A Context-Aware Data Forwarding Algorithm for Sensor Networks

Anandha Gopalan
Department of Computer Science
University of Pittsburgh
Pittsburgh, PA 15260, U.S.A
Email: axgopala@cs.pitt.edu

Taieb Znati
Department of Computer Science
University of Pittsburgh
Pittsburgh, PA 15260, U.S.A
Email: znati@cs.pitt.edu

Abstract

Recent advances in wireless communication and technologies have given rise to low-cost sensor networks. Sensor networks comprise of low-cost, low-power nodes that are densely deployed in the environment to monitor a specific state of the environment, for example: temperature, light, sound, speed or radiation. This paper presents a new data forwarding algorithm for sensor networks that takes into consideration the direction of the message, the positional relevance of a node to the message and the available power at that node. We conclude this paper by discussing an experimental study of the performance of the proposed data forwarding protocol for sensor networks.

1. Introduction

Sensor networking is an emerging technology that promises unprecedented ability to monitor and manipulate the physical world. A sensor is a small and inexpensive wireless device, with ability to sense and actuate the physical environment in a variety of modalities. Sensor nodes, deployable virtually anywhere, have the ability to self-organize into a well-connected network and cooperatively collect, aggregate and disseminate information to end users [6].

The sensor nodes are low-cost, low-power multi-functional nodes that can be easily deployed in the environment. A sensor network is a network consisting of a large number of sensor nodes that are densely deployed in the environment to monitor a specific state of the environment, for example: temperature, light, sound, speed or radiation. The position of a sensor node during deployment need not be pre-determined, hence the sensors can be deployed randomly in in-hospitable terrains, inaccessible environments or during disaster relief management. Each sensor node has a wireless link to communicate, an on board processor for some basic computation and a sensor that senses the environment to collect the data. Routing in a sensor network is quite different when compared to routing in an ad-hoc network. Unlike an ad-hoc network, which can contain several source nodes and several destination nodes; in a sensor network, several sensors collect the required data from the required environment and forward this data to a destination, which is a node that is responsible for processing the data that is received. Nodes may also carry out in-network aggregation of the data in order to reduce the amount of traffic that is generated in the network [2, 8, 11, 19].

Due to the computation and communication limitation in a sensor network, the routing algorithm must be computationally efficient. Energy efficiency is another important aspect of sensor networks; a sensor network has a large number of nodes and it is often very difficult to replace a node that has failed. Hence, it becomes imperative to reduce failures in the network due to an excess in the energy consumption. The protocols that are designed for sensor networks must take this resource constrained environment into consideration. A routing protocol for sensor networks must thus be both lightweight and energy efficient.

The focus of this paper is on the development of an energy-efficient data forwarding protocol for information dissemination in sensor networks. The communication cost in a sensor network is far greater when compared to the computation cost and therefore, communication in a sensor network must be kept to a minimum. This is necessitates the need for the routing algorithm to be energy efficient due to a paucity of resources, especially power at each sensor node. There are MAC protocols that save energy in the network by periodically setting some sensor nodes to sleep, thus conserving the energy of that node and increasing the lifetime of the network [16]. This on-off behavior of the sensor nodes leads to the problem of synchronization and the lifetime of the network needs to be extended to avoid dis-connectivity in the network. In order to extend the lifetime of the network, the routing protocol that is used in the network requires an efficient power management and resource allocation scheme.

The major contribution of this paper is a context-aware data forwarding protocol for sensor networks that uses information about the forwarding node and information gathered from the message to be forwarded, namely:

- Power capability of the node
- Directionality of the message
- Positional relevance of the node along the routing path

This information (collectively called the $context$), enables the protocol to be efficient, by tying the energy level at each node to its *probability of forwarding*, the probability with which each node can forward the message in the direction of the destination. A higher energy level at a node (higher its probability), indicates that a node is more likely to forward this message.

Another big advantage of this protocol is that it does not require the sensor nodes in the network to be synchronized with respect to their on-off schedules. The *best* node among those nodes that are awake is chosen to forward the traffic in the direction of the destination. This also leads to energy savings in the network.

The simulation results show that the protocol performs well over different network scenarios and tests. The throughput of the protocol increases as the network density increases, while the packet loss decreases. The lifetime of the network increases with network density (a desirable property for sensor networks) and the number of messages transmitted across a network over its lifetime also increased with network density.

The rest of the paper is organized as follows: section 2 briefly talks about the work related to this paper, section 3 outlines the protocol for context-aware data forwarding in sensor networks, section 4 details the simulation environment and the results while section 5 concludes the paper and identifies areas for future work.

2. Related Work

This section talks in brief about some of the past work in information dissemination in sensor networks that are related to this work.

The classical method used to disseminate information in a sensor network was using flooding and gossiping protocols [10]. These protocols assumed no knowledge about the network topology and did not rely on any routing information. The big disadvantage of flooding is that it can cause an $implosion$ due to duplicate messages traversing the network. Gossiping avoids the problem of implosion by arbitrarily selecting any node to send the packet to, rather than broadcasting the packet. These protocols are restricted by the fact that they do not use the knowledge of the topology of the network.

Among the early pioneers of data-centric routing was SPIN [15]. SPIN used the knowledge of the meta-data to route data in the network. Meta-data are exchanged between neighboring nodes in the network before transmission. Topological changes are localized since nodes need to know only their single-hop neighbors. The advantage of this scheme over flooding is that it saves energy. The disadvantage of this scheme is that the meta-data that is advertised may not reach the destination due a lack of interest for this meta-data among the intermediary nodes.

Directed Diffusion [4] is a protocol for data dissemination that relies on data propagation by using data that is named by attribute-value pairs. A node requesting data, sends out its $interest$ (an attribute-value pair that describes the information that this node is interested in) for the named data. This $interest$ is cached among the intermediary nodes and finally reaches the sensor nodes that are responsible for this information. These sensor nodes collect the required information and send it to the node requesting the data. In-network aggregation can be used to reduce the network traffic and save energy. Directed Diffusion saves energy when compared to SPIN [15] due to its on-demand nature and caching. The disadvantage with Directed Diffusion is that it cannot be used as a routing protocol for applications that require a continuous data transfer.

LEACH [17] is a hierarchical routing algorithm for sensor networks. Nodes are bunched together into local clusters based on the signal strength. The cluster-head of each cluster takes part in routing the data towards the sink. LEACH cannot be applied to sensor networks that are deployed on a large-scale since it assumes a single-hop communication between a node and its cluster-head.

GAF [20] is an energy-aware location-based routing protocol that is predominantly used in ad-hoc networks. Energy conservation is achieved by selectively turning off nodes in the network. The disadvantage of this scheme is the assumption that each node is GPS equipped. It is expensive in terms of communication and computation power to equip a sensor node with a GPS receiver.

[7, 12] mention a probabilistic forwarding protocol that probabilistically forwards the data towards the sink. They use the knowledge of a node's position with respect to the direction of the destination. The disadvantage of this scheme is that, using the angle of the node with respect to the destination is not enough to determine the node to forward the data.

In this work, we propose a context-aware data forwarding protocol that addresses the shortcomings of the other protocols. This protocol does not require a GPS receiver, rather it uses the relative position of a node with respect to the sender and the destination. The probability of forwarding is based not only on the angle of the node with respect to the destination, but also its distance and the current power at the intermediary node.

3. Context-aware Data Forwarding Protocol

3.1. Sensor Model

A sensor network comprises of a large number of heterogeneous sensor nodes. The sensor nodes are characterized by their power and energy costs for exchanging data with other nodes in the network. The sensor nodes are spread densely throughout an area as shown in Figure 1. A sensor network has a $sink$ (usually the control center), the destination where the data collected in the network must propagate to. The $sink$, upon receipt of the data performs the necessary computation on the data. We assume that upon deployment of the sensor network, the initialization protocol provides each sensor node with the direction of the $sink$.

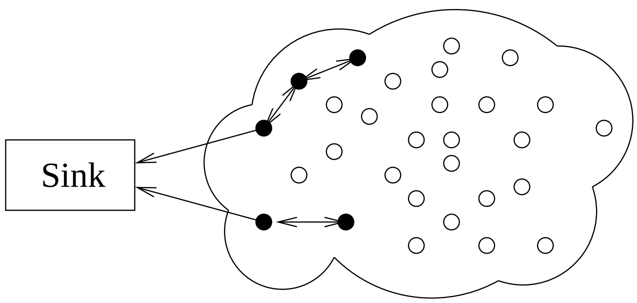

Figure 1. A Sensor Network

Each sensor node is equipped with a sensor that monitors the state of the environment with respect to different criteria like light, temperature, radiation, speed or movement. The primary mode of communication in sensor networks is through $broadcasting$. Each node can be in one of $four$ states in the network with regards to its energy consumption:

1. Sleeping
2. Sensing the environment
3. Transmitting a message
4. Receiving a message

For modeling the energy spent during the $four$ different states of a sensor node, we use the energy model from [17, 12, 13]. Let E_{elec} be the energy that the radio dissipates to run the transmitter and receiver. Let ϵ_{amp} be the transmit amplifier needed to achieve an acceptable signal-to-noise ratio. We assume the energy consumed to transmit over a distance x is proportional to x^2. Using the energy model, the energy expended to transmit and receive a `k-bit` message over a distance x is:

Transmit:

$$E_{Tr}(k,x) = E_{Tr-elec}(k) + E_{Tr-amp}(k,x) \quad (1)$$
$$E_{Tr}(k,x) = E_{elec} * k + \epsilon_{amp} * k * x^2 \quad (2)$$

Receive:

$$E_{Re}(k) = E_{Re-elec}(k) \quad (3)$$
$$E_{Re}(k,x) = E_{elec} * k \quad (4)$$

where,

- $E_{Tr-elec}$: Energy consumed to run the transmitter
- $E_{Re-elec}$: Energy consumed to run the receiver
- E_{Tr}: Energy dissipated for transmission
- E_{Re}: Energy dissipated for reception
- E_{idle}: Energy dissipated in the idle state
- $E_{powerup}$: Energy dissipated for waking up from the sleep state

The model assumes that the power consumed during waking up from sleep is three times the power consumed in the idle state.

$$E_{powerup} = 3 * E_{idle} \quad (5)$$

3.2. Protocol Description

The sensor network is a resource constrained network, where communication is very expensive. Hence, it is imperative that any data forwarding protocol designed for such a network take this into account. To reduce flooding in the sensor network, the traffic that is forwarded towards the $sink$ is sent in a cone-shaped fashion as shown in figure 2.

Consider the scenario when a sensor node S has data to send to the $sink$ D. As mentioned earlier, S has some idea as to the direction of D. Using this information, node S forwards the data in a cone-shaped fashion towards the direction of D. Any intermediate node (node F in figure 2) that is awake and receives the data sent by S calculates its probability of forwarding. Based on this probability, node F can either choose to forward this message or wait for the next time slot. In case node F chooses to wait and within a certain $timeout$, another intermediate node, G forwards the message and node F can hear this broadcast, node F does not subsequently forward the message. This avoids flooding

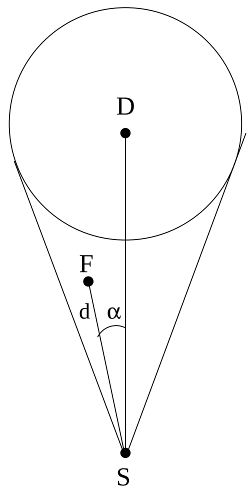

Figure 2. Directional routing

the network by sending duplicate messages. The higher the probability of forwarding, the higher the chances are that node F forwards this message towards D. Nodes outside the cone-based region listen to see if any node in the region has forwarded the message or not. If the message has not been forwarded for a certain amount of time (a $timeout$), one of these nodes forward the message towards the destination. This location-based directed forwarding algorithm is explained as a $pseudo\ code$ in algorithm 1.

Algorithm 1: Forwarding Data towards the Sink
Input: Void
Output: Result
FORWARD-DATA-TOWARDS-SINK()
Let P_n be the forwarding probability of node n
While (!success)
  Generate a random number in $[0, 1]$
  With P_n
    Forward-Data ()
  With $1 - P_n$
    Wait for the next time slot
    if Data-Sent by another node before timeout
      drop request
      return Success
    else
      continue
return Success

The intermediary nodes in the sensor network require some information about the source and the sink to calculate the probability of forwarding. An intermediary node needs to calculate the distance to the source and the angle of deviation from the source. This requires some knowledge of the position of the intermediary node. Using GPS [5] for the purpose of obtaining the position of a node is too expensive in terms of computation and communication energy for a sensor node. A node only needs to know its $relative$ position with respect to the source and this would help in calculating the distance and the angle of deviation. This can be achieved by using the scheme suggested in [18]. To enable data forwarding and to calculate the probability of forwarding, the source needs to include its own position and the position of the $sink$ in the data packet.

The probability of forwarding of a node is calculated by taking into consideration the power of the node, the angle of deviation of this node with respect to the source and the distance of this node from the source. As shown in figure 3, three interesting cases arise while calculating the probability of forwarding. In the figure, S is the source, while D is the destination ($sink$) and F and G are intermediary nodes.

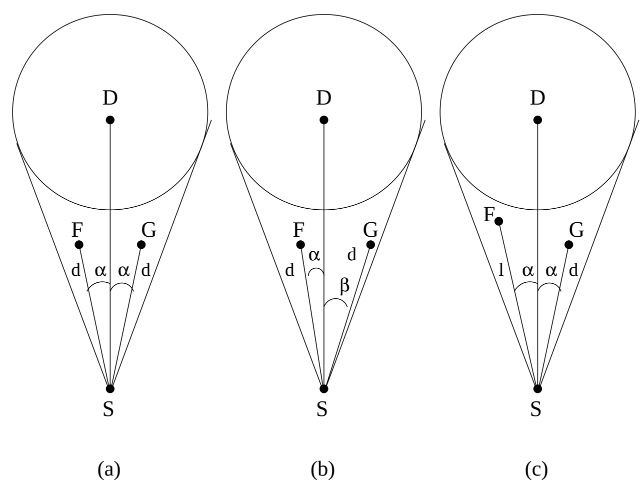

Figure 3. Different scenarios

- Case 1: As shown in Figure 3(a), nodes F and G are both equidistant from S and the angle of deviation (α) of F and G with respect to S is also the same. Assuming that node F has a higher power when compared to node G, node F should have a higher probability of forwarding than node G. Given this scenario, the probability of forwarding must be higher for the node that has higher power.

- Case 2: Figure 3(b) shows that nodes F and G are equidistant from S. Assuming that both F and G have the same power, node F should have a higher probability of forwarding than node G since the angle of deviation of F (α) with respect to S is less than the angle of deviation of G (β).

- Case 3: From Figure 3(c), we can see that nodes F and G have the same angle of deviation with respect to S, but node F is farther away from S when compared to G, since $l > d$. Assuming that both F and G have the same power, the probability of forwarding for node F must be greater than the probability of forwarding of node G.

The probability of forwarding must hence, reflect the requirements listed above. We will now mathematically derive the expression for the probability of forwarding ($P(n)$).

Let α_c = the angle of the cone (the angle between the tangent from S and the straight line between S and D)
Let α_n = the angle of deviation that an intermediary node makes with S
Let d_n = distance of intermediary node from S
Let d = radius of cone
Let $\Gamma(p)$ = function of the power (p) at the node

$$P(n) = w_i * \left(\frac{(\alpha_c - \alpha_n)}{\alpha_c}\right) + w_j * \frac{d_n}{d} + w_k * \Gamma(p) \quad (6)$$

where, $w_i,\ w_j,\ and\, w_k$ are the respective weights. It can be easily seen that probability of forwarding, $P(n) \leq 1$ iff $w_i + w_j + w_k \leq 1$ and $\Gamma(p) < 1$.

The challenge is to design a function that is not computationally expensive, (given the fact that the sensor nodes in the network are resource constrained), but still achieves the required properties as stated above.

The requirement is for a function that increases exponentially towards 1 as a function of the power at the node. The current power at the node can be calculated by using the values of E_{Tr}, E_{Re}, E_{idle} and $E_{powerup}$ from [17, 2].

The $exponential$ function bounded between 0 and 1 will work for this purpose. The higher the power at a node, the higher the value of $\Gamma(p)$, thus ensuring that this node has a higher probability. Hence, $\Gamma(p) = e^{p-1}$, (where p is the remaining power, relative to full power at a sensor node). Re-writing equation 6 using this relationship, we get:

$$P(n) = w_i * \left(\frac{(\alpha_c - \alpha_n)}{\alpha_c}\right) + w_j * \frac{d}{d_n} + w_k * e^{p-1} \quad (7)$$

where, $0 \leq e^{p-1} \leq 1$ and $0 \leq p \leq 1$

4. Simulation and Results

This section discusses the simulation environment and the results of the simulation. The objective of the simulation was to perform a sensitivity analysis of the protocol. Such an analysis would provide us with valuable information about the performance of the proposed protocol under various network and traffic scenarios.

The protocol was implemented in the Glomosim network simulator [9] on Linux and was tested by providing different network scenarios. The tests carried out measured different aspects of the protocol: throughput number of packets dropped, the number of messages traversing through the network and the normalized lifetime of the network. These tests were carried out while varying the density (number of nodes) of the sensor network. The number of nodes in the network was varied from 10 (sparse network) - 50 (dense network) nodes.

The Glomosim network simulator is a simulator that is tailor made for wireless networks. To use this simulator for a sensor network the following modifications had to be incorporated in Glomosim:

- Reduce the radio range of the node
- On-off behavior of the sensor nodes
- Energy dissipated due to communication

The basic components of the simulation are: traffic generation, sensor behavior and power model. Traffic is generated using one sensor node (node 1) as the source and another sensor node (node 2) as the destination. The type of network traffic used in the simulation was CBR traffic where each packet was of size 100 bytes. The sensor nodes were placed randomly in a grid of size 20x20m. The on-off behavior of the sensors was incorporated by having nodes sleep and wakeup for a specific amount of time. The power model that was used in the simulation is provided in section 3.1. The values used for transmit power, receive power, idle and wake power were from [2]. In all experiments, the values that were measured were measured for varying degrees of network density. Each experiment was run a total of 10 times and the results used were averaged over these experiments. Since we were simulating a sensor network, we assumed the sensor network was a network that was stationary after being deployed. Hence, no mobility was incorporated into the simulations.

The first set of experiments measured the performance of the protocol with respect to the throughput and the number of packets that were dropped in the network against varying degrees of network density.

To measure the throughput and the number of packets dropped in the network, the destination node collected the statistics with regards to the total time for packet transmission and the total number of packets that were received at the destination. The time was measured in $nano-seconds$. Traffic was generated using node 1 as the sender and node 2, the receiver. It was ensured that node 1 and node 2 were not in direct communication range of each other, hence making sure that the route between nodes 1 and 2 would follow a multi-hop path. Figure 4 and figure 5 show the results of these experiments.

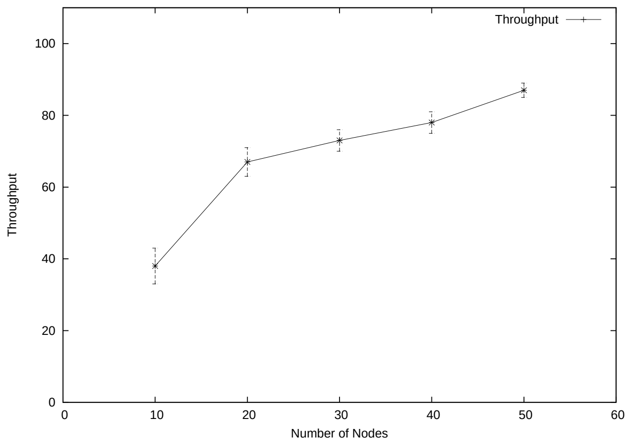

Figure 4. Throughput

From figure 4, we conclude that the throughput of the network increases as the density of the network grows. This is to be expected as the higher the number of nodes in the network, the higher the probability of path availability (since there is a higher probability of a node being in the path between the source and the destination and this node can forward the packet towards the destination).

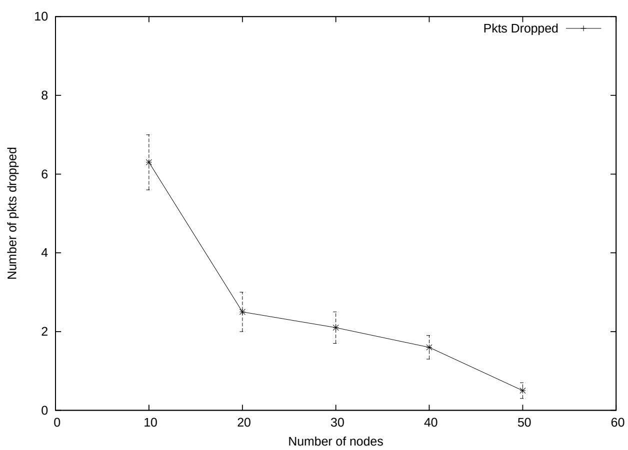

Figure 5. Number of Packets dropped

Figure 5 shows the number of packets dropped with respect to node density. We can observe from the figure that the number of packets dropped decreases (and almost becomes zero) as the network density increases. This is a desirable property in any protocol for sensor networks, since sensor networks are a densely deployed network of sensor nodes. The number of packets that are lost for a sparsely populated network is quite high due to the fact that there are very few nodes in the network and these nodes keep switching between the states of on and off. When the node is in the off state, it does not receive any packets and hence, this leads to packet loss in the network. The denser the network, the higher the probability of finding a node that is $awake$ in the forwarding region, thus ensuring that the packet reaches its destination.

The second set of experiments were conducted to measure the impact of the energy consumption on the lifetime of the network as the density of the network increases.

We measure the lifetime of the network to be the time taken for a node in the network to fail due to lack of power. This node can be any node in the network (except the source). The energy level of a node is calculated after every message is sent or received by the node based on the values provided in [2]. Traffic was generated using node 1 as the sender and node 2 as the destination and the type of traffic was CBR. Traffic was generated continuously and there was no upper limit on the number of messages sent; traffic generation stopped when the simulation ended. As before, nodes 1 and 2 were not in direct communication of each other. The results were collected after the simulation ended (which was when a node in the network died because of lack of power). Figures 5 and 7 show the results of these experiments.

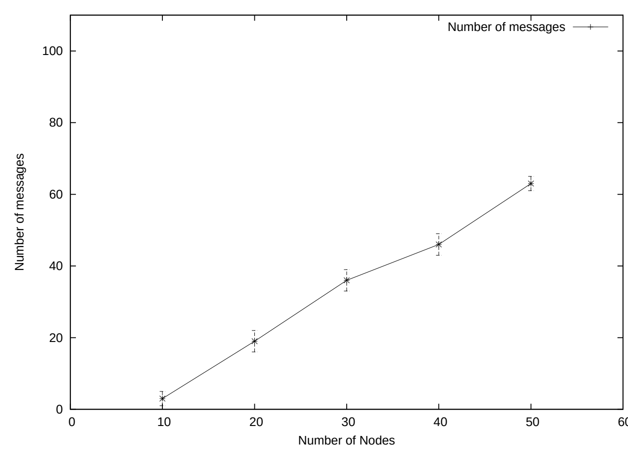

Figure 6. Number of Messages

Figure 6 shows us that the number of packets that are traversing the network increases as the network density increases. This is a desirable property for sensor networks, since this shows that the packet takes multiple paths to the destination, due to the unavailability of other nodes (which may be due to the nodes sleeping). This allows for increased longevity in the network lifetime since the same intermediary nodes are not used repeatedly.

From figure 7, we can see that the normalized lifetime of the network increases as the network density increases. The normalized lifetime of a network was calculated by normal-

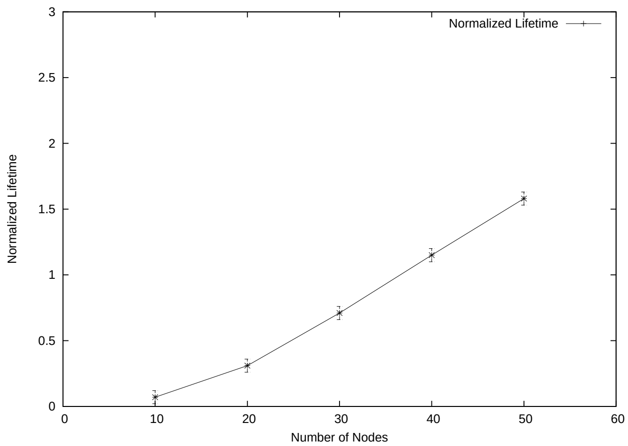

Figure 7. Network Lifetime

izing the total number of packets in the network with respect to the amount of time the network was alive. The amount of time that the network was alive was calculated to be the time since the start of the simulation till the time when a node in the network died. This test gives us an idea as to amount of time the network is alive based on the time and also on the number of packets that are traversing the network. In a network that is sparsely populated, the choice of intermediary nodes are very few and these nodes use up their power quite quickly while transferring data from the source to the destination. As the density of the network grows, the number of intermediary nodes increases, thus reducing the per-node energy usage, which in turn leads to an increase in the network lifetime.

5. Conclusion and Future work

This paper presents a new data forwarding algorithm for sensor networks that is context-aware. The algorithm takes into consideration the direction of the data, the positional relevance of an intermediary node to this data and the energy level at that node. The protocol was implemented and tested using the Glomosim network simulator. Simulation results show that the protocol performs better as the density of the network increases, thus making this a favorable data forwarding protocol for use in sensor networks. There is immense potential for future work in this area. The data forwarding protocol can be improved by adding fault-tolerance to the protocol by allowing for multiple paths to the destination to ensure that there are fewer packet losses in the network. This also increases the network lifetime by decreasing the per-node energy usage. Network caching can also be used to improve the performance of the protocol by allowing nodes to cache the information about the source, so that it does not have to re-calculate its relative position and direction each time it has to forward a packet.

6. Acknowledgments

This work has been supported in part by the National Science Foundation (under grant no: ANI-0073972).

References

[1] A. Manjeshwar and D. P. Agrawal. TEEN: A Protocol for Enhanced Efficiency in Wireless Sensor Networks. In *Proceedings of the 1st International Workshop on Parallel and Distributed Computing, Issues in Wireless Networks and Mobile Computing*, April 2001.

[2] C. Intanagonwiwat, D. Estrin, R. Govindan and J. Heidemann. Impact of Network Density on Data Aggregation in Wireless Sensor Networks. In *Proc. 22nd International Conference on Distributed Computi ng Systems, Vienna, Austria, IEEE. July*, 2002.

[3] C. Schurgers and M. Srivastava. Energy Efficient Routing in Wireless Sensor Networks, 2001.

[4] Chalermek Intanagonwiwat and Ramesh Govindan and Deborah Estrin. Directed diffusion: a scalable and robust communication paradigm for sensor networks. In *Mobile Computing and Networking*, pages 56–67, 2000.

[5] E. E. D. Kaplan. *Understanding GPS: principles and applications*. Artech House, Boston, MA, 1996.

[6] I. Akyildiz and W. Su and Y. Sankarasubramaniam and E. Cayirci. Wireless Sensor Networks: A Survey. *Journal of Computer Networks, Volume 38*, pages 393–422, 2002.

[7] I. Chatzigiannakis, T. Dimitriou, S. Nikoletseas and P. Spirakis. A probabilistic forwarding protocol for efficient data propagation in sensor networks. In *5th European Wireless (EW) Conference*, 2004.

[8] K. Dasgupta, K. Kalpakis, and P. Namjoshi. An Efficient Clustering-based Heuristic for Data Gathering and Aggregation in Sensor Networks. In *Proc. Wireless Communications and Networking (WCNC)*, pages 16–20, March 2003.

[9] L. Bajaj, M. Takai, R. Ahuja, R. Bagrodia and M. Gerla. Glomosim: A scalable network simulation environment. Technical Report 990027, UCLA, 13, 1999.

[10] S. Hedetniemi and A. Liestman. A survey of gossiping and broadcasting in communication networks. *Networks, Volume 18, No. 4*, pages 319–349, 1988.

[11] S. Madden and M. Franklin and J. Hellerstein and W. Hong. TAG: A Tiny Aggregation Service for Ad-Hoc Sensor Networks. In *Proc. 5th Symposium on Operating System Design and Implementation (OSDI)*, 2002.

[12] S. Niloletseas, I. Chatzigiannakis, H. Euthimiou, A. Kinalis, A . Antoniou, G. Mylonas. Energy Efficient Protocols for Sensing Multiple Events in Smart Dust Networks. In *Proc. 37th Annual Simulation Symposium*, Apr. 2004.

[13] T. Antoniou, I. Chatzigiannakis, G. Mylonas, S. Niloletseas, A. Boukerche. A New Energy Efficient and Fault-tolerant Protocol for Data Propagation in Smart Dust Networks Using Varying Transmission Range. In *Proc. 37th Annual Simulation Symposium*, Apr. 2004.

[14] V. Rodoplu and T. Meng. Minimum energy mobile wireless networks, July 1998.

[15] W. Heinzelman and J. Kulik and H. Balakrishnan. Adaptive protocols for information dissemination in wireless sensor networks. In *Proc. MOBICOM, 1999*, pages 174–185, 1999.

[16] W. Ye and J. Heidemann and D. Estrin. An energy-efficient MAC protocol for wireless sensor networks. In *Proc. IEEE INFOCOM*, 2002.

[17] Wendi Rabiner, Heinzelman and Anantha Chandrakasan and Hari Balakrishnan. Energy-Efficient Communication Protocol for Wireless Microsensor Networks. In *Proc. 33rd Hawaii International Conference on Mobile Computing (HICSS)*, 2000.

[18] Xiang Ju and Hongyuan Zha. Sensor Positioning in Wireless Ad-hoc Sensor Networks Using Multidimensional Scaling. In *Proc. IEEE INFOCOM*, 2004.

[19] Y. Yao and J. Gehrke. The Cougar Approach to In-Network Query Processing in Sensor Networks. In *SIGMOD*, 2002.

[20] Ya Xu and John S. Heidemann and Deborah Estrin. Geography-informed energy conservation for Ad Hoc routing. In *Mobile Computing and Networking*, pages 70–84, 2001.