CERIAS Tech Report 2004-119
Visualization of Wormholes in Sensor Networks
by W Wang, B Bhargava
Center for Education and Research
Information Assurance and Security
Purdue University, West Lafayette, IN 47907-2086

Visualization of Wormholes in Sensor Networks

Weichao Wang
wangwc@cs.purdue.edu

Bharat Bhargava
bb@cs.purdue.edu

CERIAS and Department of Computer Sciences
Purdue University

ABSTRACT

Several protocols have been proposed to defend against wormholes in ad hoc networks by adopting positioning devices, synchronized clocks, or directional antennas. In this paper, we propose a mechanism, MDS-VOW, to detect wormholes in a sensor network. MDS-VOW first reconstructs the layout of the sensors using multi-dimensional scaling. To compensate the distortions caused by distance measurement errors, a surface smoothing scheme is adopted. MDS-VOW then detects the wormhole by visualizing the anomalies introduced by the attack. The anomalies, which are caused by the fake connections through the wormhole, bend the reconstructed surface to pull the sensors that are faraway to each other. Through detecting the bending feature, the wormhole is located and the fake connections are identified. The contributions of MDS-VOW are: (1) it does not require the sensors to be equipped with special hardware, (2) it adopts and combines the techniques from social science, computer graphics, and scientific visualization to attack the problem in network security. We examine the accuracy of the proposed mechanism when the sensors are deployed in a circle area and one wormhole exists in the network. The results show that MDS-VOW has a low false alarm ratio when the distance measurement errors are not large.

Categories and Subject Descriptors

C.2.0 [**Computer Systems Organization**]: Computer-Communication Networks – *Security and protection*; I.2.9 [**Computing Methodologies**]: Robotics – *Sensors*

General Terms

Security

Keywords

Visualization, Sensor Networks, Wormhole Attacks, Multi-Dimensional Scaling

WiSe'04, October 1, 2004, Philadelphia, Pennsylvania, USA.

1. INTRODUCTION

As sensor networks are merging into the pervasive computing environment, security becomes a central requirement. In this paper, we focus on the detection of wormhole attacks in sensor networks. Since the sensors use a radio channel to send information, the malicious nodes can eavesdrop the packets, tunnel them to another location in the network, and retransmit them. This generates a false scenario that the original sender is in the neighborhood of the remote location. The tunneling procedure forms a wormhole. If a fast transmission path (e.g. a dedicated channel shared by attackers) exists between the two ends of the wormhole, the tunneled packets can propagate faster than those through a normal multi-hop route. This forms the "rushing attack" studied by Hu *et al* [14].

Wormhole attacks put severe threats to both routing protocols and some security enhancements in sensor networks. For example, the sensors may depend on the neighbor discovery procedures to construct the local network topology. If the neighbor discovery beacons are tunneled through wormholes, the good nodes will get false information about their neighbors. This may lead to the choice of a non-existent route. The impacts of a wormhole on the route discovery procedure in a sensor network have been studied in [12]. Similar condition will happen if distributed monitoring is applied to detect network misbehaviours. The wormhole can selectively tunnel the normal packets sent by the security violator to the remote side, but not the packets that will expose the violator. Therefore, the real neighbors of the violator and the sensors at the remote side will get opposite conclusions on the node.

Research efforts have been put on wormhole detection in ad hoc networks and encouraging results have been collected [13, 4, 12]. These methods usually require the mobile nodes to be equipped with some special hardware, such as positioning devices, synchronized clocks, or directional antennas. With the progresses in integrated circuit design and hardware manufacture, these devices will become cheap, small, and power efficient enough to fit in sensors in the future.

In this paper, we present a more hardware-efficient approach to defend against wormholes in sensor networks. The mechanism, MDS-VOW (Multi-Dimensional Scaling - Visualization Of Wormhole), does not require the sensors to be equipped with special hardware. It reconstructs the network using multi-dimensional scaling and detects the wormhole by visualizing the anomalies introduced by the attack. Before presenting the details of the mechanism, we use three examples to illustrate the impacts of the wormhole on the

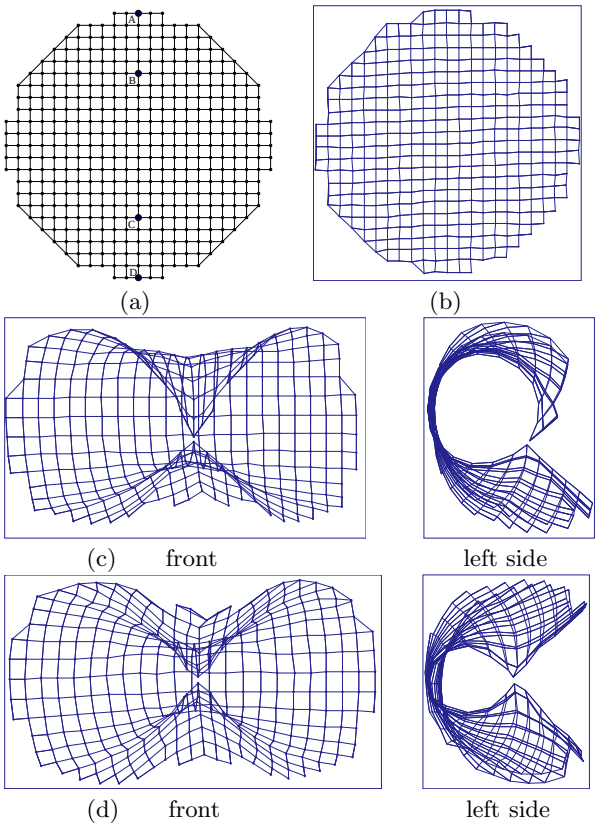


Figure 1. Reconstruct network using MDS.

The sensors are deployed on the grids in a circle area. A part of the neighbor relations are shown as the lines to assist the understanding of the figures. We assume that the distance measurements between neighbors are accurate when they are provided to MDS. (a) The original sensor network. (b) The rebuilt network when no wormhole exists. (c) The rebuilt network when a wormhole exists between sensor A and C. (d) The rebuilt network when a wormhole exists between sensor B and C.

reconstructed network in Figure 1. Figure 1.(a) shows the original sensor network and the sensors are deployed on the grids in a circle area. Figure 1.(b) shows the reconstructed network using MDS when no wormhole exists and its layout is almost the same as the original network. In Figure 1.(c) and (d), the wormhole will pull the sensors at the two ends to each other through the fake connection, and results in a bent surface. Through detecting the bending feature, MDS-VOW will identify the fake connections and locate the wormhole.

MDS-VOW mechanism consists of four steps: (1) It uses the inaccurate distance measurements between the sensors that can "hear" each other (might through a wormhole) as inputs to estimate the distance between every sensor pair. (2) Using multi-dimensional scaling, we reconstruct the network of sensors and calculate a virtual position for each of them. (3) A surface smoothing mechanism is adopted to compensate the impacts of distance estimation errors on the reconstructed network. The mechanism will preserve the features that are introduced by the wormhole. (4) The shape of the reconstructed network is analyzed and the fake neighbor connections will be identified.

As we will demonstrate, this simple method can effectively identify the fake neighbor connections. Moreover, it only requires the inaccurate distance estimations between the sensors. The reconstructed network is an arbitrary rotation and translation of the original network. Since the detection method focuses on the shape of the network instead of the coordinates of the sensors, we do not require the deployment of "anchors" that are equipped with positioning devices.

The remainder of this paper is organized as follows: In section 2, we review the previous work on the application of MDS in wireless networks, wormhole detection, and distance estimation between wireless nodes. Section 3 describes the building blocks of MDS-VOW and the algorithm in detail. The problems such as system bootstrapping, distance error compensation, and wormhole detection are studied. Section 4 presents the experimental results acquired through simulation. Two scenarios, grid placement and random placement of the sensors, are studied by varying the distance estimation error rate. Section 5 discusses the impacts of sensor density, the safety of MDS-VOW, and the future work. Section 6 concludes the paper.

2. RELATED WORK

MDS and Its Applications in Wireless Networks

Multi-dimensional scaling was originally a technique developed in the behavioral and social science for studying the structure of objects. The inputs to MDS are the measures of the difference or similarity between object pairs [7]. The output of MDS is a layout of the objects in a low-dimensional space. In this paper, the input is the distance matrix between the sensors. The mechanism can reconstruct the network and calculate a virtual position for each sensor. We adopt the classical metric MDS in the proposed mechanism, in which the distances are treated as in a Euclidean space. More details of MDS can be found in [7, 27].

MDS has been applied to solve the localization and positioning problems in wireless networks. In [26], a solution using classical metric MDS is proposed to achieve localization from mere connectivity information. The algorithm is more robust to measurement errors and requires fewer anchors than previous approaches. A distributed mechanism for sensor positioning using MDS has been presented in [15]. It develops a multi-variate optimization-based iterative algorithm to calculate the positions of the sensors. Another approach [2] for sensor network localization applies semi-definite programming relaxation to minimize the errors for fitting the distance measurements.

Wormhole Detection

Wormhole attacks on mobile ad hoc networks were independently discovered by Dahill *et al* [6], Hass *et al* [20], and Hu *et al* [13]. To defend against them, some efforts have been put on the signal processing techniques. If the data bits are transferred in some special modulating method known only to the neighbor nodes, they are resistant to the wormholes. Another approach, RF watermarking, works in the similar way. Both mechanisms prevent wormholes by increasing the difficulties to capture the signal patterns.

The adoption of directional antennas [16, 5] by mobile nodes can improve security. A solution that uses such equipments to defend against wormholes has been presented in [12]. The neighbor nodes examine the directions of the received signals from each other and a shared witness. Only when the directions of both pairs match, the neighbor relation is confirmed.

Some mechanisms proposed to locate the position of a mobile node in an indoor environment [22, 1, 28] can be applied to prevent wormholes. For example, both the original packet and the resent one will be captured by the location sensors and two conflicting positions of the same node will be detected. Either the good nodes or a centralized controller will discover this anomalous result. However, it will not be easy to port such methods to outdoor environments.

One approach to detect wormholes without clock synchronization is proposed by Capkun *et al* [4]. Every node is assumed to be equipped with a special hardware that can respond to a one-bit challenge without any delay. The challenger measures the round trip time of the signal with an accurate clock to calculate the distance between the nodes. The probability that an attacker can guess all bits correctly decreases exponentially as the number of challenges increases.

Packet leash is a solution proposed by Hu, Perrig and Johnson for wormhole prevention [13]. The leash is the information added into a packet to restrict its transmission distance. In the geographical leashes, the location information and loosely synchronized clocks together verify the neighbor relation. In the temporal leashes, the packet transmission distance is calculated as the product of signal propagation time and the speed of light.

Distance Estimation between Wireless Nodes

Several schemes have been proposed to estimate the distance between the wireless nodes. The example solutions include received signal strength [17], Time-of-Arrival and Time Difference of Arrival [21, 25, 4], and triangulation [24, 19]. Among them, Received Signal Strength Indicator is the most cost-efficient method because it does not require any extra hardware on the node. One disadvantage, however, is that the measured distance can be inaccurate. For example, an estimation error from 5% to 40% of the radio range has been assumed in [23]. The estimation accuracy can be improved by establishing a more accurate signal propagation model or using the stability of the strength difference at various points. In MDS-VOW, we adopt a mechanism from computer graphics to smooth the reconstructed network and compensate the impacts of the measurement errors.

3. VISUALIZATION OF WORMHOLES IN SENSOR NETWORKS

3.1 System Assumptions

We assume that the links among sensors are bidirectional and two neighbor sensors can always send packets to each other. This assumption will hold under most conditions when the power of the sensor has not been exhausted. We assume that two sensors are neighbors when the distance between them is shorter than r, where r is defined as the radio range.

To use MDS to reconstruct the network layout, we assume that the sensors and their neighbor relations construct a connected graph, i.e. a path exists between every sensor pair. The density of sensors may also impact the network reconstruction, and we give more discussion on this problem in section 5.

We assume that a special node exists in the sensor network, which is called the "controller". The controller can accomplish the $O(n^3)$ operations required by the MDS mechanism in a short period of time when there are n sensors in the network. In our experiments, we use a PC with 1.8G CPU and 512M RAM as the controller. When $n \approx 400$, it takes the machine a few seconds to reconstruct the network. When the controller broadcasts a message with full power, all sensors in the network can receive the data. On the contrary, only the sensors within the radio range to the controller can directly communicate to it. Others need to send their packets through the multi-hop routes. The adoption of the centralized controller impacts the scalability of MDS-VOW. Extending the mechanism to a distributed approach is discussed in the future work.

We assume that the sensors are not self-movable. Therefore, once deployed, the route changes are mainly caused by the "dead" or broken of the sensors. Extending MDS-VOW to a movable environment is discussed in section 5.

The sensors broadcast the neighbor discovery beacons at the same power level so that the neighbors can estimate the distances using the received signal strength. The sensors report the list of nodes that they can hear and the estimated distances to the controller. We assume that the controller and the sensors share a group key. This key is only used to protect the traffic generated by MDS-VOW and it is mainly used during network bootstrap. Therefore, the time duration and the amount of traffic that a malicious node can acquire to break the key is limited. The key can be determined before the sensors are deployed [3].

The sensors estimate the distances with the received signal strength. The accuracy of the estimation is impacted by various factors, such as the terrain, background noise level, or even the weather condition. We model the measurement errors as uniform noises. With an error rate e_m, if the real distance between two sensors is d ($d \leq r$), a random value drawn from the uniform distribution $[d \times (1 - e_m), d \times (1 + e_m)]$ is used as the measured distance. If the chosen value is smaller than 0 or larger than the radio range, 0 or r will be used respectively. In our experiments, we examine different values of e_m from 0 to 0.8.

3.2 Network Bootstrap

As we discussed before, the sensors send the list of nodes that they can hear and the distance estimations to the controller. The list may include both the real neighbors and the fake ones through the wormholes. Since the data is transferred through multi-hop routes and the detection has not been conducted, these routes could be controlled by wormholes and the data could become the target of the attacks. If the information cannot reach to the controller, the detection accuracy of MDS-VOW will be impacted. To prevent this problem, the network bootstrap is conducted as follows.

After sensor deployment, the controller will broadcast a route discovery packet to the sensors within the radio range r and mark the path length to itself as 0. The sensors receiving the packet will increase the path length by one and re-broadcast it. With every sensor remembers the previous hop, increases the path length by one, and re-broadcasts the packet, the routes to the controller will be established. If multiple packets are received by a sensor, it will use the one with the shortest path length. These routes can only be used to send the sensor lists and distance estimations for wormhole detection. Once the fake connections are excluded and the real neighbors are known to every sensor, other routing protocols can be adopted for sensing data transfer.

After measuring the distances to the sensors that it can hear and sending the information to the controller, every sensor will put itself in an "idle" state until a reply from the controller is received. In this state, a sensor will not forward packets for other nodes except the sensor lists and distance estimations. The controller examines the received packets and broadcasts a list with full power that includes the sensors whose information has not been received. These sensors, worrying about that their packets will get lost again, will flood the network with their information. The controller has a good chance to get the flooded packets. MDS-VOW will ignore the sensors whose packets are still not received.

For every sensor pair that can hear each other, the controller calculates the average of the two estimated distances and uses it in MDS. A connection will be ignored if none of the estimations or only one of them is received. Then the controller will execute MDS-VOW to identify the fake connections. It will send a reply with full power to every sensor and the reply includes the sensor's non-suspicious neighbors. After receiving the reply, the sensor will switch to the "operation" state and only uses the non-suspicious neighbors during route discovery and packet forwarding. Those neighbors that can be heard but fail to pass the detection will not be used. Therefore, a neighbor connection must be examined by MDS-VOW before it is adopted by the sensors. The attackers cannot hide the fake connections from the controller if they want the fake connections to be used.

After the execution of MDS-VOW, other routing protocols can be adopted to establish routes and transfer data as long as the sensors only use the non-suspicious neighbors. Since we assume that the sensors are not self-movable, there should not be any new neighbors appearing during the network operation. Therefore, MDS-VOW does not need to be run repeatedly when the sensors adapt to the route changes.

The packets transferred for MDS-VOW are protected by the group key. Message authentication code (MAC) can be calculated and attached to the packets to protect their integrity. Some sensors may stay in the "idle" state and cannot operate properly if they fail to get the replies. With the high density of sensors, their impacts on sensing coverage and routing are restricted.

3.3 Building Blocks of MDS-VOW

3.3.1 Network Reconstruction

The proposed mechanism uses the measured distances between the sensors that can hear each other to reconstruct the network layout. For every such pair, both sensors will estimate the distance and send it to the controller. The controller calculates the average value and puts the result at the suitable positions in the distance matrix. If the average value is larger than the radio range, r will be used in the matrix. The distance from a sensor to itself is 0. After the distances between the sensor pairs that can hear each other are calculated, a classical shortest-path algorithm, such as Dijkstra's algorithm, can be applied to calculate the shortest distance between every sensor pair. When all positions in the distance matrix have been filled, MDS can be applied to rebuild the network and a virtual position for every sensor will be generated.

3.3.2 Distance Error Compensation

The distance estimation errors have a significant impact on network reconstruction. As an example, Figure 2 shows

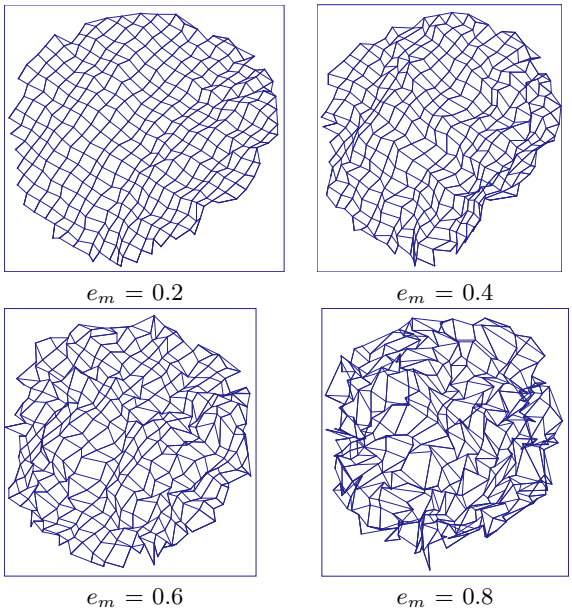


Figure 2. Impacts of measurement errors on network reconstruction.

the results of MDS when the sensors are deployed in a circle area as in Figure 1.(a). No wormhole exists in the network and the error rate e_m increases from 0.2 to 0.8. From the results, we find that mechanisms must be designed to compensate the errors while preserving the features that are introduced by the wormholes.

We propose to apply the smoothing algorithm for the reconstructed 3D surfaces [9, 11] to accomplish the task. The mechanism consists of two steps: (1) it calculates a fitting plane for every sensor based on the coordinates of itself and its neighbors, (2) a new position of the sensor is determined by the old coordinate and its projection on the fitting plane. The details are discussed as follows.

For a sensor whose position is s, the positions of its k neighbors are represented as N_0 to N_{k-1}. We first calculate the center of these $k+1$ nodes as:

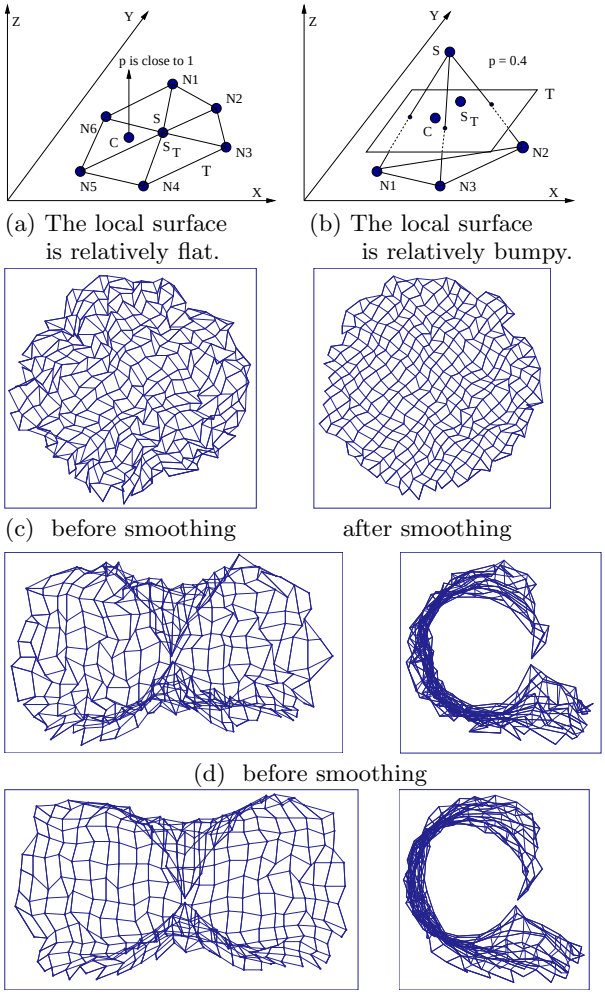
$$c = \frac{s + \sum N_j}{k+1}, j = 0 \cdots k-1 \quad (1)$$

The fitting plane of s will pass the center. Besides a point on the plane, we also need to calculate its normalized vector v_s. Using the positions of these $k+1$ sensors and the center, we can construct a 3×3 matrix as:

$$M = \sum_{j=0}^{k-1} (N_j - c)(N_j - c)^T + (s - c)(s - c)^T \quad (2)$$

M is a symmetric, positive semi-definite matrix and it has been shown in [11] that the unit eigenvector corresponding to the smallest eigenvalue of M is the normalized vector v_s of the desired plane, and the smallest eigenvalue indicates the least-squares error. The eigenvalues can be calculated by the QR factorization [10]. With the calculation of c and v_s, the fitting plane T for s is determined.

If two sensors s and s' are close to each other and the local surface is relatively flat, the fitting planes T and T' are nearly parallel. In other words, the point product of the two normal vectors has a value close to 1 or -1. On the contrary, if the surface close to s is very "bumpy", the normal vectors



(a) The local surface is relatively flat.
(b) The local surface is relatively bumpy.
(c) before smoothing
after smoothing
(d) before smoothing
after smoothing

Figure 3. Smoothing the reconstructed network.

(a) and (b) show the smoothing operations when the local surface is flat and bumpy. (c) shows the smoothing effects when no wormhole exists. (d) shows the effects when a wormhole exists in the network.

of T and T' may vary greatly. After calculating the fitting planes, we generate a value p for every sensor to describe the smoothness of the nearby surface. For s and its neighbors N_0 to N_{k-1}, we define:

$$p_s = \frac{\sum |v_s \cdot v_{N_j}|}{k}, j = 0 \cdots k-1 \quad (3)$$

Since all vectors are normalized, p_s has a value between 0 and 1. The larger the value is, the more smooth the local surface is. The projection of s on T is represented as s_T. We smooth the reconstructed network by calculating the new position of s as:

$$new_s = p_s \times s + (1 - p_s) \times s_T \quad (4)$$

Examples of the smoothing procedure are shown in Figure 3. If the local surface is flat, c will also be on the same plane and p_s is close to 1. Therefore, new_s will almost be at the same position as s, as shown in Figure 3.(a). On the contrary, if the local surface fluctuates a lot, the normal vectors of the neighbors will point to all different directions. The new position will be close to s_T, as shown in Figure 3.(b). Different from the "bumpy" features caused by the measurement errors, the bending feature of the network is caused by the fake neighbor connections through the wormhole. Its impacted area is much larger than a sensor and its neighbors. It will not be removed by the smoothing operations that focus on a small area of the network and the results can be seen in Figure 3.(c) and (d).

3.3.3 Detection of Wormhole

The results in Figure 3 show that if no wormhole exists in the network, the reconstructed surface after smoothing will be relatively flat. However, if two sensors are linked by a wormhole, the MDS mechanism will bend the reconstructed surface to fit the fake connection and minimize the fitting errors. If we imagine the network as a pie of soft rubber, the wormhole can be viewed as a "string" that pulls two sensors to each other and leads to the distortion of bending. Detecting the bending feature caused by the wormhole and locating the ends of the "string" will help us to identify the fake neighbor connections.

Figure 4 shows a reconstructed surface and the enlargement of the fake connection and its nearby areas. We find that the fake connection through the wormhole and the neighbor sensors at both ends will form a two-ended torch structure. This structure can be detected by examining the angles between the fake connection and the planes determined by the neighbor sensors. For example, the fake connection in Figure 4 is almost vertical to the plane $A_1A_3A_5$ and $B_1B_2B_3$. In MDS-VOW, we first derive a *normalized torch counter* for every connection based on the two-ended torch structures that it forms. Using the *counters* of the connections that a sensor is involved in, we define a *wormhole indicator* for every sensor. MDS-VOW then labels the connections between two sensors with large *wormhole indicators* as fake connections. The details are described as follows.

Assume that a sensor s can hear other k sensors as N_0 to N_{k-1}. For every connection sN_j $(j = 0 \cdots k-1)$, we calculate the number of two-ended torch structures that it forms. We assume that the neighbors of s can determine g different planes, and the neighbors of N_j can determine h different planes. We choose one plane from each set and examine the angles between the connection sN_j and the planes. When both angles are $\geq \frac{3\pi}{8}$, we count it as a two-ended torch structure. We examine all gh combinations. Since the number of the planes determined by the neighbors may vary greatly, the counter for sN_j is then normalized by dividing gh. We define this normalized number as the *normalized torch counter* of the connection sN_j. When we have calculated the *counters* for all the connections sN_j $(j = 0 \cdots k-1)$, we define the *wormhole indicator* of s as $\max\{counter\ of\ sN_j, j = 0 \cdots k-1\}$. As the examples of the proposed mechanism, we demonstrate the *wormhole indicators* of the sensors in different network scenarios in Figure 5.

From Figure 5 we find that the sensors close to the ends of the wormhole can be easily identified. MDS-VOW then labels the connections between two sensors that have large *wormhole indicators* as fake connections. In our experiments, we set the threshold at 0.6. The advantage of this method is that the sensors, no matter where their positions are in the network, can be handled in a uniform way. The disadvantage, however, is that some real neighbor connections may be wrongly accused as wormholes and false pos-

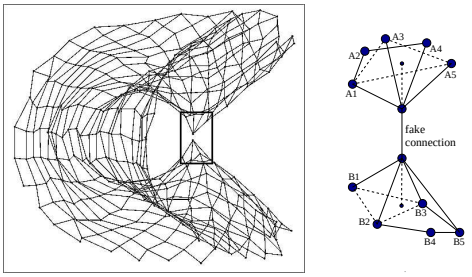


Figure 4. The two-ended torch structure caused by a wormhole.

itive alarms will be introduced into the system. We study this problem through experiments in section 4.

3.4 The MDS-VOW Algorithm

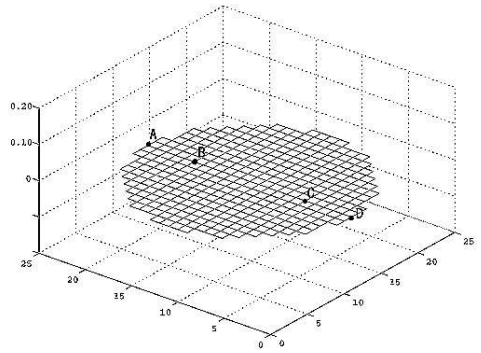
With the readiness of all building blocks, we now walk through the steps of the MDS-VOW algorithm.

1. Every sensor estimates the distances to the nodes that it can hear and reports them to the controller.
2. The Dijkstra's algorithm is applied to calculate the distance between every pair of sensors and the distance matrix of the network is constructed.
3. Using the classical metric MDS method, MDS-VOW reconstructs the layout of the network and calculates a virtual position for each sensor.
4. Smoothing mechanism is applied to compensate the impacts of the measurement errors. The mechanism will preserve the feature that is introduced by wormhole.
5. The *wormhole indicator* of every sensor in the reconstructed network is calculated. The fake neighbor connections through wormholes are identified.
6. The fake connections are distributed to the sensors by the controller. These connections will be avoided during routing and packet forwarding.

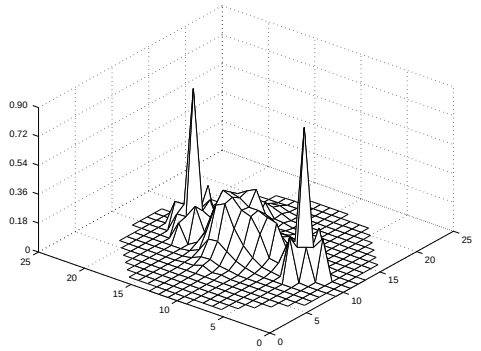
4. EXPERIMENTAL RESULTS

The performance of MDS-VOW is examined through simulation. The experiments are conducted in two phases. In the first phase, we use $ns2$ to simulate the distance estimation procedures and the report of the information to the controller. The packets may get lost because of the unreliable medium. In the second phase, a Visual C++ program executes the MDS-VOW mechanism based on the received distances and tries to identify the fake neighbor connections. The sensors are deployed in a circle area instead of a square. This choice is inspired by the scenario that a security critical location is at the center of the circle, and we need to monitor the activities within a certain range. The area of the circle is $1km^2$, and the radius of the circle is about $565m$. The radio range r of the sensors is $110m$, and any two sensors that have a distance shorter than r can directly communicate to each other.

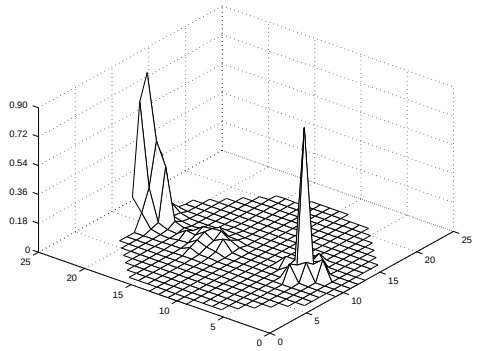
Two deployments of the sensors are examined: grid placement and random placement. In the grid placement, the sensors are deployed at an interval of $50m$ along the imaginary vertical or horizontal lines. A total number of 401 sensors are placed in the circle and the average degree of connectivity is 11.0. In the random placement, we apply the dart throwing method proposed in [18] to place the sensors randomly and roughly uniformly in the area. To maintain a similar degree of connectivity as in the grid placement, 441



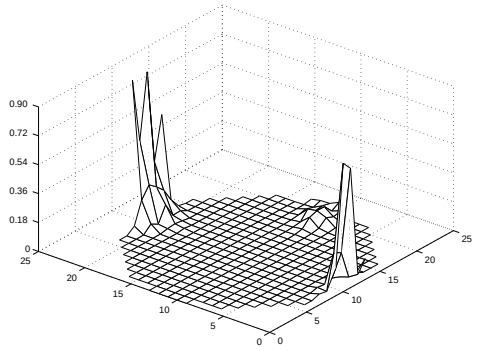
(a) No wormhole exists in the network



(b) A wormhole exists between sensor B and C



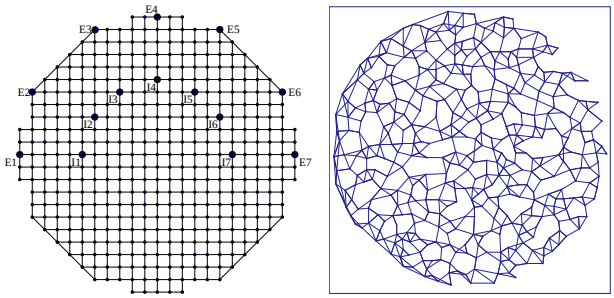
(c) A wormhole exists between sensor A and C



(d) A wormhole exists between sensor A and D

Figure 5. Wormhole indicators of the sensors in different network scenarios.

The sensors are deployed on the X-Y plane as in Figure 1.(a). The Z-axis shows the value of the wormhole indicator. The error rate e_m is 0.4 in the experiments.



(a)Grid placement (b)Random placement example

Figure 6. Experimental network topology.

In both figures, only a part of the neighbor connections are shown as the lines to assist the understanding of the figures.

sensors are used. Examples of the placements are shown in Figure 6. In both placements, the controller is located at the center of the circle. The justification of this sensor density choice is discussed in section 5.

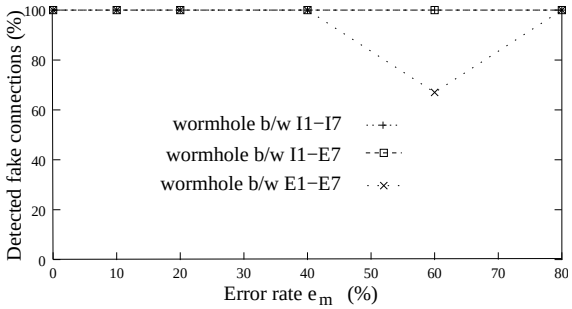
We model the distance estimation errors as uniform noises. If the accurate distance between two sensors is d ($d \leq r$) and the error rate is e_m, a random value drawn from the uniform distribution $[d \times (1 - e_m), d \times (1 + e_m)]$ will be used as the measured distance. If the selected value is smaller than 0 or larger than the radio range, 0 or r will be used respectively. In the experiments, e_m changes from 0 to 0.8. For a fake neighbor connection through a wormhole, a random value from 0 to the radio range will be first selected as d, and then the error will be added. The data points in the figures represent the averages over 15 trials using different error values.

4.1 Grid Placement

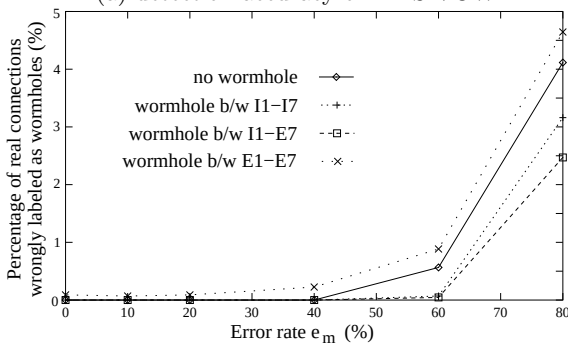
The grid placement of sensors is shown in Figure 6.(a). Seven sensors in the circle and seven sensors on the border of the area are selected as the potential victims of wormholes. They are labeled as $I1$ to $I7$ and $E1$ to $E7$ respectively. Two groups of experiments are conducted. The first group examine the MDS-VOW algorithm under different e_m rates. Four scenarios are considered: (1) no wormhole exists, (2) a wormhole links $I1$ and $I7$, (3) a wormhole links $E1$ and $E7$, and (4) a wormhole links $I1$ and $E7$. The detection accuracy of MDS-VOW and its impacts on routing are of special interest.

Figure 7 and 8 show the results. In Figure 7.(a), we find that MDS-VOW can detect the fake connections under most conditions when e_m increases from 0 to 0.8. MDS-VOW has a low false negative ratio. From Figure 7.(b) we find that when e_m is smaller than or equal to 0.6, less than 1% of the real neighbor connections will be wrongly labeled as wormholes. When e_m increases to 0.8, the false positive alarm ratio becomes larger, but still less than 5% of the real connections are wrongly accused.

The false positive alarms will lead to the breaks of the real neighbor connections and the increase in the average path length. If all connections of a sensor are broken, an isolated node will be generated and the events detected by such sensor cannot be transferred out. To examine the impacts of the false positive alarms, we show in Figure 8 the increase in the average path length between all sensor pairs. Since the degree of connectivity in the original layout is relatively large, the increase in the average path length is small. We do not detect isolated sensors in the experiments.



(a) detection accuracy of MDS-VOW



(b) false positive alarms of MDS-VOW

Figure 7. Detection accuracy of MDS-VOW in grid placement when varying e_m.

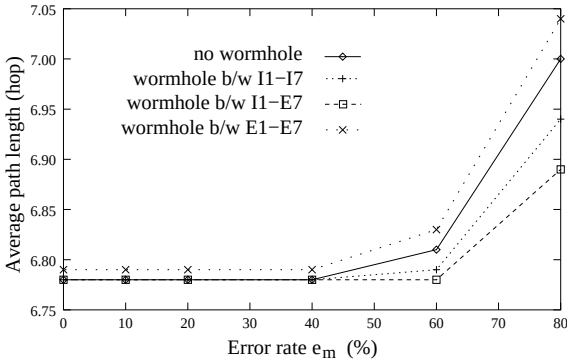


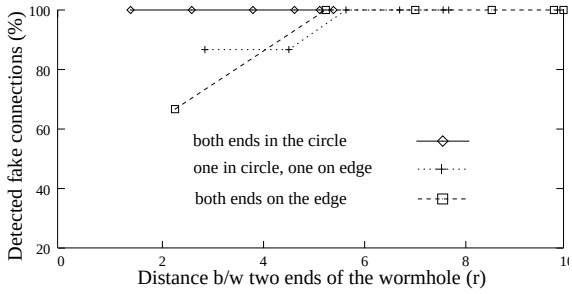
Figure 8. Increase in the average path length caused by false positive alarms.

In the second group of experiments, we fix the choice of e_m at 0.4 and examine the detection accuracy of MDS-VOW when the distance between the two ends of the wormhole changes. Seven sensors in the circle and seven on the border are selected as the potential victims. Since the increase in the average path length is small in Figure 8, we focus on the false alarm ratio in this group of experiments.

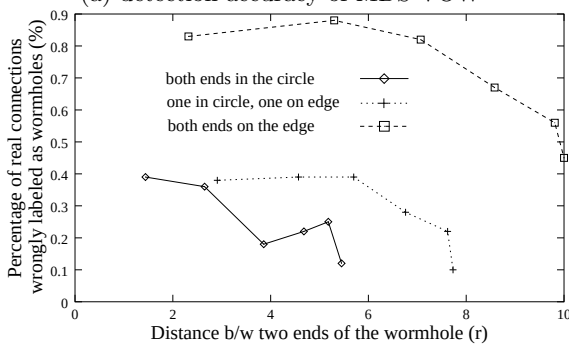
The ends of the wormhole are put at different positions in the network and three conditions of the fake connection are examined: (1) one end of the fake connection is $I1$, the other end changes from $I2$ to $I7$, (2) one end is $I1$, the other end changes from $E2$ to $E7$, and (3) one end is $E1$, the other end changes from $E2$ to $E7$. The results are shown in Figure 9 and under most conditions the fake connections can be effectively located and not many false positive alarms are introduced into the network.

4.2 Random Placement

In the random placement scenarios, we apply the dart throwing method to place the sensors randomly and roughly

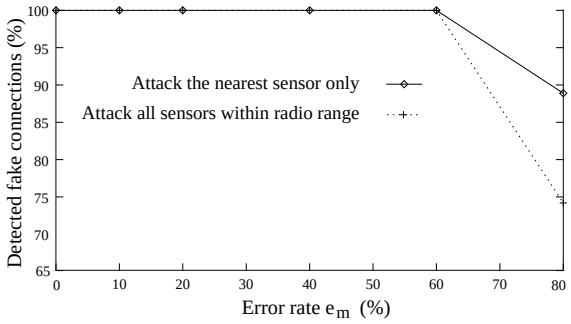


(a) detection accuracy of MDS-VOW

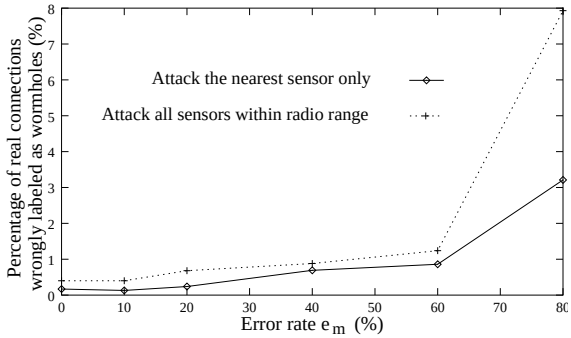


(b) false positive alarms of MDS-VOW

Figure 9. Detection accuracy of MDS-VOW when varying wormhole length.



(a) detection accuracy of MDS-VOW



(b) false positive alarms of MDS-VOW

Figure 10. Detection accuracy of MDS-VOW in random placement.

uniformly in the network. One layout is shown in Figure 6.(b). Only a part of the neighbor connections are shown in the figure to assist the understanding of the topology. Two groups of experiments are conducted to examine the detection accuracy of MDS-VOW. In group one, two random positions in the area are selected as the ends of the wormhole. The wormhole then chooses a sensor from each end that has the shortest distance to it. If the distance between the two sensors is larger than r, a fake connection between them will be established. Otherwise, the position of the wormhole will be generated again. Experiments are conducted by varying the error rate e_m.

In the second group of experiments, there is still only one wormhole in the network. But the wormhole will establish fake connections between all sensor pairs when the sensors are within the radio range to the ends of the wormhole. For example, if s_1 to s_3 are the sensors within r to one end of the wormhole, and s_4 to s_6 are within r to the other end of the wormhole, 9 fake connections will be established if the distance between every pair is larger than r. Experiments are conducted by varying e_m and the results are illustrated in Figure 10.

Studying the results shown in Figure 7 to Figure 10, we find that when $e_m \leq 0.6$, MDS-VOW has a low false positive ratio and a low false negative ratio. The proposed mechanism can detect the fake connections in the grid placement and random placement scenarios without hurting many real connections.

5. DISCUSSION AND FUTURE WORK

The proposed mechanism detects wormholes by visualizing the anomalies caused by such attacks in the reconstructed network. It avoids the requirements of special hardware and can be applied to the environments such as sensor networks. The MDS-VOW mechanism consists of multiple steps and each step uses the result from the previous one as input. The details of the method for each step are transparent to other steps and they can be improved independently. For example, when a new feature-preserving surface smoothing mechanism appears, it can replace the current one in the algorithm and helps to reduce the false positive alarm ratio.

Impacts of Sensor Density

The sensor density impacts the MDS-VOW algorithm in two ways: (1) The calculation of the distance matrix, and (2) The performance degradation caused by false positive alarms. Assuming a network with infinite sensor density. When two sensors h_1 and h_2 have a distance d, where $r < d < 2r$, we can always find a third sensor h_3 that is on the line determined by h_1 and h_2 and has a distance shorter than r to both of them. Therefore, when using the Dijkstra's method, we can add $|h_1h_3|$ and $|h_2h_3|$ to calculate $|h_1h_2|$ without introducing any error. As the density decreases, errors will be introduced into the distance matrix even when the distance estimations between neighbor sensors are accurate. For the same reason, when sensor density decreases, the degree of connectivity becomes smaller, and the node has a higher probability to become an isolated sensor when the false positive alarms break the real connections.

We refer to previous research efforts and experiments in real applications when choosing the sensor density in our experiments. Similar density has been adopted in [26]. In that paper the authors deploy 200 nodes in a $10l \times 10l$ area when the radio range changes from l to $2l$. In the vehicle classification experiments conducted by U.S. Army [8], the sensors are deployed at an interval of $30 - 40m$. In our experiments, we set the grid size as $50m$.

Security of MDS-VOW

As a security enhancement to defend against wormhole attacks, the robustness of MDS-VOW must be studied. Dur-

ing the execution of MDS-VOW, data traffic exists between the controller and the sensors and among the sensors. The malicious nodes can attack these packets by: (1) changing the contents or impersonating the senders of the packets when re-transmitting them, (2) dropping these packets, and (3) changing the re-transmission power to mislead the distance estimation. We now discuss the solutions to these attacks respectively.

For the first attack, the integrity of the packets can be protected by the group key shared by the sensors and the controller by attaching the message authentication code (MAC) to the packets. For the second attack, the analysis for system bootstrap shows that the malicious nodes cannot drop these packets to hide the fake connections because a neighbor connection must be examined by the controller before it is adopted by the sensors. The malicious nodes can still adjust the re-transmission power of the packets to mislead the distance estimation for the fake connections. But since the radio range is known to the sensors, its impacts are restricted and it can be viewed as a special distance measurement error.

Modeling Measurement Errors

The errors of the distance estimation using the received signal strength are difficult to model considering the features that may impact the measurement accuracy. Introducing the anchor nodes that know their positions into the system can reduce the positioning errors in an attack-free environment [15]. Besides the uniform noise model that is adopted in this paper, the Gaussian noise model of placement errors has been applied in [26]. We are now conducting more experiments of MDS-VOW that use the Gaussian noise model. As we discuss in the previous part, when a more accurate model of the errors appears, the current one can be replaced with limited efforts.

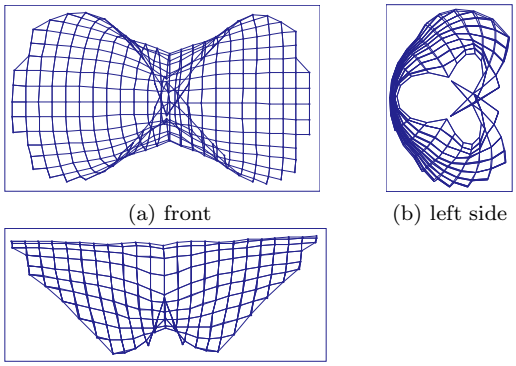
Extending MDS-VOW to Movable Environments

In this paper, we assume that the sensors are not self-movable. Therefore, unless some out-force moves the nodes, the positions and the neighbor relations of the sensors will not change. This assumption allows the controller to run MDS-VOW once during the network bootstrap and tell every sensor its non-suspicious neighbors. When extending MDS-VOW to a movable environment, we must overcome two difficulties: how to adapt to route changes, and how to reduce the communication and computation overhead.

In a movable environment, new neighbor relations and new routes may appear when the nodes change their positions. The wormhole detection mechanism must adapt to such changes by re-computing the network topology. If a proactive method is adopted, the controller needs to detect wormholes periodically. If a differential threshold coding technique is adopted, a node will update its information and activate a re-detection of wormholes only when its measurements change beyond a threshold. In both cases, only the initial ideas are available and much more research work is required to turn them into reality.

Future Work

There are several immediate extensions to the proposed mechanism. To illustrate the ideas of MDS-VOW, we assume a flat plane on which the sensors are deployed in the experiments. In the real environments, more complex conditions need to be considered. First, the terrain of the network can be non-flat and false alarms may be introduced by this



(a) front (b) left side

(c) top

Figure 11. Impacts of two wormholes on the reconstructed network.

reason. Therefore, a more robust and error-tolerant detection method needs to be designed. Second, it is difficult to deploy the sensors as a nice polygon in reality. More research efforts are required to study the impacts of wormholes when the shape of the network area is irregular.

The MDS-VOW mechanism is a centralized scheme. It makes the mechanism less adaptable and leads to the security problems such as single point of failure. There has been research work to divide the network into sub-divisions and to develop a distributed MDS method [26]. Then the pieces of the reconstructed network will be merged and the VOW method can be applied to detect wormholes.

Extended efforts are also required to study the shape of the reconstructed surface when multiple wormholes exist in the network. Figure 11 illustrates an example of the reconstructed network when two wormholes link the sensor pairs (A, C) and (B, D) as in Figure 1.(a). When multiple wormholes exist in the network, the reconstructed surface can be distorted far from the original layout and the detection of the two-ended torch structure alone may not be able to locate all fake connections. A more sophisticated mechanism based on the statistical decision theory is required.

6. CONCLUSIONS

Different from the previous efforts that require the wireless nodes to be equipped with special hardware, the proposed MDS-VOW mechanism focuses on the features that are introduced by the wormholes. It can be deployed as a hardware-efficient method to defend against wormhole attacks in a sensor network. MDS-VOW uses the inaccurate distance estimations between the neighbor sensors as the inputs, and rebuilds a layout of the sensors using multi-dimensional scaling. The analysis and experiments show that the wormhole bends the reconstructed network to pull the sensors to each other and fit the fake connections. This forms the two-ended torch structure that can be used to detect the fake neighbor connections. MDS-VOW consists of multiple steps and each step can be improved independently.

Experiments using grid placement and random placement of sensors are conducted to examine the detection accuracy of the proposed mechanism. The results show that when the distance estimation errors are uniformly distributed and the error rate is equal to or smaller than 0.6, MDS-VOW can detect most of the fake connections without introducing many false positive alarms. Since the sensors in the examined scenarios are dense and the degree of connectivity is

relatively large, breaking the wrongly accused neighbor connections will not impact the connectivity and routing of the network to a large extent.

Additional research is required to study the performance of MDS-VOW under more complex scenarios. The problems that are of special interest include: how to detect wormholes in an irregular-shaped network on a non-flat terrain, and the impacts of multiple wormholes on the reconstructed network. The results will lead to a more accurate and efficient solution that can defend against wormholes for sensor networks.

7. ACKNOWLEDGEMENT

The authors would like to thank the valuable comments from the committee and the workshop chairs. This work is supported in part by NSF ANI 0219110 and NSF IIS 0242840. The authors would also like to thank the seeding fund from CERIAS and CISCO.

8. REFERENCES

[1] P. Bahl and V. Padmanabhan, RADAR: An In-Building RF-Based User Location and Tracking System, in the *Proceedings of INFOCOM*, 2000.

[2] P. Biswas and Y. Ye, Semidefinite Programming for Ad Hoc Wireless Sensor Network Localization, in the *Proceedings of Information Processing in Sensor Networks (IPSN)*, 2004.

[3] C. Boyd and A. Mathuria, Key Establishment Protocols for Secure Mobile Communications: A Selective Survey, *Lecture Notes in Computer Science*,1998.

[4] S. Capkun, L. Buttyan, and J. Hubaux, SECTOR: Secure Tracking of Node Encounters in Multi-hop Wireless Networks, in the *Proceedings of the ACM Workshop on Security of Ad Hoc and Sensor Networks*, 2003.

[5] R. Choudhury, X. Yang, R. Ramanathan, and N. Vaidya, Using Directional Antennas for Medium Access Control in Ad Hoc Networks, in the *Proceedings of ACM MobiCom*, 2002.

[6] B. Dahill, B. Levine, E. Royer, and C. Shields, A Secure Routing Protocol for Ad hoc Networks, *Tech Report 02-32*, Dept. of Computer Science, University of Massachusetts, Amherst, 2001.

[7] M. Davison, *Multidimensional Scaling*, John Wiley and Sons, 1983.

[8] M. Duarte and Y. Hu, Vehicle Classification in Distributed Sensor Networks, to be published in *Journal of Parallel and Distributed Computing*, 2004.

[9] O. Garcia-Panyella, An easy-to-code smoothing algorithm for 3D reconstructed surfaces, *Graphics programming methods*, 139-146, Charles River Media, Inc, 2003.

[10] G. Golub and C. V. Loan, *Matrix Computations*, Johns Hopkins University Press, 1996.

[11] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle Surface reconstruction from unorganized points, in *Proceedings of ACM SIGGRAPH*, 71-78, 1992.

[12] L. Hu and D. Evans, Using Directional Antennas to Prevent Wormhole Attacks, in the *Proceedings of Network and Distributed System Security Symposium (NDSS)*, 2004.

[13] Y. Hu, A. Perrig, and D. Johnson, Packet Leashes: A Defense against Wormhole Attacks in Wireless Ad Hoc Networks, in the *Proceedings of INFOCOM*, 2003.

[14] Y. Hu, A. Perrig, and D. Johnson, Rushing Attacks and Defense in Wireless Ad Hoc Network Routing Protocols, In *Proceedings of the ACM Workshop on Wireless Security (WiSe)*, 2003.

[15] X. Ji and H. Zha, Sensor Positioning in Wireless Ad-hoc Sensor Networks with Multidimensional Scaling, in the *Proceedings of IEEE INFOCOM*, 2004.

[16] Y. Ko, V. Shankarkumar, and N. Vaidya, Medium Access Control Protocols using Directional Antennas in Ad Hoc Networks, in the *Proceedings of INFOCOM*, 13-21, 2000.

[17] A. Ladd, K. Bekris, A. Rudys, G. Marceau, L. Kavraki, and D. Wallach, Robotics-Based Location Sensing using Wireless Ethernet, in the *Proceedings of ACM MobiCom*, 2002.

[18] T. Mitsa and K. J. Parker, Digital Halftoning Using a Blue-Noise Mask, in the *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, 1991.

[19] D. Niculescu and B. Nath, Ad hoc positioning system (APS) using AoA, in the *Proceedings of INFOCOM*, 2003.

[20] P. Papadimitratos and Z. Haas, Secure Routing for Mobile Ad Hoc Networks, in the *Proceedings of SCS Communication Networks and Distributed Systems Modeling and Simulation Conference (CNDS)*, 2002.

[21] N. Priyantha, A. Chakraborty, and H. Padmanabhan, The cricket location support system, in the *Proceeding of ACM MobiCom*, 32-43, 2000.

[22] N. Sastry, U. Shanker, and D. Wagner, Secure Verification of Location Claims, in the *Proceedings of the ACM Workshop on Wireless Security (WiSe)*, 2003.

[23] C. Savarese, K. Langendoen, and J. Rabaey, Robust Positioning Algorithms for Distributed Ad-Hoc Wireless Sensor Networks, in the *Proceedings of USENIX Technical Annual Conference*, 317-328, 2001.

[24] C. Savarese, J. Rabaey, and J. Beutel, Locationing in Distributed Ad-Hoc Wireless Sensor Networks, in the *Proceedings of ICASSP*, 2001.

[25] A. Savvides, C. Han, and M. Srivastava, Dynamic fine-grained localization in ad-hoc networks of sensors, in the *Proceedings of ACM MobiCom*, 2001.

[26] Y. Shang, W. Ruml, Y. Zhang, and M. Fromherz, Localization from mere connectivity, in the *Proceedings of the 4th ACM international symposium on mobile ad hoc networking and computing*, 2003.

[27] W. Torgeson, Multidimensional scaling of similarity, *Psychometrika, (30)379-393*, 1965.

[28] A. Ward, A. Jones, and A. Hopper, A New Location Technique for the Active Office, in *IEEE Personnel Communications, 4(5):42–47*, 1997.