JAM: A Jammed-Area Mapping Service for Sensor Networks

Anthony D. Wood, John A. Stankovic, and Sang H. Son
Department of Computer Science
University of Virginia
{*wood,stankovic,son*}*@cs.virginia.edu*

Abstract

Preventing denial-of-service attacks in wireless sensor networks is difficult primarily because of the limited resources available to network nodes and the ease with which attacks are perpetrated. Rather than jeopardize design requirements which call for simple, inexpensive, mass-producible devices, we propose a coping strategy that detects and maps jammed regions. We describe a mapping protocol for nodes that surround a jammer which allows network applications to reason about the region as an entity, rather than as a collection of broken links and congested nodes. This solution is enabled by a set of design principles: loose group semantics, eager eavesdropping, supremacy of local information, robustness to packet loss and failure, and early use of results. Performance results show that regions can be mapped in 1 – 5 seconds, fast enough for real-time response. With a moderately connected network, the protocol is robust to failure rates as high as 25 percent.

1. Introduction

Wireless sensor networks (WSNs) are a continuation of the evolution of networks toward larger-scale, distributed computing. In contrast to mobile ad hoc networks (MANETs), which comprise the growing number of commercial handheld, cellular-aided, and laptop computing platforms, WSNs are made up of mostly small sensors with limited-resources and capabilities.

They are more likely to focus on a particular application rather than supporting general-purpose computation and communication. Environmental monitoring, battlefield intelligence, emergency response support, and real-time data fusion and collection are among their frequently cited applications [2, 12, 14, 8].

While WSNs benefit from a more cohesive design, purpose, and management than MANETs, they suffer from relative resource impoverishment. The large number of devices needed to provide fine-grained sensor coverage of areas measured in square kilometers dictates that each device be as simple and inexpensive as possible while still providing useful functionality.

Each sensor has an omnidirectional radio, small battery, one or more sensors, and may be environment and tamper-proofed. Individual sensors are not reliable due to mass manufacturing defects, harsh natural or urban deployments, and battery death. Aggregate behavior and robust algorithms provide the reliability that safety-critical applications demand.

1.1. Role of geographic location

Mobility in these networks may be defined in terms of the agents moving through them, rather than by the autonomous post-deployment movement of individual sensors. A number of localization services exist which can provide each sensor with an accurate estimate of its own location and provide directory services for others' locations [18, 4, 22, 10] without the expense of adding GPS capability to each node.

As WSNs are embedded in the environment, this geographical information is much more relevant and useful than for traditional wired networks. Here the topology is defined by physical proximity, even though real-world propagation and radio hardware produce one-way links [9], "gray-zones" [19], and other challenges to protocol design and simulation.

Many of the events and problems that occur in deployed WSNs exhibit strong spatiotemporal properties. Tracking intruders, vehicles, or animals explicitly provides information about where an entity is across time. Explosions leave behind a void in the network that is limited in area. Barriers too have physical properties and defined boundaries, whether walls, roads, or rivers. Even fires, which may eventually spread into large regions and leave behind pockets of conflagration, have boundaries that are dictated by physical processes.

1.2. A denial-of-service attack

In a network that is mostly homogeneous, that is, where there is little capability or functional differentiation except what is cooperative, distributed, or re-

dundant, deliberate attacks may also be strongly localized. One such attack, most likely to occur in a battlefield or urban warfare environment, is radio *jamming*.

The extent of the jamming is dictated by physical properties such as the available power, antenna design, obstacles and height above ground. Jamming is no different than normal radio propagation, except that it is unwanted and disruptive, creating a denial-of-service condition. It experiences the same probabilistic and transient propagation behavior noted in [9] and [19].

Sensor nodes inside a jammed region cannot effectively accomplish any aspect of their mission which depends on communication. The network at large may waste energy and cause further contention by trying to query or use the affected regions for routing messages. Unless accidental, as from a malfunctioning sensor node, whatever is causing the jamming may pose a hazard to sensor-network-supported human agents. Network-directed vehicles entering the region will be unable to communicate and may become stranded.

Defeating or avoiding jamming is a complicated game of one-upmanship, with the complexity and cost escalating with each counter-measure, counter-counter-measure, etc. Spread spectrum techniques such as frequency hopping and code-spreading [3, 24] are common defenses against both intentional jamming and high-noise environments. Other measures include antennas with steerable or adaptive nulls and multiple-element antenna arrays.

The costs and complexity of these solutions are prohibitive for WSNs, in which individual nodes must be cheap. Yet, jamming is particularly easy since many will use single-frequency communication.

If jamming is only a problem in military networks, perhaps the expense of prevention can be justified. While standoff and localized jamming *are* a concern in a battlefield context, jamming may also occur in commercial and industrial networks. With the promulgation of mobile handheld devices that also operate in the unlicensed Industrial, Scientific, and Medical (ISM) band, it may even be accidental. Neither must the attack be jamming in the traditional sense; other low-energy mechanisms cause similar denials-of-service [26].

Given the frequency of denial-of-service activity on the Internet, we expect to see more of this kind of attack as WSNs become more commonly and more accessibly deployed.

Though it may not be feasible to cheaply *prevent* jamming, if the network can know the location and shape of the affected area, it can mitigate its impact. Assuming that the entire network is not affected (in which case there is no hope), it can take action to avoid the area for routing and higher-layer route planning, tune energy management protocols, and report the problem to uplink control systems.

1.3. Detection: A mapping approach

We propose a mapping service for WSNs that can provide the following benefits:

- Feedback to routing and directory services
- An effective abstraction at a higher-level than local congestion, failed neighbors, and broken routes,
- Support for avoiding the region by network-controlled vehicles, military assets, emergency personnel,
- Reports to a base-station for further jamming localization, and
- Aid to power management strategies for nodes inside and around jammed regions.

The jamming detection and mapping protocol use mostly existing data and facilities in the typical sensor communication stack, making detection and mitigation a cheaper strategy than prevention.

Generally, nodes near the border of a jammed region notify neighbors outside the region of jamming. These nodes form groups and use a lightweight, low-state management mechanism to coalesce groups and map the extent of the jammed region. Bridge members aid neighborhoods of low connectivity. An eager eavesdropping strategy provides forward and backward information diffusion among mapping members.

Contributions of this work include:

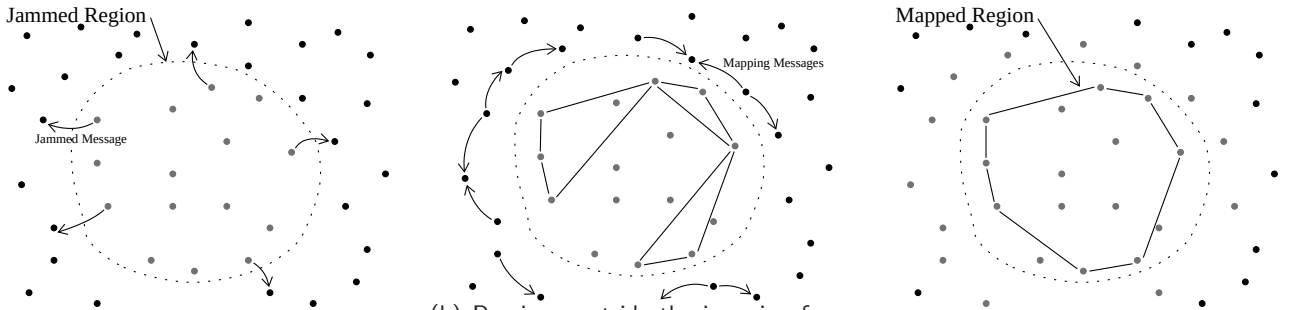
- Loose group semantics integrated with flooding and eager eavesdropping to quickly build a map of the region of interest,
- Cross-layer (MAC, routing, application) interaction to provide a useful service,
- Analysis of performance in medium-scale simulation and with failures,
- Carrier-sense defeating mechanism for high-priority message delivery, and
- Analysis of the tradeoff of time versus the amount of the region known by a portion of the group.

In the remainder of the paper we describe the Mapping Service in greater detail, including how to detect jamming and design principles employed. Then we develop evaluation criteria and show results from extensive simulation experiments. Finally, we conclude with related and future work.

2. Mapping service

Two primary components form the basis of the mapping service, shown in Figure 1: a jamming detection module, and a mapping module. Both operate on every node in the network.

The jamming detection module is responsible for monitoring the radio and medium access



(a) Upon detecting local jamming, nodes blindly report it to their neighbors.

(b) Receivers outside the jamming form groups and exchange mapping messages.

(c) Groups are coalesced to yield a mapped region.

Figure 2. Overview of nodes collaboratively mapping a jammed region in the network

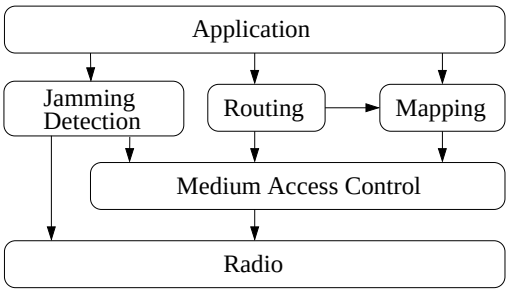


Figure 1. Architectural diagram of a mapping service. The arrows indicate a "uses" relation. Jamming Detection and Mapping interact remotely, that is, between nodes using the radio.

control (MAC) layers and applying heuristics to determine that the node is jammed. As described in Section 2.1, when it determines that the local node is most likely jammed, it sends a message to its neighbors by overriding the carrier-sense multiple access (CSMA) limitation usually enforced by the MAC, shown in Figure 2(a). It alerts the application layer, which can apply power management strategies to help the node outlast the jamming.

Mapping is initiated by the neighbors of jammed nodes who receive the jamming notifications. Each receiver forms a group, explicitly adding nearby jammed nodes as jammed members; the receiver itself becomes a mapping member. Figure 2(b) shows mapping messages, which contain information about the local group, being exchanged between neighbors. Neighboring groups are coalesced and eventually most or all of the mapping members know about the jammed region, as shown in Figure 2(c).

Details of the mapping protocol are in Section 2.2.

When the jammer(s) move or simply stop the attack, the jammed nodes recover and send notifications to their neighbors informing them of this change. The mapping members change the status of the formerly-jammed nodes and send messages to update the group. When a mapping member knows of no neighboring nodes that remain jammed, it retires from the group.

2.1. Jamming detection

Jamming is interfering with the ability of an adversary to communicate. A receiver may recognize known types of jamming by their unique energy patterns. However, this approach requires digital signal processing (DSP) capabilities and a library of patterns that may not be available except in military deployments.

We apply heuristics to determine whether the current node is experiencing non-transient interference that might be called jamming. In fact, we may expand our definition of jamming to include any kind of denial-of-service condition in which the *utility* of the communication channel drops below a certain threshold. This allows us to broaden our jamming model to include mobility, pulse jamming, and even link-layer jamming [15, 26], all based on their impact on the local ability to communicate.

The idea is that below this utility threshold, we are unable to communicate effectively enough for long enough to accomplish our objectives. This is necessarily dependent on the purpose of the sensor network.

Factors which impact this utility metric:

- Repeated inability to access wireless channel
- Bad framing
- Checksum failures
- Illegal values for addresses or other fields
- Protocol violations (e.g., missing ACKs)
- Excessive received signal level
- Low signal-to-noise ratio
- Repeated collisions
- Duration of condition

These data may be obtained from the local radio hardware, MAC layer, network layer, or other available

sensors. In the absence of dedicated jamming detection hardware, DSPs and pattern libraries, these metrics may serve as a proxy for declaring jamming. Being able to use the available facilities is an advantage on low-cost sensor nodes deployed at large scale.

Duration of the condition must be considered to provide hysteresis. The amount of damping needed will depend on the overhead and speed of mapping (see Section 3.3).

Due to the complexity of jamming detection and its sensitivity to the deployment environment, we construct it as a separate module, as shown in Figure 1, with a limited interface to the mapping protocol.

The output of the jamming detection module is a JAMMED or UNJAMMED message broadcast to the node's neighbors. Note that under normal operation a jammed node would not be able to send any message, since nearly all MAC protocols require a carrier-sense to indicate a clear channel before transmission can begin. To overcome this problem, the MAC must provide (by modification, if necessary) a way to override carrier-sense for the purpose of sending a brief, high-priority, unacknowledged broadcast message. For MACs implemented in hardware this may present an obstacle to the use of this mapping service.

2.2. Mapping protocol

2.2.1. Protocol description. When a node receives a JAMMED message, it initiates the mapping protocol. Unless the node already knows of a group with which the jammed node is *compatible* (see discussion below), it creates a new group with a random group ID and a normalized *direction* vector pointing at the jammed node. A BUILD message is not sent immediately; rather, a short announce timer is started. This allows multiple received JAMMED messages to be aggregated before sending a BUILD message, reducing the number of broadcasts containing little information.

If a node receives a BUILD message for an unknown group, it creates a group locally and stores its information. Otherwise it updates its local information by merging in any new members listed in the message; the lists in BUILD messages are always additive. If the receiver is a member of the group, it relays the message by re-broadcasting it after modification.

Relaying and back-flooding

BUILD messages always contain the sending node's state, even during relaying. This provides *forward information diffusion* to help maximize the utility of each broadcast. Each BUILD message contains an original sender field and sequence number for duplicate detection. Although these fields are copied verbatim before re-broadcasting, the member list may contain additional members about which the local node knows.

If the relaying node's neighbors are members and therefore also relay the message, the node will receive the message again. Although the message is a duplicate, in this case we allow the message to be processed once more to provide backward information diffusion—from the downstream repeater back to the previous sender. This *back-flooding* only extends one hop, as the node does not re-broadcast the message.

We use broadcast and limited flooding at each hop because routes may not be stable around the jammed region. For large regions and high node degree this leads to message explosion; this can be mitigated using smarter flooding techniques from [21].

Group compatibility

Mapping nodes associate a *direction* vector with every group. It is the normalized vector sum of the direction from the node to every jammed node from which it has received a JAMMED message. Only groups that are *compatible* may be coalesced. Two groups with *direction* vectors x and y are compatible if $\arccos(x \cdot y) \leq \alpha$, where α is the *compatibility angle*, the maximum angular difference between the two vectors beyond which groups will not be coalesced. When a node determines two groups may be coalesced, it starts a coalesce timer for the two candidate groups.

Upon timer expiry, a coalescing node chooses one to be the dominant group according to an unambiguous rule followed by every node: the group with the larger ID dominates. This avoids any need to synchronize or negotiate and ensures that the result is the same under race conditions, where two or more nodes coalesce groups at the same time. The new group inherits the dominating group's ID and the union of both groups' member lists.

The subordinate group's ID is added to a list of all such IDs in the new group and is included in BUILD messages. Any node receiving a subsequent BUILD message can check the list of subordinate IDs against its local list of active groups. If a match is found, the active group is coalesced into the dominant group. The BUILD message is then relayed, as already described.

Edge nodes

A mapping member determines that it is an *edge* node whenever it has not directly received anything from a neighboring mapping member on its right or left side with respect to its *direction* vector. In this case the edge node sends a PROBE message periodically. If a neighboring mapper does exist but was temporarily unknown due to message loss, this provides an opportunity for it to coalesce nearby groups.

Depending on failures, local topological sparseness, or jamming propagation irregularities, it may happen that adjacent mapping nodes for two different groups cannot directly communicate with each other. In these cases it may be that an intermediate node can com-

municate with both mappers, but has not directly received a JAMMED message and so is ineligible to become a mapping member.

Bridge nodes

We allow such a node to become a *bridging* member if it receives a PROBE message from a group which is compatible with another group it knows about. It's coalesce timer is longer than in the previous case, where a mapping member was coalescing. This is to prevent spurious bridging members where they are not needed.

Recovery

Teardown proceeds in a similar and obvious way. When jammed nodes recover, they send a normal (i.e., not carrier-sense overriding) UNJAMMED message to their neighbors. Mapping members notify the group of recovered nodes using a TEARDOWN message, which has the opposite membership property of a BUILD message: it is always subtractive.

When all of a mapper's neighboring jammed nodes have recovered, the mapper resigns its own implicit membership in the group. As a former member, it will continue to relay BUILD and TEARDOWN messages, so as not to partition the group. However, it will not "reactivate" the group by adding a new jammed node.

This recovery procedure is not adequate for bridging members, which by definition have not directly received a JAMMED message from any jammed nodes. Bridges resign from the group when the neighboring mapper that last informed them of the group resigns, as indicated by a flag in the TEARDOWN message.

Race conditions between adding and removing a jammed node could occur only if the node sends a JAMMED and UNJAMMED message in quick succession. We assume (see Section 2.3) that the jamming detection module includes hysteresis mechanisms to prevent this. Inconsistent membership views are discussed in Section 2.4.

2.2.2. Example sequence. A short example of the early stages of the mapping protocol operation is illustrated in Figure 3 and described below. Each step is enumerated with its corresponding subfigure.

3(a) Jammed nodes J_1, J_2, J_3 detect local jamming and blindly send JAMMED messages to their neighbors.

3(b) Nodes M_1, M_2, M_3 receive the JAMMED messages and create local groups G_1, G_2, G_3. Each group stores a *direction* vector to the reporter. All mappers set an announce timer. The local groups are shown beside each mapper.

3(c) Mapper M_2's announce timer expires and it sends a BUILD message to its neighbors containing the group ID and membership list: $\langle G_2 : J_2 \rangle$.

3(d) M_2's neighbors store group G_2's information. M_3 compares *direction* vectors of G_2 and G_3 and starts a coalesce timer since they are compatible. Also, M_1's announce timer expires and it sends a BUILD message to its neighbors.

3(e) Mapper M_3's coalesce timer expires and it sends a BUILD message containing the dominant group ID, the merged member list, and subordinate group ID list: $\langle G_3 : J_3, J_2 : G_2 \rangle$.

3(f) M_2 receives the message and also coalesces G_2 into G_3. It then relays the message to its neighbors who also coalesce the groups. Only mapping members relay the message.

3(g) Periodically mapper M_1's probe timer expires and it sends a PROBE message since it knows of no other mappers on its left or right with respect to its *direction* vector.

Node B_1 receives the PROBE and starts a long coalesce timer for groups G_1 and G_3, which are compatible.

3(h) B_1's coalesce timer expires, it coalesces groups G_1 and G_3, joining the dominant group as a bridge node, and it sends a BUILD message for the dominant group: $\langle G_3 : J_3, J_2, J_1 : G_2, G_1 \rangle$.

Bridge B_1's *direction* vector is the normalized sum of its neighbors' vectors, since it has not received a JAMMED message from a jammed node.

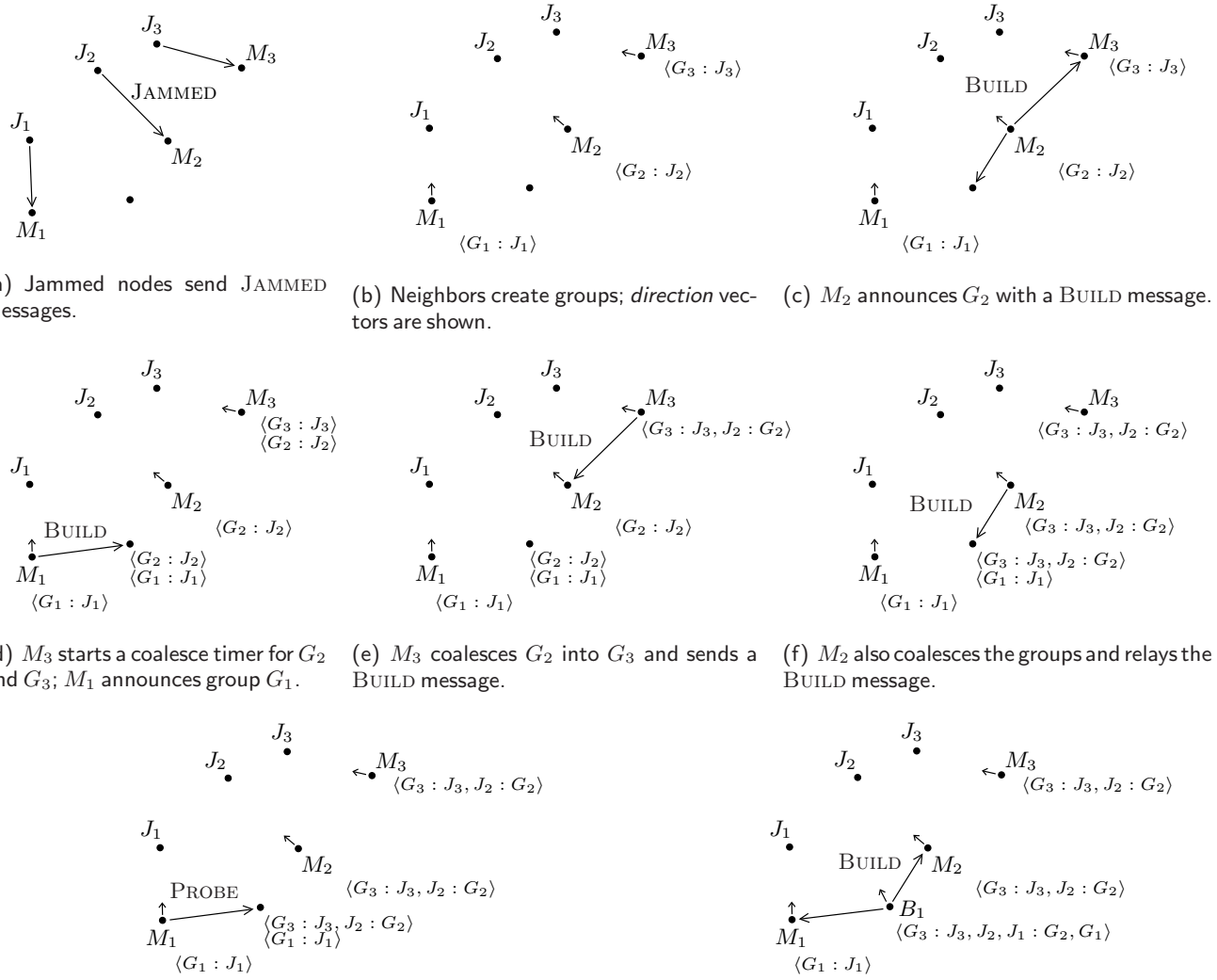
2.3. Assumptions

Our approach to mapping in sensor networks makes the following assumptions about the *network* or *environment*:

- Nodes know their own location and ID, and that of their neighbors. This information is easily exchanged upon deployment.
- Location information is used only for group compatibility calculation, so it need not be very accurate. However, less accuracy gives a greater potential for having multiple groups that are not coalesced due to incorrect compatibility decisions.
- After deployment, nodes move mainly by external environmental forces and are otherwise static.
- Radio links need not be symmetric, since nodes act on whatever messages are received and broadcast them to whichever neighbors are listening. That is, we never directly address neighbors.

The following are our assumptions and limitations on *jamming*:

- The network uses single-channel wireless communication. Specifically, no other communication channel is available to nodes except that which may be jammed by an adversary/accident.
- Though we do allow for multiple jammers, either the sensor network is large enough, or the jamming is limited enough, that attackers cannot jam



(a) Jammed nodes send JAMMED messages.

(b) Neighbors create groups; *direction* vectors are shown.

(c) M_2 announces G_2 with a BUILD message.

(d) M_3 starts a coalesce timer for G_2 and G_3; M_1 announces group G_1.

(e) M_3 coalesces G_2 into G_3 and sends a BUILD message.

(f) M_2 also coalesces the groups and relays the BUILD message.

(g) M_1 sends a PROBE because it has no neighboring mappers.

(h) Node B_1 becomes a bridge and coalesces G_1 and G_3.

Figure 3. Example mapping protocol sequence

the entire network. This would represent a complete and effective denial-of-service of the whole network, with no in-network mapping possible.

- Whether simple or complex, jamming can be quantified by sensor devices using radio signal strength, bit-error-rates, or other available information (as in Section 2.1).
- Jamming dynamics are smoothed by a hysteresis mechanism to be on the same order as the time to map a region. Our evaluation uses simple, constant-power jammers (see Section 3).
- The MAC can provide an unacknowledged, carrier-sense defeating broadcast mechanism for jamming notifications.

2.4. Design principles

The unique characteristics of WSNs present many challenges to designers of MAC, routing, and group management protocols. We suggest the following design principles and describe how each is represented in the mapping service.

1. **Loose group semantics**

   Traditional group management and consensus algorithms impose prohibitive transaction or state requirements in a WSN setting where packet loss and node casualty rates may be an order of magnitude greater than in wired networks. We use loose group semantics in *addressing*, *messaging*, and *membership*.

When creating a group in response to a JAMMED message, we randomly generate group IDs. Probabilistic uniqueness avoids the overhead of synchronizing multiple concurrent group creations, and allows the length to be tuned for variable collision probability.

An unambiguous rule is used to determine the dominant group when coalescing: the numerically larger ID always dominates. This further eliminates any negotiation or confusion about which group will be subsumed by the other.

Messaging within the group is unacknowledged and unreliable. The protocol is driven entirely by message reception and timeout events, with no synchronization or input blocking. Duplicate messages are detected using a per-sender sequence number and are discarded.

Members do not have a shared, consistent view of the global membership. Since nodes relay BUILD messages only if they are members, receivers of a message may assume that the sender is a member. Mapping and bridging members join implicitly, and aside from the local interaction just described, do not explicitly include themselves in any BUILD message.

Jammed nodes, however, are explicitly listed as members in BUILD messages. They are added by proxy, that is, by the node(s) that received their JAMMED messages.

2. **Eager eavesdropping**

Since our goal is to quickly diffuse knowledge of the jammed region to as many mapping members as possible, we eagerly eavesdrop on all received BUILD messages. As BUILD messages are relayed from member to member, each updates their local copy of the membership, adding any new jammed nodes listed in the message. When relaying the message, the original sender ID and sequence number are used, but the membership list included is the newly merged list on the local node. This provides maximum forward information diffusion.

We also use *one-hop back-flooding*. It is well recognized that under certain circumstances[1] the sender of a message may receive it again when the next upstream node relays it. Since, as described above, the relayed message may contain more information than the original, even the sender processes it again to extract any new members. Of course, it does not re-propagate the message. This provides enhanced backward information diffusion, one hop at a time.

This principle is similar to optimizations described in [20] for taking advantage of a shared channel.

3. **Supremacy of local information**

Local information is considered more *up-to-date* than that received by relayed messages. Each node maintains a separate list of neighboring jammed and mapping members in each group (these are the only mapping members stored). The list is updated only by "directly" perceived information: when a JAMMED or UNJAMMED message is received (it is never relayed), jammed member information is updated; since only members relay BUILD or TEARDOWN messages, we implicitly assume that the previous-hop sender of the message is a member, and so update the mapping member information.

We also consider local information to be more *trustworthy*, and use it to verify information received in a relayed message. For example, we trust our local jammed status for a particular node (gathered from a JAMMED or UNJAMMED message, as described above) more than the status of the same node received in a BUILD message's membership list. When there is a conflict, we use the local information.

If an unjammed node receives a BUILD message that indicates it is jammed, it will send an UNJAMMED message with its correct status. This mitigates the propagation of erroneous information, whether intentionally fabricated or simply stale.

4. **Robustness to packet loss and failure**

Individual packets, like individual sensor nodes, are not critical. Messages which indicate significant state information, like JAMMED, UNJAMMED, and PROBE messages, are resent periodically.

Others, like BUILD messages, reflect the state at the sender. If they are lost, the same or updated information will be included in the next BUILD message sent or relayed. Near the end of the mapping protocol, no event may occur to cause a new BUILD message to be sent; in this case the lost message contents will not be repeated. However, the additional information contained in such a message is likely to be limited.

We use bridging members (as described in Section 2.2) to mask areas of failure or low connectivity. They only coalesce groups in response to a PROBE message, which indicates that a nearby node considers itself on the "edge" of a group. After delaying long enough for any nearby mapping members in other groups to initiate a coalesce operation, the receiver will coalesce and join the dominant group as a bridging member, even though it has not directly received a JAMMED message.

1 When links provide two-way communication and there is no hidden terminal interference, the sender of a broadcast message will receive its upstream neighbor's subsequent relay.

5. **Early use of results**
   Nodes do not wait until a complete picture of the jammed region is available to perform avoidance strategies. Whatever local information is available is used to influence routing, power management, higher-layer planning, etc.

This collection of five properties allows the protocol to quickly converge on a single jammed region in a decentralized, fault-tolerant manner. It avoids the costs of synchronization and attempts to maximize information flow through the group.

3. Evaluation

We evaluated the mapping service by extensive simulation. Below we develop the criteria for measuring the service's performance, describe our specific simulation methodology and experimental setup, and show the results. Analysis of the protocol follows.

3.1. Criteria

The following criteria were used to evaluate the mapping protocol's performance in simulation:

Type of groups — We consider two types of groups: *member groups* and *dominant groups*.

Member groups are those held by any mapping member of a group at the end of the simulation. Ideally every member will finish with one and the same group. However, if messages are lost or the network is partitioned due to failures, some nodes may finish with multiple non-coalesced groups.

Dominant groups were determined by considering the subsumed group ID list in every group remaining at the end of the simulation. Any remaining group with an ID not contained in any other group's subsumed group ID list is considered to be a dominant group. Ideally there will be one dominant group. It represents a group that was not coalesced into another group at any time by any node.

If multiple dominant groups remain at the end of the mapping protocol, it represents a *convergence failure*. Other than jammed region shape analysis, no information is available to determine whether the multiple groups refer to the same jammed region.

Number of groups — The number of the different types of groups indicates the convergence (or lack thereof) to a single group surrounding the jammed region, which is desired.

Number of members — The number of members is bounded above by the number of nodes within range of the jammer and mapping nodes within a radio radius of a jammed node. The actual number may be smaller due to message loss.

Number of messages — Overhead of the mapping protocol is demonstrated by the number of different types of messages sent.

Activity time — The amount of time from the first protocol message until the last coalesce operation indicates the duration of useful protocol activity.

Knowledge time — Finer-grained metrics show the time at which a certain percentage of the mapping members knew about a certain percentage of the entire jammed region boundary. This indicates how quickly useful information is gathered, how quickly base station reports (if any) may be pieced together to give a global view of a region, and other time versus knowledge tradeoffs.

Faults — The above metrics for failure scenarios show the protocol's robustness.

3.2. Simulation methodology

We simulated a wireless sensor network using the GloMoSim simulator [27], with a 4000 by 4000 meter field in which 400 nodes were placed at 200 meter intervals on a grid. Radio settings are consistent with a design point between the MICA2 and WaveLAN devices' capabilities.

MACA [16] was modified to provide the carrier-sense defeating broadcast and to bound the number of RTS retransmissions consistent with the limit imposed by 802.11 [13]. We did not feel it necessary to use a more complex MAC layer since the mapping protocol uses only broadcast frames.

Three transmission power levels were used to test performance under different levels of connectivity, from minimally-connected (4 neighbors) to moderately-connected (8 and 12 neighbors).

The power of the jamming signal was varied in some experiments. The size of the jammed region ranged from affecting 5 nodes to 144 nodes. For any one configuration, the jamming began at a predetermined time and remained constant throughout the remainder of the simulation.

Jamming was detected by monitoring the number of consecutive unsuccessful attempts to capture the wireless channel. After a total elapsed time of 250 milliseconds (196 attempts), during which time every attempt indicated a busy channel, jamming was declared. We artificially induced every node to send a message (the only time channel capture is attempted) within 1.5 seconds after the actual start of jamming, so as to measure the time taken by the mapping protocol itself.

We performed two sets of experiments:

1. The first experiment tested the correctness of the jammed region mapping and performance for vary-

ing jammed region sizes, from 5 nodes affected up to 144 nodes affected. The series was repeated three times, one each for neighborhood sizes of 4, 8, and 12 nodes. For each configuration, measurements reported are the mean of 10 simulations with different random seeds.

2. Secondly, we tested the performance of the 8 and 12 neighborhood sizes under mapping node failure. The failure rate ranged from 0 to 35%. When a node became a mapper, it had a probability of failure from 0 to 35%, depending on the experiment. A failed node neither sent nor received further messages; that is, it crashed. For each configuration, measurements reported are the mean of 10 simulations with different random seeds.

3.3. Results

3.3.1. Experiment 1: Varying jammer range. One of the most important metrics for determining the success of the mapping protocol is the number of dominant groups left at the end of the simulation. Figure 4(a) shows that in the moderately-connected networks (eight and twelve neighbors), all groups were coalesced into a single dominant group.

The minimally connected network (four neighbors) did not always achieve convergence, as shown by the sole deviation at 121 nodes jammed in Figure 4(a). With a mean of 1.2 dominant groups in this case, some simulations finished without ever merging two or more groups. Standard deviation in this single case was 0.42.

Figure 4(b) shows more detail for the minimally connected case. At the 121 jammed-nodes range, there are an average of 1.7 distinct groups present in each mapper. This is a result of message-loss induced partitions, as evidenced by a peak in the number of PROBE messages sent for this case (not shown).

Moderately-connected networks experienced strong convergence. Only one member group remained in all except two cases: at 29 and 109 jammed-nodes, the eight-neighbor simulations averaged 1.1 member groups (not shown).

Especially for minimally connected networks or with significant failures (see Experiment 2), bridge nodes are important for maintaining connectivity between mappers surrounding a jammed region. Connectivity allows adjacent groups to communicate and coalesce.

Figure 4(c) shows the types of members present for the minimally connected network simulated. In this figure, *jammed nodes* are only those along the boundary which are within radio range of a non-jammed node (i.e., a mapping node). Hence the difference between the number of *jammed nodes* and *mapping nodes* is one radio range. Figure 4(c) also shows an increasing reliance on bridging nodes as the region expands. By contrast, the more well-connected networks used an average of only 0.3 bridge nodes in the worst case.

A rough measurement of the duration of the mapping protocol activity is the time from the first group formation in response to a JAMMED message, until the last coalesce operation in response to a BUILD message. Figure 4(d) shows that both moderately-connected networks converge quickly: from 1.5 seconds to just over 5 seconds for the largest jammed region. This is fast enough to allow a reasonable real-time response to jamming in the sensor network.

The gap between four and eight-neighbor times in Figure 4(d) is directly due to the four-neighbor network's need for bridge nodes, as already discussed and shown in 4(c). The first component of the delay is due to a probe timer which must expire before edge nodes send a PROBE message. Potential bridge nodes will not coalesce groups until this PROBE message is received. They also defer to existing mapping members by waiting longer to coalesce compatible groups, in an effort to avoid superfluous bridge creation.

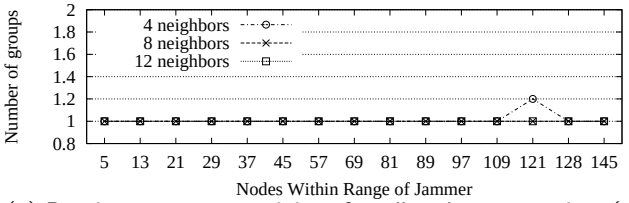
The maximum standard deviations of the activity times decrease as the network becomes more connected: 1.49, 0.90, and 0.77 for 4, 8, and 12 neighbor cases, respectively.

More fine-grained measurement and analysis of the convergence time is possible by considering how quickly some subset of the mappers know about the jammed region. In a network where we expect to have frequent message loss and node failure, it is unreasonably strict to require that every mapper know about the entire jammed region before any kind of coping strategy can be employed. We may be interested in the time necessary for only half of the mapping nodes to know about at least half of the jammed region, especially if the coping strategies depend mostly on local information, which is discovered quickly.

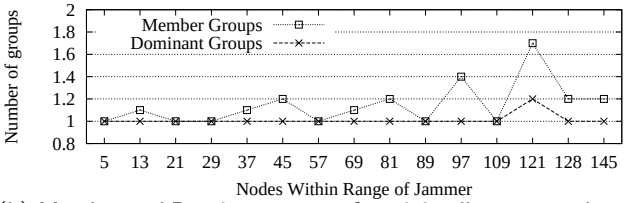
Figures 5(a) and 5(b) show time versus knowledge tradeoffs for 50 and 90 percent of mapping nodes, respectively, for the twelve neighbor case. Each series depicts the average time from the start of jamming until some amount (from 20 to 90 percent) of the jammed region is known. For example, from the figures it can be seen that a majority of the mapping nodes know up to a third of the jammed region in 0.5 to 3 seconds.

3.3.2. Experiment 2: Varying failure. Here we tested the mapping protocol's performance under varying amounts of mapping node failure. Figure 6(a) shows the number of dominant groups for the moderately-connected eight and twelve neighbor cases as the failures increase from 0 to 35% of joining mappers. Failed nodes crash, neither sending nor receiving further messages. Metrics collected do not include failed nodes.

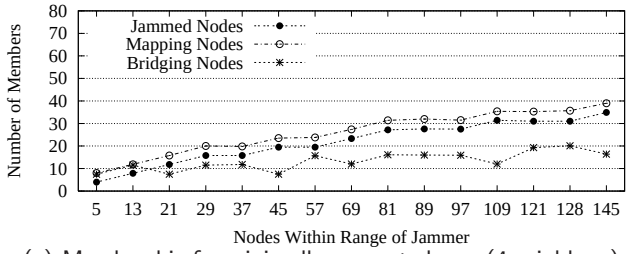
The more well-connected network is most robust to failure, showing little increase in the number of dom-



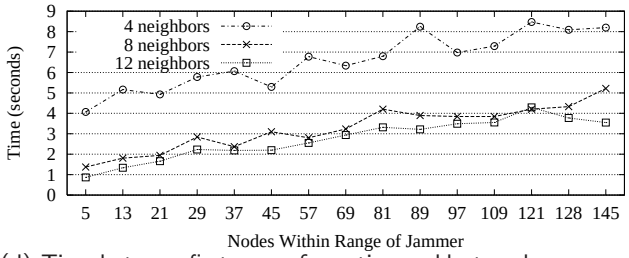
(a) Dominant groups remaining after all coalesce operations (4, 8, and 12 neighbors)



(b) Member and Dominant groups for minimally connected case (4 neighbors)

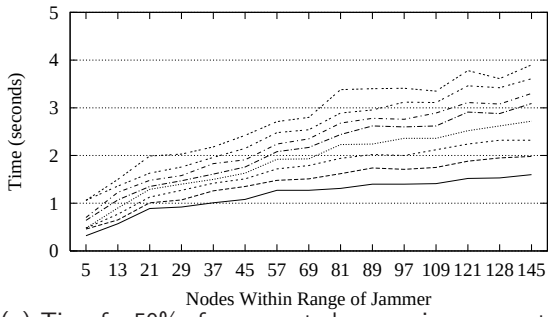


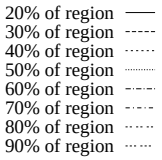
(c) Membership for minimally connected case (4 neighbors)



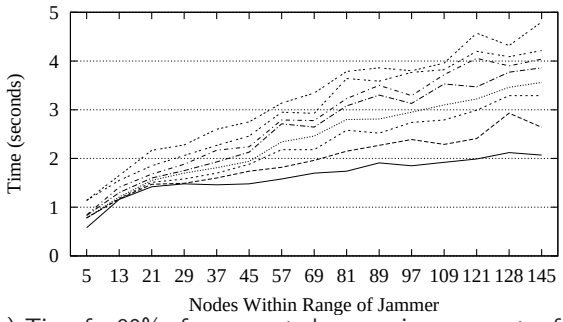
(d) Time between first group formation and last coalesce operation (4, 8, and 12 neighbors)

Figure 4. Groups, membership and activity time shown for Experiment 1, which varies the number of nodes affected by the jammer.





(a) Time for 50% of mappers to know various amounts of the jammed region



(b) Time for 90% of mappers to know various amounts of the jammed region

Figure 5. Time vs. Knowledge for Experiment 1, 12 neighbor case

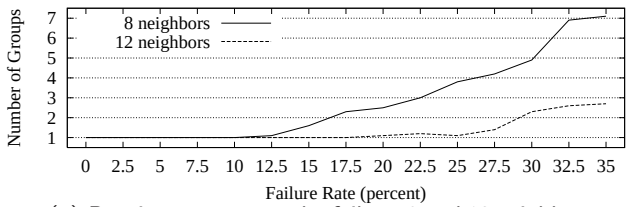
inant groups at up to 25% mapper failure. Standard deviations rise with the failure rates, to 1.24 for the twelve-neighbor case and 2.25 for the eight-neighbor case. As the number of failures rise, so do the number of partitions, which prevent adjacent groups from coalescing. The increase of PROBE messages for the eight-neighbor case in Figure 6(b) is a direct result of these partitions.

3.4. Discussion

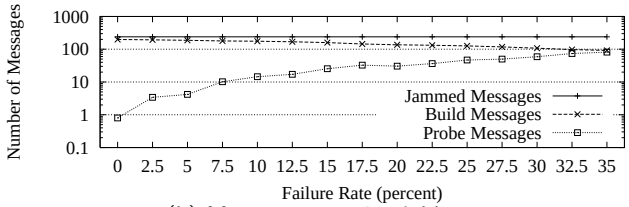
Eager eavesdropping and additive BUILD messages provide faster information diffusion by processing and updating the state contained in messages at each hop. This combination of one-hop backflooding and loose group semantics speeds convergence to a single dominant group, even with significant failure rates.

Although this information diffusion is maximized when simple flooding is employed, since every node relays its local state to its neighbors it yields large messaging overhead. The broadcast storm problem is well explained in [21], and any of the techniques given there may be used to good effect here. The messaging efficiency gained by reducing rebroadcasts will reduce the information diffusion of backflooding, however.

A potentially more important tradeoff involves the carrier-sense defeating mechanism used for sending JAMMED messages. As described in Section 2.1, a



(a) Dominant groups under failure, 8 and 12 neighbors



(b) Messages sent, 8 neighbors

Figure 6. Failure performance in Experiment 2, 121 affected nodes

jammed node may never detect a clear channel, necessitating the "blind" transmission of a short, unacknowledged message on a busy channel. This calls for restraint: eagerly or frequently ignoring carrier-sense may compound a congestion or jamming problem.

Even if jamming is reliably detected and JAMMED messages are sent infrequently, this mechanism may actually slow the mapping protocol. This may happen if many nodes detect jamming simultaneously and all send JAMMED messages that interfere with each other at the receiving mapping nodes. The jammed nodes will not be added to the group unless they randomly stagger future rebroadcasts.

In summary, we have shown that the mapping protocol correctly converges to a single dominant group for moderately-connected networks, achieves high-levels of knowledge diffusion quickly enough for a real-time response, and tolerates up to 25% mapper failure.

4. Related work

Related work includes other methods for mapping regions, boundaries, or contours. Chintalapudi and Govindan [6] present localized edge detection techniques based on statistics, image processing, and classification. Nowak and Mitra [23] describe a method for hierarchical boundary estimation using recursive dyadic partitioning (RDP). They develop an inverse-proportionality relation between energy expended and the mean-square error of the boundary and show their method is near-optimal with respect to this fundamental tradeoff. Hellerstein et al. [11] give an isobar mapping algorithm, based on a TinyDB query language, which merges container polygons having the same sensor value. Finally, Zhao et al. [28] describe the tracking problem, including that of contours, in terms of collaborative information processing.

JAM differs primarily in that does not (and cannot, due to jamming) assume nodes inside and outside of the region can communicate. The region must be inferred by surrounding nodes, who may communicate indirectly though bridge nodes, if necessary.

Byers and Kormann describe techniques for mapping wireless network coverage in which the measuring device is mobile [5]. They systematically sample the signal-to-noise (SNR) ratio of a connection with an access point to estimate its location. We also suggest using SNR, among other metrics, to detect jamming, though we assume that the sensor nodes are not mobile or as powerful as the devices they employ.

The EnviroTrack architecture [1] provides a programming abstraction for aggregate mobile entities in sensor networks. Though it is more suited for point-entities than for region-entities, this type of architecture is an example of one that could provide programmer and application access to jammed regions.

Roman, Huang, and Hazemi [25] propose a group merging and maintenance protocol for ad hoc networks that provides a consistent group membership view even with node mobility.

Various routing algorithms [17, 7] recover from voids in the network, but they may not explicitly represent or export them to other WSN services.

Finally, Anderson gives an overview of traditional methods for radio jamming avoidance and low-probability of intercept techniques in military environments [3].

5. Conclusion

Among the security problems that wireless sensor networks face is the prospect of relatively simple denial-of-service. We describe one such problem, jamming by radio interference, and propose and evaluate a protocol for mapping the extent of the jammed region.

We evaluate the mapping protocol by simulation, showing its correctness and performance for varying size jammed regions and failure rates. The results show that the mapping protocol converges quickly to provide a single dominant group of jammed nodes when the network is at least moderately connected. Further, when nodes have at least twelve neighbors in communication range, the protocol is robust to failure rates as high as 20–25 percent of mapping nodes.

Contributions of this paper include the simulation-based evaluation of a protocol that uses loose group semantics integrated with eager eavesdropping to quickly build a map of a jammed region, a carrier-sense defeating mechanism for high-priority message delivery, and an analysis of time versus knowledge tradeoffs.

Future work may apply the mapping protocol to other network hazards such as voids or obstacles, to general sensor-value contours, or may use more sophisticated jamming detection.

6. Acknowledgments

The authors gratefully acknowledge the feedback from the anonymous reviewers. This work was supported in part by the DARPA NEST program under grant F33615–01–C–1905 and by MURI award N00014–01–1–0576.

References

[1] T. Abdelzaher, B. Blum, D. Evans, J. George, L. Gu, T. He, C. Huang, P. Nagaraddi, S. Son, P. Sorokin, J. Stankovic, and A. Wood. EnviroTrack: An environmental programming model for tracking applications in distributed sensor networks, 2003. *In submission.*

[2] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless sensor networks: A survey. *Computer Networks (Amsterdam, Netherlands: 1999)*, 38(4):393–422, Mar. 2002.

[3] R. Anderson. *Security Engineering: A Guide to Building Dependable Distributed Systems*, pages 326–331. Wiley Computer Publishing, 2001.

[4] N. Bulusu, J. Heidemann, and D. Estrin. GPS-less low cost outdoor localization for very small devices. *IEEE Personal Communications Magazine*, 7(5):23–34, Oct. 2000.

[5] S. Byers and D. Kormann. 802.11b access point mapping. *CACM*, 46(5):41–46, May 2003.

[6] K. K. Chintalapudi and R. Govindan. Localized edge detection in a sensor field. In *IEEE SNPA '03*, 2003.

[7] D. D. Couto and R. Morris. Location proxies and intermediate node forwarding for practical geographic forwarding. Technical Report MIT-LCS-TR824, MIT Laboratory for Computer Science, June 2001.

[8] D. Estrin, R. Govindan, J. S. Heidemann, and S. Kumar. Next century challenges: Scalable coordination in sensor networks. In *Mobile Computing and Networking*, pages 263–270, 1999.

[9] D. Ganesan, B. Krishnamachari, A. Woo, D. Culler, D. Estrin, and S. Wicker. Complex behavior at scale: An experimental study of low-power wireless sensor networks. Technical Report UCLA/CSD-TR 02-0013, UCLA Computer Science, 2002.

[10] T. He, C. Huang, B. M. Blum, J. A. Stankovic, and T. F. Abdelzaher. Range-free localization schemes in large scale sensor networks. Technical Report CS-TR-2003-06, University of Virginia, 2003.

[11] J. M. Hellerstein, W. Hong, S. Madden, and K. Stanek. Beyond average: Towards sophisticated sensing with queries. In *IPSN '03*, Mar. 2003.

[12] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. E. Culler, and K. S. J. Pister. System architecture directions for networked sensors. In *ASPLOS*, pages 93–104, 2000.

[13] IEEE Computer Society LAN MAN Standards Committee. *IEEE Std 802.11: Wireless LAN Medium Access Control and Physical Layer Specifications*, Aug. 1999.

[14] J. Kahn, R. Katz, and K. Pister. Emerging challenges: Mobile networking for 'smart dust'. *J. Comm. Networks*, 2(3):188–196, Sept. 2000.

[15] C. Karlof and D. Wagner. Secure routing in wireless sensor networks: Attacks and countermeasures. In *First IEEE International Workshop on Sensor Network Protocols and Applications*, 2003.

[16] P. Karn. MACA – a new channel access method for packet radio. In *ARRL/CRRL Amateur Radio 9th Computer Networking Conference*, pages 134–140. ARRL, 1990.

[17] B. Karp and H. T. Kung. GPSR: Greedy perimeter stateless routing for wireless networks. In *MobiCom '00*, pages 243–254, N. Y., Aug. 2000. ACM Press.

[18] J. Li, J. Jannotti, D. De Couto, D. Karger, and R. Morris. A scalable location service for geographic ad-hoc routing. In *MobiCom '00*, pages 120–130, Aug. 2000.

[19] H. Lundgren, E. Nordstr, and C. Tschudin. Coping with communication gray zones in IEEE 802.11b based ad hoc networks. In *Proceedings of the fifth ACM international workshop on Wireless mobile multimedia*, pages 49–55. ACM Press, 2002.

[20] S. R. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong. TAG: a Tiny AGgregation service for ad-hoc sensor networks. In *OSDI*, Dec. 2002.

[21] S.-Y. Ni, Y.-C. Tseng, Y.-S. Chen, and J.-P. Sheu. The broadcast storm problem in a mobile ad hoc network. In *MobiCom '99*, pages 151–162. ACM Press, 1999.

[22] D. Niculescu and B. Nath. Ad hoc positioning system (APS). In *Proceedings of IEEE GLOBECOM '01*, pages 2926–2931, Nov. 2001.

[23] R. Nowak and U. Mitra. Boundary estimation in sensor networks: Theory and methods. In *IPSN '03*, Apr. 2003.

[24] R. L. Pickholtz, D. L. Schilling, and L. B. Milstein. Theory of spread spectrum communications—a tutorial. *IEEE Transactions on Communications*, 20(5):855–884, May 1982.

[25] G.-C. Roman, Q. Huang, and A. Hazemi. Consistent group membership in ad hoc networks. In *ICSE*, pages 381–388, May 2001.

[26] A. D. Wood and J. A. Stankovic. Denial of service in sensor networks. *IEEE Computer*, 35(10):54–62, Oct. 2002.

[27] X. Zeng, R. Bagrodia, and M. Gerla. Glomosim: A library for parallel simulation of large-scale wireless networks. In *Workshop on Parallel and Distributed Simulation*, pages 154–161, 1998.

[28] F. Zhao, J. Liu, J. Liu, L. Guibas, and J. Reich. Collaborative signal and information processing: An information directed approach. In *Proceedings of the IEEE, to appear*, 2003.