The Holes Problem in Wireless Sensor Networks: A Survey

Nadeem Ahmed, Salil S. Kanhere, Sanjay Jha
School of Computer Science and Engineering
The University of New South Wales, Sydney 2052, Australia.
Email: {nahmed, salilk, sjha}@cse.unsw.edu.au

UNSW-CSE-TR-0438

November 2004

THE UNIVERSITY OF
NEW SOUTH WALES

SYDNEY • AUSTRALIA

Abstract

Several anomalies can occur in wireless sensor networks that impair their desired functionalities i.e. sensing and communication. Different kinds of holes can form in such networks creating geographically correlated problem areas such as coverage holes, routing holes, jamming holes, sink/black holes and worm holes, etc. We detail in this report different types of holes, discuss their characteristics and study their effects on successful working of a sensor network. We present state-of-the-art in research for addressing the holes related problems in wireless sensor networks and discuss the relative strengths and short-comings of the proposed solutions for combating different kinds of holes. We conclude by highlighting future research directions.

1 Introduction

The recent advances in the MEMS (Micro-Electro-Mechanical Systems) technology has augmented research in wireless sensor networks. A wireless sensor network is composed of tiny sensor nodes each capable of sensing some phenomenon, doing some limited data processing and communicating with each other. [1]. These tiny sensor nodes are deployed in the target field in large numbers and they collaborate with each other to form an adhoc network capable of reporting the phenomenon to a data collection point called sink or base station. These networked sensors have many potential civil and military applications i.e., they can be utilized for object tracking, intrusion detection, habitat and other environmental monitoring, disaster recovery, hazard and structural monitoring, traffic control, inventory management in factory environment and health related applications etc.[2], [3].

These myriad of applications present various design, operational and management challenges for wireless sensor networks. The challenges become even more demanding if we consider the inherited constraints of wireless sensor networks such as low processing power and bandwidth, limited battery life, and short radio ranges. The error prone characteristics of wireless transmission and unexpected node failures results in unstable topologies.

Wireless sensor networks differ from adhoc networks in several ways. One of the distinguishing features is the introduction of the sensing component in sensor networks. A node in a sensor network is thus performing two demanding tasks simultaneously, sensing and communicating. To accomplish these task, we normally assume that the node not only perform required sensing of the phenomenon but is also able to communicate with neighbors in its radio range, for onward transmission of the sensed data to sink. But this assumption is often not true in real world deployment scenarios.

Several anomalies can occur in the wireless sensor network that can impair their functionality. The target field that is supposed to be 100% covered by the densly deployed nodes may have *coverage holes*, areas not covered by any node, due to random aerial deployment creating voids, presence of obstructions, and, more likely, node failures etc. Similarly, nodes may not be able to communicate correctly if *routing holes*, areas devoid of any nodes, exist in the deployed topology. Thus the network fails to achieve its objectives if some of the nodes cannot sense or report the sensed data back to the sink. Some of the anomalies may be deliberately created by adversaries that are trying to avoid the sensor network. These malicious nodes can jam the communication to form *jamming holes* or they can overwhelm regions in the sensor network by denial of service attacks such as *sink/black/worm holes* to hinder their operation normally based on trust.

We discuss in this paper such exceptional circumstances with special attention to the phenomenon occurring in a region or hole and present state of the art in research related to these holes problems in wireless sensor networks.

We group together these potential holes related problems in four categories namely coverage holes, routing holes, jamming holes and sink/worm holes. The remainder of this report is organized as follows. We introduce the holes related problems in Section 2. Proposed solutions for coverage holes are discussed in Section 3. Section 4 elaborates proposed solutions to avoid the impact of routing holes. Jamming holes are covered in Section 5 and Section 6 discusses some of the suggested countermeasures against various denial of service holes related attacks i.e. sink/black and worm holes. We suggest some future research directions in Section 7 and Section 8 concludes the report.

2 Problem Definition

We formally define here various types of holes that can occur in a wireless sensor network and discuss their distinguishing characteristics.

Coverage Holes

Although the coverage problem has been interpreted in a variety of ways in the existing literature, we follow [4] for defining the coverage hole problem as follows. Given a set of sensors and a target area, no coverage hole exists in the target area if every point in that target area is covered by at least k sensors where k is the required degree of coverage for a particular application (see Figure 1). It is pertinent to mention that the coverage hole problem defined is dependent on application requirements. Some applications may require a higher degree of coverage of a given target area for fault tolerance/redundancy or for accurate localization of targets using triangulation-based positioning protocols [5] or trilateration based localization [6].

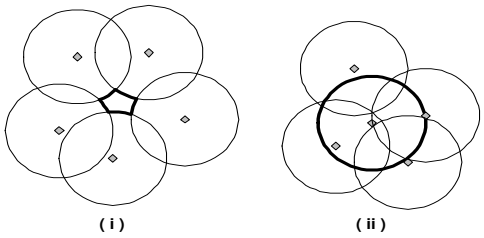

Figure 1: (i). Coverage hole with unit disk sensing model (ii). Sensor with dark gray sensing circle is necessary if degree of coverage required is 2

Thus a target area with a particular sensor deployment is said to have coverage holes when every point in that target area is not adequately covered according to the application requirements.

The sensing coverage of a sensor node is usually assumed uniform in all directions and is represented by unit disk model (Figure 1). However, the coverage of a sensor node not only depends on the sensing range of the sensor but also on the event characteristics [7] e.g. target detection of military tanks as compared to detection of movement of soldiers depends on the nature and characteristics of event (target in this case) as well as the sensitivity of the sensors involved. Most of the proposed solutions for the coverage hole problem assume that all the sensors are uniform in sensing capabilities and that they are being utilized to detect only a single type of event thus simplifying the scope of the problem to be addressed.
Nodes in the sensor network should be able to communicate with each other for transferring useful data to the sink. This implies that the network should be connected at all times. As with the multiple coverage requirement discussed earlier, multiple connectivity is also desirable to guard against single link or node failure partitioning the network. For the single coverage requirement, Wang et al. [8] proved that protocols working on assumption that communication range of sensors is $\geq$ twice the sensing range only needs to guarantee coverage and it will satisfy the connectivity constraint as well. We discuss different solutions to minimize the coverage holes in Section 3.

Routing Holes

A routing hole consist of a region in the sensor network where either nodes are not available or the available nodes cannot participate in the actual routing of the data due to various possible reasons. These holes can form due to voids in sensor deployment, sensor failure due to malfunctioning, battery depletion or an external event like fire or structure collapse physically destroying the nodes. In all such cases the routing protocol cannot draw any support from the nodes located inside the routing hole. These nodes will not participate in the routing process and the routing protocol should be able to detect and route traffic around these holes successfully.
Routing holes can also exist due to local minimum phenomenon often faced

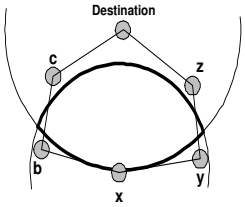

Figure 2: Local minimum phenomenon in greedy forwarding

in geographic greedy forwarding. Forwarding here is based on destination location. In Figure 2, a node x tries to forward the traffic to one of its 1-hop

neighbor that is geographically closer to the destination than the node itself. This forwarding process stop when x cannot find any 1-hop neighbor closer to the destination than itself and the only route to destination requires that packet moves temporarily farther from the destination to b or y. This special case is referred to as local minimum phenomenon and is more likely to occur whenever a routing hole is encountered.
We discuss different solutions to detect routing holes and to route around them in greedy forwarding. We also detail various multi path and single path routing solutions to provide fault tolerance against the routing holes in Section 4.

Jamming Holes

A sensor network depends on the collaborative efforts of deployed sensor nodes to accomplish its task. Individual nodes sense and report the sensed data to the sink with possible aggregation and in-network processing at intermediate nodes. An interesting scenario can occur in tracking applications when the object to be tracked is equipped with jammers capable of jamming the radio frequency being used for communication among the sensor nodes [9]. When this happens, nodes will still be able to detect the presence of the object in the area but unable to communicate the occurrence back to the sink because of the communication jamming. All the nodes located inside the zone of influence of the jammer object waste their energies in channel sensing as they all have useful data to send. This zone of influence centered at the jammer is referred to as jamming hole in this report.
The jamming can be deliberate or unintentional. Unintentional jamming results when one or more of the deployed nodes malfunction and continuously transmits and occupies the wireless channel denying the facility to other neighboring node. In deliberate jamming an adversary is trying to impair the functionality of the sensor network by interfering with the communication ability of the sensor nodes. This adversary can be a *laptop-class attacker* [10] with more resources and capable of effecting a larger area of the sensor network or *mote-class attacker* [10] i.e. one of the deployed nodes that has been compromised and is now acting maliciously to create a denial of service condition in its area of influence.
Apart from communication jamming, jamming of sensing capabilities is also possible for certain kind of sensor networks e.g. consider the case of a sensor network that relies on acoustic sampling to detect object for tracking purposes. The acoustic pattern available for the object to be tracked is compared with the sensed data. If the object that is being tracked can introduce random high power acoustic noises, the sensors cannot reliably detect its presence and would be unable to report the existence of the object. We discuss proposals to detect jamming holes in a sensor network in Section 5.

Sink Holes/Black Holes/Worm Holes

Sensor networks are highly susceptible to denial of service attacks due to their inherent characteristics i.e. low computational power, limited memory and communication bandwidth coupled with use of insecure wireless channel. Sink/black holes is a kind of denial of service attack that can be easily launched by an adversary node in the sensor network. The malicious node starts advertising very attractive routes to the data sink. The neighbor nodes thus select the malicious node as the next hop for message forwarding considering it a high quality route. The neighbors of the malicious node helps in forming the sink hole by propagating this route to other nodes. Almost all traffic is thus attracted to the malicious node that can either drop it, selectively forward it based on some malicious filtering mechanism or change the content of the messages before relaying it. This malicious node has thus formed a sink hole with itself at the center.
The sink hole is characterized by intense resource contention among neighboring nodes of the malicious node for the limited bandwidth and channel access [11]. This results in congestion and can accelerate the energy consumption of the nodes involved, leading to the formation of routing holes due to nodes failure. With sink holes forming in a sensor network, several other types of denial of service attacks are then possible [10],[11].
Worm holes is also another kind of denial of service attack [12]. Here the malicious nodes located in different part of the sensor network creates a tunnel among themselves. They starts forwarding packets received at one part of the sensor network to the other end of the tunnel using a separate communication radio channel. The receiving malicious node then replays the message in other part of the network. This causes legitimate nodes located in different parts of networks to believe that they are neighbors, resulting in incorrect routing convergence. Worm hole attack changes the routing topology of the sensor network causing the nodes to use the malicious nodes for attractive routes to nodes located in different parts of the network thus effectively creating sink holes. Section 6 details some of the counter-measures against these denial of service attacks.

3 Coverage Holes

In this section we describe various proposed solutions for finding and fixing coverage holes in sensor networks. The proposed protocols discussed here are classified based on the number of mobile nodes in the sensor network as (i) Mobile sensor networks (ii) Hybrid sensor networks (iii) Static sensor networks. For each of the protocols discussed in this section, a further differentiation is made according to whether it is designed to address single coverage or the multiple coverage requirement.

3.1 Mobile Sensor Networks

Several researchers have investigated techniques to obtain maximum single coverage of a target area using mobile sensors [13] [14] [15] [16]. A typical problem statement for this scenario is to maximize the coverage of a given target area with constraints on deployment time, the distance the sensors have to travel to maximize coverage and the complexity of the protocol [13].
In one of the proposed solutions [13], Wang et al. used Voronoi diagrams

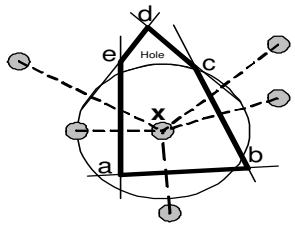

Figure 3: Voronoi diagram. *abcde* is the Voronoi ploygon for node x. Circle centered at x is the sensing disk

to discover the existence of areas not covered by any sensor once all the sensors have been initially deployed in the target area. A node needs to know the location of its neighbors to construct its Voronoi diagram. This implies that either all the nodes are GPS enabled or they use one of the GPS-less localization techniques such as [17]. The diagram partitions the whole space into Voronoi polygons. Each polygon has a single node with the property that every point in the polygon is closer to this node than any other node. A sensor node knows about its sensing capabilities, as a regular sensing disk of known radius. A sensor node compares its sensing disk with the area of its Voronoi polygon to estimate any local coverage hole (Figure 3).
Three distributed self-deployment algorithms have been proposed: Vector based(VEC), Voronoi based(VOR) and Minimax algorithm. All the algorithms are based on the principle of iteratively moving sensors from densely deployed areas to sparsely deployed areas. In VEC a node calculates the average distance between the neighbors assuming all nodes are uniformly deployed in the given target area and tries to keep the distance with its neighbor approximately equal to this distance. In VOR once the coverage hole is detected, the node moves toward the farthest vertex of the Voronoi polygon to cover the local maximum coverage hole. Minimax works like VOR with the additional check that while moving toward the farthest Voronoi vertex, it also keeps track of distances to other vertices and finds a target position inside the polygon from where the distance to the farthest vertex is minimized. Some optimizations for these protocols have also been proposed to cater for clusters forming in initial deployment that would unduly extend the number of iterations required to achieve reasonable coverage in the tar-

get area.
The authors have evaluated the performance of the proposed protocols based on deployment quality and cost. Deployment quality is measured by the total coverage achieved and time required to achieve this coverage. The cost includes the number of iterations the algorithm performs to achieve maximum coverage. Simulation results presented in [13] shows that the Minimax algorithm outperform the other two proposed algorithm in achieving maximum coverage with little increase in computational overhead. However, on average, Minimax moves the sensors longer distance than the other two algorithms to achieve this higher level of coverage. Results for lower value of communication range have shown that VEC outperforms the other two as it is least affected by the accuracy of the Voronoi diagram constructed.
Saurabh et al. in [18] proposed a protocol called Co-Fi that uses mobility capable sensors for repairing the loss of coverage due to energy depletion in a deployed sensor network. Low energy nodes, on predicting death, broadcast a Panic request message. Nodes with high energy level respond with the Panic reply message if they can move without losing existing coverage. The Panic reply message contains residual energy and the mobility cost (shortest distance the helper node has to travel to reach its final destination). The dying node receives multiple Panic reply messages and it chooses a node with maximum utility (residual enegy - mobility cost) and notifies the selected node to move. As the protocol relies on broadcast of a Panic request message by a dying node, it will not work when nodes gets physically destroyed creating un-repairable coverage holes.
In other significant research Howard et al. [14] proposed a potential fields based approach for self-deployment of mobile sensor networks. Nodes are treated as virtual particles and the virtual forces due to potential fields repel the nodes from each other and obstacles. There are viscous friction forces that will eventually bring the node to an equilibrium state. The authors assume that each sensor is capable of determining the range and bearing of both its neighbor nodes and the obstacles. This approach does not require any communication among the nodes for movement or localization information, instead the nodes only use their sensed information in making the decision to move making it a cost effective solution to the coverage problem. However, extensive simulation/experimental studies have not been conducted to test the sensitivity of the approach to changes in communication and sensing ranges and different network sizes etc.
In other related work by the same authors [15], an incremental self deployment algorithm is proposed that maintains the line of sight relationships among the nodes to solve the network coverage problem. The line of sight relationship is necessary for localizing the nodes by using existing deployed nodes as landmarks without using costly GPS. The algorithm deploys each node one at a time with the constraint that each node must be visible to at least one other node. The target position of the next node is calculated,

using the deployment algorithm, at a high processing power base station and the next node waiting to be deployed is communicated its final target position by the base station. Each deployed node is responsible for communicating its local information back to the base station for utilization in the next iteration of the deployment algorithm. This implies that each node has to maintain bidirectional communication with the base station at all times or else they are deployed so that they form a multi-hop connected network at all times. If an already deployed node obstructs the new active node then the algorithm provides for re-assignment of new targets to a number of previously deployed nodes making them active once again.
The authors have conducted simulations based on achieved coverage and time. Time metrics include the computation time and the actual deployment time. Different goal selection policies for the deployment have been proposed and evaluated. Results have shown that deterministic goal selection policy outperform stochastic based policies and can achieve about 80% of the coverage obtained by best greedy algorithm. Also the computation time is shown to be a polynomial function of the number of deployed nodes with the goal assignment phase scaling as n^2 where n is the number of deployed nodes. The incremental scheme using sequential deployment of nodes is quite slow compared with distributed self deployment proposals and will take much longer time to deploy when the number of nodes is increased.
Heo et al. [16] proposed two schemes for single coverage problem. In one scheme called Distributed Self-Spreading algorithm (DSS) the authors propose a deployment scheme similar to [14] and VEC [13]. Sensors are assumed to be randomly deployed initially. They start moving based on partial forces exerted by the neighbors. Forces exerted on each node by its neighbors depends on the local density of deployment and the distance between the node and the neighbor. Oscillation control and stability checks are applied at each iteration of the algorithm. The algorithm thus tries to minimize the size of the coverage hole in each iteration.
Another scheme called Intelligent Deployment and Clustering Algorithm (IDCA) is also proposed in [16] to utilize low energy consumption characteristics of local clustering. Local density is compared with density expected when all nodes are uniformly distributed in the target area and for close values a sensor selects the clustering mode. In this mode the relative remaining energy of the sensors dictates whether a node should move or not. The idea is to reduce the variance in remaining energy once all the nodes are uniformly distributed in the target area.
To summarize, the solutions discussed in this subsection either utilizes techniques from computational geometry e.g. Voronoi diagrams or uses virtual forces, to attract or repel neighbors, to achieve the desired coverage of the area.

3.2 Hybrid Sensor Networks

In the previous section we discussed various protocols that handled the coverage hole problem based on the assumption that all sensors are mobile and have equal resources. The coverage scenario when only some of the sensors are capable of moving has been under active research, especially in the field of robotics for exploration purposes. The movement capable sensors can help in deployment and network repair by moving to appropriate locations within the topology to achieve desired level of coverage and connectivity and to connect a possibly disconnected network.

Corke et al. [19] address the issue of network deployment with adequate connectivity using an Unmanned Aerial Vehicle (UAV). The flying robot, referred as AVATAR by the authors, can deploy the network based on a precomputed network topology. The sensors, once deployed, compute their connectivity map and relay this information to the flying UAV. The existing network connectivity is compared with the desired topology and the separation regions are identified. Deployment points within this region of separation are computed to repair the network and the UAV again deploys more sensors at desired deployment points. The deployment of additional sensors in the target area increases the sensor density(achieving multiple coverage and connectivity) or repairs failing sensor nodes in the network.

The experimental results during the network deployment stage show that it is very difficult to achieve the desired network topology using aerial deployment. The experiments to achieve a grid topology with 2m by 2m grid spacing resulted in a median deployment error value of 1.2m. This basic limitation makes it interesting to evaluate the proposed deployment scheme in terms of number of nodes required to achieve a desired degree of coverage and connectivity.

In [20], Batalin et al. suggested a combined solution for the exploration and coverage of a given target area. The coverage problem is solved with the help of a constantly moving robot in a given target area. The underlying assumption is that the area is large enough such that a constantly moving robot is required to cover the target area. The mobile robot first performs the network deployment in the target area as it explores the unknown environment. The deployed nodes then guide the robot, based on their local measurements, to poorly covered areas. The mobile robot, using its local sensing data and the recommended direction acquired from a deployed sensor node, decides about its future direction for exploration. The recommendations from the sensor nodes are like signposts, suggesting the least visited direction to the robot. If an obstruction is observed by the robot in the recommended direction, it asks the node for a new direction. When the robot does not receive a direction beacon when it has traveled a predetermined distance in one direction, it deploys another sensor node to improve the local coverage of the unknown area.

The algorithm does not consider the communications between the deployed nodes. All decisions are made by the robot by directly communicating with a neighbor sensor node. Distributed computation and in network processing is suggested in [20] as a solution for the homing problem (when a robot wishes to return to a specific point in the target area). The deployment strategy and especially the network repair policy can also benefit from the multi hop information derived out of a communicating sensor network and thus we feel that this should be explored further.
Wang et al. [21] address the coverage problem by moving the available mobile sensors in a hybrid network to heal coverage holes. The static sensors detect their local coverage holes by using Voronoi diagrams as in [13]. The mobile sensors also calculate coverage holes formed at their current position if they decide to leave their current position. The static sensors bid for the mobile sensors based on the size of their detected coverage hole. A mobile sensor compares the bids and decides to move if the highest bid received has a coverage hole size greater than the new hole generated in its original location due to its movement. The bids are broadcast locally up to two hops and the static sensors are able to direct neighboring eligible mobile sensors to a point close to the farthest vertex in their Voronoi polygon. However, the local broadcast may prevent the bid messages reaching mobile sensors if they are located farther than two hops. Also, the proposed protocol do not address the scenario when initial deployment is clustered and most of the sensors have no coverage holes to report.
Mobile sensors in the bidding protocol move iteratively in a zig-zag path before settling down to their final positions. As an improvement to the baisc bidding protocol Wang et al. in [22] proposed a proxy-based sensor deployment protocol. In each round of the bidding protocol, mobile nodes choose a static sensor closest to its logical position as its proxy. This proxy static node delivers the position realted information and process the bids received on behalf of the mobile sensors. This logical movements is shown to save energy due to lesser physical movement of the sensors.

3.3 Static Sensor Networks

In this section we cover protocols that try to achieve the desired degree of coverage and connectivity without relying on the mobility capabilities. The problem of getting the desired level of coverage and connectivity with minimum number of sensors is commonly referred to as topology/density control in sensor networks. In density control, given the requirement of coverage and connectivity, the aim is to select a minimal number of on-duty nodes that are active at any time out of the available densely deployed nodes. This node scheduling is feasible as long as no coverage holes appear due to nodes being turned off for energy savings. The duty-cycling of nodes also depends on whether all of the deployed sensors have the same sensing range or different

ranges. Efforts like [8], [4], [23] etc. have focused on providing the desired multiple level of coverage in a given target area while [24], [25], [26] etc. address single coverage problem.

Authors of [8] Wang et al. presented the Coverage Configuration Protocol (CCP) that can provide flexibility in configuration of sensor network to self-configure for different degrees of coverage. The underlying assumption is that the sensor density is enough to support the highest degree of coverage desired by the application. The idea is to maintain the minimum topology of the network that is just sufficient to provide the desired connectivity and degree of coverage by turning redundant nodes off through the protocol. The relationship between the degree of coverage and connectivity has been shown to be different for sensor nodes whose sensing circle intersects with the boundary, referred to as *boundary nodes* and the *interior nodes*, nodes have a sensing circle clear of any boundary. The authors proved that for *boundary nodes* desired connectivity is equal to the degree of coverage while for *interior nodes* the desired connectivity is twice the degree of coverage. Each deployed node runs the K_s-coverage eligibility algorithm when $R_c \geq 2R_s$, where R_c is the communication range and R_s is the sensing range. Given a requested coverage degree K_s, a sensor node is scheduled to sleep if every location within its coverage range is already K_s covered by other active nodes in its neighborhood. For cases when $R_c < 2R_s$, CCP does not guarantee connectivity along with the coverage. Authors have proposed combining CCP with an existing connectivity maintenance protocol, SPAN [27] which provides the communication connectivity.

Huang et al in [4] proposed polynomial-time algorithms to verify whether every point in the target area is covered by at least the required number of nodes. Algorithm k-UC assumes a uniform circular sensing disk while k-NC assumes a non-disk sensing range for each sensor node. The proposed algorithm requires only the location information of each deployed node to evaluate the desired multiple coverage. For k-UC, each node calculates the coverage of its neighbor only if the neighbor lies within twice the sensing range of the node. For k-NC, different neighbor selection rules are defined for cases when one of the nodes is within the sensing range of other nodes and, when one/both of the nodes are within sensing range of each other. The node then calculates its perimeter coverage by finding the sector of its coverage area occupied by the neighbor sensing range. The node thus verifies whether its whole perimeter 2π is covered by existing neighbors to the required degree or not. To detect coverage holes, the authors suggest a central controller entity that can collect the details of insufficiently covered segments from each node and can dispatch new nodes to cover that existing hole. However, this centralized approach lacks scalability.

Yan et al. in [23] proposed a distributed density control algorithm capable of providing differentiated coverage based on different requirements in different parts of the deployed sensor network. The algorithm is based on time

synchronization among the neighbors. In the initialization phase, nodes get their location and synchronize with neighbor nodes. In the sensing phase, comprising of several rounds of equal duration, each node divides its whole area into grids and advertises its reference point and, start and stop time, defined with respect to that reference point. The node sorts the neighbors covering a particular grid in ascending order of their reference points in a round. Based on the time sequence obtained, a node can decide its on-duty time such that the whole grid still gets the required degree of coverage. The results from all the covered grids are merged to find the adopted duty schedule for the node.

For single coverage requirement, Zhang et al. in [24] have proposed the Optimal Geographical Density Control (OGDC) protocol. This protocol tries to minimize the overlap of sensing areas of all sensor nodes for cases when $R_c \geq 2R_s$. The algorithm starts with all the nodes initially in "UNDECIDED" state. A node with sufficient power is randomly chosen to start the process of node selection. This starting node broadcasts a power-on message. Nodes, on reception of this power-on message, check their power level and the existing coverage of area under their sensing range. If sufficient power is available, and the area is not fully covered, the node adds the starting node as the neighbor, sets its state to "ON" and broadcasts the power-on message again. This process continues with slightly different behavior for power-on messages received from starting and non-starting nodes. Simulation results presented by the authors of [24] shows that OGDC protocol cannot always preserve the original coverage of the network completely.

The issue of single coverage with different sensing capabilities has also been addressed in [25] by Tian et al. The authors discussed the topology control for both same and different sensing ranges. A node decides to turn off after discovering that its neighbors (sponsors) can completely cover its sensing area. The node credits the sponsored area based on sectors instead of the actual crescent formed in its sensor range. Once the sponsored area becomes equal to the sensing area under the node, the node qualifies as a candidate to be switched off. To avoid the possibility of multiple neighbors turning off and creating a coverage hole, the nodes use a random back-off algorithm before going to sleep.

In the uniform sensors case, the assumption that a node's coverage is only influenced by neighbors within the sensing range of the node in our opinion is overly simplistic. One can easily prove that even if we have only one neighbor within the sensing range of node i that does not fully cover the node i coverage area, the whole sensing area of i can still be covered by other nodes not lying within the sensing range of i.

Jiang et al. in [26] extend the sponsor area based node scheduling algorithm of [4]. They identified two short-comings in the sponsored area approach. Firstly, neighbors lying outside the sensing range are not considered although they can contribute to the node coverage. Secondly, the sector based area

Category	*Approach*	*Proposed Solution*	*Main Assumptions*	*Characteristics*
Mobile Sensors	Computational Geometry	VEC, VOR, Minmax [13]	Location information	Localized, scalable, distributed.
		Co-Fi [18]	Location information. Nodes can predict their death	Single coverage based. Residual energy considerations.
	Virtual Forces	Potential Fields [14]	Range and bearing	Scalable, distributed. No local communication required for localization or movement.
		DSS,IDCA [16]	Location information	Scalable, distributed, residual energy based.
	Sequential	Incremental [15]	Line of sight for localization	Centralized. Bidirectional communication with base station.
Hybrid Sensors	Single Mobile Sensor	UAV [19]	Predetermined topology information	Flying robot for deployment and network repair. Inaccuracies using aerial deployment
		Single Robot [20]	Location information	Distributed, no multihop communication for network deployment and repair.
	Multiple Mobile Sensors	Bidding Protocol [21], Proxy-based [22]	Location information	Uses Voronoi diagram. Logical movements in proxy scheme.
Static Sensors	Multiple Coverage	CCP [8]	Location information, uniform sensing disk	Configurable degree of coverage, calculated by intersection points of sensing circles.
		k-UC, k-NC [4]	Location information	Perimeter coverage, non-disk sensing model supported.
		Differentiated [23]	Location information, time synchronization	Grid based differentiated degree of coverage.
	Single Coverage	OGDC [24]	Location information, uniform sensing disk	Residual energy consideration.
		Sponsored Area [25]	Location information	Sector based coverage calculations, non-disk sensing model supported.
		Extended-Sponsored Area [26]	Location information, time synchronization	Uniform disk sensing model.

Table 1: Comparison of proposed solutions to coverage hole problem

calculation for coverage results in a more conservative estimate of the neighbors contributions in covering the area. The authors introduced the

concept of effective neighbor nodes for calculating the nodes coverage accurately and to decide upon redundant nodes that could be put to sleep while conserving the original coverage. The neighbor set now includes all nodes within twice the sensing range of each node. Their proposed solution assumes time synchronization among neighbor nodes and regular disc node coverage. The results presented in [26] show that the proposed protocol is able to outperform the sponsored area approach by about 30% in terms of reducing the actual number of nodes required for maintaining the original coverage. Other protocols like Adaptive Self-Configuring Sensor Networks topologies (ASCENT) [28], SPAN [27], Geographical Adaptive Fidelity (GAF) [29] and Probing Environment and Adaptive Sleeping (PEAS) [30] also address the topology/density control along with routing to conserve energy. But with little or no attention to maintaining the desired coverage of the target area. There are other papers in the literature that address the coverage issue but with a different definition of the coverage problem. They consider coverage as minimum and maximum exposure paths and consider it as a metric to indicate the probability of tracking a moving object through the deployed sensor network. The discussion of such protocols is out of scope to this report.

Table 1 summarizes the proposed solutions for coverage holes problem. Most of the proposed solutions assume uniform sensing disks and that all the sensors are location aware. The single coverage problem has been the focus of study in existing literature and many solutions have been proposed to solve this problem using mobile or hybrid sensors. There have been some research effort to address the multiple coverage requirements with static sensor nodes but we didn't find any research effort addressing the multiple coverage requirement with hybrid or mobile sensor networks. The issue of using mobile or hybrid sensors to address the multiple coverage requirement is further discussed in Section 7.

4 Routing Holes

The previous section introduced various coverage and connectivity issues in wireless sensor networks. We discussed solutions to guarantee connectivity and coverage in the target area. This section details proposed solutions for the routing hole problem. Assuming that the network has been successfully deployed, the aim is now to route data packets successfully and efficiently through the sensor network, bypassing any routing hole that may appear in the network.

Information driven and data centric routing has been the focus of many research efforts in wireless sensor networks [31], [32], [33] etc. Two popular fault tolerance schemes used in proposed protocols are single-path routing with route repair and multipath routing. Geographic greedy routing has also

been proposed as a scalable and localized solution to address the routing issues associated with MANETs [34] and is a viable solution for routing in large sensor networks. In this section we classify the routing holes problem in three categories. Issues of routing holes are discussed single path routing, in multipath routing and geographic greedy routing.

4.1 Single-path Routing

This section covers single path routing protocols for wireless sensor networks where resilience is provided by different path repair strategies. Single path routing algorithms for wireless sensor networks is a well investigated area and there are a number of routing protocols that have been proposed. We only discuss some of the representative proposed routing protocols
Intanagonwiwat et al. [31] were the first to propose a data centric communication protocol for wireless sensor networks called Directed Diffusion (DD). In the data centric paradigm of DD, data is named using attribute-value pairs. The sink generates an *interest* for the named data that is diffused through the whole sensor network. The interest message contains an initially low data rate as an attribute interval. Each node establish a gradient toward its neighbor nodes from which it receives the interest. A node capable of producing the desired named data starts sending *exploratory data* to all its neighbors from which it has received interest for the data, at the rate specified in the interest message. This exploratory data follows the established gradients, in a multipath way, to reach the sink that initiated the interest message. The sink chooses to reinforce one particular neighbor to draw data, at higher data rates, by sending a *positive reinforcement* message that is unicast back to the sender. The sender will now start sending data at higher rates along the reinforced gradient to the sink. DD can be tuned to work as a multipath protocol if alternate paths are kept alive by sending reinforcement with different interval attributes to different neighbors.
Directed Diffusion works well in scenarios when there are few sinks as compared to a large number of senders but its overhead becomes high when the number of sinks increase [35]. Also DD is not suitable for short lived data flows.
In [35], Heidemann et al. proposed two new diffusion algorithms, Push and One Phase Pull(OPP) to cater for different application requirements. Push diffusion reverses the role of senders and sinks as compared to the original algorithm proposed in DD which is referred to as Two Phase Pull(TPP) by the authors. Senders in Push flood the exploratory data, sinks on receiving exploratory data send positive reinforcement to create reinforced gradient. Senders in OPP selects a preferred gradient when it receives multiple interest messages and instead of sending exploratory data, it starts sending data at a higher rate on its selected low latency path. OPP does not require positive reinforcement as in TPP.

After having explained the three variants of the directed diffusion algorithm, we now evaluate their resilience to node failures and routing holes in particular. TPP recovers from failed path by periodic flooding of interest and exploratory data. The choice of refresh rates for the interest and the exploratory data(interest refresh rate is 30s, less than the exploratory data refresh rate is 90s) thus controls how quickly a failed path is discovered and bypassed. The flooding nature of these refresh messages makes sure that the alternate path, if exists, is found though at high cost. The algorithm also provides for local repair by intermediate nodes by sending reinforcement messages. If the interleaved alternate paths are kept alive in TPP by sending reinforcement with different interval attributes to different neighbors, the alternate path can be quickly chosen and new reinforcement with higher data rate can be sent along the alternate path to setup reinforced gradient path along the new route. In Push diffusion, the recovery part depends on the periodic exploratory data flooding that typically is 3 times the interest refresh interval of TPP. Thus Push is expected to take much longer to select another route in case of route failure than TPP.
The interest flooding overhead in DD becomes high when the number of sinks increase [35]. Also the event flooding proposed in push diffusion is only viable when number of sinks interested in the event is much more than the number of sources. Braginsky et al. in [36] proposed Rumor Routing, to address these overheads. The rumor routing protocol creates paths leading to each events, instead of flooding the event in the entire network, by using long-lived protocol packets called *Agents*. Whenever a sink creates a query, it goes through random walk until it finds the established event path, which it can then follow to the event. Thus network wide flooding is avoided both for event and query as compared to the three variant of directed diffusion, TPP (both interest and exploratory data are flooded), Push diffusion (only exploratory data is flooded) and OPP (only interest is flooded).
In rumor routing if the node that originated the query finds out that the query did not find any established event path (through timeout based on TTL values), it can retransmit the query (another random walk) as chances of finding the event path grows exponentially with the number of retransmission, if such a path does exist. As a last resort, the query can be flooded to find the event at the cost of high overhead. If case of routing holes, higher TTL values or flooding can route across the hole to find the event.
Luo et al. in [32], proposed Two Tier Data Dissemination protocol (TTDD) that divides the whole topology into cells using a grid structure. The grid structure is constructed from the perspective of sender using simple geographical greedy forwarding so each sender will form its own grid to disseminate data to sinks. The protocol selects forwarding points called Dissemination Nodes (DN) at the sensors that are closest to the grid points. A query from a sink now travels two tiers to reach the sender. At the lower tier, query is flooded by the sink within its own cell only until it reaches its

nearest DN. The second tier consists of routing along the DN such that it reaches either the source or a DN that is already receiving data from the source. During query forwarding a reverse path is formed toward the sink that is used to actually forward the data to the sink.
The protocol is optimized to work with multiple mobile sinks. As long as the mobile sink is within the cell from where it first originated the query, *trajectory* data forwarding is used to deliver data to the mobile node. The mobile node selects a primary agent in its cell initially and upon moving it keeps its primary agent informed about its current location through a neighboring node called immediate agent analogous to the concept of home and foreign agents in Mobile IP. When a node moves out a cell distance from its primary agent it picks up a new primary agent and again floods a query.
Critical to the performance of the TTDD protocol is the choice of the cell size. A bigger cell size will result in more flooding area for the query and lesser number of DN for each source. The size of cell is static and assumed known to the sources as well as sinks. Sinks use this information to limit the flooding of queries. Queries and consequently data may take suboptimal paths traversing the DNs of the grid topology. The grid is not periodically refreshed during its lifetime, instead TTDD uses *upstream information duplication* by selecting various neighbors as recruited nodes to replicate the information of upstream DN. In case the DN fails, one of its recruited neighbor can take over the role of DN for its cell. The problem turns severe in case a routing hole forms covering all the recruited nodes as well as the DN.
Tian et al. [37] proposed a local pivot-initiated path repairing approach while using single path routing in wireless sensor networks. A node located immediate upstream to a failure point, called pivot node, is responsible to find alternate paths through local interaction. The pivot node broadcast Help Request (HREQ) message to its neighbors. A neighbor that can provide an alternate route to the destination replies with Help Response (HREP) message. In case no HREP message is received, the pivot node returns the packet to the node from which it initially received the packet. The upstream node now tries the local repair excluding the node that returned the packet back. This process continues until a route to destination is found or the packet is traced backed to the source without finding any suitable path. The approach seems more energy efficient but requires each intermediate forwarding node to incorporate some kind of feedback mechanism to ensure that data is delivered to its successor.
We have discussed, in this subsection, single path routing protocols that use one or more of the following path repair strategies: retransmission, source initiated path repair, intermediate node initiated local repair, routing information duplication and flooding.

4.2 Multi-path Routing

The basic idea behind multipath routing is to maintain alternate routes to provide resilience against node failures or to distribute the traffic among multiple paths for load balancing and uniform energy consumption in the network.

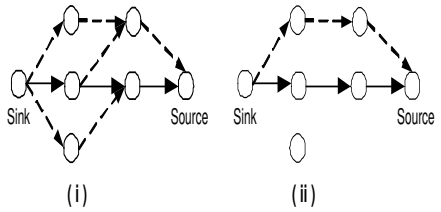

Figure 4: (i). Braided multipath (ii). Node disjoint multipath

Ganesan et al. [38] proposed the use of multipath routing for data dissemination in wireless sensor networks. They considered two approaches of multipath routing namely, node-disjoint multipath routing and braided multipath routing. The node-disjoint multipath routing requires that the alternate paths and the original path are mutually exclusive i.e. they do not intersect each other. The braided paths consists of many interleaving paths (see Figure 4). The original best path is referred as primary path in the paper. The authors has used isolated failure and patterned failure(geographically correlated failure) as the failure model to test the resilience provided by the two types of multipaths. The node-disjoint multipath can tolerate any number of nodes failure on the primary path but fails if a single node on each alternate path fails while the braided multipath can provide an alternate route in this scenario. In case of patterned failure(a routing hole), alternate paths that are geographically closer to the primary path are more likely to be affected.

The authors proposed localized algorithms for the computation of braided and node-disjoint multipaths and have shown through simulations study that for isolated failures the braided routing out performs the node-disjoint multipath routing, while in case of routing holes both braided and node-disjoint gives comparable performance but the braided multipath has about one third of the overhead for alternate path maintenance. If the braided or node-disjoint multipath fails to find an alternate path after failure, it resorts to network wide flooding to probe for other routes that may be available for use.

In [33], Ye et al. proposed a multipath routing protocol named Gradient Broadcast (GRAB) designed specifically for robust data delivery. The whole protocol relies on two parameters, the *cost field* and the *credit*. The sink propagates advertisement packets in the network and each node that re-

ceives this advertisement records the cost to reach the sink along a path. At the end of the cost field setup phase each node have the minimum cost needed to reach the sink. When a node sends a packet to the destination, it simply includes its own cost in the packet and broadcast it to its neighbors. Only the neighbor with smaller cost than the cost contained in the packet are allowed to forward it, rest all drops the packet.

The multipath aspect of the GRAB protocol is controlled by credit. Credit is the extra allowance that allows the use of interleaved paths each of which has cost less than the cost+credit alloted by the source of the packet. Thus the amount of credit actually controls the width of the mesh, the wider the mesh, the more robust it would be at the cost of increased overhead. The design of GRAB provides for extra consumption of credit in the beginning hops from the source so that the mesh is wider in the beginning and nodes closer to the sink consume less credit resulting in a narrower mesh. Thus the location of the routing hole will dictate how GRAB copes with it. If the routing hole is formed near the sender, there are more chances of bypassing it due to the extra width of mesh available near the sender, while hole forming near the sink will have more profound failure effect.

In [39], Dulman et al. explored the tradeoff between degree of reliability and amount of traffic overhead for multipath routing. The authors present a scheme for delivering data reliably in spite of route failures. They propose to split the data packet into k sub-packets using forward error correcting codes (FEC) to add redundancy to the original data. The factor k is dependent on the number of disjointed paths from source to destination and the failing probability of the alternate paths. The destination then affords to miss some sub-packets but can still reconstruct the original message. The simulation results show that instead of transmitting the whole message on all the alternate disjointed multiple paths, transmitting the sub-packets reduces the overhead significantly.

Rahul et al. [40] proposed the use of multipath routing to uniformly distribute the usage of energy in wireless sensor networks. The idea is that if the low cost path (lowest energy consuming) is always used, nodes on this path will soon deplete their energy and can even result in network partition. Their proposed protocol keeps a set of good paths for the same destination and probabilistically choose different paths at different times. The alternate paths maintained are sub-optimal paths as compared to the lowest cost path but distributing the traffic across different paths results in more even distribution of energy usage and prolongs network lifetime of deployed network. In case of nodes failures and routing holes, the protocol can switch between different maintained paths and in event all available paths fails due to wider hole, periodic flooding of interest is the last hope to find any available path to the destination.

A critical parameter affecting the resilience of the multipath routing is the choice of the degree of multipath that trades the degree of resilience with

maintenance cost. We believe the degree of resilience required is application as well as traffic type dependant. The multipath routing is a viable solution to provide robust routing keeping in mind the failure prone behavior of sensor nodes and the inherent characteristics of the wireless medium but in our opinion the degree of resilience should be configurable by the application.

4.3 Goegraphic Routing

Geographic routing relies on greedy forwarding to route packets by only making local decisions. The node making the routing decision only needs to know the geographic location of the destination(usually carried in the packet header or acquired by some location service like [41]), itself(through GPS or other GPS-less techniques) and all of its one hop neighbors(neighbor database with locations). Protocols proposed in [42] and [43] explore the use of geographical greedy forwarding assuming either no location information is available or only a subset of the nodes participating in greedy forwarding knows their location.
Karp et al. in [34] details Greedy Perimeter Stateless Routing (GPSR) for MANETs. The protocol starts in greedy forwarding mode. GPSR recovers from routing holes due to local minimum phenomenon by using perimeter routing mode. In perimeter routing the *right-hand rule* is used which states that when arriving at node x from a node y the next edge traversed is the next one sequentially counterclockwise about x from edge xy. The right-hand rule requires that all the edges are non-crossing. GPSR proposes either Relative Neighborhood Graph(RNG) [44] or Gabriel Graph(GG) [45] to get a planar network graph with no crossing edges. GG is a geometric graph in which the edge (u, v) is present if and only if circle with diameter $d(u, v)$ contains no other vertices of the graph while in RNG an edge (u, v) exists if the distance between them $d(u, v)$ is less than or equal to the distance between every other vertex w and whichever of u and v is farther from w.
Once a node finds out that it can no longer use the greedy forwarding, the perimeter forwarding mode is followed. Based on the planar graph achieved through either RNG or GG, a node first finds out the edge of the graph that crosses the line from the node to the destination using the right-hand rule. From that edge the next edge is again found using the right-hand rule. The packet can return to greedy mode anytime when the node making the forwarding decision finds out that its distance to destination is less than the point where perimeter routing started. Each node works in the promiscuous mode to capture all kinds of traffic in its radio range. All traffic carries the sender and destination location information and thus location information of neighbors can be easily extracted without additional overhead. Specific beaconing to obtain this information is not required in regions of active data forwarding.
RNG and GG offer different densities of connectivity by eliminating different

number of redundant links. Specifically RNG is a subset of GG and both take time $O(K^2)$ at each node where K is the nodes original connectivity. The maintenance of planar graph at each node introduces overhead. While all the nodes maintain the planar graph all the time, this information is only used if a local minimum phenomenon for a specific destination occurs at a specific node. Also the underlying assumption of symmetric radio ranges for the construction of planar graph is contradictory to the real world deployment.

Kranakis et al. in [46] proposed the Compass Routing II algorithm that guarantees that the destination is reached even when local minimum phenomenon occurs in greedy forwarding. They proposed the use of least deviation angle from the line joining the node to the destination when trying to route a packet to the next hop. Compass routing is augmented by the Delauney triangulation to correctly route packets in some network topologies where simple least deviation angle results in loops.

Similar to the work in [46], Bose et al. [47] proposed the FACE-1 and FACE-2 routing algorithms to guarantee packet delivery in MANETs. The suggested solution is also based on getting a connected planar subgraph by using Gabriel Graph and then traversing the edges of the graph using right-hand rule. In contrast to GPSR, all routing is done through the perimeter of the GG formed at each node. FACE-2 modifies FACE-1 in that the perimeter traversal follows the next edge whenever that edge crosses the line from the source to destination. The routing hole problem is addressed by always using the perimeter mode routing, however the nodes in the perimeter depletes more energy.

As an extension to the compass routing II algorithm, [48] introduced deterministic fall back mechanism to get back in the greedy mode from the perimeter routing mode without necessarily exploring the complete face boundary. The proposed algorithm, called GOAFR$^+$, first constructs the Gabriel Graph (GG) of the network. It starts forwarding in a greedy mode and when it reaches a local minimum point, it switches to the face routing of [46]. The algorithm uses two counters to keep track of how many visited nodes in face routing are nearer to the destination and how many are far from the destination as compared to the starting node. Based on the values of these two counters the algorithm decide whether to continue in face routing mode or fall back to the greedy phase. The authors proved that their proposed algorithm is not only efficient on practical average-case networks but also in theory it is asymptotically worst-case optimal.

In [43], De Couto et al. proposed a probabilistic solution called Intermediate Node Forwarding (INF) for routing around holes assuming non-uniform radio ranges. Negative acknowledgment packets (NAKs) has been proposed in the basic geographic forwarding to provide feedback to the source about packet drops due to local minimum occurring at any intermediate node. Once the sender knows about packet drops, it randomly selects an inter-

mediate location from a disk, of radius one quarter of the distance between the sender and the destination, centered at the midpoint of the distance between the sender and the destination. Packets are routed from the source to the intermediate point using geographic forwarding and from the intermediate node to the destination again using geographic forwarding. The idea is to find an intermediate node that does not suffer from the local minimum phenomenon and that can actually route around the routing hole. If the selected node drops the packet again, the radius of the disk is doubled and another random intermediate location is selected.

Some variations of the basic INF protocol are also suggested with timeouts replacing the NAKs and disk radius only doubling after k NAKs instead of reacting to a single NAK. Also selecting multiple intermediate locations is suggested for providing higher level of fault tolerance. Incorporating the NAKs in the geographical greedy forwarding will increase the protocol overhead. On the other hand if the information of the node dropping the packet is available then a more realistic approach could be to avoid the region around the node dropping the packet instead of trying to route around the midpoint as suggested in the INF. Furthermore the assumption of NAKs reaching the source is not valid if asymmetric links are considered. A more efficient scheme would incorporate routing around the hole without tracing back to the sender for route repair.

Fang et al. in [49] also addressed the problem of locating and bypassing the routing holes formed due to local minimum phenomenon in geographical greedy forwarding. They define *stuck nodes* as nodes in the topology where packets can possibly get stuck in greedy multi-hop forwarding due to local minimum and propose the TENT rule to test whether a node in a given topology is a stuck node. They also came up with BOUNDHOLE distributed algorithm to find the boundary of the routing hole.

The TENT rule at each node maintains location information of all 1-hop neighbors of the node in counter-clockwise order. Then for each pair of adjacent nodes u, v for node x the perpendicular bisector of ux and vx is drawn that intersect at a point O. If O is inside the communication range of x, then x is not a stuck node(see Fig 5). A node can thus calculate its stuck directions by applying the local check for all pairs of adjacent neighbors. The TENT rule is computed locally at each node with only 1-hop neighbors information in contrast with the planar graph approaches. The BOUNDHOLE algorithm uses the right-hand rule after each node has identified its stuck direction, if any. A messenger packet is initiated by a stuck node to mark the boundary of a hole in the direction of a stuck angle. This packet is sent to the second neighbor in counterclockwise order facing the direction of the stuck angle, this process continues with special processing for crossing edges till the messenger packet returns to the originating node by completing the cycle.

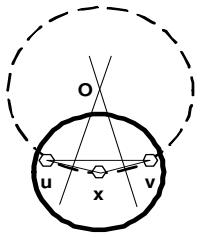

Figure 5: Illustration of the TENT rule. Solid circle represents node x's transmission range

The nodes on boundary of a hole can store either the complete information about the boundary or partial information like upstream and downstream neighbors depending on the application requirements. When routing, at a node x that is a stuck node, the packet is routed along the boundary of the hole until the packet gets to a node whose distance to destination is closer than x. From this point the packet again follows greedy forwarding. In a special case if the packets traverse the whole boundary of the hole without finding a suitable node with lesser distance to destination, restricted flooding is done to cover the chance of destination lying within the hole closer to x than any other node on boundary of the hole still outside the communication range of x.

Yu et al. [50] proposed Geographic and Energy Aware Routing (GEAR) to route a packet toward a region of interest. GEAR works in two phases, in phase I it uses energy aware next hop neighbor selection to route a packet toward a target region while phase II involves restricted flooding or recursive geographical forwarding to disseminate the packet inside the region. The protocol assumes bi-directional links and that location as well as residual energy information of all one-hop neighbors is available. For phase I, each node keeps two kinds of costs, an estimated cost and a learned cost of reaching the destination through its neighbor. Each node gets the learned cost to a region of interest from all of its one-hop neighbors and it can compute its own learned cost by adding to the selected neighbor cost, the cost to reach that neighbor. The estimated cost depends on the node's residual energy and its distance to the destination and is used as default value in case learned cost is not available.
In case there are no routing holes, the estimated cost is equal to the learned cost. When a routing hole appears, the change in the path is reflected in the new learned cost of the node that encounters the routing hole. This new learned cost is also propagated back to be utilized for early avoidance of the routing hole so that nodes surrounding the routing hole do not get depleted at a fast rate. In phase II, for high-density networks, recursive geographical forwarding is used as it is more energy efficient than restricted flooding. GEAR works well if the region to be covered is a small fraction

Category	*Protocol*	*State maintained*	*Fault tolerance with routing holes*
Single Path	DD [31]	Interest gradient and data path	Periodic flooding of interest and exploratory data. Local repair by reinforcement messages.
	OPP, Push [35]	Interest/event gradient	Periodic flooding of interest/event
	Rumor Routing [36]	Event path (Agents)	Retransmission. Flooding in extreme case.
	TTDD [32]	Per source DN information	Recruited nodes for each DN
	Pivot based [37]	Only path repair strategy specified. Uses any existing routing protocol	Local repair attempted followed by notification to upstream node
Multipath	Braided [38]	Multiple paths maintained	Braided/node-disjoint alternate paths available
	GRAB [33]	Cost field	Alternate paths based on available credit
	FEC based [39]	Multiple paths maintained	FEC for message reconstruction at destination
	EAR [40]	Multiple paths maintained	Alternate paths and localized flooding
Geographic Routing	GPSR [34]	Location information and planar graph	Right hand rule using planar graph to avoid holes
	Compass Routing [46], FACE-I, FACE-II [47], GOAFR+ [48]	Location information and planar graph	Face routing to avoid holes using planar graph
	INF [43]	Location information	NAKs and source initiated repair
	TENT rule [49]	Location information, boundary of holes(perimeter nodes only)	TENT rule to identify holes. Boundary information maintained to avoid the hole.
	GEAR [50]	Learned and estimated cost	Limited flooding in region, learned cost helps find alternate routes

Table 2: Comparison of some proposed routing protocols

of the total area covered by the sensor network but the protocol efficiency tend to diminish for cases when whole region is the area of interest.
Table 2 lists comparison of the proposed routing protocols that have been discussed in this Section. The mechanism to achieve fault tolerance against the routing hole problem and the state maintained are highlighted for each of the protocol.

5 Jamming Holes

A jamming hole differs from other types of holes that can exist in the sensor network in that it circumvents the ability of nodes in a specific area to communicate/sense. The nodes are assumed to have ample energy and resources but their ability to sense an appropriate event(jamming of sensing capabilities) or communicate the event to a sink(communication jamming) is compromised by the effect of appropriate jamming techniques.
The schemes that are usually employed to combat jamming include the use of various spread spectrum techniques for radio communications and using different transmission media like infra-red or optical combined with radio. The prohibitive factor in adapting these countermeasures is the cost and complexity. We assume for the rest of this section that no such counter-jamming technique is available to the deployed network.
Wood et al. [9] proposed a protocol called JAM, to detect and map jammed regions in a sensor network. The detection part of the protocol applies heuristics based on available data, e.g. bit-error rates etc., to distinguish jamming from normal interference. The critical part of the detection phase is the selection of a certain threshold value above which the interference is declared as jamming. Once jamming is detected, the protocol assumes a carrier sense overriding mechanism to send a brief, high-priority broadcast message referred as JAMMED to its neighbors.
The mapping part of the protocol starts when a node outside the jammed region receives a JAMMED message from a node inside the jamming hole. The receiving node now broadcasts a BUILD message to its neighbors to start building the jamming hole boundary. Multiple BUILD messages originating from neighboring nodes are combined to form the boundary of the hole.
The JAM protocol assumes that the location information and unique ID is known to each node. The protocol also relies on the availability of a carrier-sense defeating broadcast mechanism to notify the jamming to un-jammed neighbors. This is, in our opinion, the most demanding requirement. The broadcast of the JAMMED message will succeed if the jamming is intermittent. For continuous channel jamming it is hard to guarantee that the JAMMED message will reach any of the neighbor node outside the jamming hole. Thus, a protocol that can detect the presence of a jamming hole without relying on help from the nodes inside the jamming hole is more desirable. This is further discussed in future research directions in Section 7.

6 Sink /Black Holes/Worm Holes

This section describes some proposed measures to mitigate the effects of sink/black hole and worm hole denial of service attacks in wireless sensor networks.

Karlof et al. [10] analyzed the resilience of various routing protocols and energy conserving topology maintenance algorithms against sink holes. They showed that popular routing protocols like directed diffusion, rumor routing and multi-path variant of directed diffusion etc. are all vulnerable to sink holes attacks. For geographical greedy forwarding algorithms it is more difficult to create sink holes because in this case a malicious node has to advertise different attractive locations to different neighbors in order to qualify as next hop.

Authors of [10] suggest authentication and link layer encryption as countermeasures against the sink hole attacks. This is essentially to prevent malicious nodes forming part of the topology and participating in the routing protocol for injecting incorrect routing information. However, the authorization mechanism and link layer encryption can fail to protect against worm hole attacks.

Wood et al. in [11] identified four possible defenses against the sink holes. In the authorization solution, only authorized nodes can exchange routing information with each other. The solution is not scalable due to high computation and communication overhead. Also, public key cryptography is not feasible in sensor networks given the capacities and constraints of the sensor devices. In another proposed solution, nodes can monitor their neighbor behavior to verify that the next hop does transmit the message just sent by the node. This scheme also fails when a malicious node is simply altering the contents of the message and forwarding it or forwarding the packets to another malicious node (in worm holes, if using the same communication channel) to avoid monitoring. The third proposed solution suggests that redundancy introduced by using multi path routing can help in avoiding the sink holes and worm holes especially when dis-joint multi path routes are maintained and actually utilized for data transmission. In another proposed solution in [11], probing by geography based protocols is carried out to detect the presence of sink holes. The nodes can periodically send probes across the network diameter to check the routes.

Almost all of the proposed protocols for wireless sensor networks assume a trust relationship among nodes. Security considerations are minimal, if any. Some applications i.e. Health monitoring and battlefield tracking etc. require a higher degree of security and resilience against various denial of service attacks. We agree with authors of [10] that security should be a built-in feature of these proposed protocols rather than an add-on.

7 Future Research Directions

This survey reveals a lack of research effort to address the multiple coverage requirement in sensor networks using mobility capable sensors. Also, we suggest that there is a need of more efficient localized solutions to address the detection and mapping of jamming holes in wireless sensor networks. In this section we discuss these future research directions in more detail.

7.1 Using Mobile Sensors to Provide Multiple Coverage

The basic assumption in static sensor networks that nodes are available to cover every point in the target region to provide the desired level of coverage is overly simplistic. This is particularly true for hostile or harsh environments, like battlefields and fire or chemical spill, where the network cannot be carefully deployed to a predetermined regular topology. Random aerial deployment of sensor nodes is a possible solution but in this case it is very difficult to guarantee that required multiple coverage is achieved.
We propose the use of mobile sensors in the sensor networks to target the multiple coverage problem. Mobile sensors can help achieve the desired multiple coverage in scenarios where it is impossible to guarantee perfect deployment. A significant reduction in required number of sensors is also expected compared to static sensors trying to cover the area using random aerial deployment. Use of mobile sensors can be considered in two different scenarios, namely hybrid and mobile networks. In hybrid networks, the sensor nodes can rely on coverage hole detection from static nodes while in mobile network solution all the sensor nodes dynamically select their final locations in the network with local interaction.

7.2 Efficient Detection of Moving Jamming Holes

We discussed the jamming holes and one proposed protocol, JAM, to detect and map the boundary of the jamming holes in Section 5. The assumption in the JAM protocol that nodes located within the jamming hole can actually override the jamming signals is unrealistic. Also, a moving jammer in the target field will leave the proposed protocol with incomplete information about the boundary of the hole. It would thus be interesting to explore the scenario of a moving jamming hole changing its position in the wireless sensor network. The jamming hole detection protocol should not only be able to report the existence of the jamming hole but also predict, in real time, the possible direction of the movement. We identify the need of an efficient jamming hole detection and reporting protocol capable of dealing with moving jamming holes.

8 Conclusion

We have provided a snapshot of the current state of the art of research in sensor networks dealing with various holes related problems. We discussed existing solutions and listed their behavior with holes such as coverage, routing, jamming, sink/black and worm holes. The literature review has also revealed some possible future research directions. These include utilizing mobile sensors for achieving multiple coverage and finding an efficient solution to tracking moving jamming holes in wireless sensor networks.

References

[1] Ian F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. A survey on sensor networks. *IEEE Communications Magazine*, pages 102–114, 2002.

[2] Chee-Yee Chong and Srikanta P. Kumar. Sensor networks:evolution, opportunities, and challenges. In *IEEE*, Volume 91 No 8, pages 1247–1256, August 2003.

[3] Deborah Estrin, Ramesh Govindan, John Heidemann, and Satish Kumar. Next century challenges: Scalable coordination in sensor networks. In *IEEE/ACM International Conference on Mobile Computing and Networking MobiCom'99*, pages 263–270, August 1999.

[4] Chi-Fu Huang and Yu-Chee Tseng. The coverage problem in a wireless sensor network. In *Proceedings of the 2nd ACM International Conference on Wireless Sensor Networks and Applications(WSNA)*, Sep 2003.

[5] D. Nicules and B. Nath. Ad-hoc positioning system (APS) using AoA. In *Proceedings of the IEEE INFOCOM*, 2003.

[6] Jerry D. Gibson. *Mobile Communications Handbook*. Springer-Verlag, 1999.

[7] Sachin Adlakha and Mani Srivastava. Critical density thresholds for coverage in wireless sensor networks. In *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC)*, 3, pages 1615–1620, March 2003.

[8] Xiaorui Wang, Guoliang Xing, Yuanfang Zhang, Chenyang Lu, Robert Pless, and Christopher Gill. Integrated coverage and connectivity configuration in wireless sensor networks. In *Proceedings of the ACM First International Conference on Embedded Networked Sensor Systems(SenSys '03)*, pages 28–39, Nov 2003.

[9] Anthony D. Wood, John A. Stankovic, and Sang H. Son. Jam: A jammed-area mapping service for sensor networks. In *24th IEEE Real Time System Symposium (RTSS'03)*, pages 286–298, Dec 2003.

[10] Chris Karlof and David Wagner. Secure routing in wireless sensor networks: Attacks and countermeasures. In *1st IEEE International Workshop on Sensor Network Protocols and Applications SNPA'03*, May 2003.

[11] Anthony D. Wood and John A. Stankovic. Denial of service in sensor networks. *IEEE Computer*, 35(issue 10):48–56, Oct 2002.

[12] Yih-Chun Hu, Adrian Perrig, and David B. Johnson. Wormhole detection in wireless adhoc networks. Technical Report TR01-384, Department of Computer Science, Rice University, June 2002.

[13] Guiling Wang, Guohong Cao, and Tom La Porta. Movement-assisted sensor deployment. In *IEEE INFOCOM 2004*, June 2004.

[14] Andrew Howard, Maja J Mataric, and Gaurav S Sukhatme. Mobile sensor network deployment using potential fields:a distributed, scalable solution to the area coverage problem. In *6th International Symposium on Distributed Autonomous Robotics Systems (DARS02)*, June 2002.

[15] Andrew Howard, Maja J Mataric, and Gaurav S Sukhatme. An incremental self-deployment algorithm for mobile sensor networks. *Autonomous Robots,Special Issue on Intelligent Embedded Systems*, 13(2):113–126, Sep 2002.

[16] Nojeong Heo and Pramod K. Varshney. An intelligent deployment and clustering algorithm for a distributed mobile sensor network. In *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, volume 5, pages 4576–4581, Oct 2003.

[17] Nirupama Bulusu, John Heidemann, Deborah Estrin, and Tommy Tran. Self-configuring localization systems: Design and experimental evaluation. *ACM Transactions on Embedded Computing Systems*, May 2003.

[18] Saurabh Ganeriwal, Aman Kansal, and Mani B. Srivastava. Self aware actuation for fault repair in sensor networks. In *Proceedings of the IEEE International Conference on Robotics and Automation(ICRA)*, May 2004.

[19] Peter Corke, Steven Hrabar, Ronald Peterson, Daniela Rus, Srikanth Saripalli, and Gaurav Sukhatme. Autonomous deployment and repair of a sensor network using an unmanned aerial vehicle. In *Proceedings of*

the IEEE 2004 International Conference on Robotics and Automation, Volume 4, pages 3602–3608. IEEE Computer Society Press, May 2004.

[20] Maxim A. Batalin and Gaurav S. Sukhtame. Coverage, exploration and deployment by a mobile robot and communication network. *Telecommunication Systems Journal, Special Issue on Wireless Sensor Networks*, 26(2):181–196, 2004.

[21] Guiling Wang, Guohong Cao, and Tom La Porta. A bidding protocol for deploying mobile sensors. In *11th IEEE International Conference on Network Protocol ICNP '03*, pages 315–324, Nov 2003.

[22] Guiling Wang, Guohong Cao, and Tom La Porta. Proxy-based sensor deployment for mobile sensor networks. In *IEEE International Conference on Mobile Adhoc and Sensor Systems (MASS) 2004*, 2004.

[23] T. Yan, T. He, and J. Stankovic. Differentiated surveillance for sensor networks. In *Proceedings of the ACM First International Conference on Embedded Networked Sensor Systems(SenSys '03)*, Nov 2003.

[24] Honghai Zhang and Jennifer C. Hou. Maintaining sensing coverage and connectivity in large sensor networks. Technical Report UIUDCS-R-2003-2351, Department of Computer Science, University of Illinois at Urbana Champaign, 2003.

[25] Di Tian and Nicolas D. Georganas. A coverage-preserving node scheduling scheme for large wireless sensor networks. In *Proceedings of the first ACM International Conference on Wireless Sensor Networks and Applications(WSNA)*, Sep 2002.

[26] Jie Jiang and Wenhua Dou. A coverage-preserving density control algorithm for wireless sensor networks. In *3rd International Conference on AD-HOC Networks and Wireless ADHOC-NOW'04*, pages 42–55. Springer-Verlag, July 2004.

[27] B.Chen, K. Jamieson, H. Balakrshnan, and R. Morris. Span: An energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks. In *Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking(MobiCom)*, July 2001.

[28] Alberto Cerpa and Deborah Estrin. ASCENT: adaptive self-configuring network topologies. *IEEE Transcations on Mobile Computing, July-Sep 2004.*, 3(3):272–285, 2004.

[29] Y. Xu, J. Heidemann, and D. Estrin. Geography-informed energy conservation for adhoc routing. In *Proceedings of the ACM/IEEE Inter-*

national Conference on Mobile Computing and Networking(MobiCom '01), pages 70–84, 2001.

[30] F. Ye, G. Zhong, S. Lu, and L. Zhang. PEAS: a robust energy conserving protocol for long-lived sensor networks. In *Proceedings of International Conference on Distributed Computing Systems(ICDCS '03), May 2003.*, 2003.

[31] Chalermek Intanagonwiwat, Ramesh Govindan, and Deborah Estrin. Directed diffusion: A scalable and robust communication paradigm for sensor networks. In *Proceedings of the Sixth Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom '2000)*, pages 56–67, Boston, MA, USA, August 2000. ACM Press.

[32] Haiyun Luo, Fan Ye, Jerry Cheng, Songwu Lu, and Lixia Zhang. Two-tier data dissemination model for large-scale wireless sensors networks. In *Proceedings ofACM/IEEE International Conference on Mobile Computing and Networking (MobiCom 2002)*, Atlanta, Georgia, USA, September 2002.

[33] Fan Ye, Gary Zhong, Songwu Lu, and Lixia Zhang. Gradient broadcast: A robust data delivery protocol for large scale sensor networks. In *IPSN*, Palo Alto, CA, USA, April 2003.

[34] Brad Karp and H. T. Kung. GPSR: Greedy perimeter stateless routing for wireless networks. In *Mobile Computing and Networking*, pages 243–254, 2000.

[35] John Heidemann, Fabio Silva, and Deborah Estrin. Matching data dissemination algorithms to application requirements. In *Proceedings of the ACM First International Conference on Embedded Networked Sensor Systems(SenSys '03)*, Nov 2003.

[36] David Braginsky and Deborah Estrin. Rumor routing algorithm for sensor networks. In *1st ACM International Conference on Wireless Sensor Networks and Applications(WSNA '02)*, Sep 2002.

[37] Di Tian and Nicolas D. Georganas. Energy efficeint routing with guaranteed delivery in wireless sensor networks. In *IEEE Wireless Communications and Networking Conference (WCNC '03)*, Apr 2003.

[38] Deepak Ganesan, Ramesh Govindan, Scott Shenker, and Deborah Estrin. Highly-resilient, energy-efficient multipath routing in wireless sensor networks. *Mobile Computing and Communications Review*, 1(2), 2001.

[39] Stefan Dulman, Tim Nieberg, Jian Wu, and Paul Havinga. Trade-off between traffic overhead and reliability in multipath routing for wireless sensor networks. In *IEEE Wireless Communications and Networking Conference (WCNC '03)*, Apr 2003.

[40] Rahul C. Shah and Jan M. Rabaey. Energy aware routing for low energy ad hoc sensor networks. In *IEEE Wireless Communications and Networking Conference (WCNC '02)*, 1, pages 350–355, March 2002.

[41] Sylvia Ratnasamy, Brad Karp, Li Yin, Fang Yu, Deborah Estrin, Ramesh Govindan, and Scott Shenker. GHT: A geographic hash table for data-centric storage. In *Proceedings of the 1st ACM international workshop on Wireless sensor networks and applicationsWSNA '02*, pages 78–87. ACM Press, 2002.

[42] Ananth Rao, Sylvia Ratnasamy, Christos Papadimitriou, Scott Shenker, and Ion Stoica. Geographic routing without location information. In *Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking(MobiCom '03)*, pages 96–108, Sep 2003.

[43] Douglas S. J. De Couto and Robert Morris. Location proxies and intermediate node forwarding for practical geographic forwarding. Technical Report MIT-LCS-TR-824, MIT Laboratory for Computer Science, June 2001.

[44] G. Toussaint. The relative neighborhood graph of a finite planar set. *Pattern Recognition 12*, 4:261–268, 1980.

[45] K.R. Gabriel and R. R. Sokal. A new statistical approach to geographic variation analysis. *Systematic Zoology*, 18:259–278, 1969.

[46] Evangelos Kranakis, Harvinder Singh, and Jorge Urrutia. Compass routing on geometric networks. In *Proceedings of the 11th Canadian Conference on Computational Geometry*, pages 51–54, 1999.

[47] P. Bose, P. Morin, I. Stojmenovic, and J. Urrutia. Routing with guaranteed delivery in ad hoc wireless networks. *Wireless Networks*, 7:609–616, 2001.

[48] Fabian Kuhn, Roger Wattenhofer, Yan Zhong, and Aaron Zollinger. Geometric ad-hoc routing: Of theory and practice. In *23rd ACM Symposium on Principles of Distributed Computing (PODC '03)*, July 2003.

[49] Qing Fang, Jie Gao, and Leonidas J. Guibas. Locating and bypassing routing holes in sensor networks. In *IEEE INFOCOM 2004*, June 2004.

[50] Y. Yu, R. Govindan, and D. Estrin. Geographical and energy aware routing: A recursive data dissemination protocol for wireless sensor networks. Technical Report UCLA/CSD-TR-01-0023, UCLA Computer Science Department, May 2001.