Secure Routing in Wireless Sensor Networks: Attacks and Countermeasures

Chris Karlof David Wagner
University of California at Berkeley
{ckarlof,daw}@cs.berkeley.edu

Abstract— **We consider routing security in wireless sensor networks. Many sensor network routing protocols have been proposed, but none of them have been designed with security as a goal. We propose security goals for routing in sensor networks, show how attacks against ad-hoc and peer-to-peer networks can be adapted into powerful attacks against sensor networks, introduce two classes of novel attacks against sensor networks — sinkholes and `HELLO` floods, and analyze the security of all the major sensor network routing protocols. We describe crippling attacks against all of them and suggest countermeasures and design considerations. This is the first such analysis of secure routing in sensor networks.**

I. Introduction

Our focus is on routing security in wireless sensor networks. Current proposals for routing protocols in sensor networks optimize for the limited capabilities of the nodes and the application specific nature of the networks, but do not consider security. Although these protocols have not been designed with security as a goal, we feel it is important to analyze their security properties. When the defender has the liabilities of insecure wireless communication, limited node capabilities, and possible insider threats, and the adversaries can use powerful laptops with high energy and long range communication to attack the network, designing a secure routing protocol is non-trivial.

One aspect of sensor networks that complicates the design of a secure routing protocol is in-network aggregation. In more conventional networks, a secure routing protocol is typically only required to guarantee message availability. Message integrity, authenticity, and confidentiality are handled at a higher layer by an end-to-end security mechanism such as SSH or SSL. End-to-end security is possible in more conventional networks because it is neither necessary nor desirable for intermediate routers to have access to the contents of messages. However, in sensor networks, in-network processing makes end-to-end security mechanisms harder to deploy because intermediate nodes need direct access to the contents of the messages. Link layer security mechanisms can help mediate some of the resulting vulnerabilities, but it is not enough: we will now require much more from our routing protocols, and they must be designed with this in mind.

A. Our contributions

We present crippling attacks against all the major routing protocols for sensor networks. Because these protocols have not been designed with security as a goal, it is unsurprising they are all insecure. However, this is non-trivial to fix: it is unlikely a sensor network routing protocol can be made secure by incorporating security mechanisms after design has completed. Our assertion is that sensor network routing protocols must be designed with security in mind, and this is the only effective solution for secure routing in sensor networks.

We make five main contributions.

- We propose threat models and security goals for secure routing in wireless sensor networks.
- We introduce two novel classes of previously undocumented attacks against sensor networks[1] – sinkhole attacks and `HELLO` floods.
- We show, for the first time, how attacks against ad-hoc wireless networks and peer-to-peer networks [1], [2] can be adapted into powerful attacks against sensor networks.
- We present the first detailed security analysis of all the major routing protocols and energy conserving topology maintenance algorithms for sensor networks. We describe practical attacks against all of them that would defeat any reasonable security goals.
- We discuss countermeasures and design considerations for secure routing protocols in sensor networks.

II. Background

We use the term *sensor network* to refer to a heterogeneous system combining tiny sensors and actuators with general-purpose computing elements. Sensor networks may consist of hundreds or thousands of low-power, low-cost nodes, possibly mobile but more likely at fixed locations, deployed en masse to monitor and affect the environment. For the remainder of this paper we assume that all nodes' locations are fixed for the duration of their lifetime.

For concreteness, we target the Berkeley TinyOS sensor platform in our work. Because this environment is so radically different from any we had previously encountered, we feel it is instructive to give some background on the capabilities of the Berkeley TinyOS platform.

A representative example is the Mica *mote*[2], a small (several cubic inch) sensor/actuator unit with a CPU, power source, radio, and several optional sensing elements. The processor is a 4 MHz 8-bit Atmel ATMEGA103 CPU with 128 KB

[1] These attacks are relevant to some ad-hoc wireless networks as well.
[2] We use the terms *mote* and *sensor node* interchangeably.

Protocol	Relevant attacks
TinyOS beaconing	Bogus routing information, selective forwarding, sinkholes, Sybil, wormholes, HELLO floods
Directed diffusion and its multipath variant	Bogus routing information, selective forwarding, sinkholes, Sybil, wormholes, HELLO floods
Geographic routing (GPSR, GEAR)	Bogus routing information, selective forwarding, Sybil
Minimum cost forwarding	Bogus routing information, selective forwarding, sinkholes, wormholes, HELLO floods
Clustering based protocols (LEACH, TEEN, PEGASIS)	Selective forwarding, HELLO floods
Rumor routing	Bogus routing information, selective forwarding, sinkholes, Sybil, wormholes
Energy conserving topology maintenance (SPAN, GAF, CEC, AFECA)	Bogus routing information, Sybil, HELLO floods

Fig. 1. Summary of attacks against proposed sensor networks routing protocols.

of instruction memory, 4 KB of RAM for data, and 512 KB of flash memory. The CPU consumes 5.5 mA (at 3 volts) when active, and two orders of magnitude less power when sleeping. The radio is a 916 MHz low-power radio from RFM, delivering up to 40 Kbps bandwidth on a single shared channel and with a range of up to a few dozen meters or so. The RFM radio consumes 4.8 mA (at 3 volts) in receive mode, up to 12 mA in transmit mode, and 5 μA in sleep mode. An optional sensor board allows mounting of a temperature sensor, magnetometer, accelerometer, microphone, sounder, and other sensing elements. The whole device is powered by two AA batteries, which provide approximately 2850 mA hours at 3 volts.

Sensor networks often have one or more points of centralized control called *base stations*. A base station is typically a gateway to another network, a powerful data processing or storage center, or an access point for human interface. They can be used as a nexus to disseminate control information into the network or extract data from it. In some previous work on sensor network routing protocols, base stations have also been referred to as *sinks*.

Base stations are typically many orders of magnitude more powerful than sensor nodes. They might have workstation or laptop class processors, memory, and storage, AC power, and high bandwidth links for communication amongst themselves. However, sensors are constrained to use lower-power, lower-bandwidth, shorter-range radios, and so it is envisioned that the sensor nodes would form a multi-hop wireless network to allow sensors to communicate to the nearest base station. See Figure 3 for a picture illustrating a representative architecture for sensor networks.

A base station might request a steady stream of data, such as a sensor reading every second, from nodes able to satisfy a query. We refer to such a stream as a *data flow* and to the nodes sending the data as *sources*.

In order to reduce the total number of messages sent and thus save energy, sensor readings from multiple nodes may be processed at one of many possible *aggregation points*. An aggregation point collects sensor readings from surrounding nodes and forwards a single message representing an aggregate of the values. Aggregation points are typically regular sensor nodes, and their selection is not necessarily static. Aggregation points could be chosen dynamically for each query or event, for example. It is also possible that every node in the network functions as an aggregation point, delaying transmission of an outgoing message until a sufficient number of incoming messages have been received and aggregated.

Power management in sensor networks is critical. At full power, the Berkeley Mica mote can run for only two weeks or so before exhausting its batteries. Consequently, if we want sensor networks to last for years, it is crucial that they run at around a 1% duty cycle (or less). Similarly, since the power consumption of the radio is three orders of magnitude higher when transmitting or listening than when in sleep mode, it is crucial to keep the radio in sleep mode the overwhelming majority of the time.

It is clear that we must discard many preconceptions about network security: sensor networks differ from other distributed systems in important ways. The resource-starved nature of sensor networks poses great challenges for security. These devices have very little computational power: public-key cryptography is so expensive as to be unusable, and even fast symmetric-key ciphers must be used sparingly. With only 4 KB of RAM, memory is a resource that must be husbanded carefully, so our security protocols cannot maintain much state. Also, communication bandwidth is extremely dear: each bit transmitted consumes about as much power as executing 800–1000 instructions [3], and as a consequence, any message expansion caused by security mechanisms comes at significant cost. Power is the scarcest resource of all: each milliamp consumed is one milliamp closer to death, and as a result, nearly every aspect of sensor networks must be designed with power in mind.

Lest the reader think that these barriers may disappear in the future, we point out that it seems unlikely that Moore's law will help in the foreseeable future. Because one of the most

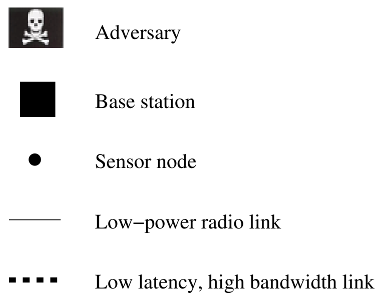

Fig. 2. **Sensor network legend.** All nodes may use low power radio links, but only laptop-class adversaries and base stations can use low latency, high bandwidth links.

important factors determining the value of a sensor network comes from how many sensors can be deployed, it seems likely there will be strong pressure to develop ever-cheaper sensor nodes. In other words, we expect that users will want to ride the Moore's law curve down towards ever-cheaper systems at a fixed performance point, rather than holding price constant and improving performance over time.

This leaves us with a very demanding environment. How can security possibly be provided under such tight constraints? Yet security is critical. With sensor networks being envisioned for use in critical applications such as building monitoring, burglar alarms, and emergency response, with the attendant lack of physical security for hundreds of exposed devices, and with the use of wireless links for communications, these networks are at risk.

III. SENSOR NETWORKS VS. AD-HOC WIRELESS NETWORKS

Wireless sensor networks share similarities with ad-hoc wireless networks. The dominant communication method in both is multi-hop networking, but several important distinctions can be drawn between the two. Ad-hoc networks typically support routing between any pair of nodes [4], [5], [6], [7], whereas sensor networks have a more specialized communication pattern. Most traffic in sensor networks can be classified into one of three categories:

1) Many-to-one: Multiple sensor nodes send sensor readings to a base station or aggregation point in the network.
2) One-to-many: A single node (typically a base station) multicasts or floods a query or control information to several sensor nodes.
3) Local communication: Neighboring nodes send localized messages to discover and coordinate with each other. A node may broadcast messages intended to be received by all neighboring nodes or unicast messages intended for a only single neighbor[3].

Nodes in ad-hoc networks have generally been considered to have limited resources, but as we have seen in Section II, sensor nodes are even more constrained. Of all of the resource

[3]By neighbor we mean a node within normal radio range.

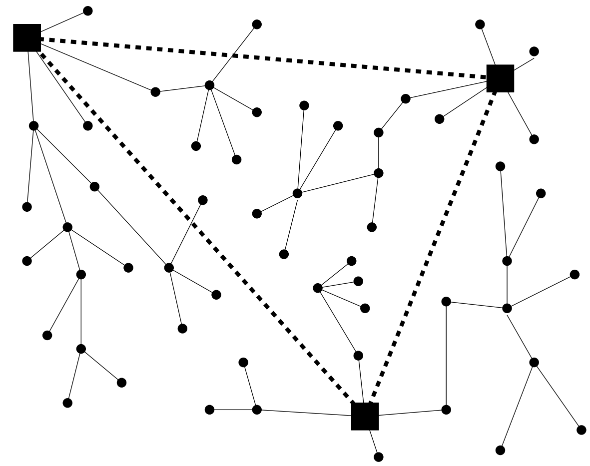

Fig. 3. A representative sensor network architecture.

constraints, limited energy is the most pressing. After deployment, many sensor networks are designed to be unattended for long periods and battery recharging or replacement may be infeasible or impossible.

Nodes in sensor networks often exhibit trust relationships beyond those that are typically found in ad-hoc networks. Neighboring nodes in sensor networks often witness the same or correlated environmental events. If each node sends a packet to the base station in response, precious energy and bandwidth are wasted. To prune these redundant messages to reduce traffic and save energy, sensor networks require in-network processing, aggregation, and duplicate elimination. This often necessitates trust relationships between nodes that are not typically assumed in ad-hoc networks.

IV. RELATED WORK

Security issues in ad-hoc networks are similar to those in sensor networks and have been well enumerated in the literature [8], [9], but the defense mechanisms developed for ad-hoc networks are not directly applicable to sensor networks. There are several reasons for why this is so, but they all relate to the differences between sensor and ad-hoc networks enumerated in the previous section.

Some ad-hoc network security mechanisms for authentication and secure routing protocols are based on public key cryptography [8], [10], [11], [12], [13], [14], [15], [16]. Public key cryptography is too expensive for sensor nodes. Security protocols for sensors networks must rely exclusively on efficient symmetric key cryptography.

Secure routing protocols for ad-hoc networks based on symmetric key cryptography have been proposed [17], [18], [19], [20]. These protocols are based on source routing or distance vector protocols and are unsuitable for sensor networks. They are too expensive in terms of node state and packet overhead and are designed to find and establish routes between *any* pair of nodes—a mode of communication not prevalent in sensor networks.

Marti et al. [21] and Buchegger and Boudec [22] consider the problem of minimizing the effect of misbehaving or selfish nodes on routing through punishment, reporting, and holding grudges. These application of these techniques to sensor networks is promising, but these protocols are vulnerable to blackmailers.

Perrig et al. [23] present two building block security protocols optimized for use in sensor networks, SNEP and μTESLA. SNEP provides confidentiality, authentication, and freshness between nodes and the sink, and μTESLA provides authenticated broadcast.

V. Problem Statement

Before diving into specific routing protocols, it helps to have a clear statement of the routing security problem. In the following sections we outline our assumptions about the underlying network, propose models for different classes of adversaries, and consider security goals in this setting.

A. Network Assumptions

Because sensor networks use wireless communications, we must assume that radio links are insecure. At the very least, attackers can eavesdrop on our radio transmissions, inject bits in the channel, and replay previously overheard packets. We assume that if the defender can deploy many sensor nodes, then the adversary will likely also be able to deploy a few malicious nodes with similar hardware capabilities as the legitimate nodes. The attacker may come upon these malicious nodes by purchasing them separately, or by "turning" a few legitimate nodes by capturing them and physically overwriting their memory. We assume that the attacker might have control of more than one node, and these malicious nodes might collude to attack the system. Also, in some cases colluding nodes might have high-quality communications links available for coordinating their attack (see, e.g., Section VI-E for one way in which attackers might put such a capability to use).

We do not assume sensor nodes are tamper resistant. We assume that if an adversary compromises a node, she can extract all key material, data, and code stored on that node. While tamper resistance might be a viable defense for physical node compromise for some networks, we do not see it as a general purpose solution. Extremely effective tamper resistance tends to add significant per-unit cost, and sensor nodes are intended to be very inexpensive.

The physical and MAC layers are susceptible to direct attack. Adversaries can jam radio links by transmitting without stop or try to cause collisions by leveraging the "hidden terminal" problem [24]. With a MAC protocol using Clear-to-Send/Receive-to-Send (CTS/RTS) frames, adversaries can send frequent CTS frames with long "Duration" fields, effectively preventing other nodes from using the channel. In addition, MAC protocols using randomized backoff are susceptible to attack if nodes have poor entropy management or predictable pseudo-random number generation. Adversaries able to predict backoff times (and thus when a node will transmit) can cause long backoff times or collisions.

Physical layer threats are typically countered by frequency hopping or spread spectrum communication [25], and MAC layer attacks can be alleviated by using a less susceptible protocol (Slotted Aloha [26], for example), good entropy management, and a cryptographically secure pseudo-random number generator [27]. It is possible for adversaries to exploit weaknesses in these layers to mount attacks whose goals are similar to those discussed in Section VI (for example, an adversary could try to corrupt packets selectively by well timed collisions or jamming), but we will not consider attacks on the physical and MAC layers any further.

B. Trust Requirements

Since base stations interface a sensor network to the outside world, the compromise of a significant number of them can render the entire network useless. For this reason we assume that base stations are *trustworthy*, in the sense that they can be trusted if necessary and are assumed to behave correctly. Most, but not all routing protocols depend on nodes to trust messages from base stations.

Aggregation points may be trusted components in certain protocols. Nodes may rely on routing information from aggregation points and trust that messages sent to aggregation points will be accurately combined with other messages and forwarded to a base station. Aggregation points are often regular sensor nodes. It is possible that adversaries may try to deploy malicious aggregation points or attempt to turn currently compromised nodes into aggregation points. For this reason aggregation points may not necessarily be trustworthy.

C. Threat Models

An important distinction can be made between *mote-class attackers* and *laptop-class attackers*. In the former case, the attacker has access to a few sensor nodes with similar capabilities to our own, but not much more than this. In contrast, a laptop-class attacker may have access to more powerful devices, like laptops or their equivalent. Thus, in the latter case, malicious nodes have an advantage over legitimate nodes: they may have greater battery power, a more capable CPU, a high-power radio transmitter, or a sensitive antenna.

An attacker with laptop-class devices can do more than an attacker with only ordinary sensor nodes. An ordinary sensor node might only be able to jam the radio link in its immediate vicinity, while a laptop-class attacker might be able to jam the entire sensor network using its stronger transmitter. A single laptop-class attacker might be able to eavesdrop on an entire network, while sensor nodes would ordinarily have a limited range. Also, laptop-class attackers might have a high-bandwidth, low-latency communications channel not available to ordinary sensor nodes, allowing such attackers to coordinate their efforts.

A second distinction can be made between *outsider attacks* and *insider attacks*. We have so far been discussing outsider attacks, where the attacker has no special access to the sensor network. One may also consider insider attacks, where an authorized participant in the sensor network has gone bad.

Insider attacks may be mounted from either compromised sensor nodes running malicious code or adversaries who have stolen the key material, code, and data from legitimate nodes, and who then use one or more laptop-class devices to attack the network.

D. Security Goals

In an ideal world, we would like to guarantee the confidentiality, integrity, authenticity, and availability of all messages in the presence of resourceful adversaries. Every eligible receiver should receive all messages intended for it and be able to verify the integrity of every message as well as the identity of the sender. Adversaries should not be able to infer the contents of any message, even if they participate in the routing of it.

However, the question remains to which of these goals should be the responsibility of the routing protocol and which goals are handled better at higher (e.g., application) or lower (e.g., link) layers. In more conventional networks, the primary security goal of a routing protocol is reliable delivery of messages, i.e., protection against denial of service, and message authenticity, integrity, and confidentiality are usually achieved by an end-to-end security mechanism such as SSH or SSL. The reason for this stratification of responsibilities is because the dominating traffic pattern is end-to-end communication, where it is neither necessary nor desirable for the contents of the message (beyond the necessary headers) to be available to the intermediate routers.

This is not the case in sensor networks. The many cases, the dominant traffic pattern in sensor networks is many-to-one, with many sensor nodes needing to communicate sensor readings or network events back to a central base station. As discussed in Section III, in-network processing such as aggregation, duplicate elimination, or data compression is needed to do this in an energy efficient manner. Since in-network processing requires intermediate nodes to access, modify, and possibly suppress the contents of messages, it is highly unlikely that end-to-end security mechanisms between a sensor node and a base station can be used to guarantee integrity, authenticity, and confidentiality of such messages[4].

In the presence of outsider adversaries, link layer security mechanisms can guarantee integrity, authenticity, and confidentiality of messages because they deny an outsider access to the network. However, we still must rely on the routing protocol to guarantee availability.

The presence of insiders significantly lessens the effectiveness of link layer security mechanisms. By definition, an insider is allowed to participate in the network. Link layer security can still prevent a compromised node from interfering with messages between other nodes, but such a node will have complete access to any messages routed through it and is free to modify, suppress, or eavesdrop on the contents. The conclusion then is that link layer security is not enough: since insiders may be able to exploit features in the routing protocol to violate the security goals, the routing protocol itself must be considered security critical.

In the presence of only outsider adversaries, it is conceivable to achieve these idealized goals. However, in the presence of compromised or insider attackers, especially those with laptop-class capabilities, it is most likely that some if not all of these goals are not fully attainable. Rather, instead of complete compromise of the entire network, the best we can hope for in the presence of insider adversaries is *graceful degradation*. The effectiveness of a routing protocol in achieving the above goals should degrade no faster than a rate approximately proportional to the ratio of compromised nodes to total nodes in the network.

Finally, in our view, protection against the replay of data packets should not be a security goal of a secure routing protocol. This functionality is best provided at the application layer because only the application can fully and accurately detect the replay of data packets (as opposed to retransmissions, for example).

[4]End-to-end security mechanisms are useful in sensor networks. We will see in Section VIII how end-to-end security can be used to help create more secure routing protocols. Also, end-to-end security can be used to protect messages after aggregation has been completed.

VI. Attacks on Sensor Network Routing

Many sensor network routing protocols are quite simple, and for this reason are sometimes susceptible to attacks from the literature on routing in ad-hoc networks. Most network layer attacks against sensor networks fall into one of the following categories:

- Spoofed, altered, or replayed routing information
- Selective forwarding
- Sinkhole attacks
- Sybil attacks
- Wormholes
- `HELLO` flood attacks
- Acknowledgement spoofing

In the descriptions below, note the difference between attacks that try to manipulate user data directly and attacks that try to affect the underlying routing topology.

We start with some general discussion of these types of attacks; in Section VII, we show how these attacks may be applied to compromise routing protocols that have been proposed in the literature.

A. Spoofed, altered, or replayed routing information

The most direct attack against a routing protocol is to target the routing information exchanged between nodes. By spoofing, altering, or replaying routing information, adversaries may be able to create routing loops, attract or repel network traffic, extend or shorten source routes, generate false error messages, partition the network, increase end-to-end latency, etc.

B. Selective forwarding

Multi-hop networks are often based on the assumption that participating nodes will faithfully forward received messages. In a selective forwarding attack, malicious nodes may refuse to forward certain messages and simply drop them, ensuring that they are not propagated any further. A simple form of this attack is when a malicious node behaves like a black hole and refuses to forward every packet she sees. However, such an attacker runs the risk that neighboring nodes will conclude that she has failed and decide to seek another route. A more subtle form of this attack is when an adversary selectively forwards packets. An adversary interested in suppressing or modifying packets originating from a select few nodes can reliably forward the remaining traffic and limit suspicion of her wrongdoing.

Selective forwarding attacks are typically most effective when the attacker is explicitly included on the path of a data flow. However, it is conceivable an adversary *overhearing* a flow passing through neighboring nodes might be able to emulate selective forwarding by jamming or causing a collision on each forwarded packet of interest. The mechanics of such an effort are tricky at best, and may border on impossible[5]. Thus, we believe an adversary launching a selective forwarding attack will likely follow the path of least resistance and attempt to include herself on the actual path of the data flow. In the next two sections, we discuss sinkhole attacks and the Sybil attack, two mechanisms by which an adversary can efficiently include herself on the path of the targeted data flow.

C. Sinkhole attacks

In a sinkhole attack, the adversary's goal is to lure nearly all the traffic from a particular area through a compromised node, creating a metaphorical sinkhole with the adversary at the center. Because nodes on, or near, the path that packets follow have many opportunities to tamper with application data, sinkhole attacks can enable many other attacks (selective forwarding, for example).

Sinkhole attacks typically work by making a compromised node look especially attractive to surrounding nodes with respect to the routing algorithm. For instance, an adversary could spoof or replay an advertisement for an extremely high quality route to a base station. Some protocols might actually try to verify the quality of route with end-to-end acknowledgements containing reliability or latency information. In this case, a laptop-class adversary with a powerful transmitter can actually *provide* a high quality route by transmitting with enough power to reach the base station in a single hop, or by using a wormhole attack discussed in Section VI-E. Due to either the real or imagined high quality route through the compromised node, it is likely each neighboring node of the adversary will forward packets destined for a base station through the adversary, and also propagate the attractiveness of the route to its neighbors. Effectively, the adversary creates a large "sphere of influence", attracting all traffic destined for a base station from nodes several (or more) hops away from the compromised node.

One motivation for mounting a sinkhole attack is that it makes selective forwarding trivial. By ensuring that all traffic in the targeted area flows through a compromised node, an adversary can selectively suppress or modify packets originating from any node in the area.

It should be noted that the reason sensor networks are particularly susceptible to sinkhole attacks is due to their specialized communication pattern. Since all packets share the same ultimate destination (in networks with only one base station), a compromised node needs only to provide a single high quality route to the base station in order to influence a potentially large number of nodes.

D. The Sybil attack

In a Sybil attack [2], a single node presents multiple identities to other nodes in the network. The Sybil attack can significantly reduce the effectiveness of fault-tolerant schemes such as distributed storage [28], dispersity [29] and multipath [30] routing, and topology maintenance [31], [32]. Replicas, storage partitions, or routes believed to be using disjoint nodes could in actuality be using a single adversary presenting multiple identities.

Sybil attacks also pose a significant threat to geographic routing protocols. Location aware routing often requires nodes to exchange coordinate information with their neighbors to efficiently route geographically addressed packets. It is only reasonable to expect a node to accept but a single set of coordinates from each of its neighbors, but by using the Sybil attack an adversary can "be in more than one place at once".

E. Wormholes

In the wormhole attack [1], an adversary tunnels messages received in one part of the network over a low latency link and replays them in a different part[6]. The simplest instance of this attack is a single node situated between two other nodes forwarding messages between the two of them. However, wormhole attacks more commonly involve two distant malicious nodes colluding to understate their distance from each other by relaying packets along an out-of-bound channel available only to the attacker.

An adversary situated close to a base station may be able to completely disrupt routing by creating a well-placed wormhole. An adversary could convince nodes who would normally be multiple hops from a base station that they are only one or two hops away via the wormhole. This can create a sinkhole: since the adversary on the other side of the wormhole can artificially provide a high-quality route to the base station, potentially all traffic in the surrounding area will be drawn through her if alternate routes are significantly less attractive. This will most likely always be the case when the endpoint of the wormhole is relatively far from a base station. Figure 6 shows an example of a wormhole being used to create a sinkhole.

More generally, wormholes can be used to exploit *routing race conditions*. A routing race condition typically arises when

[5] It may be extremely difficult for an adversary to launch such an attack in a network where every pair of neighboring nodes uses a unique key to initialize frequency hopping or spread spectrum communication, for example.

[6] Specifically, packets transmitted through the wormhole should have lower latency than those packets sent between the same pair of nodes over normal multi-hop routing.

a node takes some action based on the first instance of a message it receives and subsequently ignores later instances of that message. In this case, an adversary may be able to exert some influence on the resulting topology if it can cause a nodes to receive certain routing information before it would normally reach them though multi-hop routing. Wormholes are a way to do this, and are effective even if routing information is authenticated or encrypted. Wormholes can also be used simply to convince two distant nodes that they are neighbors by relaying packets between the two of them.

Wormhole attacks would likely be used in combination with selective forwarding or eavesdropping. Detection is potentially difficult when used in conjunction with the Sybil attack.

F. `HELLO` *flood attack*

We introduce a novel attack against sensor networks: the `HELLO` flood. Many protocols require nodes to broadcast `HELLO` packets to announce themselves to their neighbors, and a node receiving such a packet may assume that it is within (normal) radio range of the sender. This assumption may be false: a laptop-class attacker broadcasting routing or other information with large enough transmission power could convince every node in the network that the adversary is its neighbor.

For example, an adversary advertising a very high quality route to the base station to every node in the network could cause a large number of nodes to attempt to use this route, but those nodes sufficiently far away from the adversary would be sending packets into oblivion. The network is left in a state of confusion. A node realizing the link to the adversary is false could be left with few options: all its neighbors might be attempting to forward packets to the adversary as well. Protocols which depend on localized information exchange between neighboring nodes for topology maintenance or flow control are also subject to this attack.

An adversary does not necessarily need to be able to construct legitimate traffic in order to use the `HELLO` flood attack. She can simply re-broadcast overhead packets with enough power to be received by every node in the network. `HELLO` floods can also be thought of as one-way, broadcast wormholes.

Note: "Flooding" is usually used to denote the the epidemic-like propagation of a message to every node in the network over a multi-hop topology. In contrast, despite its name, the `HELLO` flood attack uses a single hop broadcast to transmit a message to a large number of receivers.

G. Acknowledgement spoofing

Several sensor network routing algorithms rely on implicit or explicit link layer acknowledgements. Due to the inherent broadcast medium, an adversary can spoof link layer acknowledgments for "overheard" packets addressed to neighboring nodes. Goals include convincing the sender that a weak link is strong or that a dead or disabled node is alive. For example, a routing protocol may select the next hop in a path using link reliability. Artificially reinforcing a weak or dead link is a subtle way of manipulating such a scheme. Since packets sent along weak or dead links are lost, an adversary can effectively mount a selective forwarding attack using acknowledgement spoofing by encouraging the target node to transmit packets on those links.

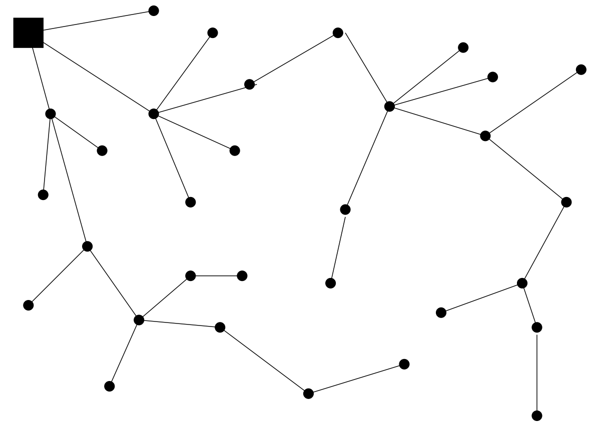

Fig. 4. A representative topology constructed using TinyOS beaconing with a single base station.

VII. Attacks on specific sensor network protocols

All of the proposed sensor network routing protocols are highly susceptible to attack. Adversaries can attract or repel traffic flows, increase latency, or disable the entire network with sometimes as little effort as sending a single packet. In this section, we survey the proposed sensor network routing protocols and highlight the relevant attacks.

A. TinyOS beaconing

The TinyOS beaconing protocol constructs a breadth first spanning tree rooted at a base station. Periodically the base station broadcasts a route update. All nodes receiving the update mark the base station as its parent and rebroadcast the update. The algorithm continues recursively with each node marking its parent as the first node from which it hears a routing update during the current *time epoch*. All packets received or generated by a node are forwarded to its parent (until they reach the base station).

Attacks: The TinyOS beaconing protocol is highly susceptible to attack. Since routing updates are not authenticated, it is possible for any node to claim to be a base station and become the destination of all traffic in the network (see Figure 5).

Authenticated routing updates will prevent an adversary from claiming to be a base station, but a powerful laptop-class adversary can still easily wreak havoc. An adversary interested in eavesdropping on, modifying, or suppressing packets in a particular area can do so by mounting a combined wormhole/sinkhole attack. The adversary first creates a wormhole between two colluding laptop-class nodes, one near the base station and one near the targeted area. The first node forwards (authenticated) routing updates to the second through

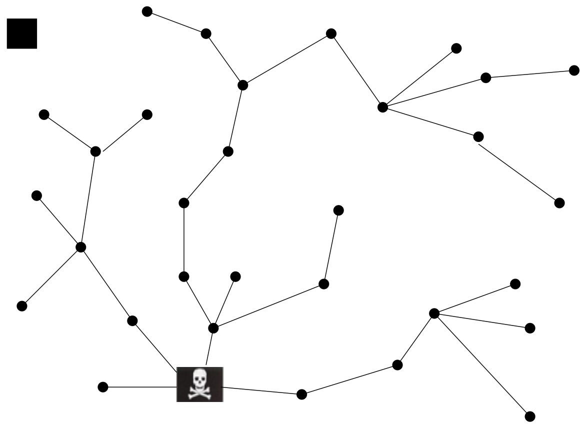

Fig. 5. An adversary spoofing a routing update from a base station in TinyOS beaconing.

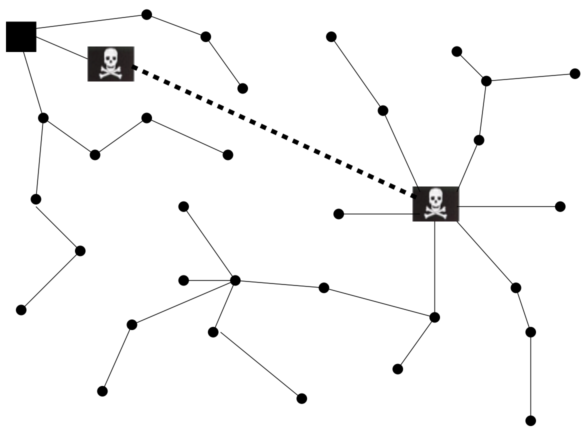

Fig. 6. A laptop-class adversary using a wormhole to create a sinkhole in TinyOS beaconing.

the wormhole, who participates normally in the protocol and rebroadcasts the routing update in the targeted area. Since the "wormholed" routing update will likely reach the targeted area considerably faster than it normally would have through multi-hop routing, the second node will create a large routing subtree in the targeted area with itself as the root. As seen in Figure 6, all traffic in the targeted area will be channeled through the wormhole, enabling a potent selective forwarding attack.

If a laptop-class adversary has a powerful transmitter, it can use a `HELLO` flood attack to broadcast a routing update loud enough to reach the entire network, causing every node to mark the adversary as its parent. Most nodes will be likely out of normal radio range of both a true base station and the adversary. As shown in Figure 7, the network is crippled: the majority of nodes are stranded, sending packets into oblivion. Due to the simplicity of this protocol, it is unlikely there exists a simple extension to recover from this attack. A node that realizes its parent is not actually in range (say by using link layer acknowledgements) has few options short of flooding every packet. Each of its neighbors will likely have the adversary marked as its parent as well.

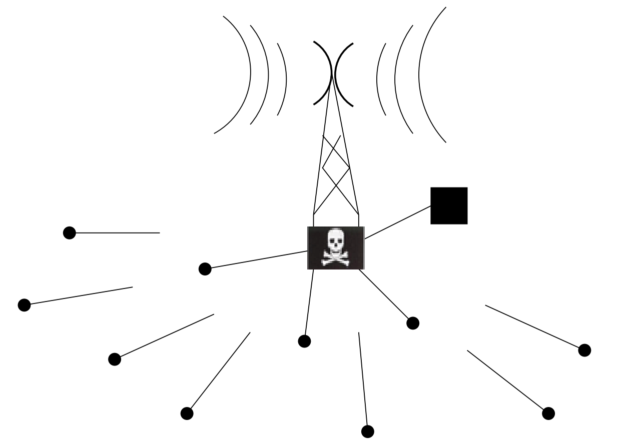

Fig. 7. `HELLO` **flood attack against TinyOS beaconing.** A laptop-class adversary that can retransmit a routing update with enough power to be received by the entire network leaves many nodes stranded. They are out of normal radio range from the adversary but have chosen her as their parent.

Routing loops can easily be created by mote-class adversaries spoofing routing updates. Suppose an adversary can determine that node A and node B are within radio range of each other. An adversary can send a forged routing update to node B with a spoofed source address indicating it came from node A. Node B will then mark node A as its parent and rebroadcast the routing update. Node A will then hear the routing update from node B and mark B as it is parent. Messages sent to either A or B will be forever forwarded in a loop between the two of them.

B. Directed diffusion

Directed diffusion [33] is a data-centric routing algorithm for drawing information out of a sensor network. Base stations flood interests for named data, setting up gradients within the network designed to draw events (i.e., data matching the interest). Nodes able to satisfy the interest disseminate information along the reverse path of interest propagation. Nodes receiving the same interest from multiple neighboring nodes may propagate events along the corresponding multiple links. Interests initially specify a low rate of data flow, but once a base station starts receiving events it will reinforce one (or more) neighbor in order to request higher data rate events. This process proceeds recursively until it reaches the nodes generating the events, causing them to generate events at a higher data rate. Alternatively, paths may be negatively reinforced as well.

There is a multipath variant of directed diffusion [34] as well. After the primary dataflow is established using positive reinforcements, alternate routes are recursively established with maximal disjointedness by attempting to reinforce neighbors not on the primary path.

Attacks: Due to the robust nature of flooding, it may be difficult for an adversary to prevent interests from reaching targets able to satisfy them. However, once sources begin to generate data events, an adversary attacking a data flow might have one of four goals:

Suppression: Flow suppression is an instance of denial-of-service. The easiest way to suppress a flow is to spoof negative reinforcements.

Cloning: Cloning a flow enables eavesdropping. After an adversary receives an interest flooded from a legitimate base station, it can simply replay that interest with herself listed as a base station. All events satisfying the interest will now be sent to both the adversary and the legitimate base station.

Path influence: An adversary can influence the path taken by a data flow by spoofing positive and negative reinforcements and bogus data events. For example, after receiving and rebroadcasting an interest, an adversary interested in directing the resulting flow of events through herself would strongly reinforce the nodes to which the interest was sent while spoofing high rate, low latency events to the nodes from which the interest was received. Three actions result: (1) data events generated upstream by legitimate sources will be drawn through the adversary because of her artificially strong positive reinforcements, (2) alternate event flows will be negatively reinforced by downstream nodes because the adversary provides (or spoofs) events with the lowest latency or highest frequency, and (3) the adversary's node will be positively reinforced due the high quality spoofed and real data events she is able to provide. With this attack, an adversary is able to ensure any flow of events propagates through herself on the way to the base station that originally advertised the associated interest.

Selective forwarding and data tampering: By using the above attack to insert herself onto the path taken by a flow of events, an adversary can gain full control of the flow. She can modify and selectively forward packets of her choosing.

A laptop-class adversary can exert greater influence on the topology by creating a wormhole between node A located next a base station and node B located close to where events are likely to be generated. Interests advertised by the base station are sent through the wormhole and rebroadcast by node B. Node B then attracts data flows by spoofing strong positive reinforcements to all neighboring nodes while node A broadcasts spoofed negative reinforcements to its surrounding nodes. The combination of the positive and negative reinforcements pushes data flows away from the base station and towards the resulting sinkhole centered at node B.

The multipath version may appear more robust against these attacks, but it is just as vulnerable. A single adversary can use the Sybil attack against her neighbors. A neighbor will be convinced it is maximizing diversity by reinforcing its next most preferred neighbor not on the primary flow when in fact this neighbor is an alternate identity of the adversary.

C. Geographic routing

Geographic and Energy Aware Routing (GEAR) [35] and Greedy Perimeter Stateless Routing (GPSR) [36] leverage nodes' positions and explicit geographic packet destinations to efficiently disseminate queries and route replies. GPSR uses greedy forwarding at each hop, routing each packet to the neighbor closest to the destination. When holes are encountered where greedy forwarding is impossible, GPSR recovers by routing around the perimeter of the void. One drawback of GPSR is that packets along a single flow will always use the same nodes for the routing of each packet, leading to uneven energy consumption. GEAR attempts to remedy this problem by weighting the choice of the next hop by both remaining energy and distance from the target. In this way, the responsibility for routing a flow is more evenly distributed among a "beam" of nodes between the source and base station. Both protocols require location (and energy for GEAR) information to be exchanged between neighbors, although for some fixed, well-structured topologies (a grid for example) this may not be necessary.

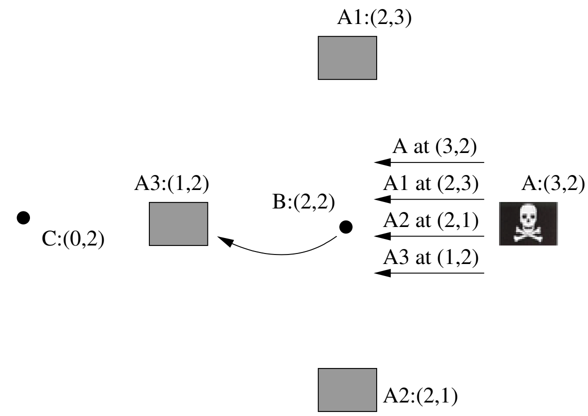

Fig. 8. **The Sybil attack against geographic routing.** Adversary A at actual location (3,2) forges location advertisements for non-existent nodes A1, A2, and A3 as well as advertising her own location. After hearing these advertisements, if B wants to send a message to destination (0,2), it will attempt to do so through A3. This transmission can be overheard and handled by the adversary A.

Attacks: Location information can be misrepresented. Regardless of an adversary's actual location, she may advertise her location in a way to place herself on the path of a known flow. GEAR tries to distribute the responsibility of routing based on remaining energy, so an appropriate attack would be to always advertise maximum energy as well.

Without too much additional effort, an adversary can dramatically increase her chances of success by mounting a Sybil attack. As depicted in Figure 8, an adversary can advertise multiple bogus nodes surrounding each target in a circle (or sphere), each claiming to have maximum energy. By intercepting transmissions sent to each of the bogus nodes, the adversary maximizes her chances for placing herself on the path of any nearby data flow. Once on that path, the adversary can mount a selective forwarding attack.

In GPSR an adversary can forge location advertisements to create routing loops in data flows without having to actively participate in packet forwarding. Consider the hypothetical

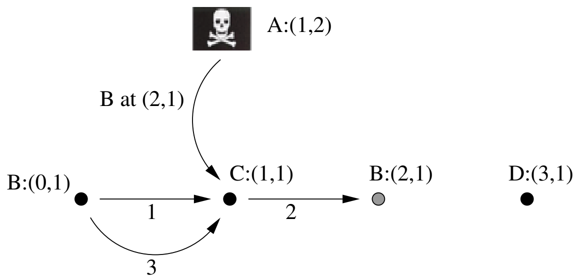

Fig. 9. **Creating routing loops in GPSR.** By forging a location advertisement claiming B is at (2,1), an adversary can create a routing loop as described in Section VII-C.

topology in Figure 9 and flow of packets from B to location (3,1). Assume the maximum radio range is one unit. If an adversary forges a location advertisement claiming B is at (2,1) and sends it to C, then after B forwards a packet destined for (3,1) to C, C will send it back to B because it believes B is close to the ultimate destination. B and C will forever forward the packet in a loop between each other.

D. Minimum cost forwarding

Minimum cost forwarding [37] is an algorithm for efficiently forwarding packets from sensor nodes to a base station with the useful property that it does not require nodes to maintain explicit path information or even unique node identifiers. It works by constructing a cost field starting at the base station. The base station has cost zero. Every other node maintains the minimum cost required to reach the base station. Cost can represent any application dependent metric: hop count, energy, latency, loss, etc.

Every node except the base station starts with cost ∞. Cost values are established by flooding a beacon starting from the base station. The beacon advertises the base station's cost (zero) and is propagated throughout the network. Upon hearing an advertisement from node M containing M's cost, node N now knows of a path of cost $C_M + L_{N,M}$. Node N compares its current cost C_N to $C_M + L_{N,M}$, where C_M is M's cost carried in the advertisement and $L_{N,M}$ is the cost of the link between N and M. If the new cost is smaller, then it sets $C_N = C_M + L_{N,M}$ and rebroadcasts an advertisement containing its new cost. In essence, this is a distributed shortest-paths algorithm.

As a node's cost converges to its minimum cost, the node will immediately send out a new advertisement every time its cost is updated. The authors present an optimization to the above algorithm which reduces the number of messages sent to establish the minimum cost field. After a node updates its cost, it delays rebroadcasting the advertisement containing its new cost for a time proportional to the link cost in the advertisement it received.

A message initiated by a source contains a cost budget initialized to the calculated minimum cost from the source to the base station. At each hop, the link cost of the hop is subtracted from the budget. The message is broadcast without specifying a specific next hop. A neighboring node hearing the message will forward the message only if the packet's remaining cost budget is equal to that node's own minimum cost. The authors also present a multipath version called *credit-based mesh forwarding* [38] which works by giving a message an extra amount of "credit" beyond the minimum cost of the source, enabling possibly multiple receivers to forward the message.

Attacks: Minimum cost forwarding is extremely susceptible to sinkhole attacks. A mote-class adversary can create a large sinkhole by simply advertising cost zero anywhere in the network. The optimization described above may cause confusion when a node receives a (spoofed) cost lower than what it had previously believed to be minimum. A laptop-class adversary can use a wormhole to help synchronize this attack with base station-initiated cost updates.

By using the `HELLO` flood attack, a laptop-class adversary can disable the entire network by transmitting an advertisement with cost zero powerful enough to be received by every node in the network. Assuming the adversary can force the link cost of this advertisement to be close to the average link cost between two neighboring nodes, it will likely minimize the cost of all nodes in the network. When a node broadcasts a future message destined for a base station, a neighboring node would be required to have a cost of nearly zero in order for it to take the responsibility for forwarding the message. This makes the adversary the sole "destination" of all messages from nodes within radio range and leaves nodes outside radio range "stranded".

E. LEACH: Low-Energy Adaptive Clustering Hierarchy

LEACH [39] leverages clustering to efficiently disseminate queries and gather sensor readings to and from all nodes in the network. LEACH assumes every node can directly reach a base station by transmitting with sufficiently high power. However, one hop transmission directly to a base station can be a high power operation and is especially inefficient considering the amount of redundancy typically found in sensor networks. LEACH organizes nodes into clusters with one node from each cluster serving as a *cluster-head*. Nodes first send sensor readings to their cluster-head, and the cluster-head aggregates or compresses the data from all its "children" for transmission to a base station. If cluster-head selection is static, those unlucky nodes chosen as cluster-heads would quickly run out of energy and die. LEACH uses randomized rotation of nodes required to be cluster-heads to evenly distribute energy consumption over all nodes in the network.

LEACH operation is broken into rounds, with each round having a set-up phase and a steady-state phase. In the beginning of the set-up phase, each node probabilistically decides whether or not to be a cluster head based on its remaining energy and a globally known desired percentage of cluster-heads. Each node electing itself as a cluster-head broadcasts an advertisement message announcing its intention. Non-cluster-head nodes receive possibly several advertisements and pick one cluster to join based on the largest received signal strength of the advertisement from the corresponding cluster-head. Nodes inform the cluster-head of the cluster they intend to join, and each cluster-head sends back a TDMA schedule for sending data to it for each node in its cluster. In the steady-state phase, each cluster-head waits to receive data from all nodes in its cluster and then sends the aggregated or compressed result back to a base station.

Attacks: Since nodes choose a cluster-head based on received signal strength, a laptop-class adversary can disable the entire network by using the HELLO flood attack to send a powerful advertisement to all nodes in the network. Due the large signal strength of the advertisement, every node is likely to choose the adversary as its cluster-head. The adversary can selectively forward those data transmissions that actually reach her, while the rest of the network is effectively disabled.

The adversary can use the same technique to mount a selective forwarding attack on the entire network using only a small number of nodes if the target number of cluster-heads or the size of the network is sufficiently small. Simple countermeasures such as refusing to use the same cluster-head in consecutive rounds or randomized selection of a cluster-head (rather than strongest received signal strength) can easily be defeated by a Sybil attack.

The authors also describe using LEACH to form hierarchical clusters. In this case, it is in the adversary's best interest to use the above techniques against the top-most layer of clustering. Other clustering protocols [40] and protocols optimizing or extending LEACH such as TEEN [41] and PEGASIS [42] are also susceptible to attacks similar to those described above.

F. Rumor routing

Rumor routing [43] is a probabilistic protocol for matching queries with data events. Flooding and gossiping [44] of events and/or queries throughout the network are robust mechanisms for doing this, but both have relatively high associated energy costs. However, flooding can be used to create a network-wide gradient field [33], [37], which is useful in routing frequent or numerous events or queries and amortizes the initial setup cost. Rumor routing offers a energy-efficient alternative when the high cost of flooding cannot be justified. Examples include posing a query on a very small cluster of nodes and advertising an observed event of possibly limited interest.

In rumor routing, when a source observes an event, it sends an agent on a random walk through the network. Agents carry a list of events, the next hop of paths to those events, the corresponding hop counts of those paths, a time to live (TTL) field, and a list of previously visited nodes and those nodes' neighbors (used to help "straighten" paths and eliminate loops). When an agent arrives at a new node, it informs that node of events it knows of (and the next hop on the path to those events), adds to its event list any events the node might know of, and decrements its TTL. If the TTL is greater than zero, the node probabilistically chooses the agent's next hop from its own neighbors minus the previously seen nodes listed in the agent. When a base station wants to disseminate a query, it creates an agent that propagates in a similar way. A route from a base station to a source is established when a query agent arrives at a node previously traversed by a event agent that satisfies the query.

Attacks: The establishment of routes is entirely dependent on nodes properly handling agents. An adversary can mount a denial-of-service attack by removing event information carried by the agent or by refusing to forward agents entirely. Query or event information in agents can also be modified.

Mote-class adversaries can mount a selective forwarding attack by extending tendrils in all directions like a jellyfish to create a sinkhole. An adversary creates tendrils by forwarding multiple copies of a received agent. The motivation for creating tendrils is this. The easiest way to mount a selective forwarding attack is to be on the path of the data flow. Thus, the intersection of the query and events agents must occur downstream from the adversary (towards the base station) at a node that one of the agents visited *after* the adversary. If the intersection occurs upstream of the adversary, she will be "cut out" of the path of data flow. An adversary can maximize the chances of this intersection occurring downstream from herself by creating many tendrils to "catch" query agents, i.e., by sending out multiple copies of a received agent. If these tendrils can cover a significant portion of the network, a query agent is more likely to intersect a downstream tendril than a node upstream from the adversary.

Regardless of how many tendrils an adversary creates, it is advantageous for them to be as long as possible and to advertise the shortest possible path to events of interest to the adversary. Thus, in the copies of the agent the adversary creates, the TTL field should be reset to maximum, the hop counts of paths to interesting events should be reset to zero, but unlike in the routing loop attack, the recently visited node list should remain unchanged.

Resetting the TTL field will clearly maximize the length of the tendrils, but the reason for zeroing the hop counts of paths to interesting events while maintaining the recently visited list in each agent may be non-obvious. If the adversary zeros the hop count of known paths to interesting events carried in the agent, it is very likely a node receiving the agent that already knows of a path to an event carried by the agent will now choose to use the new path since the adversary has artificially made it appear to be shorter. However, an adversary does not want *all* nodes to use this new path. The nodes that the agent traversed from the event source to adversary must *not* update their path. The adversary is relying on those nodes to forward events to her, and if those nodes were to use the artificially short path created by the adversary, a loop would be created. By including this list in each outgoing agent, the adversary assures that each agent will not be forwarded to one of these upstream nodes.

What then is motivation for resetting the hop counts at all? It is possible for other agents to intersect the agent path upstream from the adversary and carry information regarding those events throughout the network. It is these nodes that an adversary wants to "turn" and cause them to choose a new path through the adversary for those interesting events. A good metaphor is a river with tributaries. The adversary relies on the river for events to flow downstream to her from the source, but tributaries branching off the river (i.e., other agents that intersected the agent's path upstream) can be re-routed through the adversary without effecting the main flow.

The above attack is subtle and complicated, but a laptop-class adversary can make things easier by creating a wormhole between a node near a potential source and a node near a base station, and then using the Sybil attack to maximize each nodes' chance of being chosen as the initial destination of a event or query agent. Queries are immediately matched with events via the wormhole, and the adversary can then selectively forward events of her choosing.

G. Energy conserving topology maintenance

Sensor networks may be deployed in hard to reach areas and be meant to run unattended on long periods of time. It may be difficult to replace the batteries on energy-depleted nodes or even add new ones. A viable solution in such contexts is to initially deploy more sensors than needed, and make use of the additional nodes to extend network lifetime. SPAN [32] and GAF [31] adaptively decide which nodes are required to be active in order to maintain an acceptable level of routing fidelity while allowing the remaining nodes to turn off their radios and sleep [7].

1) GAF: GAF [31] places nodes into virtual "grid squares" according to geographic location and expected radio range. Any pair of nodes in adjacent grid squares are able to communicate. Nodes are in one of three states: *sleeping*, *discovery*, and *active*. Active nodes participate in routing while discovery nodes probe the network to determine if their presence is needed. Sleeping nodes have their radio turned off. Nodes are ranked with respect to current state and expected lifetime. Discovery messages are used to exchange state and ranking information between nodes in the same grid. GAF attempts to reach a state in which each grid square has only one active node.

Attacks: Nodes in the discovery or active state that receive a discovery message from a higher ranking node will transition to *sleeping*, and after some period of time will wake up and transition back to *discovery*. An adversary can easily disable other nodes (i.e., ensure they are sleeping) in her grid by periodically broadcasting high ranking discovery messages. The adversary can then mount an selective forwarding attack or choose to ignore incoming packets completely. It is also possible for a laptop-class adversary with a loud transmitter to disable the entire network. Using the Sybil attack and a `HELLO` flood attack, the attacker can target individual grids by broadcasting a high ranking discovery message from a bogus, non-existent node in each grid. Done frequently enough, the adversary can ensure the entire network remains sleeping.

2) SPAN: In SPAN [32], nodes decide whether to sleep or join a backbone of "coordinators" that attempt to maintain routing fidelity in the network. Coordinators stay awake continuously while the remaining nodes go into "power saving" mode and periodically send and receive `HELLO` messages to determine if they should become a coordinator. In a `HELLO` message, a node announces its current status (coordinator or not), its current coordinators, and its current neighbors. A node's current coordinators are those neighbors which are coordinators.

A node becomes eligible to become a coordinator if two of its neighbors cannot reach other directly or via one or two coordinators. In order to prevent broadcast storms if multiple nodes discover the need of a coordinator and were simultaneously to announce their intention to become one, each node delays its announcement of becoming a coordinator by a randomized backoff. While in the backoff stage, it continues to listen for additional `HELLO` messages and coordinator announcements. If at the end of the backoff stage, the coordinator eligibility condition still holds, the candidate node announces its intention to become a coordinator. The randomized backoff is a function of *utility* and remaining energy. *Utility* is a measure of the number of pairs of nodes (among a node's neighbors) that would become connected if that node were to become a coordinator. A node with high utility and energy is more likely to calculate a shorter backoff time. Nodes eventually withdraw from being a coordinator for two reasons: (1) the eligibility requirement no longer holds, or (2) in order to ensure fairness, after some time a node will withdraw from being a coordinator if it discovers every pair of neighboring nodes can reach each other through some other neighbor. A node will then announce its intention to withdraw, but will continue to forward packets for a short period of time until a new coordinator is elected.

Attacks: A laptop-class adversary may attempt to disrupt routing in the network by preventing nodes from becoming coordinators when they should. An attack to cripple routing in the entire network works as follows: First, the adversary partitions the targeted area into cells $C_1, C_2, \ldots C_n$ of reasonable size [8]. For each cell C_i, the adversary chooses a bogus coordinator node id ID_i. The adversary broadcasts n `HELLO` messages with enough transmit power to be heard by every node in the network announcing that ID_i ($i = 1$ to n) is a coordinator and has neighbors $\{C_{i1}, C_{i2}, \ldots, C_{ik_i}, ID_1, ID_2, \ldots, ID_n\}$ where $C_{i1}, C_{i2}, \ldots, C_{ik_i}$ are the nodes in cell C_i. Every node in cell C_i believes 1) it has $ID_1, ID_2, \ldots, ID_n$ as neighbors, and 2) it can "reach" each of its real and bogus neighbors through ID_i. Each bogus coordinator must declare $ID_1, ID_2, \ldots, ID_n$ as its neighbors otherwise a real node will become a coordinator to create connectivity between them. The adversary has effectively disabled the entire network since no real nodes are actively participating in routing. To enable a selective forwarding attack, an adversary (possibly even mote-

[7] SPAN and GAF were originally proposed for more general ad-hoc networks, but are applicable to sensor networks as well.

[8] "Reasonable size" should be around the maximum number of neighbors any one node can be expected to have without causing alarm.

class) can scale down this attack to ensure it is the sole coordinator actively engaged in routing for a smaller area.

Cluster-based Energy Conservation (CEC) [45] and the Adaptive Fidelity Energy-Conserving Algorithm (AFECA) [46] are two other proposed energy conserving topology management algorithms. CEC creates clusters and selects cluster-heads based on the highest advertised remaining energy. Networks using CEC can be disabled by a `HELLO` flood attack similar to that one described against GAF. AFECA allows each node to sleep for randomized periods based on the number of (overheard) neighbors it has. A node using AFECA can be made to sleep for abnormally long periods of times by using the Sybil and `HELLO` flood attack to inflate the number of perceived neighbors.

VIII. Countermeasures

A. *Outsider attacks and link layer security*

The majority of outsider attacks against sensor network routing protocols can be prevented by simple link layer encryption and authentication using a globally shared key. The Sybil attack is no longer relevant because nodes are unwilling to accept even a single identity of the adversary. The majority of selective forwarding and sinkhole attacks are not possible because the adversary is prevented from joining the topology. Link layer acknowledgements can now be authenticated.

Major classes of attacks not countered by link layer encryption and authentication mechanisms are wormhole attacks and `HELLO` flood attacks. Although an adversary is prevented from joining the network, nothing prevents her from using a wormhole to tunnel packets sent by legitimate nodes in one part of the network to legitimate nodes in another part to convince them they are neighbors or by amplifying an overheard broadcast packet with sufficient power to be received by every node in the network.

The attacks against TinyOS beaconing described in Section VII-A illustrate these techniques, and link layer security mechanisms can do nothing to prevent them. If a wormhole has been established, encryption may make some selective forwarding attacks against packets using the wormhole more difficult, but clearly can do nothing to prevent "black hole" selective forwarding.

Link layer security mechanisms using a globally shared key are completely ineffective in presence of insider attacks or compromised nodes. Insiders can attack the network by spoofing or injecting bogus routing information, creating sinkholes, selectively forwarding packets, using the Sybil attack, and broadcasting `HELLO` floods. More sophisticated defense mechanisms are needed to provide reasonable protection against wormholes and insider attacks. We focus on countermeasures against these attacks in the remaining sections.

B. *The Sybil attack*

An insider cannot be prevented from participating in the network, but she should only be able to do so using the identities of the nodes she has compromised. Using a globally shared key allows an insider to masquerade as *any* (possibly even non-existent) node. Identities must be verified. In the traditional setting, this might be done using public key cryptography, but generating and verifying digital signatures is beyond the capabilities of sensor nodes.

One solution is to have every node share a unique symmetric key with a trusted base station. Two nodes can then use a Needham-Schroeder like protocol to verify each other's identity and establish a shared key. A pair of neighboring nodes can use the resulting key to implement an authenticated, encrypted link between them. In order to prevent an insider from wandering around a stationary network and establishing shared keys with every node in the network, the base station can reasonably limit the number of neighbors a node is allowed to have and send an error message when a node exceeds it.

Thus, when a node is compromised, it is restricted to (meaningfully) communicating only with its verified neighbors. This is not to say that nodes are forbidden from sending messages to base stations or aggregation points multiple hops away, but they are restricted from using any node except their verified neighbors to do so. In addition, an adversary can still use a wormhole to create an artificial link between two nodes to convince them they are neighbors, but the adversary will not be able to eavesdrop on or modify any future communications between them.

C. `HELLO` *flood attacks*

The simplest defense against `HELLO` flood attacks is to verify the bidirectionality of a link before taking meaningful action based on a message received over that link. However, this countermeasure is less effective when an adversary has a highly sensitive receiver as well as a powerful transmitter. Such an adversary can effectively create a wormhole to every node within range of its transmitter/receiver. Since the links between these nodes and the adversary are bidirectional, the above approach will unlikely be able to locally detect or prevent a `HELLO` flood.

One possible solution to this problem is for every node to authenticate each of its neighbors with an identity verification protocol (Section VIII-B) using a trusted base station. If the protocol sends messages in both directions over the link between the nodes, `HELLO` floods are prevented when the adversary only has a powerful transmitter because the protocol verifies the bidirectionality of the link. Although this does not prevent a compromised node with a sensitive receiver and a powerful transmitter from authenticating itself to a large number of nodes in the network, an observant base station may be able to detect a `HELLO` flood is imminent. Since such an adversary is required to authenticate itself to every victim before it can mount an attack, an adversary claiming to be a neighbor of an unusually large number of the nodes will raise an alarm.

D. *Wormhole and sinkhole attacks*

Wormhole and sinkhole attacks are very difficult to defend against, especially when the two are used in combination.

Wormholes are hard to detect because they use a private, out-of-band channel invisible to the underlying sensor network. Sinkholes are difficult to defend against in protocols that use advertised information such as remaining energy or an estimate of end-to-end reliability to construct a routing topology because this information is hard to verify. Routes that minimize the hop-count to a base station are easier to verify, however hop-count can be completely misrepresented through a wormhole. When routes are established simply based on the reception of a packet as in TinyOS beaconing or directed diffusion, sinkholes are easy to create because there is no information for a defender to verify.

A technique for detecting wormhole attacks is presented in [1], but it requires extremely tight time synchronization and is thus infeasible for most sensor networks. Because it is extremely difficult to retrofit existing protocols with defenses against these attacks, the best solution is to carefully design routing protocols which avoid routing race conditions and make these attacks less meaningful.

For example, one class of protocols resistant to these attacks is geographic routing protocols. Protocols that construct a topology initiated by a base station are most susceptible to wormhole and sinkhole attacks. Geographic protocols construct a topology on demand using only localized interactions and information and without initiation from the base station. Because traffic is naturally routed towards the physical location of a base station, it is difficult to attract it elsewhere to create a sinkhole. A wormhole is most effective when used to create sinkholes or artificial links that attract traffic. Artificial links are easily detected in geographic routing protocols because the "neighboring" nodes will notice the distance between them is well beyond normal radio range.

E. Leveraging global knowledge

A significant challenge in securing large sensor networks is their inherent self-organizing, decentralized nature. When the network size is limited or the topology is well-structured or controlled, global knowledge can be leveraged in security mechanisms.

Consider a relatively small network of around 100 nodes or less. If it can be assumed that no nodes are compromised during deployment, then after the initial topology is formed, each node could send information such as neighboring nodes and its geographic location (if known) back to a base station. Using this information, the base station(s) can map the topology of the entire network. To account for topology changes due to radio interference or node failure, nodes would periodically update a base station with the appropriate information. Drastic or suspicious changes to the topology might indicate a node compromise, and the appropriate action can be taken.

We have discussed why geographic routing can be relatively secure against wormhole, sinkhole, and Sybil attacks, but the main remaining problem is that location information advertised from neighboring nodes must be trusted. A compromised node advertising its location on a line between the targeted node and a base station will guarantee it is the destination for all forwarded packets from that node. Probabilistic selection of a next hop from several acceptable destinations or multipath routing to multiple base stations can help with this problem, but it is not perfect. When a node must route around a "hole", an adversary can "help" by appearing to be the only reasonable node to forward packets to.

Sufficiently restricting the structure of the topology can eliminate the requirement for nodes to advertise their locations if all nodes' locations are well known. For example, nodes can be arranged in a grid with square, triangular, or hex shaped cells. Every node can easily derive its neighbors' locations from its own, and nodes can be addressed by location rather than by an identifier.

F. Selective forwarding

Even in protocols completely resistant to sinkholes, wormholes, and the Sybil attack, a compromised node has a significant probability of including itself on a data flow to launch a selective forwarding attack if it is strategically located near the source or a base station.

Multipath routing can be used to counter these types of selective forwarding attacks. Messages routed over n paths whose nodes are completely disjoint are completely protected against selective forwarding attacks involving at most n compromised nodes and still offer some probabilistic protection when over n nodes are compromised. However, completely disjoint paths may be difficult to create. Braided paths [34] may have nodes in common, but have no links in common (i.e., no two consecutive nodes in common). The use of multiple braided paths may provide probabilistic protection against selective forwarding and use only localized information. Allowing nodes to dynamically choose a packet's next hop probabilistically from a set of possible candidates can further reduce the chances of an adversary gaining complete control of a data flow.

G. Authenticated broadcast and flooding

Since base stations are trustworthy, adversaries must not be able to spoof broadcast or flooded messages from any base station. This requires some level of asymmetry: since every node in the network can potentially be compromised, no node should be able to spoof messages from a base station, yet every node should be able to verify them. Authenticated broadcast is also useful for localized node interactions. Many protocols require nodes to broadcast `HELLO` messages to their neighbors. These messages should be authenticated and impossible to spoof.

Proposals for authenticated broadcast intended for use in a more conventional setting either use digital signatures and/or have packet overhead that well exceed the length of typical sensor network packet. μTESLA [23] is a protocol for efficient, authenticated broadcast and flooding that uses only symmetric key cryptography and requires minimal packet overhead. μTESLA achieves the asymmetry necessary for authenticated broadcast and flooding by using delayed key disclosure and one-way key chains constructed with a publicly computable cryptographically secure hash function. Replay is prevented because messages authenticated with previously disclosed keys are ignored. μTESLA also requires loose time synchronization.

Flooding [47] can be a robust means for information dissemination in hostile environments because it requires the set of compromised nodes to form a vertex cut on the underlying topology to prevent a message from reaching every node in the network. The downsides of flooding include high messaging and corresponding energy costs, as well as potential losses caused by collisions. SPIN [48] and gossiping algorithms [44] are techniques to reduce the messaging costs and collisions which still achieve robust probabilistic dissemination of messages to every node in the network.

H. Countermeasure summary

Link-layer encryption and authentication, multipath routing, identity verification, bidirectional link verification, and authenticated broadcast can protect sensor network routing protocols against outsiders, bogus routing information, Sybil attacks, `HELLO` floods, and acknowledgement spoofing, and it is feasible to augment existing protocols with these mechanisms.

Sinkhole attacks and wormholes pose significant challenges to secure routing protocol design, and it is unlikely there exists effective countermeasures against these attacks that can be applied after the design of a protocol has completed. It is crucial to design routing protocols in which these attacks are meaningless or ineffective. Geographic routing protocols are one class of protocols that holds promise.

IX. ULTIMATE LIMITATIONS OF SECURE MULTI-HOP ROUTING

An ultimate limitation of building a multi-hop routing topology around a fixed set of base stations is that those nodes within one or two hops of the base stations are particularly attractive for compromise. After a significant number of these nodes have been compromised, all is lost.

This indicates that clustering protocols like LEACH where cluster-heads communicate directly with a base station may ultimately yield the most secure solutions against node compromise and insider attacks.

Another option may be to have a randomly rotating set of “virtual” base stations to create an overlay network. After a set of virtual base stations have been selected, a multi-hop topology is constructed using them. The virtual base stations then communicate directly with the real base stations. The set of virtual base stations should be changed frequently enough to make it difficult for adversaries to choose the “right” nodes to compromise.

X. CONCLUSION

Secure routing is vital to the acceptance and use of sensor networks for many applications, but we have demonstrated that currently proposed routing protocols for these networks are insecure. We leave it as an open problem to design a sensor network routing protocol that satisfies our proposed security goals. Link layer encryption and authentication mechanisms may be a reasonable first approximation for defense against mote-class outsiders, but cryptography alone is not enough. The possible presence of laptop-class adversaries and insiders and the limited applicability of end-to-end security mechanisms necessitates careful protocol design as well.

XI. ACKNOWLEDGMENTS

We gratefully acknowledge DARPA NEST contract F33615-01-C-1895 for supporting this work.

REFERENCES

[1] Y.-C. Hu, A. Perrig, and D. B. Johnson, “Wormhole detection in wireless ad hoc networks,” Department of Computer Science, Rice University, Tech. Rep. TR01-384, June 2002.

[2] J. R. Douceur, “The Sybil Attack,” in *1st International Workshop on Peer-to-Peer Systems (IPTPS ’02)*, March 2002.

[3] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. Culler, and K. Pister, “System architecture directions for networked sensors,” in *Proceedings of ACM ASPLOS IX*, November 2000.

[4] V. D. Park and M. S. Corson, “A highly adaptive distributed routing algorithm for mobile wireless networks,” in *IEEE INFOCOM ’97*, 1997, pp. 1405–1413.

[5] C. Perkins and E. Royer, “Ad-hoc on-demand distance vector routing,” in *MILCOM ’97 panel on Ad Hoc Networks*, 1997.

[6] D. B. Johnson and D. A. Maltz, “Dynamic source routing in ad hoc wireless networks,” in *Mobile Computing*, Imielinski and Korth, Eds. Kluwer Academic Publishers, 1996, vol. 353.

[7] C. Perkins and P. Bhagwat, “Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers,” in *ACM/SIGCOMM’94 Conference on Communications Architectures, Protocols and Applications*, 1994, pp. 234–244.

[8] L. Zhou and Z. Haas, “Securing ad hoc networks,” *IEEE Network Magazine*, vol. 13, no. 6, November/December 1999.

[9] F. Stajano and R. J. Anderson, “The resurrecting duckling: Security issues for ad-hoc wireless networks,” in *Seventh International Security Protocols Workshop*, 1999, pp. 172–194.

[10] J. Hubaux, L. Buttyan, and S. Capkun, “The quest for security in mobile ad hoc networks,” in *Proceedings of the ACM Symposium on Mobile Ad Hoc Networking and Computing (MobiHOC 2001)*, 2001.

[11] J. Kong, P. Zerfos, H. Luo, S. Lu, and L. Zhang, “Providing robust and ubiquitous security support for mobile ad-hoc networks,” in *ICNP*, 2001, pp. 251–260.

[12] M. G. Zapata, “Secure ad-hoc on-demand distance vector (SAODV) routing,” IETF MANET Mailing List, Message-ID: 3BC17B40.BBF52E09@nokia.com, Available at ftp://manet.itd.nrl.navy.mil/pub/manet/2001-10.mail, October 8, 2001.

[13] H. Luo, P. Zefros, J. Kong, S. Lu, and L. Zhang, “Self-securing ad hoc wireless networks,” in *Seventh IEEE Symposium on Computers and Communications (ISCC ’02)*, 2002.

[14] J. Binkley and W. Trost, “Authenticated ad hoc routing at the link layer for mobile systems,” *Wireless Networks*, vol. 7, no. 2, pp. 139–145, 2001.

[15] B. Dahill, B. N. Levine, E. Royer, and C. Shields, “A secure routing protocol for ad-hoc networks,” Electrical Engineering and Computer Science, University of Michigan, Tech. Rep. UM-CS-2001-037, August 2001.

[16] J. Kong, H. Luo, K. Xu, D. L. Gu, M. Gerla, and S. Lu, “Adaptive security for multi-layer ad-hoc networks,” *Special Issue of Wireless Communications and Mobile Computing, Wiley Interscience Press*, 2002.

[17] Y.-C. Hu, D. B. Johnson, and A. Perrig, “SEAD: Secure efficient distance vector routing for mobile wireless ad hoc networks,” in *Proceedings of the 4th IEEE Workshop on Mobile Computing Systems and Applications (WMCSA 2002)*, June 2002, pp. 3–13.

[18] Y.-C. Hu, A. Perrig, and D. B. Johnson, “Ariadne: A secure on-demand routing protocol for ad hoc networks,” Department of Computer Science, Rice University, Tech. Rep. TR01-383, December 2001.

[19] S. Basagni, K. Herrin, E. Rosti, and D. Bruschi, “Secure pebblenets,” in *ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc 2001)*, October 2001, pp. 156–163.

[20] P. Papadimitratos and Z. Haas, "Secure routing for mobile ad hoc networks," in *SCS Communication Networks and Distributed Systems Modeling and Simulation Conference (CNDS 2002)*, January 2002.

[21] S. Marti, T. J. Giuli, K. Lai, and M. Baker, "Mitigating routing misbehavior in mobile ad hoc networks," in *Sixth annual ACM/IEEE Internation Conference on Mobile Computing and Networking*, 2000, pp. 255–265.

[22] S. Buchegger and J.-Y. L. Boudec, "Nodes bearing grudges: Towards routing security, fairness, and robustness in mobile ad hoc networks," in *Proceedings of the Tenth Euromicro Workshop on Parallel, Distributed and Network-based Processing*. Canary Islands, Spain: IEEE Computer Society, January 2002, pp. 403–410.

[23] A. Perrig, R. Szewczyk, V. Wen, D. Culler, and J. Tygar, "SPINS: Security protocols for sensor networks," in *Proceedings of Mobile Networking and Computing 2001*, 2001.

[24] A. Demers, S. Shenker, V. Bhargavan, and L. Zhang, "Macaw: A media access protocol for wireless lans," in *ACM SigComm '94*, 1994.

[25] R. L. Pickholtz, D. L. Schilling, and L. B. Milstein, "Theory of spread spectrum communications - a tutorial," *IEEE Transactions on Communications*, vol. 20, no. 5, pp. 855–884, May 1982.

[26] N. Abramson, "The ALOHA system - another alternative for computer communications," in *Proceedings of the Fall 1970 AFIPS Computer Conference*, November 1970, pp. 281–285.

[27] M.Blum and S. Micali, "A simple unpredictable pseudo-random number generator," *SIAM J. Computing*, vol. 15, no. 2, pp. 364–383, May 1986.

[28] Castro and Liskov, "Practical byzantine fault tolerance," in *OSDI: Symposium on Operating Systems Design and Implementation*. USENIX Association, Co-sponsored by IEEE TCOS and ACM SIGOPS, 1999.

[29] A. Banerjea, "A taxonomy of dispersity routing schemes for fault tolerant real-time channels," in *Proceedings of ECMAST*, vol. 26, May 1996, pp. 129–148.

[30] K. Ishida, Y. Kakuda, and T. Kikuno, "A routing protocol for finding two node-disjoint paths in computer networks," in *Internation Conference on Network Protocols*, November 1992, pp. 340–347.

[31] Y. Xu, J. Heidemann, and D. Estrin, "Geography-informed energy conservation for ad hoc routing," in *Proceedings of the Seventh Annual ACM/IEEE International Conference on Mobile Computing and Networking*, 2001.

[32] B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris, "Span: An energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks," *ACM Wireless Networks Journal*, vol. 8, no. 5, September 2002.

[33] C. Intanagonwiwat, R. Govindan, and D. Estrin, "Directed diffusion: A scalable and robust communication paradigm for sensor networks," in *Proceedings of the Sixth Annual International Conference on Mobile Computing and Networks (MobiCOM '00)*, August 2000.

[34] D. Ganesan, R. Govindan, S. Shenker, and D. Estrin, "Highly-resilient, energy-efficient multipath routing in wireless sensor networks," *Mobile Computing and Communications Review*, vol. 4, no. 5, October 2001.

[35] Y. Yu, R. Govindan, and D. Estrin, "Geographical and energy aware routing: A recursive data dissemination protocol for wireless sensor networks," University of California at Los Angeles Computer Science Department, Tech. Rep. UCLA/CSD-TR-01-0023, May 2001.

[36] B. Karp and H. T. Kung, "GPSR: greedy perimeter stateless routing for wireless networks," in *Mobile Computing and Networking*, 2000, pp. 243–254.

[37] F. Ye, A. Chen, S. Lu, and L. Zhang, "A scalable solution to mininum cost forwarding in large sensor networks," in *Tenth Internation Conference on Computer Communications and Networks*, 2001, pp. 304–309.

[38] F. Ye, S. Lu, and L. Zhang, "GRAdient broadcast: A robust, long-live large sensor network," University of California at Los Angeles Computer Science Department, Tech. Rep., 2001.

[39] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in *33rd Annual Hawaii International Conference on System Sciences*, 2000, pp. 3005–3014.

[40] D. Estrin, R. Govindan, J. Heidemann, and S. Kumar, "Next century challenges: Scalable coordination in sensor networks," in *5th annual ACM/IEEE International Conference on Mobile Computing and Networking*, 1999, pp. 263–270.

[41] A. Manjeshwar and D. Agrawal, "TEEN: A routing protocol for enhanced effeciency in wireless sensor networks," in *1st International Workshop on Parallel and Distributed Computing Issues in Wireless Networks and Mobile Computing*, 2001.

[42] S. Lindsey and C. Raghavendra, "PEGASIS: Power-efficient gathering in sensor information systems," in *IEEE Aerospace Conference*, March 2002.

[43] D. Braginsky and D. Estrin, "Rumour routing algorithm for sensor networks," in *First ACM International Workshop on Wireless Sensor Networks and Applications*, 2002.

[44] L. Li, J. Halpern, and Z. Haas, "Gossip-based ad hoc routing," in *IEEE Infocom 2002*, 2002.

[45] Y. Xu, J. Heidemann, and D. Estrin, "Energy conservation by adaptive clustering for ad-hoc networks," in *Poster Session of MobiHoc 2002*, 2002.

[46] Y. X. et al., "Adaptive energy-conserving routing for multihop ad hoc networks," USC/ISI, Research Report TR-2000-527, October 2000.

[47] D. Ganesan, B. Krishnamachari, A. Woo, D. Culler, D. Estrin, and S. Wicker, "An empirical study of epidemic algorithms in large scale multihop wireless networks," Intel Research, Tech. Rep. IRB-TR-02-003, March 2002.

[48] J. Kulik, W. R. Heinzelman, and H. Balakrishnan, "Negotiation-based protocols for disseminating information in wireless sensor networks," *Wireless Networks*, vol. 8, no. 2-3, pp. 169–185, 2002.