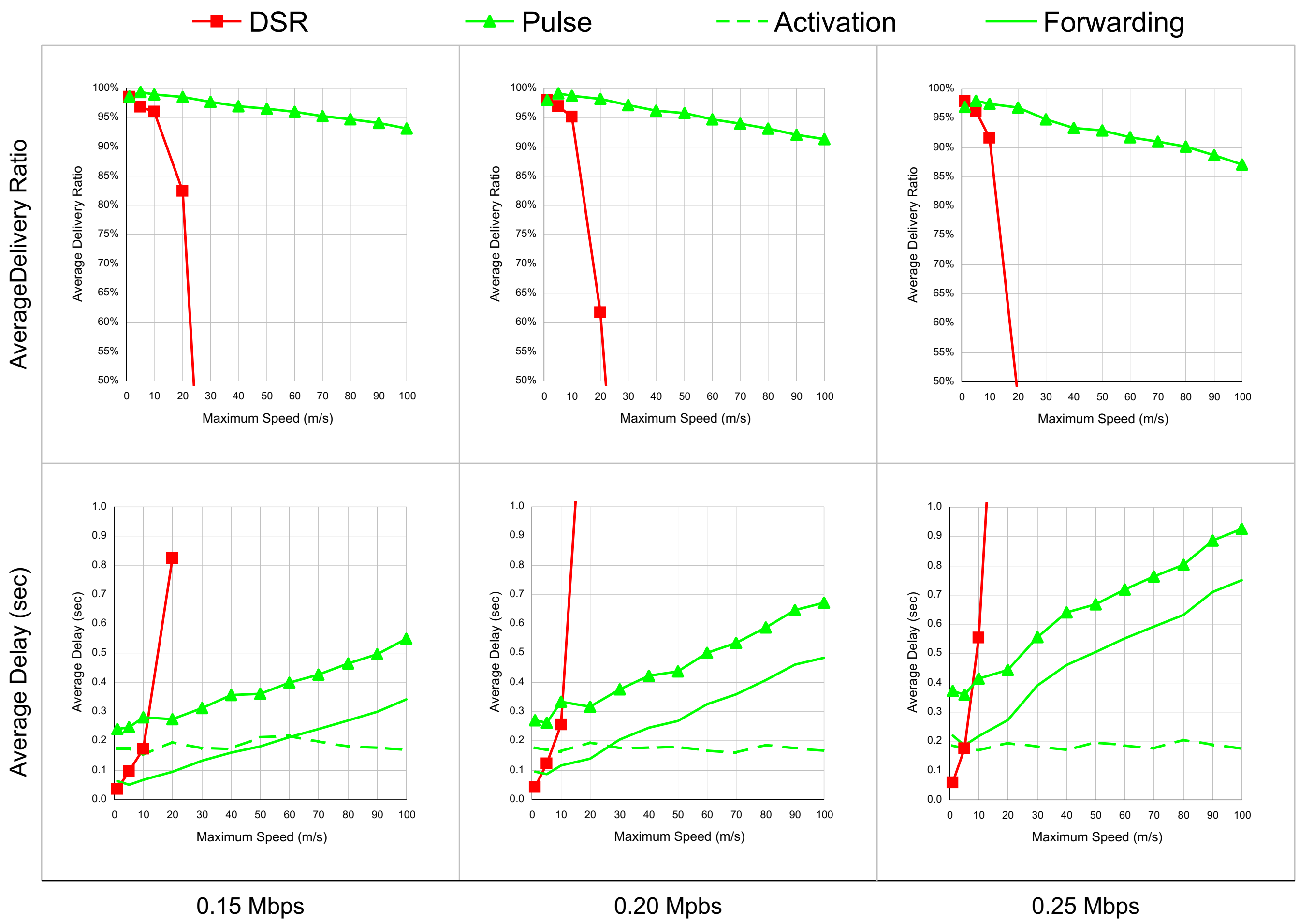

Fig. 6. Mobility scalability in the 1km x 1km - 50 node scenario

flood grows making it more difficult for the protocol to function. In addition, as the offered load increases, the number of reservation packets increases making it even more difficult to meet the timing requirements. If power saving was not required, the pulse flood would consume additional overhead, but the protocol would be able to operate successfully at much higher node densities. If scalability to high node density and energy efficiency are both priorities, the protocol can be tuned by increasing the pulse period at the expense of reduced power savings due to the increased duty cycle.

G. Energy Efficiency Evaluation

Figure 8 shows the average per node power consumption versus the average offered load in the 1km x 1km - 100 node - 5 m/s max scenario. The graph is composed of five lines which help visualize the energy consumption results. The *Sleep State line* and *Idle State line* are provided for reference. The Sleep State line indicates the power consumption of a node which never powers on its radio, and represents the lower bound of any power saving protocol. The Idle State line represents the power consumption of a node which has its radio powered on, but neither sends or receives packets. This is the lower bound of any protocol which does not de-active nodes for power saving.

The *DSR line* indicates the average per node power consumption of nodes running the DSR protocol under varying traffic loads. The error bars on this line show the average minimum and maximum power consumption. The average maximum node power consumption is computed by taking the node which consumes the most power from each random scenario, and averaging the power consumptions of these nodes together. The average minimum is computed similarly using the nodes which consume the least amount of power in each random scenario. DSR's power consumption is completely dominated by idle energy consumption. The additional energy used by transmission and reception of packets is

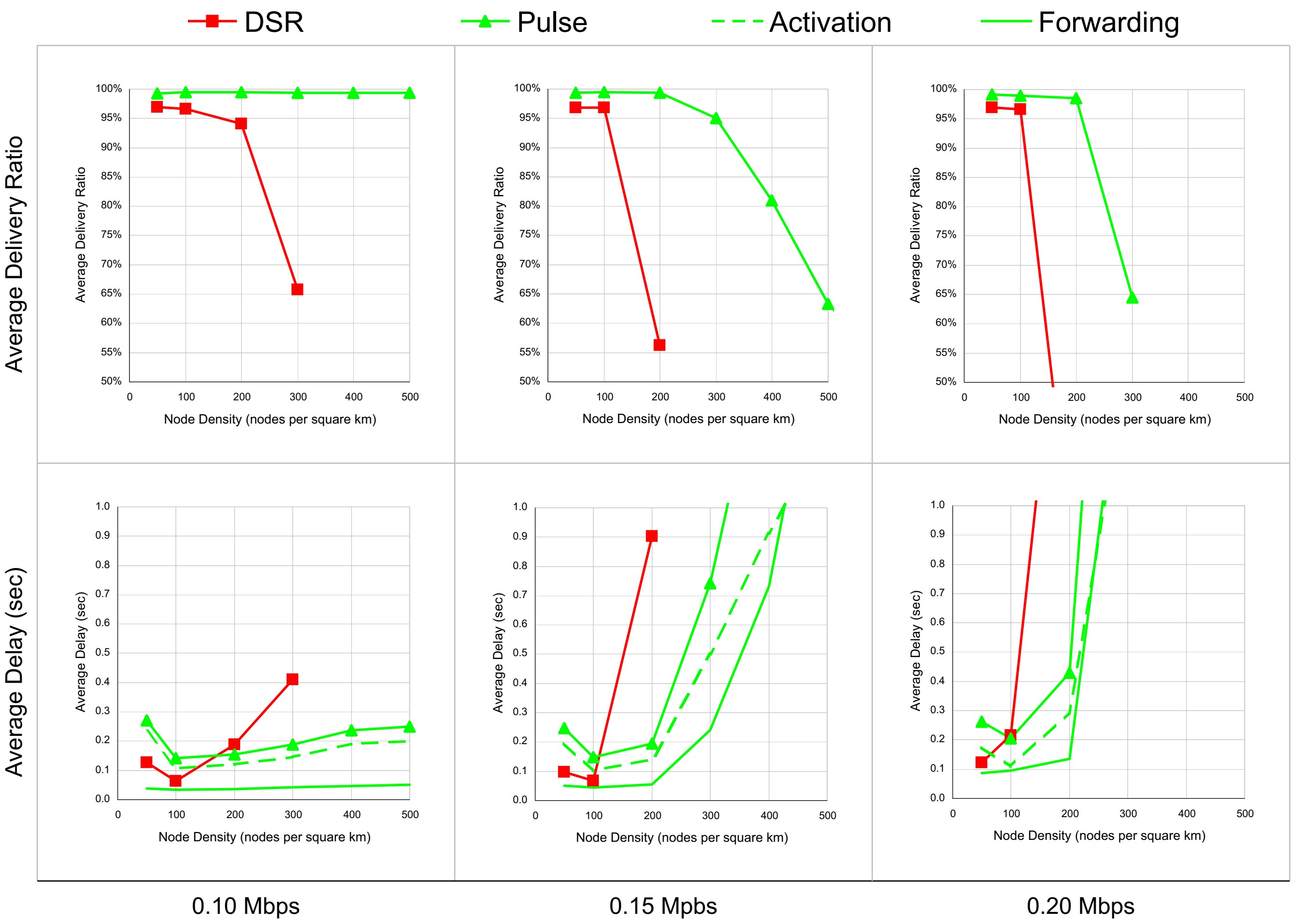

Fig. 7. Density scalability in the 1km x 1km - 5 m/s max scenario

less than 10% of the overall consumption in all simulated cases.

The *Pulse line* shows the overall average per node power consumption including the power consumption of the pulse source. The error bars on this line are computed similarly to the DSR case, except that the pulse source is excluded from the average maximum calculation. Instead the average power consumption of the pulse source is indicated separately on the graph by the *pulse source line*. The average power used by a node running the Pulse protocol is substantially less than that of a node running DSR. We see a savings over the DSR protocol of between 20% and 64% 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 is 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 have the maximum effect.

One interesting aspect of the results is that even though a large fraction of the network traffic may move through the pulse source, its average power consumption is only marginally higher than the average node running DSR (less than a 5% increase). Also, since the Pulse line

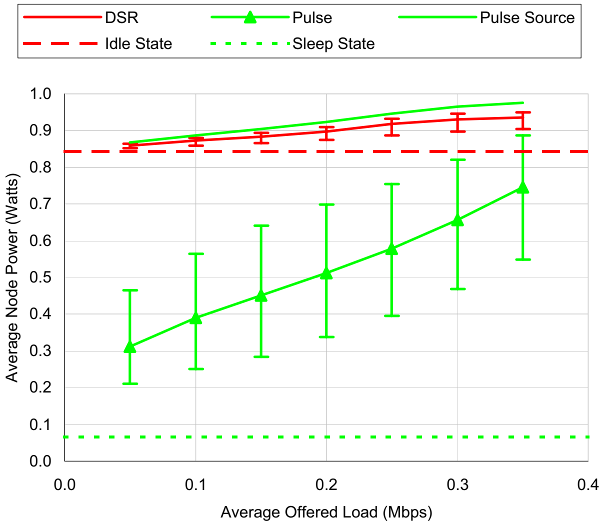

Fig. 8. Energy consumption in the 1km x 1km - 100 node - 5 m/s max scenario

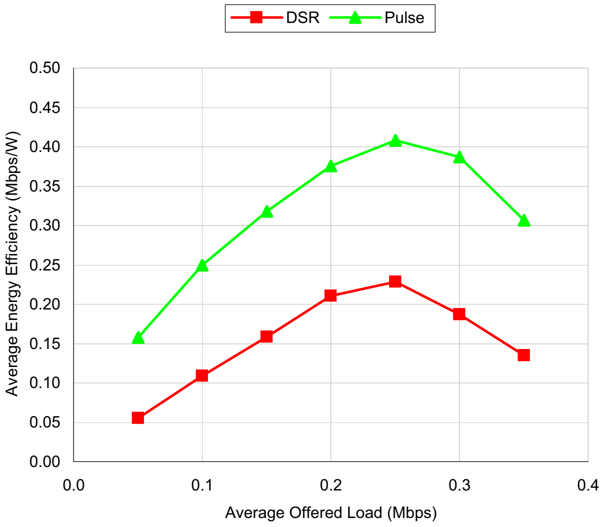

Fig. 9. Energy efficiency in the 1km x 1km - 100 node - 5 m/s max case

average includes the contribution of the pulse source, it represents an accurate estimation of the average node power consumption had the pulse source been rotated. While it would not effect the average, rotating the pulse source allows all nodes to fall within the error bars with no node consuming as much power as a node running DSR.

Figure 9 plots energy efficiency (megabits per second per watt of power consumption) 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 rate 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 consumes energy at an almost constant rate regardless of load, the energy efficiency is directly related to the obtained throughput. Thus, a linear increase in efficiency with offered load is observed until the protocol reaches saturation. The Pulse protocol achieves approximately 2 to 3 times the energy efficiency of the DSR protocol in the simulated scenarios.

IV. RELATED WORK

A number of routing protocols have been proposed by the ad hoc networking community. These protocols generally fall into two categories: proactive and reactive. Hybrid approaches have also been suggested.

Proactive routing protocols actively send routing updates to maintain topology information regardless of whether data is being sent on the network. These protocols typically do not scale well with large numbers of nodes. DSDV [12] for example would require extremely large update packets to track a network with hundreds of nodes. OLSR [4] would flood massive numbers of link state updates as the density and mobility of the network increased. Since these protocols continuously send updates, the nodes in the network need to always remain powered on in order to learn about topological changes. This makes them undesirable for both high density large networks as well as for energy efficiency.

In wireless networks, routing protocol design focuses on providing the best possible performance under the set of constraints imposed by these networks. These constraints generally include but are not limited to bandwidth, power capacity, range, and the medium which is by nature a shared resource. With these resources in mind, protocols were designed which function on-demand. The advantage of this methodology was that when no data needed to be routed, none of the limited resources were consumed. While this is true with respect to bandwidth, it is not true with respect to power consumption.

On-demand protocols, such as AODV [3] and DSR [2], only initiate a route discovery process when a data packet needs to be routed. They are designed for low communication overhead in an idle network. The general structure of existing on-demand protocols requires that nodes remain in an idle but listening state continuously. When a data packet needs to be routed, a route request is flooded through the network. At least a subset of the nodes need to be active in order to guarantee the full flood propagation. According to the power consumption numbers published for commonly available 802.11 cards the power consumed by sending or receiving a packet is only slightly greater then that of an idle node. What this means is that with regard to power consumption, on-demand protocols are saving little to no power over pro-active protocols, and the actual power differences come as a result of the number of route requests versus the number of periodic topology updates, which is normally a result of how active the network is.

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 [13][14][15] 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 [16] or GAF [17], 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 [18]. 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 goodput by cutting into available medium time.

C. *Asynchronous Wake-up*

The idea behind the asynchronous wake-up strategy [8] 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. [19] 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 differers from the existing synchronized protocols in that the time scale is much larger, and that a pro-active routing service is provided simultaneously to the power saving functionality. The larger time scale of the Pulse protocol allows it to operate with much courser time synchronization (on the order of 10 milliseconds) which can be implemented without MAC layer integration.

V. CONCLUSION

We have shown that the Pulse protocol is an effective and energy efficient mobile ad hoc network routing protocol. The Pulse protocol was able achieve high delivery ratios under a wide range of network densities, mobilities, and traffic loads. The results indicate that the protocol is particularly effective when scalability, mobility, and energy efficiency are simultaneously desired.

REFERENCES

[1] Baruch Awerbuch, David Holmer, and Herbert Rubens, “The pulse protocol: Energy efficient infrastructure access,” in *Proceedings of The 23rd Conference of the IEEE Communications Society INFOCOM*, 2004.

[2] 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.

[3] Charles E. Perkins and Elizabeth M. Royer, *Ad hoc Networking*, chapter Ad hoc On-Demand Distance Vector Routing, Addison-Wesley, 2000.

[4] T. Clausen, P. Jacquet, A. Laouiti, P. Muhlethaler, and a. Qayyum et L.Viennot, “Optimized link state routing protocol,” in *IEEE INMIC*, 2001.

[5] R. Ogier, F. Templin, and M. Lewis, “Topology dissemination based on reverse-path forwarding (tbrpf) draft,” .

[6] N. Malpani and J. L. Welch.and N. H. Vaidya, “Leader election algorithms for mobile ad hoc networks,” in *Fourth International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications*, August 2000.

[7] Elizabeth M. Royer and Charles E. Perkins, “Transmission range effects on aodv multicast communication,” in *ACM Mobile Networks and Applications special issue on Multipoint Communication in Wireless Mobile Networks*.

[8] 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.

[9] “The network simulator - ns2,” http://www.isi.edu/nsnam/ns/.

[10] Jungkeun Yoon, Mingyan Liu, and Brian Noble, “Random waypoint considered harmful,” in *IEEE INFOCOM*, 2003.

[11] 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.

[12] Charles E. Perkins and Pravin Bhagwat, “Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers,” in *ACM SIGCOMM’94 Conference on Communications Architectures, Protocols and Applications*, 1994.

[13] V. Rodoplu and T. Meng, “Minimum energy mobile wireless networks,” in *EEE International Conference on Communications, ICC’98*, June, vol. 3, pp. 1633–1639.

[14] 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.

[15] Jae-Hwan Chang and Leandros Tassiulas, “Energy conserving routing in wireless ad-hoc networks,” in *INFOCOM*, 2000, pp. 22–31.

[16] 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.

[17] 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.

[18] 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.

[19] Rong Zeng and Robin Kravets, “On-demand power management for ad hoc networks,” in *IEEE INFOCOM*, 2003.