A Note on Scaling Wireless Sensor Networks

Gary Nutt
Department of Computer Science
Technical Report No. CU-CS-1001-05
December, 2005
University of Colorado
Gary.Nutt@colorado.edu

Abstract

Wireless sensor networks have rapidly emerged as an exciting new technology that challenges hardware, systems, and application software technologies. In half a dozen years, revolutionary changes have already occurred in all three of these areas with the emergence of mote computers, the TinyOS [7] operating system, and a spectrum of applications. The technology is based on self-organizing networks of computers with very limited resources. The leading edge work has focused on networks with relatively small numbers of nodes. This paper looks at the fundamental routing approaches used in this first wave of system with an eye toward how such networks will scale. Basic simulation studies suggest that it may be necessary to exploit machine hierarchies in sensor networks where nodes have limited radio range, yet where the networks span geographical areas larger than, say, a quarter of a square mile.

1 Introduction

In only a few years, wireless sensor network (WSN) research has made remarkable strides in developing node machines that have just enough hardware and software to read sensing devices, to perform simple data analyses, to use a low-power radio network to transmit the data to a conventional computer, and to route traffic through the sensor network on behalf of other WSN nodes that are unable to directly communicate with the conventional computer. Researchers have speculated on far-reaching applications for WSNs, including application domains where a sensor node is very small – smart dust [10] – and the WSN scaling to hundreds, thousands, or even millions of nodes. Sensor node hardware continues to evolve, e.g, see [8]; there is activity in specializing network protocols for WSNs, e.g, see [1][2], and there continues to be activity in system software to support WSN applications, e.g., see [3][4][5][7][9].

Culler, Estrin, and Srivastava introduce a special edition of the IEEE Computer that focuses on WSNs in which they provide a circa 2004 snapshot of the state of the art of the spectrum of technologies [3]. They identify the following challenges:

- WSN nodes are resource constrained, and that collections of nodes must work together to provide substantial processing in the aggregate rather than as individual nodes.

- To conserve battery power, "… most of the device's components, including the radio, will likely be turned off most of the time."
- Because there are potentially so many nodes, the nodes must be self-organizing and provide a means for group (rather than individual node) administration.

All WSNs face these challenges. However the way one approaches the challenges is influenced by how the WNS will be used. For example:

- Will the WSN incorporate very large numbers (e.g., thousands or even millions) of nodes, or will it be composed of more modest numbers (e.g. up to a few hundred nodes)?
- Will the WSN be used to actively monitor an environment, reporting the occurrence of events, or will a centralized server implement the strategy of the network, querying portions of the WSN when it needs information from that part of the network?
- Will the WSN designer have the freedom to carefully place the sensor nodes in the environment, or will the situation dictate that the nodes be scattered somewhat randomly over the geographic area being sensed?
- What is the expected lifetime of he WSN?
- When a WSN begins to fail, what is the expected pattern of failure – sudden or gradual?

The self-organizing challenge, in conjunction with limited resources and power at each node, suggests that WSNs employ dynamic routing technology. The individual nodes participate in a communal phase in which they exchange information among themselves in order to determine routes for moving data through the network. Culler, Estrin, and Srivastava generally describe how dissemination is used to determine routes:

> *A basic capability in such networks involves disseminating information over many nodes. This can be achieved by a flooding protocol in which a root broadcasts a packet with some identifying information. Receiving nodes retransmit the packet so that more distant nodes can receive it. … Each packet identifies the transmitter and its distance from the root. To form a distributed tree, nodes record the identity of a node closer to the root. The network can use this reverse communication tree for data collection by routing data back to the root or for data aggregation by processing data at each level of the tree.*
> (page 47 of [3])

Flooding protocols define a proof-of-existence that WSNs can organize themselves into an effective distributed system. Brute force flooding protocols propagate many messages across the network, causing nodes to expend battery power to determine the shortest route from itself to a data collection server. Thus the requirement to be a self-organizing network is clearly in tension with the other two challenges for resource constrained node operation, and for battery power conservation. Further, once the nodes have organized themselves, some fraction of the nodes will need to expend additional battery power to forward messages from nodes that are remote from the data collection server – this is an important aspect of WSN operation.

This paper describes a simulation study of a fundamental set of characteristics, primarily related to routing, that are likely to affect scaling in a WSN. We focus on several fundamental questions about network behavior under various application domain requirements:

- What are the general performance characteristics of routing in a WSN?
- How does the placement of sensor nodes influence the size, performance, and lifetime of the network?
- Once a single node in the WSN exhausts its battery, what is the pattern for failure of the entire network?
- How well will fundamental routing approaches scale?

Finally, we propose a slightly heretical approach to WSN organization – using a hierarchy of WSN nodes to better accommodate routing in larger networks. This destroys the beauty of homogeneity of network nodes, but provides a simple way to address the network scaling barriers.

In Section 2 the nature of WSN flooding-based routing is reviewed, including the fundamental limitations due to the approach. Section 3 focuses on differences between the distribution patterns used to place individual nodes. Section 4 considers the failure patterns for WSNs that depend on routing. Section 5 considers the practical aspects of building a WSN to cover a relatively large geographic area, when each of a set of homogeneous nodes has limited radio range. As a practical solution to scaling, in Section 6 we propose the use of hierarchies of machines within the sensor network.

2 The Problem

The desire for self-organizing networks is important, but not trivial to support. The basic technique for determining routes is to use a flooding protocol. As described above, the root node initiates a routing phase by broadcasting a routing command to all nodes in the network. Receiving nodes retransmit so that nodes that were out of range of the initial route command can "hear" the second. Retransmitting may need to be repeated, since each wave of retransmissions potentially expands the routed portion of the network so that previously unreachable nodes can now find a path to the root.

With a flooding protocol, each node can receive $O(n^2)$ routing messages. Flooding can exhibit surprisingly complex behavior, leading to the use of various heuristics to refine the approach [6]. Other improvements can be made to the routing protocols, leveraging the fact that they are used in a WSN rather than a more general ad hoc wireless network e.g., see [9]. Nevertheless, such improvements do not result in algorithms with linear complexity. Because of this complexity, flooding clearly will not work in a network with millions of nodes.

In a sensor network, the goal is for nodes to be small and inexpensive; to be able to operate as a low-power, wireless unit; and yet to be able to organize themselves with a set of routes by which each node can communicate with a single *root server*. Each node can collect information, filter it, and then transmit it (directly or indirectly) to the root server.

Logically the sensor network is a point-to-point network (see Figure 1). However because of the general resource constraint on node machines – particularly battery capacity – the range of a radio in a contemporary node is about 100 feet. A direct translation of the logical model to a physical model would require that all nodes be within

100 feet of the root server. There are many important applications that can be addressed within this constraint, but there are many more that require sensors to be distributed over a larger area. This provides the motivation for the WSN to able to organize itself to route information to/from nodes that are out of radio range from the root server.

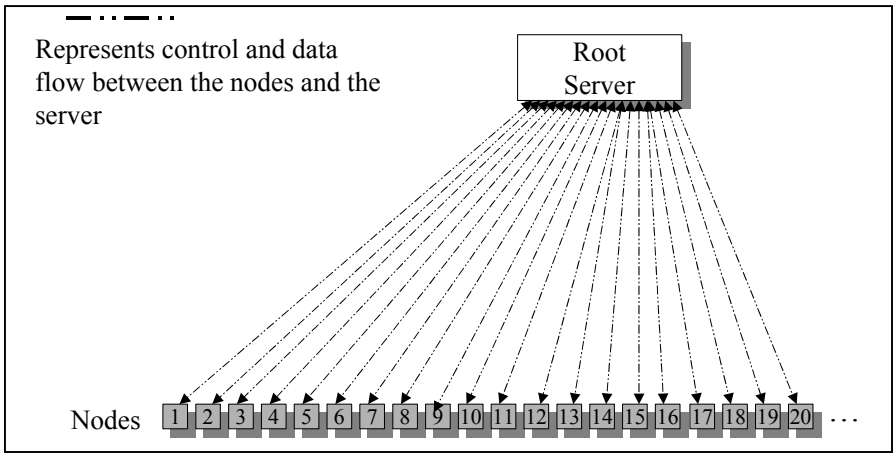

Figure 1: A Logical Sensor Network

Nodes can be physically distributed with a carefully planned placement strategy e.g., as strain gauge sensors on a bridge or airplane wing, or incident-light sensors on a redwood tree. Regular, uniform distribution of the sensor nodes throughout an area of interest is c common example of careful node placement. The nodes can also be *randomly distributed*, e.g, by scattering a collection of nodes into a hazardous area such as a forest fire or in the vicinity of a leaking nuclear reactor. In terms of routing, one would expect uniformly distributed nodes to be better behaved than randomly distributed nodes, since each node is assured of having at least one neighbor within a fixed distance. Of course there are no such assurances in a WSN with randomly distributed sensor nodes. The behavior of the routing configuration can be heavily influenced by the nature of the node distribution.

WSN strategies typically assume homogeneous nodes, i.e., all sensor nodes are the same. This simplifies the complexity of the self-organizing algorithm, and facilitates the use of random distribution techniques. By assuming homogeneous nodes, the WSN shown in Figure 1 might organize itself as shown in Figure 2. Notice that each node is responsible for sensing phenomena that occur in its immediate geographic area. In this example some nodes (numbers 2, 5, 7, 10, 12, 15, and 20) have an additional responsibility of routing information between other nodes and the root server.

The amount of routing traffic that a node supports depends on the distribution of the nodes and the algorithm used to organize the network (determine the routes). Independent of the distribution and organizing algorithm, the WSN routes will ultimately define a hierarchy on the nodes as suggested by the figure. Some of the nodes are leaves in the routing tree, while interior sensor nodes forward traffic to/from the root server, i.e., they are each the root of a subtree. Suppose that a WSN has n nodes, and each node is equally likely to sense an event occurrence. Then if a node is the root of a subtree with k nodes in it, that node will participate in the reporting of ~k/n of the event occurrences in

the monitored area. Since radio operations are relatively expensive compared to pure computation, a node can be expected to consume power at a rate proportional to k. That is, roots of subtrees that are close to the root server participate in relatively large fractions of the sensing activity, and therefore consume power at a proportionally higher rate than do leaf nodes.

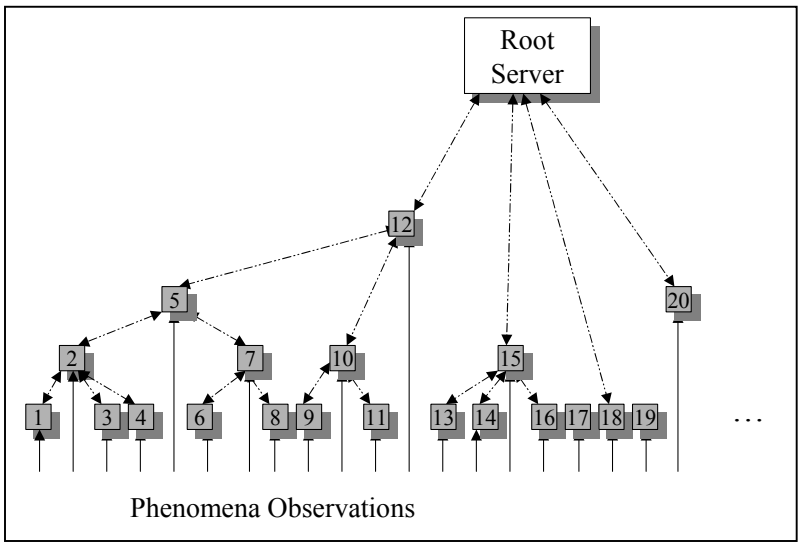

Figure 2: Sensor Network Routing

This is the fundamental premise that makes directed diffusion attractive [3][9]. In directed diffusion, subtree root nodes cache information to preserve power in nodes in the subtree. That is, subtree roots essentially tradeoff their power in exchange for power in many other nodes. Directed diffusion algorithm designers dynamically determine the subtree level at which information will be cached, and determine when cached data is stale.

Directed diffusion is applicable in a *pull networks*, or ones in which sensors react to data requests from the root server. In a *push network*, the sensors operate asynchronously (with respect to the root server), transmitting data according to the local node algorithm (e.g., when a phenomenon occurs in the observable range of the sensor node's devices). In both push and pull WSNs, the interior nodes that either respond to the most server requests, or forward information from the most active sectors of the WSN coverage area, will be required to perform the most computation, and more importantly to forward the largest number of messages. That is, they are the most critical to the overall data flow in the WSN, and will consume power at the highest rate of all sensor nodes.

How bad can the problem be? Consider a small push network: There are at least two considerations in determining how to place the nodes. First, the collection of sensor nodes must be close enough together so that they can collectively perceive all relevant phenomena, and second, the nodes most also be sufficiently close so that each is in radio contact with at least one sibling node. The first property is specific to the application since it relates to the nature of the phenomena and the sensing mechanism. Regarding radio range, at the time of this writing the commercial version of the Berkeley Mote (the Crossbow Mica2) has a radio range of about 100' (see www.xbow.com for current specifications on these commercial products).

Let's suppose we wish to distribute 64 sensor nodes uniformly over an 800'×800' area in a push network. Each node is expected to monitor an area that is a square that is 100' on each side. (A circle with 100' radius overlaps the centers of adjacent square areas.) With ideal sensor net placement, each node would be within 100' of at least one neighbor. For the purposes of this discussion, suppose that each sensor node has a relatively low capacity battery (100 mAh,) and a radio range is 100', and information will be forwarded a maximum of 8 hops. Further, suppose that the root server is at the center of the area. Finally, assume that phenomena occur at random points in the subject area with random magnitude. In a self-organizing WSN that uses the flooding protocol described in [3], a message from a sensor node to the server will experience about 3.94 hops and cover about 364.3'. After 21,667 phenomena occurrences, we estimate that there will have been about 39,115 messages transmitted from sensor nodes to the server (this depends on the magnitude of the individual phenomena and the nature of the sensor devices), and about 106,530 forwarding operations including the flooding protocol forwarding transmissions, i.e., about 73% of the node transmit operations are to forward messages.

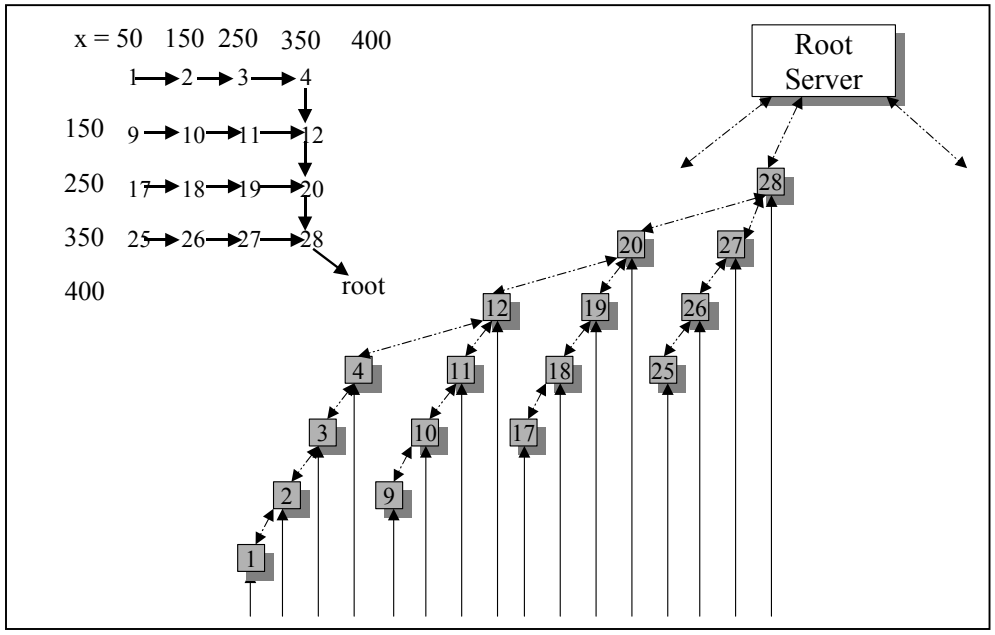

Figure 3: Example Routes

Figure 3 shows two views of the routes in the example WSN: the left side of the figure represent the geographical relationships and routes for nodes 1-4, 9-12, 17-20, 25-28, and the root. The root is at the center of the space at (400, 400), node 1 is at (50, 50, node 2 is at (150, 50), etc. The right side of the figure represents the same information, but it is intended to illustrate the routing relationships in the same form as Figure 2. Node 28 at (350,350) is the root of a subtree containing nodes 1-4, 9-12, 17-20, and 25-28. Both views show that node 28 forwards messages from all nodes in its subtree to/from the root server. Assuming that a node uses 8 mAseconds to transmit a message, and 16 mAseconds to receive one, plus adding a relatively small amount of time for local

processing at each node. Our analysis predicts that (with a 100 mAh battery) node 28 will be the first of the 64 nodes to fail due to battery exhaustion after about 21,667 phenomena occurrences. As one would expect, nodes 29, 36, and 37 also have very low remaining capacity at the time that node 28 failed.

Notice that once a critical node such as 28 fails, then the network must reorganize, since the failure left 15 nodes with no path to the server. During reorganization every node in the network incurs the cost of self-(re)organization cost. After reorganization the routes will generally be longer (i.e., node-server communication will be more costly in power) than they were before the first node failed. In the example, the average number of hops increased from 3.93 to 4.17, with a corresponding increase in the average distance of the route from 364' to 387'. As in the initial configuration, a small subset of the nodes will be close to the root, and will participate in the majority of the network traffic, thereby causing another wave of failures.

3 Planned versus Random Distribution

The example in Figure 3 suggests the importance of planned, regular node placement. It is easy to understand the routing pattern, when the structure of the node placement is a uniform grid. Suppose the application domain created barriers that prevented planned, uniform placement of the sensor nodes, i.e., the nodes could only be placed by scattering. Intuition suggests that the routing properties will probably be less desirable than for the uniform case, but how much less desirable? For the purposes of comparing placement patterns, let's make a few assumptions about the network: Each sensor node radio has a transmit/receive range of 100 feet. In order to assure that every node can communicate with at least one other node, the nodes can each be assigned a square area within the WSN coverage area. The node is at the center of its coverage area

Consider a WSN in which nodes are uniformly distributed, and each node is positioned at 100' from its 4 nearest neighbors. By assuming a 100' radio range, we know that a WSN will not behave well as the size of a node's coverage area exceeds 10,000 square feet, since no node would be able to contact its nearest neighbor (except for the few nodes that are near the root server). Conversely, if the coverage area is less than or equal to 10,000 square feet, the WSN can organize itself so that there is a path from every node to the root server, even though the placement is conservative.

Next consider the routes that would be established using the flooding protocol for various coverage area sizes and for nodes that are randomly scattered across a fixed size area (equal to N times the square feet per node). For example, if the average square feet per node is 2,500 square feet, then the total area in which N sensors are distributed is 2,500×N. Figure 4 shows the results for various 64 node distributions. Once the average amount of area exceeds about 5,000 square feet, the algorithm cannot route every node. For the maximum coverage area for uniform distributions (100' × 100'), random distributions were able to route less than 15% of the nodes.

Consider the average number of hops and the average distance of a route from each sensor node to the root server, while varying the coverage area. Since random distributions were generally unable to route more than a half to two thirds of the nodes for larger coverage areas, we did not consider the average routes for these cases above 80' × 80'. Figure 5 plots the average number of hops in a route versus the size of the coverage area for a node. In uniform distributions, the maximum number of hops for

N=64 is 7 (see Figure 3), but the average number of hops stabilizes at about 4 hops until the coverage area is too large for the radio range. Routes for scattered nodes are slightly better than uniform for small coverage areas, but deteriorate rapidly once the coverage area becomes too large.

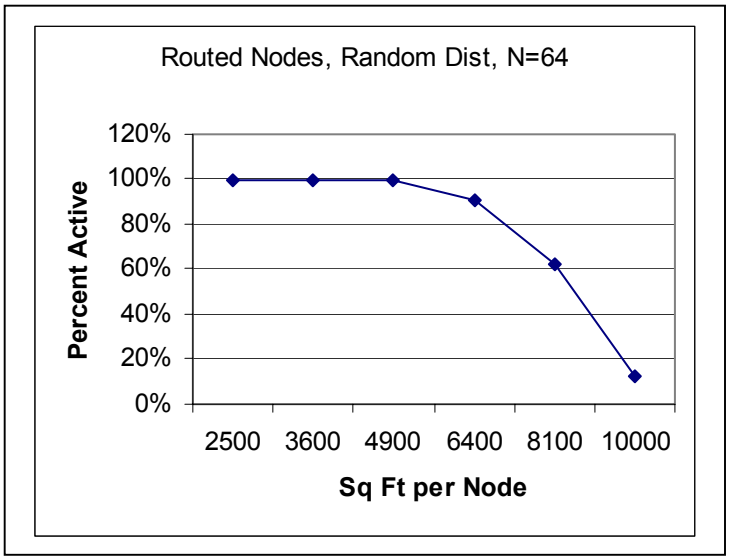

Figure 4: Fraction of Routed Nodes with Random Distribution, 64 Nodes

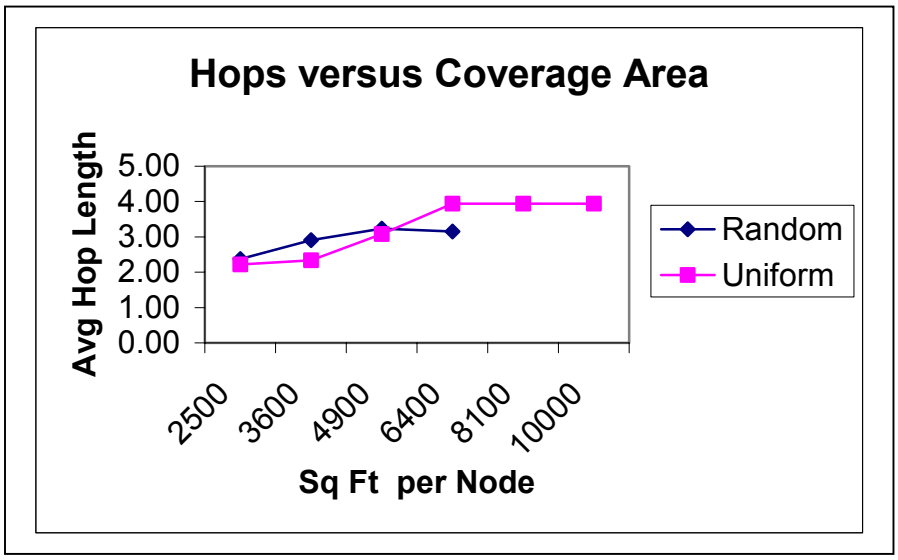

Figure 5: Effect of Distribution on Routing Hops

The average routing path length (see Figure 6) reflects the average number of hops in the network for uniform and scattered distributions, although for uniform distribution patterns the average length of a hop continues to increase as the coverage area reaches the maximum.

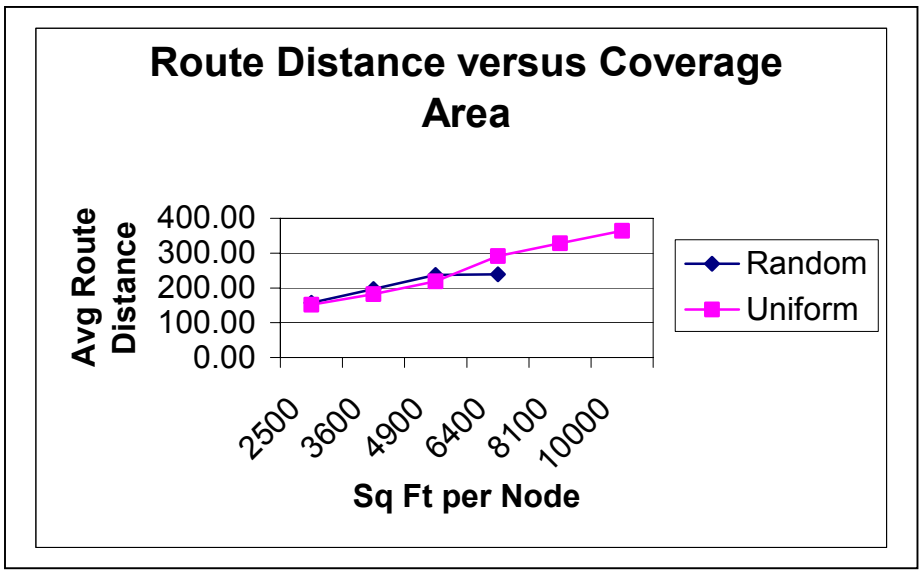

Figure 6: Effect of Distribution on Routing Distance

Based on these studies, it is easy to discern the significance of node placement in the WSN area. Regular, uniform placement ensures that the WSN can detect phenomena occurrences over the largest geographical area. When the nodes are randomly scattered across an area, the WSN could completely miss phenomena occurrences (depending on the nature of the phenomena, the range of observability, etc.). In terms of self-organization, random scattering causes some subset of the nodes to be isolated if the average area. Our model predicts that a significant fraction of the nodes will be isolated as soon as they are placed, when the average coverage area for a node exceeds about two thirds of the maximum area for complete coverage with regular placement. Further, as the average coverage area increases, the number of isolated nodes increased more at a faster than linear rate.

Provided that the WSN is able to organize itself, randomly scattered nodes do not seem to cause abnormally excessive hops nor routing distances. Random scattering of the nodes is acceptable, provided that the average coverage area of a node is not excessive.

4 Net Failure Profile

WSN nodes depend on limited battery power, implying that repeated self-organization or excessive responsibility for extra computation, sensing, or routing can all contribute to the nodes failure due to battery exhaustion. In this section we explore the overall effect on a WSN when a subset of the nodes fail due to the extra work they incur.

In a WSN with routing, the most likely node to fail is the one that is the most critical to any particular routing strategy, i.e., the larger the routing subtree is for a node, then the more traffic it will incur, and the faster its battery will be exhausted. Once such a strategic node fails, then the WSN will self-(re)organize, electing other strategic nodes – a new candidate will replace the failed node by reconfiguration in that part of the WSN, but in remote parts of the net, the heavily-loaded nodes will continue to be heavily-loaded, and will soon fail. How will this pattern of failures effect the overall operation of

the WSN? In this part of the study we examined a profile for how the WSN is likely to collapse due to repeated failures of over-worked nodes.

We considered a push-style WSN in which nodes are carefully placed with a regular, uniform policy. The average coverage area for each node is 4,900 square feet (the maximum area for a radio with 100' range is about 100,000 square feet). If the WSN has 49 nodes (and covers a total area that is 490' × 490'), we predict that the first node will fail of battery exhaustion after about 13,300 phenomena occurrences (randomly occurring over the entire area). The second node will fail after about 15,000 occurrences, the third after about 16,600 occurrences, then 3 additional nodes will fail by about 18,300 occurrences, and the remaining nodes will become inaccessible (because all strategically placed nodes have used up their batteries) by the time 20,000 occurrences have taken place. Of course there are a number of assumptions in this analysis, but it is clear that the organization of the routes will cause this type of failure, even though the number of phenomena occurrences (or other measures of activity the stimulate the sensors) will cause this kind of behavior.

Figure 7 shows our results for analysis of several different numbers of nodes where each node covers a 70' × 70' area (so the increased number of nodes covers correspondingly larger WSN areas). First, notice that WSN configurations with an even number of nodes collapse much earlier than those with an odd number of nodes. We will discuss this behavior before considering the larger trend shown in the figure.

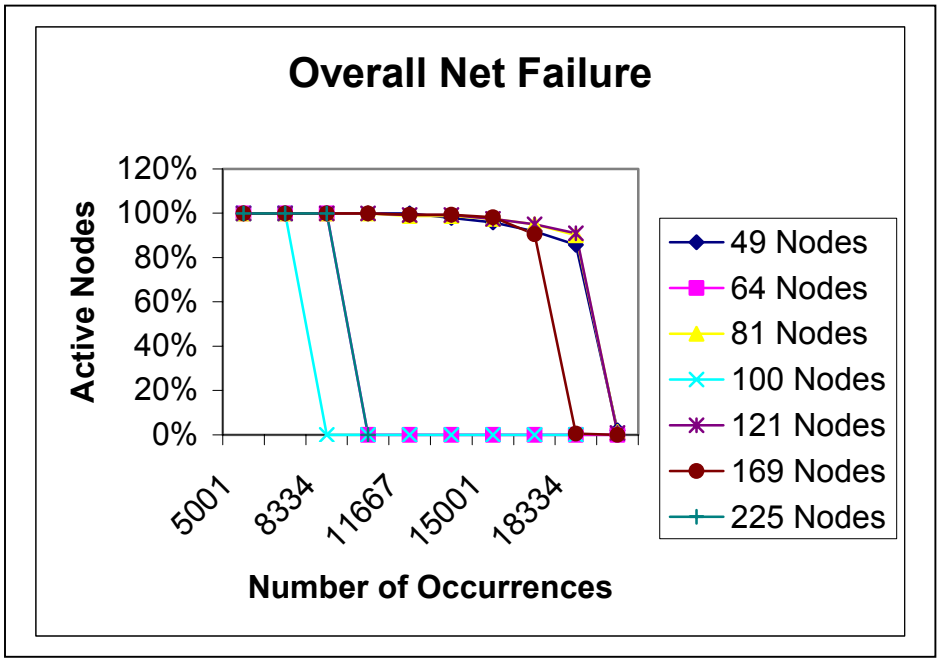

Figure 7: Uniformly Distributed Nodes over Increasing Coverage Area

In the case of an even number of uniformly distributed nodes a single node will end up forwarding a relatively larger amount of traffic than in the odd number case. To see why this is true, consider the case where there are 64 nodes uniformly distributed over a 560' × 560' area (see Figure 8(b)). In the model, the root is placed in the center of the

observation area, e.g., at x=280, y=280 for the case with 64 nodes. A node that is close to the center of the area, such as node 28, will end up with the data from an entire quadrant moving through it.

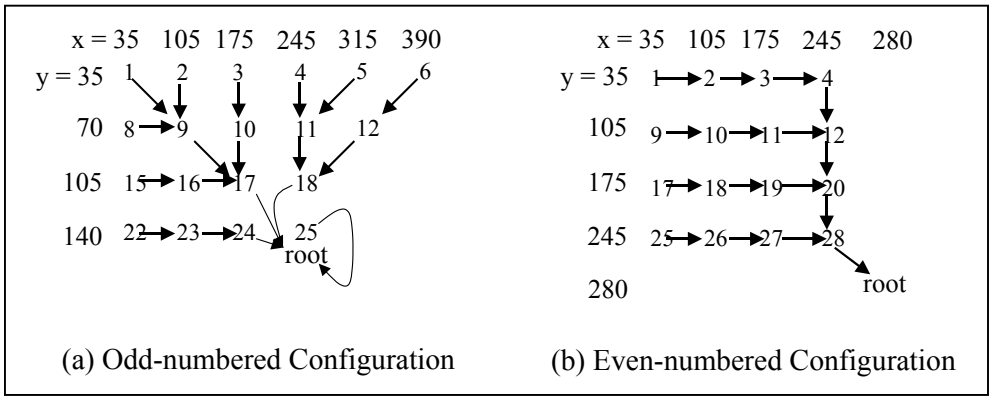

Figure 8: Odd-Even Routing

Depending on the exact implementation of the flooding protocol, the odd case will distribute the routes more equitably, e.g., see the 49-node routing pattern shown in Figure 8(a). In this case, the node 25 and the root server are both located at the center of the observation area (if the flooding protocol is working correctly, node 25 will never forward traffic to the root). Roughly speaking, nodes 17, 18, and 24 share the routing load for the NW quadrant. Now reconsider the data from Figure 7; the consequence of such sharing is significant in the cases where there are 64 or 100 nodes in the WSN, with collapse occurring with a relatively small number of phenomena occurrences because the nodes near (within radio range of) the root server exhaust their batteries and the net collapses.

In WSNs with an odd number of nodes, ultimately all those nodes that are within radio range of the root server exhaust their batteries before any of the out-of-range nodes exhaust theirs, although it will take longer than in a similar even number case. Notice that the models illustrate that as the number of nodes increase, the odd number configurations begin to face earlier and earlier. This is due to the fact that even though a node close to the root server shares the load with others, its total load depends on the number of nodes in its routing subtree, which grows as the net grows. Holding the average coverage area constant, but growing the overall WSN coverage area will require increasing numbers of nodes, thereby placing greater loads on those node in close proximity to the root server, and will lead to collapse after fewer and fewer phenomena occurrences.

5 Scaling the Coverage Area

Researchers hypothesize WSNs with thousands, even millions, of sensor nodes, communicating with a tethered server (that can communicate with other servers across the Internet), eg. See [10]. This class of researchers is generally focused on exploiting technology gains such as Moore's law for computers, and corresponding trends in radio technology, to hold node functionality constant while reducing size and power use. In this domain, contemporary node computing and communication characteristics can be

used to think about future nodes. For example, at some point in the future, there could be a viable, commercial sensor node that has the same characteristics as a specialized sensing platform [8], but be packaged (with battery) in a form factor that is the size of a pin head.

Consider a coverage area that is a quarter of a mile on each side, i.e., an area that is 1320' × 1320' = 1,742,400 square feet. If each sensor had a radio range of 100' and we decided to distribute the sensors uniformly over the area – the best possible case with contemporary sensor node radios – then we would need at least 175 nodes. If we used a grid of 14 × 14 evenly spaced nodes, each covering a 94.3' × 94.3' area, we would need 196 nodes. Presuming that the server has been placed in the center of the coverage area, the longest route would be 13 hops (because of the odd-even phenomenon of the routing algorithm, the longest route in a 15 × 15 grid would be 7 hops).

Suppose that one were to randomly scatter the sensors rather than carefully place them. In this case, the average distance between nodes must be reduced from the full radio range in order to connect nodes that fall in sparsely populated areas. Our empirical study indicated that for most nodes to have a route to the server, it would be necessary to reduce the average distance between nodes to say, 77.6'. In this configuration, the WSN would use 289 sensor nodes on a 17 × 17 grid to monitor behavior in the target coverage area. The simulation model predicted that, in the initial configuration, that average number of hops in this configuration is 11.6, covering an average distance of 868.5'.

Next consider a slightly larger WSN to monitor activity in a square mile (27,878,400 square feet). 2,788 sensors would be required to cover the area if they were uniformly distributed at 100' between nodes. If we used a grid of 53 × 53 evenly spaced nodes, each covering a 99.6' × 99.6' area, we would need 2,809 nodes. Presuming that the server has been placed in the center of the coverage area, the longest route would be 26 hops. If the nodes were randomly scattered, then using the above assumptions, the WSN would have about 4,356 nodes.[1]

Despite the fact that computationally acceptable routing algorithms still seem to be infeasible, it is possible to compute routes using techniques other than pure self-organization approaches, e.g., the server might have a very powerful radio that it uses to transmit information to all nodes in the network during routing determination. For example, the server could simultaneously tell each node to determine a new route; or perhaps it could determine explicit routes for each node in the network, and then transmit it to the nodes. Ignoring the complexity of computing the routes, the sensor nodes will still need to be able to transmit information toward the server, using the limited power within each node.

There is one other assumption in this example that might not be applicable in various applications: the server is assumed to be located in the center of the observation area. If the observation area were hazardous, this may not be possible, although perhaps it could be suspended (by using the z axis in 3-space) above the observation area. Of course, it would still have to be within 100' of the surface that contained the sensor nodes. This could be avoided by placing the root sensor on the same x-y plane as the sensor, but at

[1] We did not bother to simulate the flooding protocol for this configuration. In our tests with fewer than 100 nodes, the simulator would run in perhaps 10 seconds, but the case with 289 scattered nodes took over three hours to compute the initial routes.

the boundary of the observation area. This will influence the lengths of routes in the network, exacerbating the battery exhaustion problem.

Suppose instead that one attempted to scale the WSN by adding multiple servers to the periphery, thereby providing alternative destinations for routes. Figure 9 is a simple case, with three tethered ("root") servers, X, Y, and Z being placed on the periphery of the observation area. The WSN self-organizes by having the servers communicate via an internet to determine a time at which to organize the routes in the WSN; at the prescribed time, they all initiate the flooding protocol. Presuming that the flooding protocol is written carefully enough so that it stabilizes, we would expect to see cases where three adjacent nodes would route information to the three different servers, e.g., node i would route information to server X, node j would route information to server Y, and node k would route information to server Z. Logically, each server services its own logical WSN, and the union of the area covered by the WSNs is the desired coverage area. The routing behavior within each subnet is the same as for the cases discussed above – a subnet cannot easily scale beyond a certain point because of the characteristics of routing. The obvious solution is to insert server around the periphery between servers existing servers. In this case, the radius of the circle (or sphere in 3-space) establishes a practical limit on scalability. This is because once the radius of the coverage area exceeds the acceptable hop range of nodes in the WSN (larger hops cause forwarding nodes to fail more rapidly), interior parts of the coverage area will not be able to reach any server.

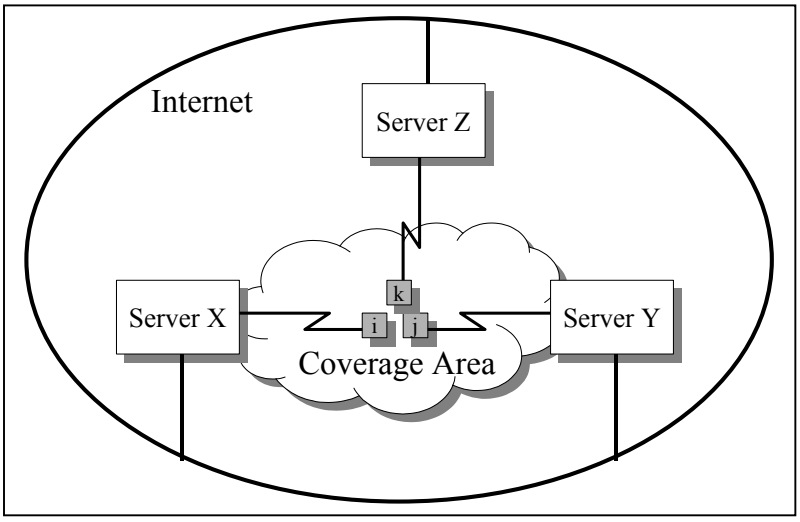

Figure 9: Multiple Server Configurations

Suppose that the WSN application domain does not preclude the placement of servers within the observation area. Then we could scale the WSN as suggested by Figure 10. In this case, we are essentially assuming that a wired internet subnet can be superimposed on the WSN coverage area. This may be a suitable solution in some application domains, but it is not likely to be applicable in cases where, for example, the coverage area is hazardous.

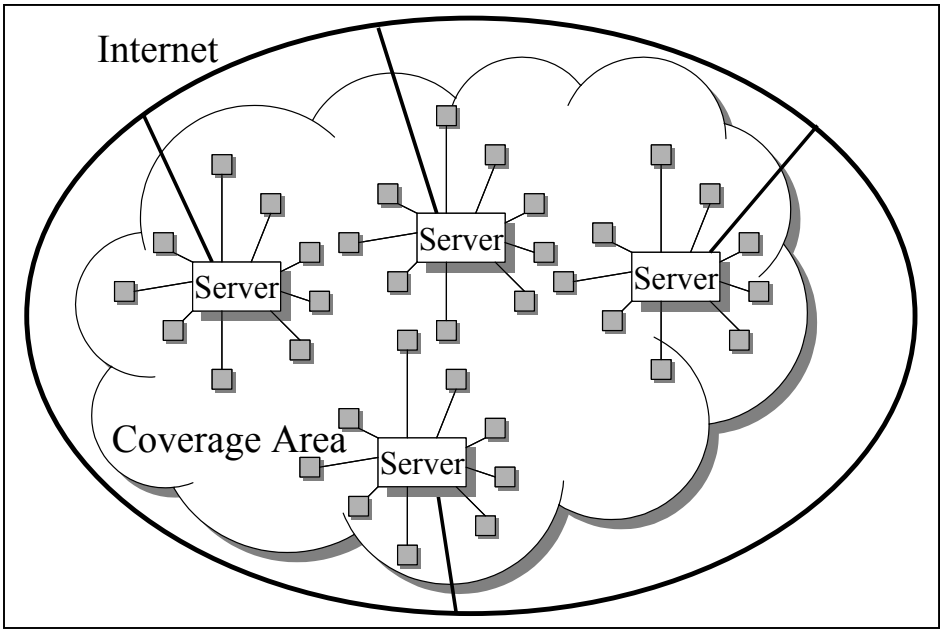

Figure 10: A Scalable WSN Configuration

6 Hierarchical WSN Node Architectures

WSNs are typically defined with heterogeneous sensor nodes, thereby simplifying self-organization – and establishing a criterion for general distributed solutions. However, these requirements impose a significant barrier on WSN scalability; by relaxing the homogeneity requirement, it is possible to provide solutions for large application domains that would not otherwise be possible.

For example, suppose that sensor nodes were all configured with the same processor, radio, devices, etc., except some nodes had larger battery capacity than other nodes. That is, the difference in cost of the two types of nodes is the difference in cost of the batteries. The strategy would then be to ensure that the nodes with a high-capacity battery be placed so that they would be near the root in routing paths, even if the nodes were scattered instead of being carefully placed. However, such as solution would not be very general, since the radio range on the high battery capacity machines would be the same as on generic nodes.

Suppose that selected nodes had higher battery capacity and increased radio range – i.e., part of the extra power would be used for extra routing, and another part would be used to increase the radio range. This would allow one to superimpose a larger grained grid of high capacity battery nodes over the grid of generic nodes.

It would not always possible to place the nodes with high capacity batteries so that they are in the "right place" in the observation area. However, there are likely to be heuristics in the scattering technique that can increase the chance of success. For example, suppose that the nodes with high capacity batteries are first randomly scattered across the space, then in a second pass the normal nodes are scattered over the space

using the same technique as in the first pass. With large enough numbers of nodes, and a well-behaved scattering mechanism, the high battery capacity nodes will be distributed proportionally to the normal nodes.

We have also modeled hierarchical organizations to compare their predicted behavior with the conventional WSN architectures. For example, suppose that a WSN was configured as a 3-layered hierarchy with 64 generic nodes, 4 nodes at the second level of the hierarchy, and a single root server at the third level of the hierarchy – this is the same as the 64-node case discussed earlier in the paper, with the addition of 4 intermediate nodes, uniformly distributed over the 560' × 560' WSN coverage area. The generic nodes were configured with 100 mAhr batteries and a radio range of 100'. The intermediate machines had a battery capacity of 1,000 mAhrs and a 300' radio range. In the conventional configuration, the initial average number of hops in a route is 3.23, covering an average distance of 238'; the first node fails after ~9,000 event occurrences. The 3-level hierarchy average number of hops is 3.01, covering a distance of 328'; the first node fails after ~106,667 event occurrences – the failed node was a generic node rather than an intermediate level node.

We analyzed the average number of hops and the average route distance for various area/sensor values, similar to those shown in Figure 4 and Figure 5. The average hop length for various area/sensor values is shown in Figure 11. The average converged on 3.94 hops, since the intermediate level nodes are uniformly distributed over the sensor net area.

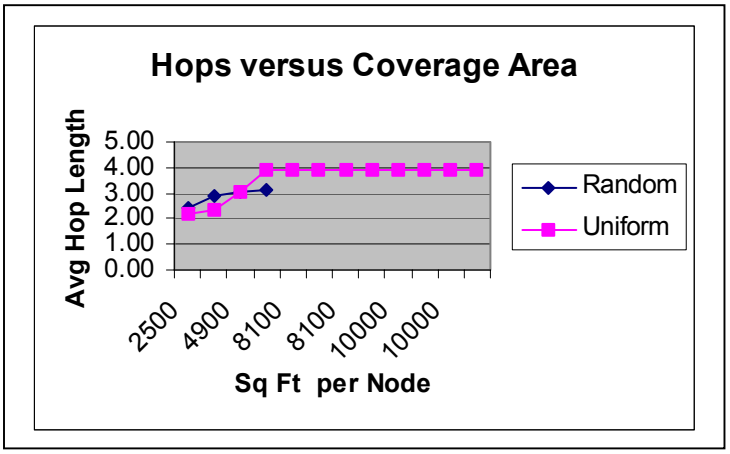

Figure 11: 3-Level Hierarchy, 64-node Average Hop Length

The average route distance, shown in Figure 12, reflects similar behavior. Of course both of these measures would change radically if the radio range of the intermediate nodes were different (thereby changing the number and distribution of intermediate layer nodes).

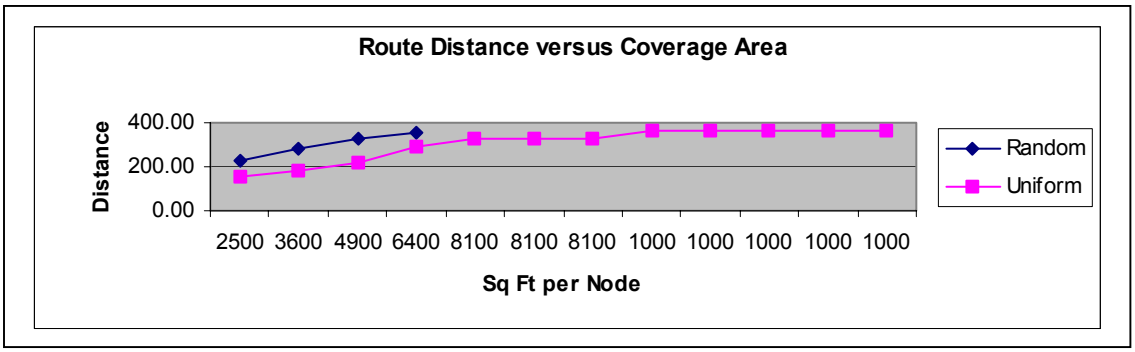

Figure 12: 3-Level Hierarchy, 64-node Average Route Distance

In essence the 3-layer network is similar to the solution suggested by Figure 10, i.e., a "backbone network} is superimposed on top of the first level sensor network. It differs from the figure in that the superimposed network is also a wireless network that uses technology similar to that used in the sensor nodes – with increased radio range and battery capacity. Of course, the intermediate level – or levels – could also be designed to leverage directed diffusion ideas, so that they were also especially well prepared to act as agents for the sensor nodes in their routing subtrees. In short, once a network hierarchy and specialized nodes are introduced, there are a substantial number of new possibilities for WSN approaches.

However the introduction of another level of hierarchy in the network substantially alters the character of the network, and also increases the difficulty in distributing the nodes. In terms of the complexity of behavior of the network, it also aggravates the complexity of the network behavior as described in [6]. However, the apparent longevity of the WSN is significant – in the 64-node case, the WSN will be active ~12 times longer than the conventional WSN organization at the cost of 4 additional nodes. The specific attributes for the intermediate nodes – the battery capacity and radio range – are not so important as the radical difference in network longevity.

The purpose of this paper has been to suggest the benefit of hierarchical WSNs, rather than to provide a closed analysis of their characteristics. The analysis highlights critical aspects of WSN routing as it relates to scaling. The preliminary analysis of even 3-layer networks suggests the cost-effectiveness of the approach.

7 Conclusion

WSNs have been defined as 2-level hierarchies, with the collection of homogeneous sensor nodes constituting the first layer, and a single root server being the second level of the hierarchy. The analysis of such networks indicates that, because of the limited radio range, the cost of self-organization, and ongoing routing costs, it will be difficult to grow such networks beyond modest coverage areas.

Flooding algorithms can cause difficulty in networks with thousands of sensor nodes, since the base line algorithm is $O(n^2)$. Even improved versions do not seem to have reached linear complexity (which may still be excessive in a sensor node with a million smart dust computers).

Perhaps worst of all, even after the routes have been established, the nodes closest to the root server will incur excessive forwarding traffic, meaning that they have extra radio receive and transmit operations. They will be the first to exhaust their batteries. Eventually, all nodes within range of the root server will exhaust their batteries, at a rate proportional to the number of nodes for which they forward messages. At this point the WSN will become inoperative.

References

[1] Akylidiz, Ianb F., Weilian Su, Yogesh Sankarasubramaniam, and Erdal Cayirci, "A Survey on Sensor Networks," *IEEE Communications Magazine*, August, 2002, pp 102-114.

[2] Culler, David, Prabal Dutta, Cheng Tien Ee, Rodrigo Fonseca, Jonathan Hui, Philip Levis, Joseph Polastre, Scott Shenker, Ion Stoica, Gilman Tolle, and Jerry Zhao, "Towards a Sensor Network Architecture: Lowering the Waistline," *HotOS X, Tenth Workshop on Hot Topics in Operating Systems*, Santa Fe, NM, 2005.

[3] Culler, David, Deborah Estrin, and Mani Srivastava, "Overview of Sensor Networks, *IEEE Computer*, August 2004, pp 41-49.

[4] Desnoyers, Peter, Deepak Ganesan, Huan Li, Ming Li, Prashant Shenoy, *HotOS X, Tenth Workshop on Hot Topics in Operating Systems*, Santa Fe, June 2005.

[5] Estrin, Deborah, Ramesh Govindan, John Heidemann, and Satish Kumar, "Next Century Challenges: Scalable Coordination in Sensor Networks," *Proceedings of the ACM/IEEE Mobicom*, Seattle, WA, 1999, pp 263-270.

[6] Ganesan, Deepak, Deobrah Estrin, Alec Woo, David Culler, Bhaskar Krishnamachari, and Stephen Wicker, "Complex Behavior at Scale: An Experimental Study of Low-power Wireless Sensor Networks," UCLA Computer Science technical report No. UCLA/CSD-TR 02-0013, 2002.

[7] Hill, Jason, Robert Szewczyk, Alec Woo, Seth Hollar, Davide Culler, and Kristofer Pister., "System Architecture Directions for Network Sensors," ASPLOS-IX Proceedings, ACM, Cambridge, MA, 2000, pp. 93-104.

[8] Hill, Jason, Mike Horton, Ralph Kling, and Lakshman Krishnamurthy, "The Platforms Enabling Wireless Sensor Networks," *Communications of the ACM*, 47, 6 (June 2004), pp. 41-46.

[9] Intanagonwiwat, R. Govindan, and D. Estrin, "Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks," *Proceedings of the ACM/IEEE Mobicom*, Boston, MA, Autust, 2000.

[10] Kahn J. M., R. H. Katz, and K. S. J. Pister, "Next Century Challenges: Mobile Networking for 'Smart Dust'," *Proceedings of the ACM/IEEE Mobicom*, Seattle, WA, 1999, pp 263-270.