Fault Tolerant Energy Aware Data Dissemination Protocol in Sensor Networks

Gunjan Khanna, Saurabh Bagchi, Yu-Sung Wu
Dependable Computing Systems Lab
School of Electrical & Computer Engineering, Purdue University
465 Northwestern Avenue, West Lafayette, IN 47907.
Email: {gkhanna,sbagchi,yswu}@purdue.edu

Abstract

In this paper we present a data dissemination protocol for efficiently distributing data through a sensor network in the face of node and link failures. Our work is motivated by the SPIN protocol which uses metadata negotiation to minimize data transmissions. We propose a protocol called Shortest Path Minded SPIN (SPMS) in which every node has a zone defined by its maximum transmission radius. A node which is a data source advertises the availability of data to all the nodes in its zone using a metadata descriptor. Any interested node requests the data and gets sent the data using multi-hop communication via the shortest path. The failure of any node in the path is detected and recovered using backup routes. We build simulation models to compare SPMS against SPIN. Different scenarios including mobility and node failures are simulated. The simulation results show that SPMS reduces the delay over 10 times and consumes 30% less energy in the static failure free scenario. Even with the addition of mobility, SPMS outperforms SPIN by energy gains between 5% and 21%. An analytical model is also constructed to compare the two protocols under a simplified topology.

Keywords: Sensor network, Energy efficient data distribution, FT Communication, Modeling, FT Mobile Computing/Networking.

1 Introduction

Sensor networks are a particular class of wireless ad hoc networks in which the nodes have micro-electro-mechanical (MEMS) components, including sensors, actuators and RF communication components. Sensor nodes are randomly dispersed over the area of interest and are capable of RF communication and contain signal processing engines to manage the communication protocols and data processing tasks. Sensor nodes are typically battery-powered and since replacing or recharging batteries is often very difficult, reducing energy consumption is an important design consideration for sensor networks.

Sensor nodes are frequently used for gathering and disseminating data about the physical conditions of the environment they are embedded in. This has spurred extensive research in protocols for distributing sensor data in sensor networks. Energy aware routing protocols optimize the number of transmissions required to set up routing paths, the amount of state maintained at each node, and the cost of transmitting data packets. There has also been interest in complementing the network level routing protocols with higher layer data dissemination protocols that take the data semantics into account. The objective is to minimize the transmission of redundant data in the network. The baseline protocol can be considered to be flooding or broadcast, where each node retransmits the data it receives to all its neighbors, except the neighbor that it received the data from. This is a simple protocol that does not keep any state at intermediate nodes and disseminates data quickly in a network. However, it results in data implosion with the destination getting multiple data packets from multiple paths. Also, consider that two sensor nodes monitor an overlapping region of the environment. The classic flooding approach (as any other low level routing protocol) cannot recognize the data overlap and optimize for it by preventing multiple transmissions of the overlapping data region.

The protocol called SPIN (**S**ensor **P**rotocols for **I**nformation via **N**egotiation) [5][13] grew out of the idea that a sensor node should handshake with its neighbors and decide on the data that it already has and the data that it needs to obtain before initiating the operation to get the data. Nodes in SPIN label their data using high-level data descriptors called *meta-data* and use meta-data negotiation to determine if a node needs the data and therefore eliminates redundant transmissions. In this paper, we propose a protocol called SPMS (**S**hortest **P**ath **M**inded **S**PIN) that reduces the energy consumption and the end-to-end delay of SPIN. We achieve this by using the fact that sensor nodes can operate at multiple power levels and once negotiation of meta-data is initiated, the remainder of the protocol and the data transfer can occur in multiple hops using the lowest energy level. At first glance, it would appear that SPMS would increase the data latency between the destination and the source. However, this turns out not to be the case. The effect of reducing the power level of transmission causes a smaller level of MAC layer contention for the shared wireless channel and therefore reduces the MAC layer backoff delay which contributes to a lower overall delay in SPMS. SPMS is also resilient to node and link failures since the data is exchanged through intermediate nodes and they may cache the data to tolerate failures of the source or another intermediate node. In order to do multi-hop routing, we run a distributed Bellman Ford algorithm among the nodes in a zone; a zone being defined as the area a node can reach transmitting at its maximum power level. Each node maintains routes to other nodes in the zone. In this paper, we quantify the cost of the algorithm under mobility scenarios where the Bellman-Ford algorithm needs to be re-executed.

We perform a theoretical analysis to show the energy and delay improvements. We also conduct simulations to show the improvements over SPIN under static and mobility scenarios, with and without failures. The simulation results show that SPMS reduces the delay over 10 times and consumes on an average 30% less energy in the static failure free scenario. Even with the addition of mobility, SPMS outperforms SPIN by energy gains between 5% and 21%.

The rest of the paper is organized as follows. In section 2 we discuss related work. Section 3 describes our protocol design and algorithm. In section 4 we provide a theoretical analysis comparing the delay reduction and energy gains in SPMS over SPIN. In section 5 we provide the simulation based evaluation of SPMS. Section 6 concludes the paper and addresses future work.

2 Related Research

Several routing mechanisms have been proposed in the literature to address the data dissemination problem in sensor networks, such as, broadcast and gossip. To reduce the expense of routing table creation at each node, Haas and Pearlman [4] have proposed a zone routing protocol. Each node proactively maintains routes for nodes within its zone and reactively acquires the routes to nodes outside the zone only when it needs to transmit outside the zone.

Communicating data in an energy-efficient manner from a sensor node to the base station, in particular, and another sensor node in the sensor field, in general, has received enormous attention of late. Fusing the energy efficiency with data latency and fault tolerance concerns has received some, but markedly less, attention. The current state-of-the-art and the unanswered questions that motivate SPMS are summarized here. Hari *et. al.* have proposed a protocol called LEACH [3] where the nodes communicate directly with the respective cluster head and the cluster heads communicate with the base station. The cluster head role is taken by different nodes, either in a round-robin manner or depending on the level of remaining energy. The protocol does not consider the end-to-end latency for the data since a fixed time division multiplexed schedule is enforced on the nodes in a cluster for data transmission. It assumes that all nodes are capable of performing direct sequence spread spectrum communication and that the base station is within communicating distance of all nodes. These assumptions respectively limit the economic feasibility and scalability of the solution. The solution briefly mentioned for the scalability challenge uses a backbone of cluster heads to reach the base station. This has several complexities that need to be addressed, e.g., efficient backbone construction is equivalent to constructing a minimum connected dominating set which is known to be NP-Complete.

There has been follow-up work by Raghavendra *et. al.* in a system called PEGASIS [6], which further minimizes the energy by sending all the data through only one node. All the cluster head data is fused and then only one node sends the entire data to the sink node. The clustering approach is an orthogonal method to that in SPMS or SPIN. We feel the clustering approach is feasible if the network has a fairly regular structure and the structure can be easily deduced and this information distributed throughout the network. Nuggehalli and Srinivasan propose a protocol called POACH (**Po**wer **A**ware **C**aching **H**euristics) [10] where they address the problem of determining the servers in a sensor network at which the data should be cached so that the overall cost of data dissemination from the sink node is minimized. The paper provides a closed form solution for deciding the placements of the data caches. But the paper does not address the issue of failures of these data caches. Also, placing the caches at specific points requires those nodes to have more memory and computation power, which further requires careful placement of nodes. SAFE [1] is a protocol for data dissemination from stationary sensor nodes to mobile sink nodes. It is a pull-based protocol where a path is set up between a sink and the source (which the sink knows *a priori*) when the sink needs the data. If any other sink needs data from the same source, the protocol finds an efficient path that overlaps with the previous paths from the source. The node nearest to the latest sink that is on the common path is called a junction node. This has the flavor of multicast tree formation; but here the requestors arrive at different times. A concern with the protocol is the amount of state that needs to be maintained at intermediate nodes (distance from a large number of sources for all the flows that are flowing through it). Also, the multiple rounds of message exchange required to set up a path add to delay in the critical path of data dissemination (5 rounds of message exchange before a junction node can start serving a sink). **T**wo **T**ier **D**ata **D**issemination protocol (TTDD) [2] is another protocol for disseminating data from stationary sensor sources to multiple mobile sinks. The goal is to prevent the explosion of messages due to the sinks broadcasting their change of position information. TTDD proactively sets up a grid structure for each data source with sensors (called dissemination nodes) having forwarding information to reach the source. When a sink needs data, it floods the query to a local region and any dissemination node in the local region picks up the query and routes data from the source. The cost of proactively creating and maintaining the grid structure from all potential sources to the edge of the sensor field can be high. The sizes of the cells and their setup which determine the performance of the protocol are sensitive to the movement patterns of the sinks which may be difficult to predict.

There has been considerable interest in distributed topology control algorithms that seek to find minimum energy paths between any two nodes in a sensor network ([18],[19],[20]). The protocols find the nodes that need to be active for minimum energy paths between a given set of nodes, or the transmission power levels of the individual nodes to maintain certain properties such as connectivity in the network. This approach typically does not consider the combination of energy, data latency, and reliability, and also does not address the issues of data implosion and overlap which are mentioned next.

3 SPMS Protocol

3.1 Background: SPIN

Hari *et. al.* [5] have proposed a class of protocols called **S**ensor **P**rotocols for **I**nformation via **N**egotiation (SPIN) that is motivated by two problems with existing wireless sensor network data distribution protocols – *Implosion*, i.e. a node always sends data to its neighbor irrespective of whether it already has it or needs it, and *Overlap*, i.e. nodes often transmit redundant information because of overlap in their sensing regions. In SPIN, nodes *name* their data using high-level data descriptors, called *meta-data*. Then, the source and the destination participate in meta-data exchange prior to data exchange which ensures that only useful information gets transmitted. A source node initiates the meta-data transfer by sending an advertisement packet and data is sent to only those nodes which request the data after seeing the advertisement. Since advertisement packets are much smaller than data packets, SPIN has energy savings over the basic flooding protocol. Each node in SPIN also monitors its resource availability (e.g., available battery power) and decides on its data dissemination activities accordingly, e.g., if it would forward a third-party node's packets.

3.2 Design of SPMS

We know from path loss models that energy spent in wireless communication is proportional to d^{α}, where d is the distance between the source and the destination and α is a constant between 2 and 4. The idea that going over long distances incurs an exponentially increasing energy cost was the motivating factor behind multi-hop communication in ad-hoc networks. This idea also forms the basis for our SPMS. SPIN suffers from the drawback of transmitting all packets at the same power level and not using the distance to a neighbor to adjust the power level. SPMS borrows the concept of meta-data exchange from SPIN, and uses a multi-hop model for data transmission among the nodes with variable transmission power levels.

However, using multiple hops to get to the destination throws two major challenges – knowing the route to the destination and dealing with failures of intermediate nodes. Regarding the first problem, there have been various routing protocols proposed for ad-hoc networks like DSR and AODV and for sensor networks like directed diffusion. Since sensor networks may consist of hundreds of thousands of redundant nodes, it is infeasible to maintain a routing table at each node with routes to all other nodes in the network.

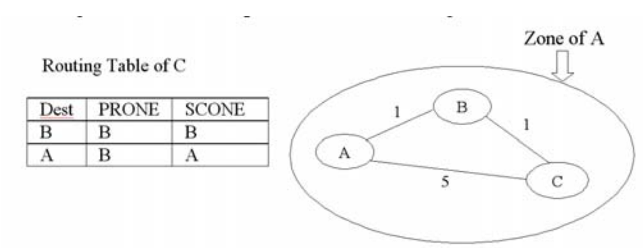


Figure 1: A sample network. Each link has a cost associated with it which represents the transmission power needed to reach the neighbor.

To reduce the cost of route discovery and maintenance, we define a *zone* for each node. A zone for a node is the region that the node can reach by transmitting at the maximum power level. The nodes which lie within a node's zone are called its *zone neighbors*.

Each node in the network maintains a routing table for each of its zone neighbors. The Distributed Bellman Ford (DBF) algorithm is executed in each zone to form the routes. Each entry of the routing table at each node has a destination field and the cost of going to the destination through each of its neighbors. Maintaining n entries for each destination enables the protocol to tolerate concurrent failures of n intermediate nodes. For a given node, the next hop node in the path to a given destination as decided by DBF is called its *next hop neighbor*. The convergence time of DBF with n nodes is $O(n.e)$, where e is the number of edges. (which is equivalent to $O(n^3)$ in a fully connected graph). The zone's size is expected to be much smaller than the entire sensor network size for the small transmission radius of sensor nodes and reasonable node densities of most sensor network deployments. Therefore, the cost of DBF is not considered prohibitive for the sizes of zones in practice (5-50 nodes). When a node moves or a failure occurs, the routing tables of its zone neighbors get updated through re-execution of the DBF. If a graphical representation of the network is considered where the weight w on an edge (i,j) denotes the minimum power at which i needs to transmit to reach j, DBF finds the shortest path between any two nodes in the weighted graph.

The nodes start transmitting after the routing converges, i.e., a run of DBF terminates. The first phase of data transmission involves meta-data exchange as in SPIN. When a node (the source) has some data to transmit, it advertises its data using an ADV (advertisement) packet broadcast to its zone neighbors. On receiving the ADV packet from its zone neighbor, a node first checks if it needs that data by reading the meta-data in the ADV packet. If it does, the node sends a REQ (request) packet to the source. In SPIN a REQ packet is sent directly to the source, but in SPMS the node sends the REQ packet to the source through the shortest path. Thus, if the source is not the next hop neighbor, the REQ packet is sent through multiple hops. If the destination realizes it has to do multi-hop communication for its REQ packet, it waits for a pre-determined fixed period of time before sending the request packet. The logic is that every node should request the data from nodes which are close by and hence can be reached by transmitting at the lowest possible power level. If there are relay nodes between the destination node and the source node, the destination node waits, expecting to hear the ADV of the data from a relay node. The SPMS protocol requires a node to advertise its own data as well as all received data once amongst its neighbors. Hence, if any intermediate relay node gets the data, it advertises the data. To handle the case when the relay node does not request the data, the destination node starts a timer on hearing the advertisement and on its expiry sends the REQ packet to the source through the shortest route. The timer for this purpose is τ_{ADV} and the timeout value is set to $TOut_{ADV}$. In this case, the REQ packet still goes through the relay node but it is destined for the source node. By sending the REQ packet through the shortest path in multi-hop fashion, SPMS saves energy compared to direct transmission to the source node. Each node after sending the REQ packet starts a timer τ_{DAT} with value $TOut_{DAT}$ to avoid waiting indefinitely for the data. Finally, the data is sent in exactly the same manner as the received request, i.e., direct from the source to the destination if they are next hop neighbors, otherwise through multi-hop communication.

Note that in SPMS no node needs global state information, either for routing or for failure status. The routing information is maintained at a node only for its zone neighbors. The failure information is obtained transiently for a node that it tries to communicate with and gets no response.

3.3 Example for Failure Free Case

Consider an example scenario where there are 3 nodes A (source), B and C. Each node is a zone neighbor of the other. The routing tables have been formed as described above using DBF. The shortest route from A to B is a direct transmission to B. The shortest route from A to C goes through B. Node A broadcasts the ADV packet to all its zone neighbors.

Case I: Both nodes B and C need the data. After receiving the ADV from A, B requests the data by sending a REQ packet to A directly. On receiving the REQ packet, A sends the DATA packet to B, again directly. C on receiving the ADV packet checks in its routing table and goes into a waiting state. It starts the timer τ_{ADV} and waits for B to advertise the same piece of data. Node B on receiving the data advertises it in its zone. Suppose C's timer τ_{ADV} has not expired yet. Then it receives the ADV packet from B lying in its zone. Since B is a next hop neighbor, C sends a REQ packet to B directly, cancels its timer τ_{ADV} and starts the timer τ_{DAT}. In the failure free situation, C gets the data from B in response to its request.

Case II: B does not request the data from A and hence will not advertise the data. Now, as before, C goes into the waiting state with its timer τ_{ADV} and waits for an ADV packet from B. After the timer expires, C sends a REQ packet to A but through the shortest route, i.e., routed through B. B relays the REQ packet to A and A finally sends the DATA packet back to C through B.

3.4 Design for Failure Cases

SPMS relies on the relay nodes for data delivery and should be resilient to intermediate node and link failures. At any stage of the protocol, the destination node maintains a **Pr**imary **O**riginator **N**od**e** (PRONE) and a **S**e**c**ondary **O**riginator **N**od**e** (SCONE). The PRONE is the first choice node for requesting the data from, while the SCONE is the second choice to be used in case the PRONE is unreachable because of a link failure or because the PRONE itself is down. In a general scenario, multiple SCONES may be maintained for tolerating more than one concurrent failure. At the beginning of the protocol, both PRONE and SCONE are initialized to the data source node. If the destination node receives an ADV packet from a closer node, then it sets the PRONE to be the closer node and the SCONE to be the PRONE from the earlier stage. In , B is the PRONE and A is the SCONE for C after it receives the ADV packet from B.

If the τ_{DAT} timer expires before the node gets the data, it sends the REQ packet to the SCONE. If the τ_{DAT} timer of a node expires because the closer node (call it N_C) failed after getting the data, the destination node has no way of distinguishing between this case and the case where the closer node did not request for the data. The destination node sends a REQ packet to its PRONE using multi-hop routing which may go through N_C. If N_C has failed, the destination node's τ_{DAT} timer expires and it finally requests the data directly from the PRONE, using a higher transmission power. Note that it is guaranteed to reach its PRONE using an available transmission power since they are each other's zone neighbors. Thus, SPMS can tolerate

1. Failure of the source node after its data has been received by any of its zone neighbor nodes
2. Failure of any intermediate node during the entire protocol.

3.5 Example for Failure Case

In Figure 2, the nodes r1, r2 and C are A's zone neighbors. A broadcasts an ADV packet in its zone, which is received by the three nodes. Assume that all the nodes request for the data. A is the PRONE and the SCONE for each of the other nodes.

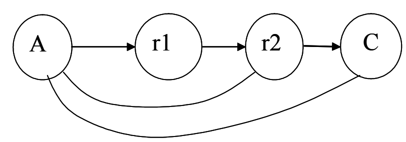


Figure 2: The above figure illustrates routing of data packet from A to C.
Arrows indicate the shortest route from A to C. All lines are the links existing among the nodes.

On receiving the ADV packet, nodes r2 and C go into waiting with timer τ_{ADV}, but r1 goes ahead and requests the data from A. After receiving the data, r1 re-advertises it in its zone. C on receiving the ADV packet from r1 resets its timer τ_{ADV} and sets its PRONE to r1 and SCONE to A. Node r2 cancels its timer and requests the data from r1 since it is its next hop neighbor.

Case 1. Suppose r2 fails before sending out an ADV packet. Node C's timer τ_{ADV} expires because it does not see any ADV packet from r2. C sends a REQ packet to r1 using its shortest path routing table, which means it would go through r2. As r2 has failed, C's τ_{DAT} timer expires. Now C requests the data from the PRONE (r1) directly using a higher transmission power. Node r1, on receiving the REQ packet from C, sends the data as direct transmission because that was the route followed by the REQ packet.

Case 2. In the second case, r2 fails after sending the ADV packet. In this case, C sends a REQ packet to r2 as it is its next hop neighbor. Since r2 has failed, C's timer τ_{DAT} expires. It then sends a REQ packet to the SCONE (r1) directly. Similar to the previous case, r1 sends the data to C through a direct transmission.

4 Theoretical Evaluation

We compare SPMS against SPIN using a detailed mathematical analysis. The analysis deals with the delay and the energy consumption.

4.1 Delay Analysis

Let R,D,A be the lengths of REQ, DATA and ADV packets, T_{tx} the time for transmission of one unit of data, $TOut_{ADV}$ and $TOut_{DAT}$ are the timeouts respectively for getting an advertisement from a relay node and data in response to a request. T_{proc} is the processing delay at a node receiving a data or control packet. This is independent of the number of bits processed. This eliminates the unrealistic simplification in the SPIN simulations where the data is taken to be processed instantaneously. The propagation delay is assumed to be zero. T_{csma} is the delay to access a channel, which is taken to be proportional to n^2, where n is the number of nodes in the transmission radius ([8],[9])[1]. Let G be the proportionality constant. Let $n1$, ns be the number of nodes reachable when the node transmits respectively at the maximum power level and at the lowest power level and ρ the density of nodes.

We derive the delay for a simple scenario (Figure 2) and then extend it to a more general scenario. Node A is the source node and sends the advertisement.

4.1.1 *Analysis of SPIN*

*a) Failure Free Case***:** Let T_b be the time for B to receive the data measured from the time (and including) A sending out the ADV. All the nodes are transmitting at their single maximum power level with a transmission

[1] Other models for MAC layer delay have used higher powers of *n*, or exponential function of *n*. These can be incorporated into our analysis directly with just replacement of the MAC delay term(s) and this would bias the analysis and results more in favor of SPMS.

radius of *n1* nodes. There are transmission delays associated with transmissions of ADV, REQ and DATA packets. Also there is processing delays at B for ADV and at A for REQ.

In general: Delay for any transmission = Delay due to MAC layer contention for the channel + Transmission delay of the packet + Processing delay

$$T_b = G\,n1^2 + AT_{tx} + T_{proc} + G\,n1^2 + DT_{tx} = 3G\,n1^2 + (A+R+D)T_{tx} + 2T_{proc} \text{ ——— } (1)$$

The same calculation holds for C, since both B and C request the data independently.

b) Failure Case: Consider that nodes may fail. The time window for failure is $(0,T_b)$. In case of failure of source node A, it is not able to transfer data to any of the nodes. Then the nodes, which have the data, re-advertise and the nodes which could not get the data eventually get the data from them but it is not possible to do an analysis for this scenario as it depends upon the network topology.

4.1.2 Analysis of SPMS

a) *Failure-free case*

We assume that in-order to get to B, A transmits at a lower level covering only a radius of *r2* which has *n2* (<*n1*) nodes. The advertisement from A is at the highest power level as earlier.

$$T_b = G\,n1^2 + AT_{tx} + T_{proc} + G\,n2^2 + RT_{tx} + T_{proc} + G\,n2^2 + DT_{tx} = G\,n1^2 + G\,n2^2 + (A+R+D)T_{tx} + 2T_{proc} \text{ ——— } (2)$$

End-to-end delay at C is dependent on whether B requests for the data or not.

Case a.a) B also requests the data. The entire A-B sequence is repeated twice for the two hops. Here we assume that $TOut_{ADV}$ is adjusted properly so that the timer does not go off before B sends ADV.

The delay for C to get the packet is given by $T_{c1} = 2(G\,n1^2 + 2G\,ns^2 + (A+R+D)T_{tx} + 2T_{proc})$

An approximation for the timeout value is $TOut_{ADV} > ns^2 + RT_{tx} + T_{proc} + DT_{tx} + ns^2 + T_{proc}$

Let $T_{round} = G.n1^2 + 2G.ns^2 + (A+R+D)T_{tx} + 2T_{proc}$

Case a.b) B does not request the data. C does a timeout $TOut_{ADV}$ after which it requests the data through B, i.e. $2 \cdot R \cdot T_{tx}$ and processing delay at A and B of that REQ packet which is $2 \cdot T_{proc}$. Finally A routes the data through B which takes $2 \cdot D \cdot T_{tx}$ and a T_{proc}. C receives the data and incurs a delay of T_{proc}. With each channel access, there is a CSMA/CA delay to access the channel.

$$\begin{aligned} T_{c2} &= G.n1^2 + 4G.ns^2 + AT_{tx} + TOut_{ADV} + 2RT_{tx} + 2T_{proc} + 2DT_{tx} + 2T_{proc} \\ &= G.n1^2 + 4G.ns^2 + (A + 2R + 2D)T_{tx} + 4T_{proc} + TOut_{ADV} \end{aligned}$$

For a completely general topology, we can assume that the distances AB and BC are different, hence require different power levels of transmission. Let the number of nodes in the transmission radii for the two power levels be *n2* and *n3* respectively. Since ADV is sent at the maximum power so the accessing delays for it remains at $G.n1^2$ (the first term).

Case a.c) K relay nodes between A and C. The worst case delay occurs when the last relay node doesn't request the data. For the first *(k-1)* nodes the data ripples through for a time of $(k\text{-}1)\ T_{round}$ and then it is the same case as analyzed in the previous section when B doesn't request the data and we calculate the delay for C.

$$T_C \le (K-1)T_{round} + TOut_{ADV} + T_{c2} \text{ ——— (3)}$$

b) Failure case

Case b.a) B fails before sending the ADV. There is a $TOut_{ADV}$ incurred at C. It would not matter if B would have requested the data or not because we have taken $TOut_{DAT}$, which counts all the delays occurred at B. Now again, C still requests through B and since B has failed so it runs through the complete $TOut_{DAT}$ and finally requests the data from A. So there are the processing delays at A (REQ) and C (DATA).

$$T_{c1} = G.n1^2 + G.ns^2 + 2G.n2^2 + (A + R + D)T_{tx} + TOUT_{ADV} + TOUT_{DAT} + 2T_{proc}$$, where, ns<n2<n1

Case b.b) B fails after sending the ADV. C sees the ADV from B and does not time out. But it still does a $TOut_{DAT}$ timeout as B does not respond to the REQ. Later C requests the packet from A.

$$T_{c2} = T_{round} + G.ns^2 + AT_{tx} + T_{proc} + G.ns^2 + RT_{tx} + TOut_{DAT} + 2G.ns^2 + (A + D)T_{tx} + 2T_{proc}$$

If there are K relay nodes and the (K-1)th node fails, the time is given by $T_c = (K-1)T_{round} + T_{c1}\ (or)\ T_{c2}$ depending on case (a.a) or case (a.b).

Consider Figure 4 where in a chain of *k* intermediate nodes, one intermediate node fails, which is not the last one, say the j^{th} node from the last one fails. There are *(k-j)* rounds for data to get to the $(k-j)^{th}$ node. Then there is a $TOut_{ADV}$ as C does not hear the ADV. C sends the REQ through the shortest route (having the failed node) and in the process incurs a delay of $TOut_{DAT}$ and finally it requests through the last heard node which is the $(k-j)^{th}$ node. Considering *nj* is the number of nodes covered in the power level when for transmission from node C to node *j*, where $ns<nj<n1$.

$$Delay = (k-j)T_{round} + TOut_{ADV} + G.ns^2 + TOut_{DAT} + 2G.nj^2 + (R+D)T_{tx} + 2T_{proc}$$

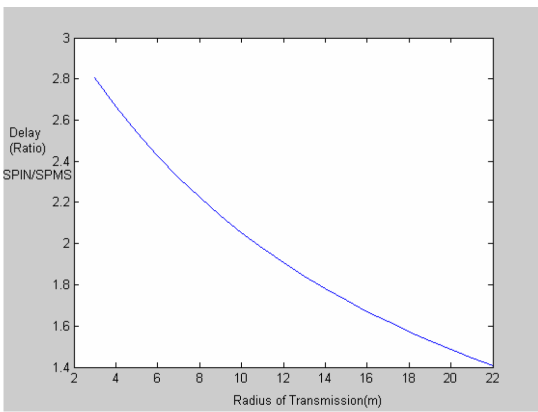


Figure 3: Graph of ratio of end-to-end latency for SPIN to SPMS as the transmission radius varies (obtained from theoretical analysis)

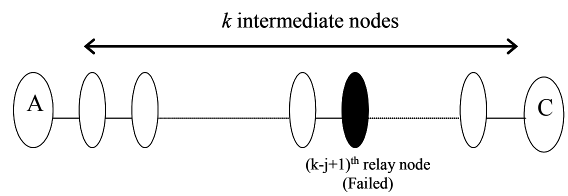


Figure 4: A sample scenario with k intermediate relay nodes

In order to compare the delay between SPMS and SPIN for a single source-destination pair in the failure free case, we take expressions (1) and (2) and plot the ratio of the delays with respect to increasing transmission radius in Figure 3. Consider sample values of $T_{tx} = 0.05$, $T_{proc} = 0.02$, A:D = 1:30, G= 0.01 and, as given by [9], n1 = 45 and ns = 5. These values assume a uniform density of nodes on the grid and consider that by increasing each power level, the number of links between two nodes is reduced by one.

$Delay_{SPIN} : Delay_{SPMS} = 2.7865$

4.2 Energy Analysis

Let the energy expended per bit corresponding to the different transmission power levels be E_1, E_2, E_3, E_4....... E_m, where $E_i > E_{i+1}$. Let E_r be the energy required to receive the packet. For simplification we can assume that this is equal to E_m which is valid for many sensor nodes [16].

Consider a simple example with two nodes – A (source) and B (destination) with *(k-1)* relay nodes in between. In case of SPIN it does not matter how many relay nodes there are, since the source always transmits at the maximum power level. In this calculation, we omit the energy wasted in redundant reception by nodes that do not wish to participate in the protocol. Since the number of uninterested receivers is higher in SPIN because of a larger transmission radius, the gain in SPMS will be higher if we take this into account.

$$E_{SPIN} = (A + D + R)E_1 + (A + D + R)E_r$$

$$E_{SPMS} = k.A.E_1 + k(D + R)E_m + k(A + D + R)E_r$$

If we consider the actual value from our experiments with the Berkeley motes, D~32·A =32·R

Assuming f = A/(A+D+R), $E_{SPIN} : E_{SPMS} = \dfrac{E_1 + E_r}{k.f.E_1 + k.E_m + k.E_r}$

If we take k relay nodes in a straight line and are equally spaced then D(ab) = d_0 + d_0….k times = k.d_0. Let us assume the energy model where energy is proportional to the distance as $d^{3.5}$, e.g., the 2-ray ground propagation model α is close to 3.5 beyond 7 meters **Error! Reference source not found.**.

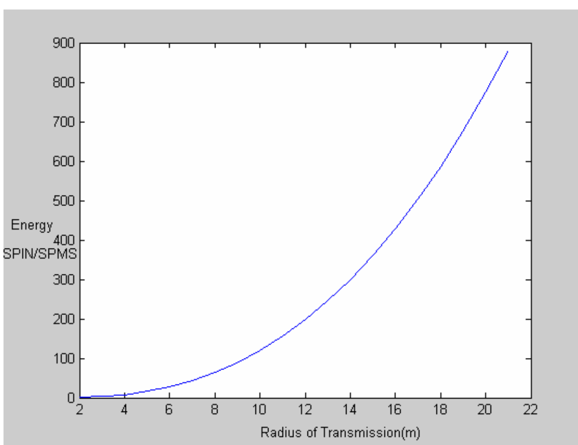


Figure 5: Ratio of Energy (SPIN/SPMS) with varying radius of transmission.

Hence putting in all the values we get

$$E_{SPIN} : E_{SPMS} = \frac{1 + k^{3.5}}{k(f.k^{3.5} + (2 - f))}$$

Figure 5 shows the plot of energy ratio with varying radius of transmission (for grid granularity of 1 unit and a node on every grid point, $k = r$). We can see from the graph that as the radius increases, SPMS does substantially better in saving energy compared to SPIN. The increase in the radius contributes to the increase in the zone size which leads to increase in the number of intermediate hops.

5 Experimental Evaluation

We carried out a simulation based study of SPIN and SPMS to bring out the difference in energy saving and delay. In our experiments we use a sensor field with uniform density of nodes. This implies that as the number of nodes increases, the sensor field area increases. The input parameters are either taken from the MICA2 Berkeley mote datasheet (e.g., the five transmission power levels) or are influenced by our practical experiments with the motes (e.g., the sizes of ADV, REQ and DATA packets).

λ (Packet Arrivals)	1 /ms	λ (Failures)	50ms	Processing Time	0.02 ms
Slot Time	0.1	MTTR	10ms	$TOut_{ADV}$	1.0 ms
No. of Slots	20	Power level (1-5)	3.1622, 0.7943, 0.1995, 0.05, 0.0125mW	$TOut_{DAT}$	2.5 ms
Time of transmission	0.05ms/byte	Distance (1-5)	91.44, 45.72, 22.86, 11.28, 5.48m	Size of DATA:REQ	20
Size of REQ or ADV	2 Bytes				

Table 1: Simulation Parameters

5.1 All-to-All Communication

In the first set of experiments we consider all-to-all communication. In this model each node generates 10 new packets and every other node in the network is interested in receiving each packet. We consider Poisson arrivals for the new packets. All-to-all communication is simulated since it is the most general communication pattern, special cases of which are given by sink to source or source to sink communication.

5.1.1 Static Failure Free Case

We consider energy and delay metrics varying the number of nodes in the sensor field. The energy plot is shown in Figure 6. Total energy consumption is calculated for the entire network and divided by the total number of packets. SPMS saves 26-43% of energy compared to SPIN. As the number of nodes increases, the number of packets sent increases. The energy consumed curve for SPIN has a higher slope than that for SPMS and hence the difference increases with increasing sensor field size.

We compare the effect of varying the transmission radius on energy consumption for both the protocols in Figure 7. It is important to consider this metric as the nodes have the capability of transmitting to different ranges. Also, as the transmission radius increases, the number of zone neighbors considered in SPMS increases and hence the overhead of the Bellman-Ford algorithm increases. In spite of this, as the transmission radius increases, SPMS increasingly outperforms SPIN. At low values of the radius, the difference between SPMS and SPIN is not substantial because the zone has very few neighbors and mostly one hop away. However, as the transmission radius increases, the zone size increases and SPMS uses longer multi-hop routes to reach the outlying nodes in the zone.

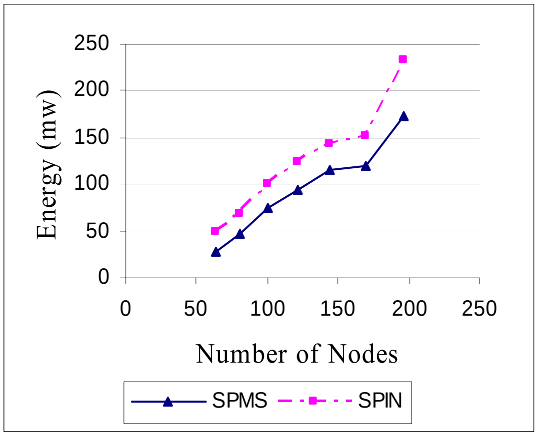


Figure 6: Energy consumed by SPIN and SPMS with varying number of sensor nodes (transmission radius=20 m)

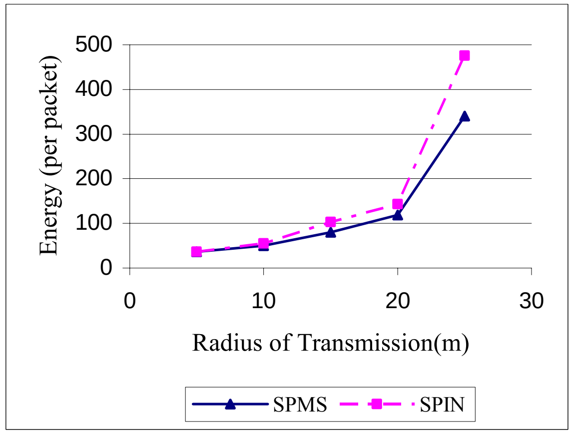


Figure 7: Energy consumed by SPIN and SPMS with different transmission radii (number of nodes=169)

We compare the delay incurred in transmitting packets under SPIN and SPMS. The delay is measured from the time the ADV packet is sent out by the source to the time that the data packet is received at the destination. The delay is obviously different for different source destination pairs and for the results; the average delay across all the packets is plotted. We can see from Figure that the delay increases with the number of nodes for both SPMS and SPIN but the delay in sending packets is much less in SPMS. SPMS gets the packet across almost 10 times faster than SPIN. The delay difference between SPIN and SPMS widens with increasing number of nodes. This is because the delay to a node increases faster in SPIN since each round of SPIN where the data is transmitted to all the nodes in a zone takes longer than a corresponding round of SPMS. With increasing sensor field size, more number of rounds has to be executed to reach a given destination.

The effect of the transmission radius on the delay is shown in Figure. As the radius increases, the delay drops for both SPIN and SPMS. At first, one may think that since the number of nodes is increasing, the delay would increase. As the radius increases, the number of nodes in the zone increases. This increases the number of nodes that hear the ADV and acquire the data packet in response in the same round. As opposed to this, for a smaller transmission radius, the data has to ripple through multiple zones before reaching the destination. Rippling through each zone takes place in a round which adds to the delay. This decrease in delay with increasing radius offsets the increase due to increased contention at the MAC layer as the number of nodes in a zone increases.

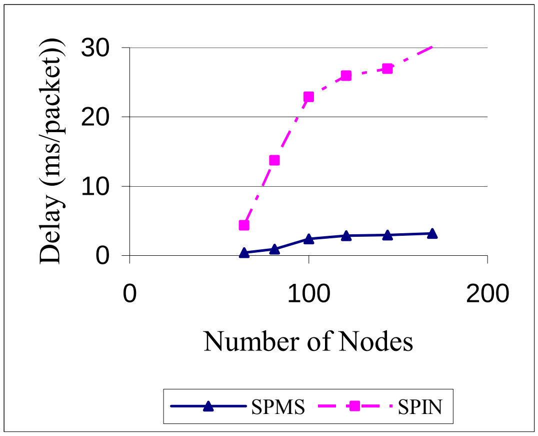


Figure 8: End-to-end delay with varying number of nodes (transmission radius = 20m)

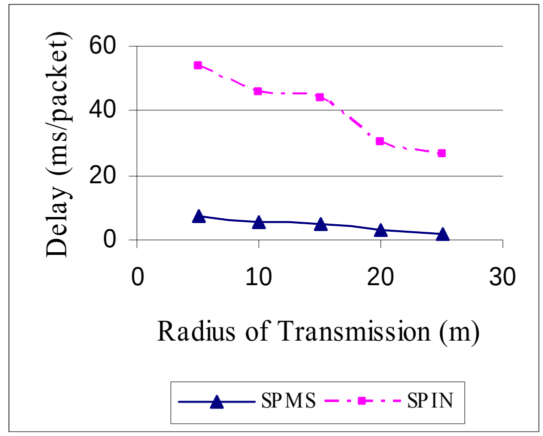
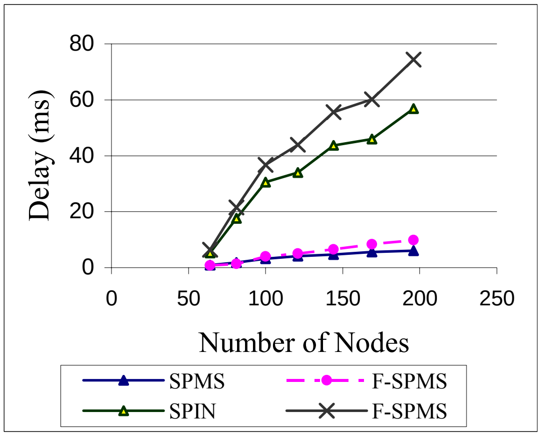


Figure 9: End-to-end delay variation with transmission radius (number of nodes = 169)

5.1.2 Static Failure Case

We test SPMS and SPIN under failure scenarios, called F-SPMS and F-SPIN respectively. The type of failures considered is transient node failures. Nodes fail with an exponential inter-arrival time (mean λ) and stay failed for a time drawn from a uniform distribution ($repair_{min}$, $repair_{max}$). During the time of repair, any received message is dropped and any scheduled packet transfer is cancelled. We assume recovery is always successful. In our implementation, the routing table keeps only the shortest (i.e., least cost) and the second shortest path to the destination which tolerates only one failure during the recovery window.

In order to compare the results against the failure-free runs, the number of new packets generated by each node is kept at the same. As expected, the delay increases (Figure 10) in the failure cases because now some nodes in the zone have to wait for the timer τ_{ADV} or τ_{DAT} to go off and then request the packet through the alternate path. The delay difference between the failure and the failure free runs for the small radii is small as there are less intermediate hops. As the radius increases there are relay nodes whose failure induces the delay in SPMS in getting the packet. Scalability test was also carried out for the failure cases. For small number of nodes, the length of the path is smaller and therefore the number of failures being activated is smaller. Therefore, the difference between the failure free and failure cases is not substantial, but it becomes pronounced as the number of nodes increases.



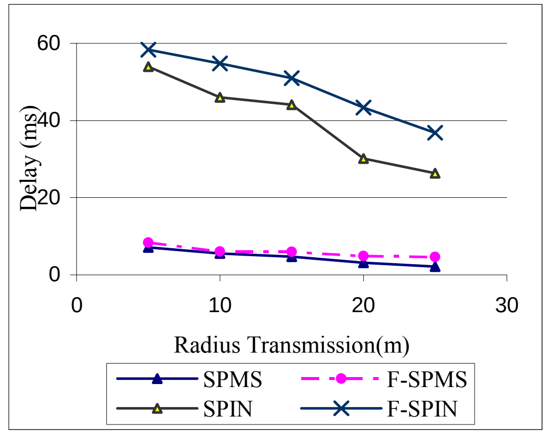


Figure 10: End-to-end delay with varying number of nodes for static nodes with transient failures

Figure 11: End-to-end delay with transmission radius for static nodes with transient failures.

5.1.3 Mobile Failure Free Cases

In this set of experiments, nodes are allowed to move. As nodes move, the routing tables have to be modified and no packet transfer can take place until the routing tables converge. At some discrete times in the simulator clock, a predefined fraction of nodes move. The nodes which are to move and their destination are chosen randomly. Once the routing tables converge, the data transmission starts all over again. The energy expended in SPMS in forming routing tables is included in the energy measurement.

Figure12 shows that even with mobility SPMS performs better than SPIN although the percentage of energy savings go down (varies from 5%-21%). As the frequency of node motion increases, energy is required for setting up the routing tables. Our calculations with the cost of running Bellman Ford and the energy gain of SPMS over SPIN lead us to conclude that at least 239.18 packets must be successfully transmitted between two instances of network mobility for SPMS to save energy compared to SPIN.

5.2 Cluster Based Hierarchical Communication

The cluster heads are responsible for collecting the data and so request the data if they need it. The other nodes in the zone of the source node can also be interested in data with a probability of 5%. The energy consumption is measured by varying the zone radius. In SPMS the communication between the nodes and the cluster head is multi-hop compared to a direct transmission in SPIN. In Figure 13 we see that SPMS consumes 35-59% less energy than SPIN for the failure-free case. As the transmission radius increases a cluster has a larger radius with more scope for the multi-hop communication of SPMS and hence, the energy difference increases. Next, node failures are injected in the same manner as for all-to-all communication. As one can see from the Figure 13, in failure cases, the energy expended by the protocols is much more than for the failure-free runs.

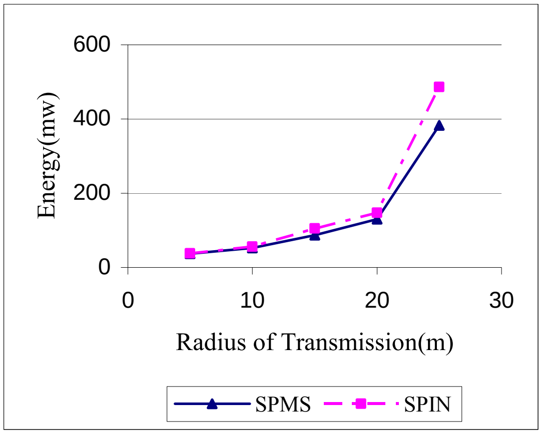


Figure 12: Energy consumed with transmission radius for mobile nodes in all-to-all communication

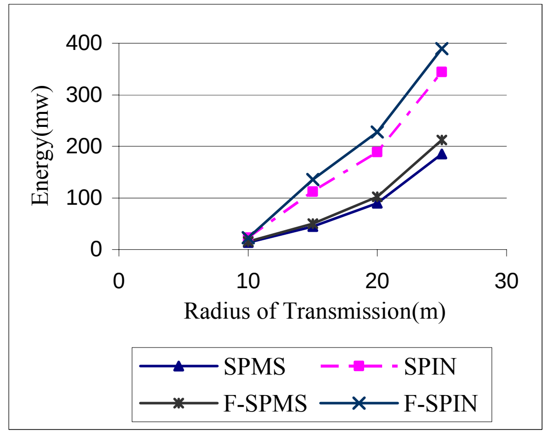


Figure 13: Energy consumed with transmission radius for cluster-based hierarchical communication

6 Conclusion

In the paper, we have presented a protocol called SPMS for data dissemination in energy constrained wireless sensor networks. The protocol uses meta-data exchange prior to exchange of data to decide if a node requires the data. It uses a hybrid of push-pull – pushing the advertisement of the data from the source followed by a pull from the interested destination. Unlike an earlier published protocol SPIN that also uses meta-data exchange, SPMS uses shortest distance multi-hop routing for the request and data transfers. This allows it substantial energy savings. Also, somewhat counter-intuitively, SPMS reduces the end-to-end data latency. Since transmissions need to reach only the next hop, they can happen at low transmission power levels. This reduces contention for the shared wireless MAC channel and thus reduces the delay due to MAC layer backoff.

We present a theoretical analysis of the energy expended and delay in SPMS and compare it to that in SPIN. As the transmission radius increases, the delay in SPIN approaches that in SPMS, but its energy consumption increases substantially reaching up to three orders of magnitude higher. We conduct a simulation based comparative study of the properties of SPMS and SPIN. The simulation study includes two communication patterns and both failure and node failure cases. SPMS performs better for the static cases, while in the mobile cases; it incurs the cost of routing table formation but is still shown to outperform SPIN.

For future work, we propose to consider an extension to SPMS to disseminate data when the source and the destination are in separate zones with no interested nodes in the intermediate zones. This would require the use of zone routing of [4] and the request phase of the protocol to go across zones. We are also investigating the issue of data caching at intermediate nodes which route the data but are not receivers. This can improve the fault tolerant property of the protocol.

References

[1] Kim, S., Son, S. H., Stankovic, J., Li, S., Cho, Y., "SAFE: A Data Dissemination Protocol for Periodic Updates in Sensor Network", *In Proceedings of International Workshop on Data Distribution for Real-Time Systems*, 2003.

[2] Ye, F., Luo, H., Cheng, J., Lu, S., and Zhang, L., "A two-tier data dissemination model for large-scale wireless sensor networks", *In Proc. of the Eighth ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom 2002)*, Atlanta, Georgia, USA, September 2002.

[3] Heinzelman, W.B., Chandrakasan, A..P.., Balakrishnan, H., "An application-specific protocol architecture for wireless microsensor networks", *IEEE Transactions on Wireless Communications*, October 2002, pp. 660-670.

[4] Hass, Z., Pearlman, M. R., "The Performance of Query Control Schemes for the Zone Routing Protocol*", IEEE/ACM Transactions on Networking*, vol 9, No.4, pp.427-438, August 2001.

[5] Heinzelman, W. R., Kulik, J. and Balakrishnan, H. "Adaptive Protocol for information Dissemination in Wireless Sensor Networks", *Mobicom* 1999.

[6] Lindsey, S., Raghavendra, C. and Sivalingam, K. M., "Data Gathering Algorithms in Sensor Networks Using Energy Metrics", *IEEE Transactions on Parallel and Distributed Systems*, vol 13 No.9,pp. 924-935, September 2002.

[7] Chhaya, H. S., Gupta, S., "Wireless Merging onto the Information Superhighway", *Sixth IEEE international Symposium on Personal Indoor and Mobile radio Communication, PIMRC'95*, Volume 2, 27-29, sept 1995 pages 613-617.

[8] Kim, J. H. and Lee, J. K., "Performance analysis of Mac protocols for wireless LAN in Rayleigh and shadow fast fading*", Global Communications Conference, 1997 GLOBECOM'97* IEEE, Vol 1, 3-8 Nov 1997.

[9] Khattab, Hadidi and Mourad, "Analysis of Wireless CSMA/CA Network using Single Station superposition (SSS)", *International Journal of Electronics and Communications(AEU)* . Vol 56 No. 2, Mar/Apr 2002.

[10] Nuggehalli, P., Srinivasan,V., Chiasserini, C. F..,"Energy Efficient Caching Strategies in Ad Hoc Wireless Networks", *MobiHoc 2003,* pp 25-34, June1-3, 2003.

[11] Estrin, D., Srivastava, M., "Wireless Sensor Networks", *Mobicom 2002 Tutorial T5*, 2002.

[12] Perkins, C., Royer, E., "Ad hoc On demand distance vector routing", Proc. 2nd IEEE Workshop on Mobile Comp. Sys. & Apps., Feb 1999.

[13] Kulik, J., Heinzelman, W., Balakrishnan, H.,., "Negotiation based protocols for disseminating information in wireless sensor networks", ACM Wireless Networks, Vol. 8, Mar-May 2002, pp.169-185.

[14] Hill, J., Szewczyk, R., Woo, A., Hollar, S., Culler, D., and Pister, K., "System Architecture Directions for Network Sensors", *ASPLOS 2000*.

[15] Johnson, D. B.., Maltz, D. A., "Dynamic source routing in ad hoc wireless networks", Mobile Computing, Kluwer, 1996, pp. 153-181.

[16] Savvides, A., Srivastava, M.B.., "A Distributed Computation Platform for Wireless Embedded Sensing", Proceedings of ICCD 2002.

[17] Tamminen, J., "2.4 Ghz Wireless Radio Interface", Radionet, www.radionet.fi/res/277/TUT_WLAN_Seminar.pdf.

[18] Rodoplu, V., Meng, T., "Minimum energy mobile wireless networks," In Proceedings of the 1998 IEEE International Conference on Communications (ICC), June 1998.

[19] Ramanathan, R., Hain, R., "Topology Control of Multihop Wireless Networks Using Transmit Power Adjustment," pp. 404-413, INFOCOM 2000.

[20] Li, L., Halpern, J., "Minimum energy mobile wireless networks revisited," In Proc. IEEE International Conference on Communications (ICC), June 2001.