

The Pulse Protocol: Sensor Network Routing and Power Saving

Baruch Awerbuch, David Holmer, and Herbert Rubens
Department of Computer Science
Johns Hopkins University
Baltimore, MD. 21218
{baruch, dholmer, herb}@cs.jhu.edu

Kirk Chang¹
Applied Research
Telcordia Technologies
Red Bank, NJ. 07701
changki@telcordia.com

I.-J. Wang
Applied Physics Lab
Johns Hopkins University
Laurel, MD. 20723
i-jeng.wang@jhuapl.edu

Abstract—We present a performance evaluation of the Pulse protocol operating in a sensor network. In this work, a number of modifications are made to the original Pulse protocol to provide efficient operation in a sensor 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 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. A new mechanism called *intermediate wake-up periods* is introduced in this work in order to reduce the energy costs of low delay applications. Through simulation we explore the performance of both the protocol and the modifications with respect to energy efficiency and delay.

I. INTRODUCTION

In a sensor network, due to the stringent requirements on both size and power consumption, radios usually have a limited transmission range. Therefore, in addition to collecting and processing information, each sensor also needs to act as a transit node to relay information from its neighbors to the data collector. In addition, some applications, such as target tracking, require accurate timing information as well as the sensor data. As a result, routing, energy consumption, and time synchronization have become three main topics of interest for the sensor networking research community. Unfortunately, these three research topics have traditionally been handled independently, and separate protocols have been proposed for each. However, in sensor networks where network, computational, storage, and energy resources are all at a

premium, running three separate protocols is undesirable. In this paper we propose a powerful framework, based on the Pulse protocol, which simultaneously address all three of these important issues. The Pulse protocol was originally designed for mobile multi-hop infrastructure access networks [1] and mobile ad hoc networks [2], but has many properties that make it well suited for sensor networks.

The Pulse protocol utilizes a periodic flood (a *pulse flood*) initiated at the pulse source, to provide both routing and synchronization to the network. During this pulse flood, each node is updated with the best route to the pulse source. Thus the periodic pulse flood forms a periodically updated spanning tree rooted at the pulse source. In addition to providing periodically updated routing, timing information contained in the pulse flood packets serves to provide and maintain network wide synchronization. Moreover, as nodes are able to synchronize with the periodic pulse, nodes that are not required for packet forwarding can power off their radios for the time between pulses, resulting in substantial energy savings.

Such a framework can be naturally applied to a sensor network paradigm, where the information gathering collectors/sinks can act as pulse sources. Thus the Pulse protocol is highly suited for the many-to-one data gathering communication pattern seen in many sensor networks. In addition, the synchronized tree based nature of the Pulse protocol provides a favorable environment for existing sensor network data aggregation and local coordination schemes.

In this work, a number of modifications are made to the original Pulse protocol to provide efficient operation in a sensor network environment. Since the topology of a sensor network is relatively stable, the full pulse flood is run infrequently, drastically increasing energy efficiency

¹This work was performed while the author was with JHU APL.

and decreasing overhead. In addition, a new mechanism of intermediate wake-up periods is introduced in order to reduce the energy costs of lower delay applications. Different sensing applications have different delay requirements (e.g. a fire detection system and an environmental monitoring system may have drastically different delay requirements). Through simulation, we investigate the trade-off between event delivery delay and energy efficiency in various settings. We show that the Pulse protocol can be tuned to provide excellent energy efficiency while maintaining high packet delivery ratios and meeting the latency requirements of a wide range of applications.

Our Contribution: We present a performance evaluation of the Pulse protocol operating in a sensor network. In this work, a number of modifications are made to the original Pulse protocol to provide more energy efficient operation in a sensor network environment. This results in substantial energy savings over the original protocol. Through simulation we show how the Pulse protocol can be specifically tuned for the target network environment and application in order to achieve maximum energy savings.

II. 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 the sensor collector (*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 network's 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 [3]. 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 a packet for a specified time, 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.

B. Sensor Network Applicability

The Pulse protocol architecture is ideal for many sensor network applications. By pro-actively maintaining a spanning tree rooted at the sensor collector, all of the sensors in the network have a continuously updated route. In addition, since nodes only need to track their best route to the sensor collector, the routing table storage requirements are extremely small. While the protocol pro-actively maintains a route to the sensor collector, reverse routes are established on-demand by utilizing reservations and paging. This enables bi-directional communication between the sensors and the sensor collector, while minimizing local storage requirements.

While the protocol relies on loose time synchronization, it is able to establish and maintain the synchronization without any additional overhead above the standard tree maintenance. Furthermore, the protocol is capable of being enhanced when external time synchronization protocols are present. By establishing tighter time synchronization, additional power can be saved by reducing the early turn on time.

The Pulse protocol also provides a fast activation feature where nodes keep track of whether they overhear

a reservation during the reservation period. If they overhear a reservation, they are able to turn on immediately, to send an update to the sensor collector; since they know that adjacent nodes are already powered on. This greatly reduces latency when multiple events occur within close proximity with respect to both time and space. This occurs frequently in sensor networks since events are localized and are sensed by multiple sensors within an area. For example, in a target tracking application using vibration sensors, a person walking through a field of sensors would be activating different sensors every few seconds, but the activated sensors would be in close proximity. In addition, many sensor applications require local coordination for data aggregation. By having synchronized wake-up periods nodes are able to activate and coordinate locally before reporting data to the sensor collector.

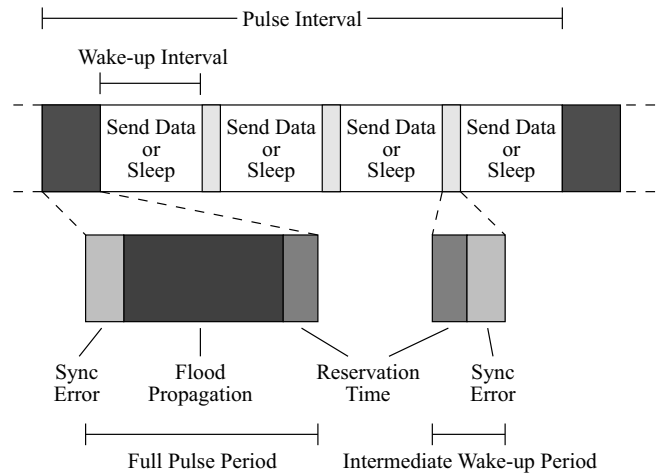


Fig. 1. Pulse Protocol Timing Diagram

III. SIMULATION

A. Simulation Model

In order to evaluate the performance of the Pulse protocol in a sensor network environment a number of simulations were conducted in the version 2.1b9a of the NS2 [4] network simulator. In order to accurately model the MICA2 [5] sensor network radio, modifications were made to the 802.11 medium access control (MAC) protocol timings. The mica radio operates at a speed of 38.4 kbps, has a nominal range of 250m, and uses a CSMA MAC protocol which returns a MAC layer acknowledgement to increase reliability over the wireless channel. The 802.11 RTS CTS mechanism was disabled since it is not used by the sensor radios. In addition, the power consumption tracking code was updated to match the MICA2 radio specifications (see Table I).

C. Protocol Modifications

In order to exploit the specific characteristics of sensor networks, modifications were made to the original Pulse protocol. In a sensor network the nodes typically exhibit extremely low levels of mobility. In a static network configuration, topological changes generally occur as a result of a node exhausting its battery power or from variations in the environment. The routing protocol must be able to repair these infrequent route failures within a grace period dictated by the latency requirements of the sensing application.

In order to minimize energy consumption, the Pulse protocol was modified to enable intermediate wake-up periods. Fig. 1 shows the timing of the full pulse and intermediate periods as well as the pulse and wake-up intervals. It can be observed that full pulse periods are dominated by the pulse propagation time. By utilizing the loose time synchronization provided by the pulse flood, nodes are able to schedule intermediate wake-up periods during which all of the nodes power on their radios in order forward reservation packets. During intermediate reservation periods nodes only need to power on long enough to account for synchronization error and reservation propagation which is considerably shorter than a full pulse period. By decoupling the reservation period from the pulse flood, the protocol is able to save power by allowing path activation without the expense of the costly pulse flood.

The simulations were conducted within square topologies of varying sizes. In all configurations the designated *sensor collector* node was placed in the middle of the network and also served as the Pulse source. The remaining sensor nodes were placed randomly within the simulation area and remained stationary throughout the duration of the simulation. Network traffic events were generated using an exponential random traffic model at average rates of 1 per second, 1 per 10 seconds, and 1 per minute, depending on the scenario. Each traffic event consisted of a multi-packet burst of packets destined for the sensor collector. The number of packets in the burst was exponentially randomly distributed with a mean of 3 full size 29 byte payload packets per burst. The individual simulations were run for one hour of simulation time. The results are the average over multiple

TABLE I
MICA2 RADIO POWER CONSUMPTION

Transmit	Receive / Idle	Sleep
81 mW	30 mW	0.003 mW

random seeds.

The configurations simulated are representative of a *cell* which consists of a sensor collector and some number of surrounding sensors. A real sensor network deployment would most likely consist of multiple cells which form a large network. In these networks the sensor collector might have special capabilities, such as higher power communication channels. The Pulse protocol would serve as the lowest level in the sensor network communication hierarchy.

B. Flood Parameter Tuning

An initial set of experiments were conducted in order to find appropriate values for the protocol flood timings. The purpose of these experiments is to show the relationship between network scenarios and the timing values required for protocol operation. In order to accomplish this, we use a set of 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 from 40 to 60 milliseconds, and jitters from one to ten milliseconds were all simulated. The results of these simulations indicate that a flood delay of 60 milliseconds and a flood jitter of 10 milliseconds work well in the low bandwidth sensor environment. The resulting ninety-ninth percentile values for the total propagation time and synchronization error for a number of different network sizes and densities are shown in Table II. For the remainder of this work, we will use a 1000m x 1000m topology with 200 nodes as our example network.

TABLE II
SYNCHRONIZATION ERROR / PULSE FLOOD PROPAGATION
TIMING PARAMETERS

Topology Size (km)	50 nodes/km ²	100 nodes/km ²	200 nodes/km ²
1 x 1	52ms / 384ms	54ms / 227ms	49ms / 221ms
2 x 2	67ms / 513ms	68ms / 450ms	90ms / 470ms
4 x 4	99ms / 1014ms	123ms / 936ms	167ms / 1008ms

C. Intermediate Ratio Tuning

The second stage of tuning the pulse protocol is dedicated to determining an appropriate *intermediate ratio*. For this discussion, we shall keep the wake-up interval fixed at one every five seconds. The intermediate ratio refers to what percentage of these wake-ups are intermediate wake-up periods as opposed to full pulse periods. The advantage of having a large ratio of intermediate wake-up periods is that these periods are significantly shorter than full pulse periods, because no pulse flood is propagated. As a result, a higher intermediate period ratio allows us to maintain the same latency bound at a lower power consumption level. Intuitively, it would seem that a large intermediate ratio should be used in order to achieve the maximum power savings, however an arbitrarily large ratio also has several disadvantages.

These disadvantages stem from the fact that an increased intermediate ratio will by definition decrease the frequency of the pulse flood. Reducing the frequency of the pulse flood has several disadvantages. One of the primary concerns is that since the pulse flood is responsible for routing, the protocol will take longer to react to route changes. This could be a severe disadvantage in mobile networks, however most sensor networks are stationary. The most common cause of route changes in sensor networks is node death from energy exhaustion.

Another concern is that a decrease in the pulse flood frequency will result in less accurate time synchronization between nodes as a result of clock drift. As a result, larger synchronization error parameters, which consume more energy, may need to be used. In this work we focus on many-to-one sensor originated traffic, however a sensor collector could also initiate contact with sensors in a polling pattern. In the pulse protocol, this type of connection initiation can only occur during a full pulse period, and thus would have higher latency when intermediate periods are used.

A number of simulations were conducted to investigate the energy savings provided by a variety of different intermediate ratios. Fig. 2 shows the average

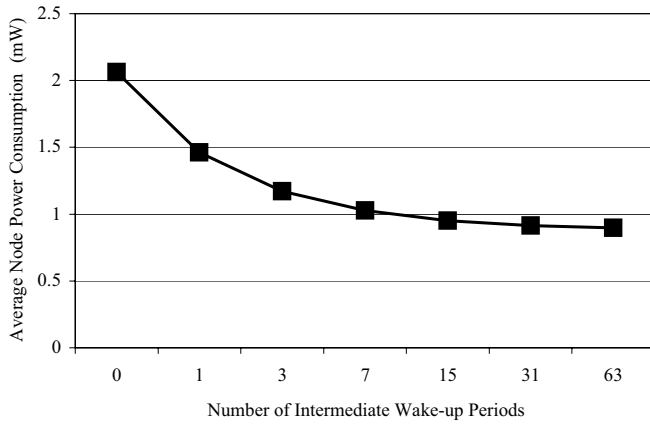


Fig. 2. Intermediate Ratio Tuning Results

node power consumption the 1000 x 1000 meter 200 node case using the previously selected flood tuning parameters. An average event generation rate of one per ten seconds and a reservation time of 100ms are used. In all cases a full pulse or intermediate period occurs every five seconds, and only the ratio of full to intermediate periods is changed. The x-axis indicates the number of intermediate periods that occur for every one full pulse period. Zero indicates that all the periods are full pulse periods, and represents the performance of the protocol without the intermediate period enhancements. As seen in Fig. 2, intermediate wake-up periods can provide substantial power savings, however arbitrarily high intermediate ratios offer decreasing returns. For the remainder of this paper we have selected the ratio of 15 intermediate periods for every full pulse period. In this scenario, this ratio operates at less than half the power of the protocol without intermediate wake-ups, and keeps the interval between full pulses reasonable. Using a higher intermediate ratio would only result in minimal additional power savings and would unduly decrease the pulse flood frequency.

D. Interval Tuning

The final stage of tuning is to determine the wake-up period interval. This sets how often all the nodes in the network must turn on to allow connections to the collector to be initiated. The value of this parameter will have a profound impact on both the event delivery latency as well as the energy consumption. A number of simulations were conducted in our example network with a variety of wake-up intervals and three different traffic loads. Reservation time was set to 50ms in all simulations. Fig. 3 shows the average node power con-

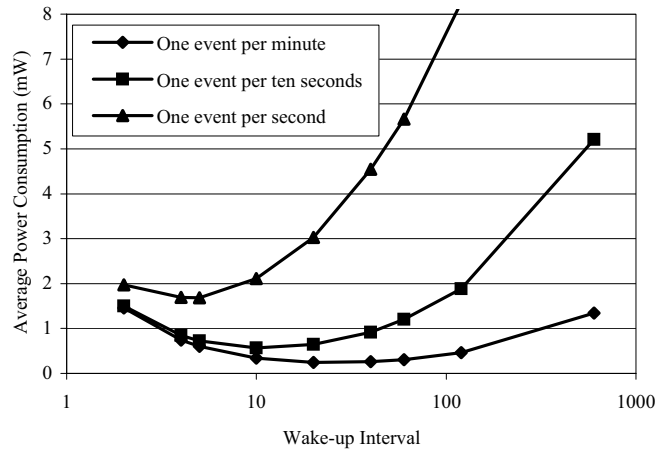


Fig. 3. Interval Tuning Power Results

sumption.

Consider a sensor node in the network that has not generated an event for a long amount of time. When the sensor generates an event, several packets must be sent to the collector. However, since the node has been idle for a long time, it must wait for the next wake-up period before it can activate a route to the collector. Thus, on average the node must wait for half the wake-up interval until the next period, and in the worst case the node may need to wait the full wake-up interval if it just missed the previous period. The wake-up interval is the primary factor that determines event delivery latency.

Due to the synchronized turning on and off of nodes, there will be one optimal interval for a given traffic pattern that yields the lowest energy consumption. In general, a shorter interval will consume more power by causing all nodes to turn on more often. A longer interval can consume more power by increasing the time a node spends in the active state after it has delivered its event but is waiting for the next period to turn off. An interval that is close to the optimum for the expected traffic pattern should be selected as long as that interval meets the application event delivery latency requirements. If an application requires tighter latencies, then a shorter interval can be used, but at the expense of increased energy consumption.

The results show that the optimal points are different for each traffic load. For the three simulated average traffic loads of one event per minute, one event per ten seconds, and one event per second, the best pulse intervals were 320, 160, and 80 seconds respectively. The average node power consumption for each of these cases represent 99%, 98%, and 93% savings respectively over

a node that keeps its radio on but never sends packets. In addition to low power consumption, the pulse protocol yields high reliability with greater than 99.5% delivery ratios in all three cases.

IV. RELATED WORK

The SMAC protocol [6] uses a single radio approach to save energy in sensor networks. The protocol uses a scheduling technique which allows nodes to turn off their radios while other transmissions are taking place. It also uses a synchronization technique which allows nodes to power off their radios and perform synchronized wake-ups.

The STEM [6] protocol uses a two radio approach, where the primary radio remains in a powered off state in order to save power, and a second wake-up radio is employed to initiate communication. When a node needs to communicate, it sends a continuous signal on the wake-up radio. All of its neighbors sense the signal and activate. Broadcasts can be sent when all nodes are active, or point-to-point communication can be initiated at which point the idle nodes will power their primary radios off. The PAMAS [7] approach also utilizes two radios in order to allow nodes to save power. The PAMAS protocol turns nodes off that are not actively communicating. By utilizing the secondary radio, inactive nodes can quickly power on their primary radio with minimal delay. A number of other two radio approaches have also been suggested by the community [8] [9].

V. FUTURE WORK

While intermediate wake-up periods allowed the Pulse protocol to operate more efficiently in a sensor network environment, additional protocol modifications could further increase both performance and efficiency. One additional modification which we would like to explore is a path deactivation capability. Currently, when nodes activate paths, they remain active until the next intermediate period. By keeping paths active the protocol is able to provide a fast activation capability which reduces activation latency for nodes which are in range of an active path. While path deactivation would eliminate the fast activation feature, it would allow energy efficient operation with arbitrarily long intervals. This capability could be implemented using a reservation reference count. When a node forwards a reservation it keeps track of how many reservations it has forwarded. When a node has no more packets to send, it can send a deactivation packet to its parent and decrement its reference count.

When a node's reference count reaches zero it can power off its radio and sleep until the next wake-up interval. Generally, sensor networks are designed for specific applications, which have specific traffic patterns and latency requirements. The Pulse protocol framework allows the routing protocol to be customized to meet these needs.

VI. CONCLUSION

In this work, a number of modifications were made to the original Pulse protocol in order to provide energy efficient operation in a sensor network environment. A new mechanism consisting of intermediate wake up periods was introduced in order to reduce the energy costs of lower delay applications. Through simulation, we illustrated the trade-off between event delivery delay and energy efficiency in various configurations. We showed that the Pulse protocol can be tuned to provide excellent energy efficiency while maintaining high packet delivery ratios and meeting the latency requirements of a wide range of applications.

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] Baruch Awerbuch, David Holmer, and Herbert Rubens, "The pulse protocol: Mobile ad hoc network performance evaluation," in *Technical Report*, 2004.
- [3] Charles E. Perkins and Elizabeth M. Royer, *Ad hoc Networking*, chapter Ad hoc On-Demand Distance Vector Routing, Addison-Wesley, 2000.
- [4] "The network simulator - ns2," <http://www.isi.edu/nsnam/ns/>.
- [5] "Mica 2 mote data sheet," <http://www.xbow.com>.
- [6] W. Ye, J. Heidemann, and D. Estrin, "An energy-efficient mac protocol for wireless sensor networks," in *21st International Annual Joint Conference of the IEEE Computer and Communications Societies*. INFOCOM 2002, June 2002.
- [7] S. Singh and C. Raghavendra, "Pamas: Power aware multi-access protocol with signalling for ad hoc networks," in *ACM Computer Communications Review*, 1999.
- [8] C. Gou, L. C. Zhong, and J. M. Rabaey, "Low power distributed mac for ad hoc sensor radio networks," in *Globecom*, November 2001.
- [9] E. Shih, P. Bahl, and M. J. Sinclair, "Wake on wireless: An event driven energy saving strategy for battery operated devices," in *MOBICOM*, September 2002.