Multi-Resolution State Retrieval in Sensor Networks

Budhaditya Deb, Sudeept Bhatnagar and Badri Nath
Dept. of Computer Science, Rutgers University
110 Frelinghuysen Road, Piscataway, NJ 08854-8019, USA
{bdeb, sbhatnag, badri}@cs.rutgers.edu

***Abstract*- Large-scale sensor networks require mechanisms to extract topology information that can be used for various aspects of sensor network management. It is critical for any topology discovery algorithm in sensor networks to adhere to the resource constraints of bandwidth and energy. In this paper, we describe a distributed parameterized algorithm for Sensor Topology Extraction at Multiple Resolutions (*STEM*), which makes a tradeoff between topology details and resource expended. The algorithm retrieves network state at multiple resolutions at a proportionate communication cost. We also define various classes of topology queries and show how the parameters in the algorithm can be used to support queries specific to sensor networks. We show that the topology determined, albeit at a low resolution, is sufficient for approximating actual network properties. Finally we show how STEM can be used for general-purpose multi-resolution information retrieval in sensor networks.**

I. INTRODUCTION

Network Topology is an important attribute of the network state as it aids in network management and performance analysis. It refers to the physical connectivity and logical relationships of network elements. For IP networks, accurate knowledge of network topology is a prerequisite to many critical network management tasks, including proactive and reactive resource management and utilization, server siting, event correlation, root cause analysis, growth characteristics and even for use in simulation for network research. In this paper, we describe a distributed parameterized algorithm for Sensor Topology Extraction at Multiple Resolutions (*STEM*), which makes a tradeoff between topology details and resource expended. The algorithm retrieves network state at multiple resolutions at a proportionate communication cost. The rest of the introduction section provides the motivation, main contributions, overview, and related work.

A. Motivation

We first discuss the motivation for a new topology discovery algorithm for sensor networks. Later we discuss the need for multi-resolution topology retrieval for sensor networks.

Researchers have proposed different mechanisms for topology discovery of data networks [3,8,9,10]. The fundamentally different architecture of sensor networks [1,2] poses new challenges to the problem of determining topology. This is described as follows:

- Using routing tables for aggregating topology information is not feasible because traditional routing tables may not be available if data centric model of routing is used [6]. Further, in ad-hoc deployments, routing tables are often inaccurate or incomplete.
- Probing every node using algorithms used for Internet mapping [3] is not possible in sensor networks since nodes operate in various levels of doze mode, may often be disconnected.
- Incurred overhead using probing techniques is unsuitable for energy constrained sensor networks.
- Finally, topology discovery algorithms for wired networks do not use the broadcast nature of wireless media called the *Wireless Multicast Advantage* [29].

STEM is designed to return topology at a required resolution at proportionate costs. The following lists factors due to which multi-resolution topology retrieval is desirable for sensor networks.

- Our simulation studies show that different topology resolutions are required for different applications to perform at a desired level. For example, on retrieving only 10% of the network edges (for a 1000 node network with average degree of 25) using STEM, the average single source shortest path length increased by only 6% of the hops from the optimal. On the same graph, to bound the increase in average all pairs shortest path lengths to 6%, we have to retrieve 25% of the network edges. As the granularity of the recovered topology increases, the network properties like coverage area, connectivity, average node density, and expected shortest path lengths converge towards the real values for the network.
- A low resolution topology can provide a good estimate of the actual network properties since the density of a sensor network is higher than the critical densities required to maintain connectivity [14].[1]

This research work was supported in part by DARPA under contract number N-666001-00-1-8953 and a grant from CISCO systems.

[1] Typically, *Sensing Range* of nodes is much smaller than the *Communication range* (E.g. in heat sensors and motion sensors). A network barely sufficient to provide sensing coverage could be sufficiently dense for providing communication coverage.

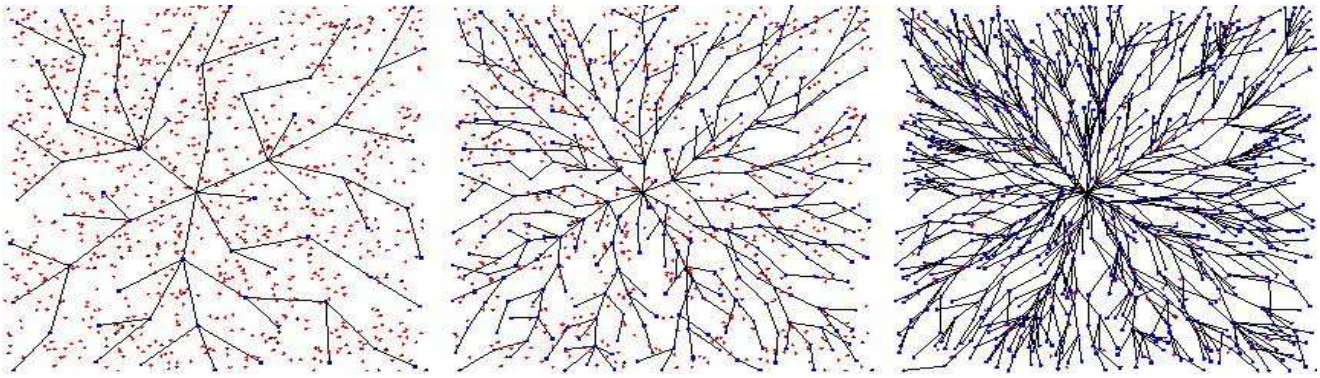

Figure 1: Topology of network with (1000 nodes in a 400x400 m^2 field, communication range 40m) retrieved at multiple resolutions. In all cases topology discovery is initiated at the center node

- As has been observed before in [14], we found that some network properties show phase transitions when tested on topologies at different resolutions. For example, to verify if each node in a network (with node degree of around 25) has at least three disjoint paths, STEM only needs to recover 17% of the edges. Thus, retrieving topology at a resolution slightly higher than the critical resolution gives a very good approximation to the property exhibited by the underlying graph. STEM provides a method to retrieve topology at critical resolutions. Our experiments were motivated by studies on critical density thresholds in [14]. However, we look at critical thresholds when the same network graph is observed at various resolutions.
- STEM provides mechanisms to retrieve topology at any desired resolution at proportionate costs. Given the above observations on multi-resolution topologies, such a tradeoff is essential for energy constrained sensor networks.
- Finally, for sensor networks the structure of topology in physical space is as important as the logical connectivity of its elements (E.g. to compute field exposure and coverage [12,13]). A topology discovery algorithm that can return the location information of nodes at a desired resolution would be very useful to approximate exposure and coverage of the network within specified error bounds.

B. *Main Contributions:*

Our first contribution is a *Distributed Parameterized Algorithm* for sensor topology extraction at ***multiple resolutions (STEM)***. We introduce the notion of *Minimal Virtual Dominating Set (MVDS*), which is the minimal set of nodes required for extracting topology at a desired resolution. STEM uses three parameters to select a suitable MVDS for required resolutions to retrieve topology at proportional cost. Retrieved network topology ranges from the minimal backbone to the complete network graph. (Section IIIB).

Our second contribution is a description of various types of topology discovery queries relevant for sensor networks and applications that can use these queries. We give formal analyses (Section IV) to describe the expected behavior of the algorithm for uniform random topologies. These results are used to map the queries to the parameter values to be used in the algorithm. (Section V)

Finally, we show that our algorithm is not just limited to topology discovery. The basic framework of the algorithm can be used to retrieve many different kinds of information at multiple resolutions (Section VI).

C. *Overview:*

STEM utilizes the broadcast property of wireless medium called as the *Wireless Multicast Advantage [29]*. A node can detect the presence of its neighbors by eavesdropping on the communication channel. Thus by selecting a subset of nodes, approximate topology can be created by merging their neighborhood lists. The resolution of the topology depends on the cardinality and structure of the chosen set of nodes. For example, to construct a minimal backbone tree of the network, we only need to merge the neighborhood lists of the minimal dominating set of the network graph.

The algorithm runs in two stages. First a monitoring node, which requires the topology, sends a topology discovery request to all the nodes in the network by controlled flooding. The request contains two parameters called *virtual range* and *resolution factor.* These parameters are used to select a minimal set of nodes required to retrieve topology at a desired resolution. We define this set as the *Minimal Virtual Dominating Set (MVDS).* During this stage, the nodes are colored *red* or *black* such that each *red* node is a neighbor of some *black* node and the black nodes form the MVDS. Further, at the end of the first phase, a *black* node tree rooted at the monitoring node is set up. In the second

phase, the *black* nodes reply back to the request with a subset of its neighborhood list, determined by the resolution factor. Each black node aggregates the data received from its children black nodes and sends it to its parent in the tree.

Figure 1 illustrates the results of applying STEM on a topology of 1000 nodes randomly spread in 400x400m^2 field with communication range of 40m and three different virtual ranges. The nodes on the tree form the responding MVDS. The effectiveness of STEM in selecting the MVDS is clearly depicted by the uniform distribution of the MVDS across the sensor field.

D. Related Work:

In [27], using end-to-end Bayesian probing in IP networks, properties are inferred using correlations. Our work focuses on *selecting a representative set* of nodes in a sensor networks to estimate network properties whereas in [27], *given a set of network endpoints*, the network is compactly characterized with respect to certain metrics.

For mobile ad hoc networks, in [16] authors propose a mobile agent based framework to distribute topology information. The agents migrate to least visited nodes and update topology caches in visited nodes. Getting the network topology at the initiating node would entail either waiting for an agent to visit it or querying every node for their topology caches. The authors show that the optimal number of agents required is half the number of nodes. This would not scale for sensor networks where the number of nodes is large and consequently the overhead of the agents would be large. Reducing the agents would increase the expected time that any agent takes to reach the initiating node.

References [19,20] introduced the concept of virtual backbones (which essentially is a dominating set) for wireless networks. References [18,19,20,21,22,23,24] describe routing in ad hoc networks using minimal connected dominating sets. In [21], authors give bounds on approximation algorithms for finding Connected Dominating Sets (CDS). Using each of the above methods, a simple topology discovery algorithm can be designed to query the dominating nodes, which provide their neighborhood lists. However, our algorithm differs significantly from the above in its *multi-resolution* nature. Hence, the topology returned is not limited to the minimal backbone. In fact, the topology that each of the above can provide is only a special case of lowest resolution topology recovered by our algorithm. Moreover, the following factors make STEM extremely suitable for sensor networks:

- The MVDS is created using message complexity of N (number of nodes in the network), i.e. each node sends *only one* packet and compares well to a centralized greedy approximation scheme to find an MVDS. Such an algorithm using only one packet per node makes it very useful for energy constrained sensor networks.
- The cardinality of the MVDS is dependent only on the network field dimensions, the communication radius, and required resolution and is almost constant with respect to density of the network. Hence the message complexity to retrieve topology does not increase with increase in density of the network.
- *STEM* creates a tree rooted at the monitoring node, which is optimal in the number of hops from monitoring node. Thus if any node is actually *h* hops away from the monitoring node, it is also *h* hops away from the monitoring node in the aggregation tree. Thus the topology response packets travel the minimal number of hops to reach the monitoring node.
- The above characteristics are achieved using local timer mechanisms to forward topology discovery request packets and select the responding nodes. The timer mechanism actually makes it possible to do the above using only one packet per node. The timers are based completely on local information and do not require any *time synchronization.* Hence it is very practical for sensor networks.

II. TOPOLOGY DISCOVERY ALGORITHMS

We first give the basic assumptions and the topology resolution metrics. Then we list two trivial approaches for topology discovery and their probabilistic variations, which are based on breadth first search. Finally we describe the STEM algorithm.

A. Assumptions and Network Model

In this section, we describe the assumptions for purposes of describing the algorithms. Later, for complete evaluation of these algorithms, we relax some of the assumptions.

- The sensor network topology is a unit disk graph. The network is connected and all nodes are active. All nodes have a fixed circular communication range *R* (the radius of the disk).
- Channels are error-free such that all packets are reliably transmitted.
- Each node maintains a neighborhood list by eavesdropping on the communication channel.

All the algorithms follow three basic stages of execution.

- A *monitoring node,* also called an *initiating node,* requiring the topology of the network initiates a *topology discovery request*.

- This request diverges throughout the network reaching all active nodes.
- A response action is set up which converges back to the initiating node with the topology information. Nodes respond to the topology request with topology information (such as adjacency lists and node positions)

We assume that the request divergence is through controlled flooding so that each node forwards the discovery request exactly once.[2] Since wireless is a broadcast medium of communication, a node can collect its neighborhood list by eavesdropping on the communication channel. Note that each node must send at least one packet for other nodes to know its existence. A topology discovery request from each node ensures: 1) If the network is connected, all nodes receive a packet; 2) They have complete neighborhood lists. When nodes are sleeping, neighboring nodes use cached information of nodes.

This work presents a mechanism to retrieve network topology at multiple resolutions. The resolution of the topology is determined by two measures:

1. *Edge Resolution:* Ratio of the number of *desired/retrieved* edges and the actual number of edges in the network.

2. *Node Resolution:* Ratio of the number of *desired/retrieved* nodes and the actual number of nodes in the network.

B. Trivial Approaches

We describe two trivial approaches and their probabilistic variations for topology discovery. The overhead of these approaches is governed by a fixed per packet overhead and overhead proportionate to the number of edges returned by them.

1. *Direct Response:* When a node receives a topology discovery request, it forwards this message and immediately sends back a response with its neighborhood list along the reverse path.

2. *Aggregated Response:* A node receives a packet, it forwards the request immediately but waits for its children nodes to respond before sending its own response. On receiving responses from its children, it aggregates the data and sends it to its own parent.

The inflexible nature of these approaches causes them to incur a large overhead irrespective of the desired resolution. These methods can be made adaptive by using their probabilistic variations to get multi-resolution topologies. In the probabilistic variation of the direct response mechanism, nodes decide to reply back with some probability p and report all their edges. In the aggregated response case, *all* nodes respond but report each of their edges with probability p. In the first case, since nodes respond back randomly, aggregation is not guaranteed. In the second case, since all nodes are reporting back, the number of responses is same as its non-adaptive counterpart, albeit with a reduced per-response cost. Also in the probabilistic variations, no guarantees can be provided that *each* node will be covered. In our proposed approach we provide such guarantees with lesser overhead than all these approaches.

C. STEM Algorithm

While the previous approaches return the complete network topology (assuming that all nodes are active), the overhead incurred is quite high (this is verified in our simulations presented later in the paper). We found that many graph properties, when computed on the actual graph, are not significantly different when computed on a sparse representation of the graph Thus a topology discovery algorithm could benefit by retrieving only a low-resolution topology without paying too much in terms of the graph property degradation. STEM selects a subset of nodes to reply to the topology discovery query. These nodes reply back with their neighborhood information. The number of these nodes determines the resolution of retrieved topology. The overhead is proportional to the resolution retrieved.

The key to the STEM algorithm is to have a mechanism to control the cardinality of the responding set. The conceptual framework, which allows STEM to do this, is described below:

Consider a graph $G(V, E(R))$ where an edge exists between nodes v_i and v_j if their distance is at most R.[3] For wireless networks, having communication range as R defines the network graph. We define the following terms on this graph:

– *Virtual Edge Set (E(r)):* The subset of $E(R)$ such that each edge in the set has endpoints at most distance r apart. In this case, r is called the *Virtual Range.*

[2] A probabilistic forwarding approach to flooding as described in [15] can be used as well.

[3] Such a graph is referred to as a unit disk graph in literature

- *Virtual Graph G(V, E(r)):* The sub-graph of G which has only edges from *E(r)*.

- *Minimal Virtual Dominating Set MVDS(r):* A minimal dominating set on *G(V, E(r))*.

STEM is a coloring algorithm, which creates an *MVDS(r)* based on the *virtual range r*. Since a decrease in *r* causes a decrease in the number of virtual edges, the cardinality of *MVDS(r)* increases as *r* decreases. STEM selects the nodes in *MVDS(r)* to respond to topology discovery queries with *r* providing a resolution control parameter. Note that finding a minimum dominating set of unit disk graph is NP-Complete [26]. STEM is a distributed approximation algorithm to find *MVDS*.

1) STEM Request Propagation Phase

The request propagation in STEM is similar to the other approaches using controlled flooding. The algorithm takes three user-specified parameters that control the topology resolution from the minimal backbone tree to the complete network graph. The parameters are defined as follows:

1. *Virtual Range (r∈[0,R] where r is virtual range and R is the Communication range):* The virtual range controls the cardinality of the *MVDS* as described earlier. Only the members of *MVDS* reply to the topology discovery request.

2. *Resolution Factor (f∈[0,1]):* A member of the *MVDS* reports this fraction of edges originating from it. For example, for *f=0.4* a node should return 40% of its neighborhood list.

3. *Query Type (Q):* This defines the set of queries, which STEM can support. The *query type* maps to specific filters and aggregating functions that may be required for general-purpose multi-resolution information retrieval.

Thus, each query is of the form: ***STEM (r, f, Q).***

Now we describe the coloring algorithm assuming that the parameters are known. We later describe how to compute the parameter values based on the required resolution.

To find the *MVDS* we use four colors. As the topology discovery request propagates, different nodes are colored according to their definitions given below:

- ***White:*** Yet undiscovered node, or a node which has not received any topology discovery request packet.

- ***Black*****:** A node in the *MVDS* which replies to topology discovery request with its neighborhood set. After becoming *black*, a node discards all other request packets.

- ***Red*****:** A node which is *virtually dominated* by at least one *black* node, i.e., it is inside the virtual range *r* of the *black* node. After becoming *red*, a node discards all other request packets. The node is said to be *attached to* the corresponding black node.

- ***Blue:*** A node which receives a packet from a *red* or *blue* node or a node which is within communication range of a *black* node but outside its *virtual range*. It waits for a time period for some other node in its virtual range to become *black*. Otherwise it itself becomes a *black* node.

The request packet contains the following fields:

- Sending Node ID and Node Color
- Black Node it is attached to
- Number of Hops from Monitoring Node.
- Parameters (virtual-range, resolution factor, query type)

Each node uses two timer functions defined as follows:

1. *Request Forwarding Timer (R_F_T (distance))*: The time between receiving a discovery request and forwarding it, is the forwarding delay. The parameter *distance* is the distance between the receiving node and the sending node[4]. The timer is inversely proportional to the distance i.e. the farthest node forwards the earliest

2. *Black Node Formation Delay (B_F_T(distance)):* This is the time period after which a blue node changes to black. The timer is proportional to its deviation from *2r* distance from the black node at previous hop. Thus a node which is closest to the *2r* distance becomes black first.

Initially all nodes are *white*. The *initiating node* is colored *Black* and initiates the process by broadcasting a topology discovery request. As the topology discovery request propagates, each node is colored *black*, *red* or *blue* according to the coloring algorithm. Figure 4 describes the action taken by a node on receiving a discovery request.

[4] We assume that the distance between nodes can be approximated by using GPS, signal strengths, etc

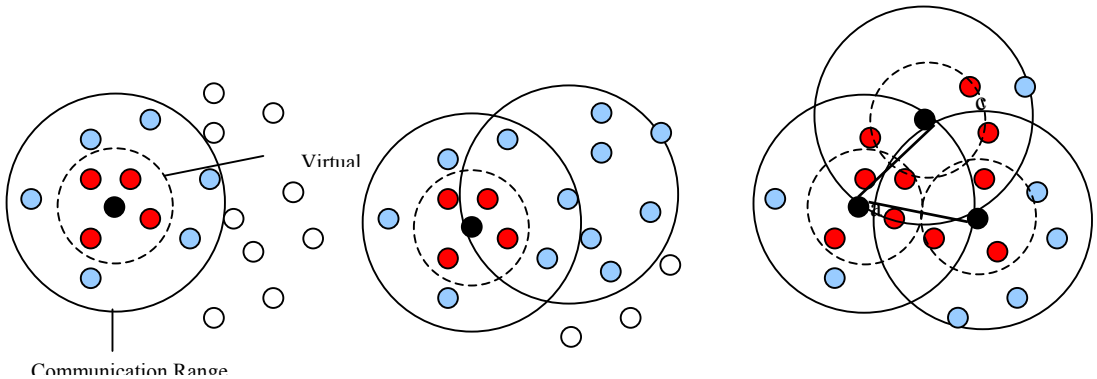

Figure 3. Illustration of the Coloring Algorithm. Node *a* becomes *black* and forwards a topology discovery request. Nodes within its virtual range *r* are colo*red red* and nodes outside its virtual range but within its communication range *R* are colo*red blue*. Node *b*, which is farthest from node *a* forwards the topology discovery request the earliest since forwarding delay is inversely proportional to the distance. Since node *b* was blue, all its *white* neighbors also become blue. All blue nodes start a timer to become *black*. Nodes *c* and *d* become *black* earlier than other blue nodes since they are closer to *2r* distance from previous *black* node *a*. They color their neighbors similar to earlier step. The *black* nodes *c* and *d* are children *black* nodes of node *a*.

```
RECEIVE_REQUEST_PACKET( recvColor, distance, r )
 1.  if ( (recvColor==BLACK)  & ( selfColor==WHITE) )
 2.   if (distance < r)
 3.        selfColor = RED
 4.          FORWARD_REQUEST_PACKET(R_F_T(distance))
 5.       if (distance>r)
 6.            selfColor= BLUE
 7.            B_F_T(distance)
 8.          FORWARD_REQUEST_PACKET(R_F_T(distance))
 9.  if ((recvColor==BLACK) & ( selfColor==BLUE ) & ( distance<r ))
10.      selfColor=RED
11.      cancel (B_F_T)
12.      if (a request packet has not been Forwarded )
13.          FORWARD_REQUEST_PACKET (R_F_T(distance))
14. if ( (recvColor==RED) OR ( recvColor==BLUE ) )
15.      if (selfColor == WHITE)
16.            selfcolor = BLUE
17.            B_F_T(distance)
18.          FORWARD_REQUEST_PACKET(R_F_T(distance))
```

Figure 4 Coloring Algorithm when node receives a topology request Packet

The algorithm is described below:

- The node that initiates the topology discovery request, is colo*red black* and broadcasts a topology discovery request packet.

- Lines 1-13 describe the operations when nodes receive packet from a *black* node. All *white* and *blue* nodes within virtual range of a *black* node become *red*. Other nodes that are in communication *range* but outside *virtual range* become *blue*. After ***R_F_T****(distance)* time a node forwards the discovery request if it has not already done so. All *blue* nodes start a timer to become *black* with a *Black Node Formation delay* function ***B_F_T****(distance)*.

- Lines 14-18 describe operations when a node receives a packet from a *red* or *blue* node. When a *white* node receives a packet it becomes *blue.* It then starts a timer ***B_F_T****(distance),* to become *black.*

- If any *blue* node receives a packet from a black node in its *virtual range*, it cancels its B_F_T and becomes *red*. Once nodes are *red* or *black*, they ignore other topology discovery request packets. Figure 3 illustrates the algorithm with an example.

Ideally a minimum number of nodes should reply back to provide a desired resolution (in this case MVDS). Since the problem of finding minimum dominating set is known to be NP-complete (ideal MVDS is a minimum dominating set on the *virtual graph*), we use a greedy approach for finding the set of nodes to reply back, i.e. at each step a node that covers the maximum number of *yet* uncovered nodes is chosen to become *black*. It is not possible to know the best candidate to become *black* without global knowledge of neighborhood sets. Since global knowledge is not available at runtime, the greedy approach cannot be implemented. We use timer mechanisms (based on only local knowledge) to approximate the greedy approach.

The timer mechanism tend to have the following effect:

1. Maximal number of nodes become blue so that the probability of having the best candidate to become *black* is higher. (using R_F_T delay)

2. The better candidates among the candidate *blue* nodes become *black* with a lower delay. (using B_F_T delay)

These are illustrated using Figure 5. When any node forwards, a node with a lower overlap with a covered region is expected to reach lesser number of uncolored nodes. Consider the example in Figure 5. The communication range of node *a* has a lesser overlap with that of *b* as compared to that of *c*. When *b* is further from *a* than *c,* it is expected to cover a larger unexplored region. R_F_T ensures that *b* forwards before *c.*

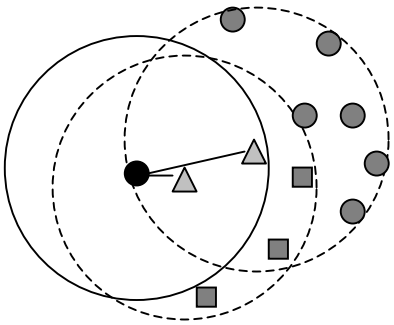

Figure 5. Illustration for the delay heuristic.

The second condition is achieved with a *black node formation delay* at each blue node proportional to its nearness to *2r* distance from a *black* node in previous step. This means that a blue node closest to *2r* distance from previous *black* node would become *black* the earliest. A node at *2r* distance from previous black node, is expected to have minimum overlap of virtual region and hence *cover* larger uncovered *blue* nodes.

Note that end of the coloring phase each node is either *red* or *black* if the network is connected. This is because a blue node either becomes *black* if its B_F_T expires or becomes *red* if a node in its virtual range becomes black. Moreover the timer mechanisms try to reduce the cardinality of the black node set. Hence the black node set gives the required *MVDS(r)*.

2) STEM Response Phase

The first phase of the algorithm sets up the node colors. Each node in the network is either a *black* node or a neighbor of a *black* node (i.e. *red* node). The initiating node becomes the root of the *black* node tree where the parent *black* nodes are at most two hops away from its children *black* node. Each node has the following information at the end of this period:

- Each node knows its *parent black node*, which is the last *black* node from which the topology discovery was forwarded.
- Each *black* node knows the *default node* to which it should forward packets in order to reach the *parent black node*. This node is essentially the node from which it had received the topology discovery request.
- All nodes have their neighborhood information by eavesdropping on the communication channel.

Using the above information, the response action is described below:

- When a node becomes *black*, it sets up an *acknowledgement timer* (described later) to reply to the discovery request. Each *black* node waits for this time period during which it receives responses from its children *black* nodes.
- It aggregates all topology information from its children and adds *f fraction* of edges in its communication region.
- When its time period for acknowledgement expires, it forwards the aggregated neighborhood list to the *default node* to its parent *black* node.
- In the general framework for information retrieval, the parameter *query-type*, is used to filter or aggregate the information. However for topology discovery we just merge the neighborhood lists.

We note that for the algorithm to work properly, timeouts of acknowledgements should be properly set. The *acknowledgement timer* of a *black* node should always expire before its *parent black node* so that each *black* node forwards only after receiving responses from its children. For this we set a timeout value inversely proportional to the number of *hops* a *black* node is away from the monitoring node (the number of hops is obtained from the discovery request packet). We need an upper bound on the number of hops between extreme nodes. If the extent of deployment region and communication range of nodes is known initially, the maximum number of hops can be easily calculated. However, if that information is not available to the nodes, we can assume that the topology discovery runs in stages where it discovers only a certain extent of area at each stage. In this work we assume that the initiating node has knowledge of the extent of node deployment area.

III. ANALYSIS OF STEM

In this section we state some of the analytical results related to STEM. All the results assume that the node density is uniform and that the network is connected. We verify our analytical results with simulations, for which we modified the NS-2 simulator [28] to incorporate details of STEM.

A. Expected number of Black nodes

To find number of black nodes (*B)* for a given *communication range R* and *virtual range r,* we associate with each node a probability to become black based on the definition of node colors.

Let *p = probability of each node to become black.*

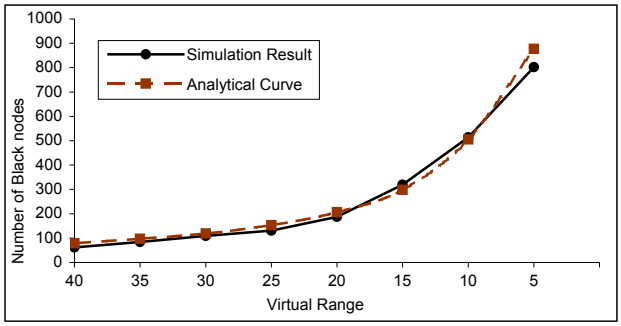

Figure 7: Comparison of Analytical expectation of the number of black nodes and simulations. Both simulation results are for 400x400m^2 field with 1000 nodes for a communication range 40m. The results show the number of black nodes formed for different virtual ranges

Each node would become black if no other node within its *virtual* range becomes black, i.e., if the expected number of nodes in any virtual area is *n,* then for a given node, all its *n-1* virtual neighbors should not be black. Then the following equation is satisfied:

$$p = (1-p)^{n-1} \quad \ldots 1$$

where,

$$n = \frac{N\pi r^2}{A}, n \geq 1, \quad 0 < p < 1$$

N = Total number of nodes.

A = Area of the sensor field

We solve the above equation for p to get the probability of each node becoming black. Then the expected number of black nodes, B is simply pN. Equation 1 is only valid when the expected number of nodes within a virtual region is greater than 1. As the virtual range reduces, this value can become less than 1. This means that each node has less than one node in its virtual range in the expected.

Let *E = Expected virtual degree of a node*

$$E = \sum_{x=1}^{N-1} x \binom{N-1}{x} \left(1 - \frac{\pi r^2}{A}\right)^x \left(\frac{\pi r^2}{A}\right)^{N-x-1} \quad \ldots 2$$

The total number of virtual edges in the network is half the sum of the expected degrees of all nodes, which is *0.5NE*. Since the number of *virtual* edges is less than the number of nodes (recall *n<1* in this case), nodes with no edges would always become black. Out of the rest of the nodes, only half would become black with the remaining being neighbors of these nodes. The expected number of black nodes is given by:

$$B = \left(N - \frac{NE}{2}\right) + \left(\frac{NE}{4}\right) = N - \frac{NE}{4} \quad \ldots 3$$

Thus the expected number of black nodes is:

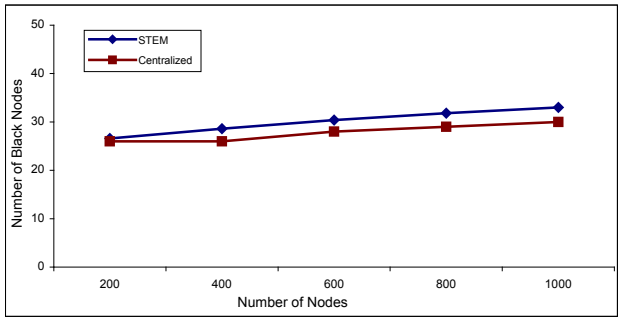

Figure 8. Number of *Black* nodes vs. total number of nodes for STEM and Centralized greedy algorithm. The communication range is 30m and sensor field dimensions are is 200x200m^2

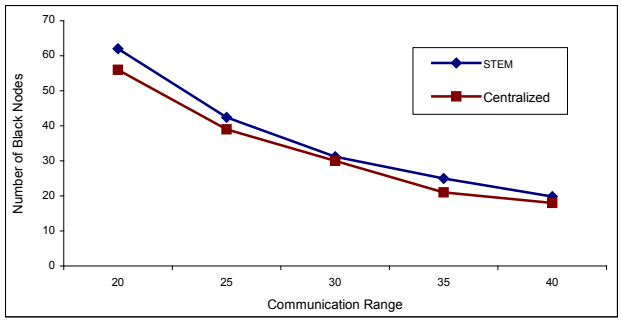

Figure 9. Number of *Black* nodes vs communication range for STEM and centralized greedy algorithm. The size of sensor field is 200x200m^2 and the field has 1000 nodes.

$$B = \begin{cases} pN & for \quad n > 1 \\ N - \frac{NE}{4} & for \quad n \leq 1 \end{cases} \quad \ldots 4$$

Note that Equation 4 does not take into account the heuristics and assumes a random formation of black nodes with equal probability of all configurations. However the heuristics in STEM make a black node configuration with lower overlap in virtual regions a more likely event. Hence the expected number of black nodes formed would be lesser than that estimated by equation 4. Figure 7. shows the plot of number of *black* nodes for different *virtual ranges* with communication radius of 40m. The average number of *black* nodes formed in simulations is compared to the number of *black* nodes from the analytical expectation of the number.

Figures 8 and 9 show comparison of STEM against centralized *log(n)-approximate* solution provided by the greedy algorithm [15] for set cover to find the black nodes. The nodes for this simulation are uniformly spread in 200m x 200m field. For this simulation, the virtual range is equal to the communication range. Figure 8 shows the impact of increasing the number of nodes in the field. Figure 9 shows the effect of communication range for 1000 nodes in the field. In both cases, STEM performs almost as well as the centralized

solution which has global knowledge. This result shows that the heuristics used in STEM timers perform well.

B. Retrieved Topology Resolution

STEM takes two parameters, which control the resolution of the returned topology. Thus for a given *virtual range(r)* and *resolution-factor(f),* the returned topology resolution is computed as follows:

The number of black nodes formed for a *virtual range r* is given by equation 4. Each of these black nodes formed, returns *f%* of its edges. Recall that two or more black nodes may be formed in the same communication range if *virtual range* is smaller than R. Since we assume edges to be symmetric, we should account for the edges which may be reported by both its endpoints. Let,

p = probability of a node becoming black, as given by equation 4.

Thus both nodes associated with any edge have probability *p* of becoming black. When a black node reports any edge with probability *f* (the resolution factor), the actual probability of that edge being reported is:

$$x = p^2\left(f^2 + 2f(1-f)\right) + 2pf(1-p) \qquad \ldots 5$$

Thus if *d= average node degree, then*
e= number of edges reported back = Nxd.

Figure 10 shows the plot of expected topology resolution as a function of STEM parameters *virtual range(r)* and *resolution-factor (f)*. The contours at the base are the *iso-resolution* curves obtained by intersection of a resolution plane with the curve. Each contour shows the set of *(r,f)* pairs which return the same resolution. We show in section V, how these can be used for selecting the parameter values for specific queries.

C. Overhead

In this section, we compute the overhead incurred by the response mechanism of STEM and compare it with the direct and aggregate response mechanisms.

Let
C = constant packet header overhead with each packet.

C_f *= Overhead per edge information sent.*

$B_i(r)$ *= number of Black nodes at i hops from the initiating node, a function of virtual range.*

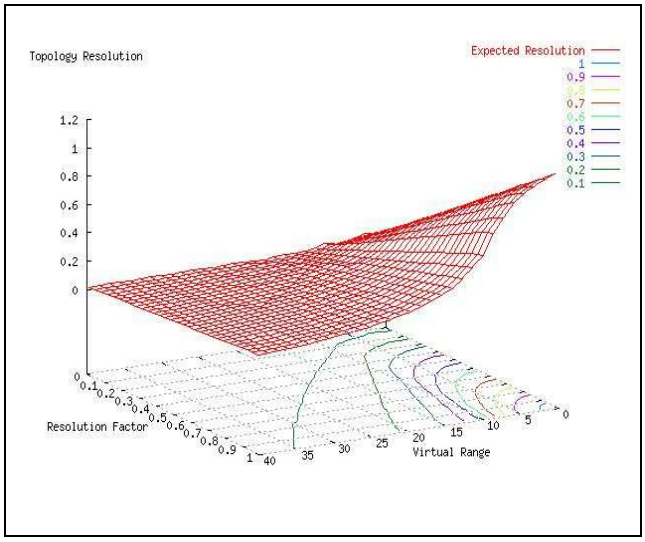

Figure 10: Expected Resolution of topology as a function of parameters *Virtual Range* and *Resolution Factor*. Each contour on the base shows the range of parameter values for topology returned at a particular resolution.

Each black node sends *f%* of the edges in its neighborhood. This information travels to the initiating node which is *i* hops away. Thus the expected byte overhead O_T of the response process is given by:

$$O_T = C_f x d \sum_{i=1}^{H} iB_i + C\sum B_i \qquad \ldots 6$$

Where, *T=maximum number of hops from initiating node*

d = average node degree
x is given by equation5

Equation 6 uses the maximum hop distance *H* in computing the overhead. However, *H* is not known in advance at the initiating node. The following observation enable us to estimate *H*: The R_F_T timer ensures that a node *h+1* hops away from the initiating always receives topology discovery request from a node *h* hops away. This is because each node at *h* hops forwards request before any node at *h+1*. This implies that the request propagates as a wave encompassing a circular region of radius *iR* in the i^{th} step. This means that the maximum number of hops can be approximated as:

$$H = ceil(D/R) \qquad \ldots 7$$

D = distance of farthest node from initiating node.

From the assumption in equation 7, we see that any black node *i* hops away from the initiating node lies in the band between the circles of radii *R(i+1)* and *Ri* centered at the initiating node. If we know the area of this i^{th} band, the expected number of black node in that region can be computed.

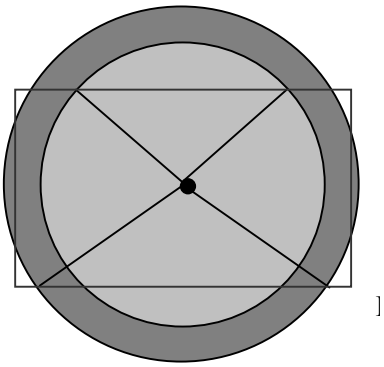

Figure 11. Computing the area of circle inside the rectangle.

For exposition we consider the sensor field to be a rectangle with the initiating node somewhere inside the rectangle as shown in figure 11. The i^{th} band is shown using the two circles, which cut the rectangle at maximum of eight points. If the area for each of the circles inside the rectangle is $A_{R(i+1)}$ and A_{Ri}, the number of black nodes in the band between the two circles (dark shaded region inside the rectangle) is given by:

$$B_i = \left(A_{Ri} - A_{R(i-1)}\right)\frac{B}{A} \qquad \ldots 8$$

Note that calculating this is trivial once we get the intersection points of the circles with the sides of the rectangle. We can use the above procedure to find the number of black nodes for a sensor field of any general polygonal shape. However in this work we only consider field to be rectangular.

Figure 12. shows the plot of STEM overhead against different *virtual-ranges* and *resolution-factors*. The overhead plot is used in conjunction with Figure 10 to get the highest resolution topology at a given byte overhead. We show later how this helps is selecting the parameter values to support queries in section V.

Now we discuss some simulation results for STEM overhead. In evaluating the number of bytes, we assume a constant packet header of 5 bytes and an additional 2 bytes of information per edge reported in the packet. The number of bytes transmitted during the entire operation characterizes overhead.

We compare STEM against direct response and aggregated response approaches and their probabilistic variations. Figure 13 shows the extra overhead incurred by these approaches over STEM. Note that the x-axis is non-uniform because the resolution values correspond to different virtual radii for STEM. Responding probability for the probabilistic variations was computed using these resolutions. Since direct and aggregated response mechanisms are not adaptive to the desired resolution,

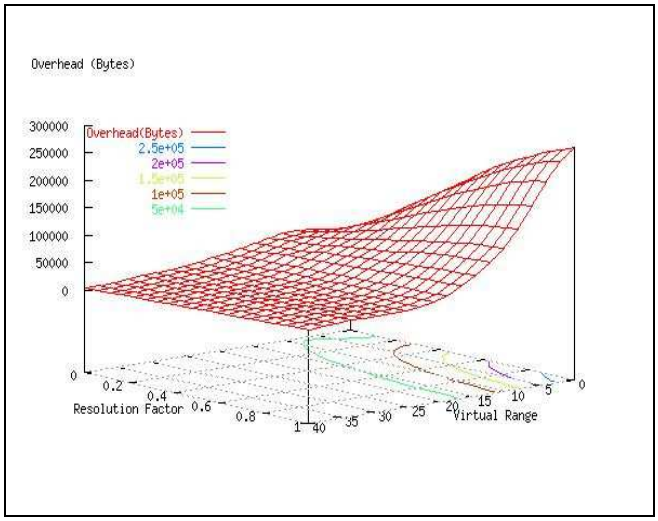

Figure 12. Expected Byte overhead of STEM for different values of parameters (, N=1000, R=40m, 400x400m). Constant packet overhead C=5 bytes, Overhead per edge information C_f = 2 bytes. The contours show the range of parameter values which require equal overhead to return topology.

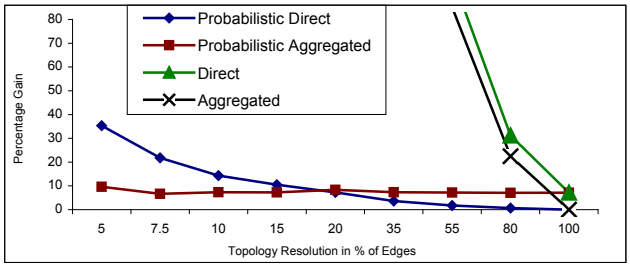

Figure 13. Relative Gain in Overhead of STEM over Aggregated, Direct and their probabilistic variations.

they have a huge overhead compared to STEM. The probabilistic variations perform better than their non-adaptive counterparts, but are significantly worse than STEM. Moreover, STEM guarantees that each node would be reported whereas the probabilistic variations cannot provide such a guarantee.

D. Impact of Sleeping Nodes

One of the distinguishing features of sensor networks is that nodes would sleep periodically. We analyze the impact of sleeping nodes on the recovered topology. Specifically, we compute the fraction of sleeping nodes that would be reported by the responding active nodes. Note that active nodes cache data about its neighbors and would respond with information about sleeping nodes as well.

This discussion assumes that the network of active nodes is connected. Let,

x=fraction of nodes sleeping.

=>(1-x) N = N' is the number of active nodes.

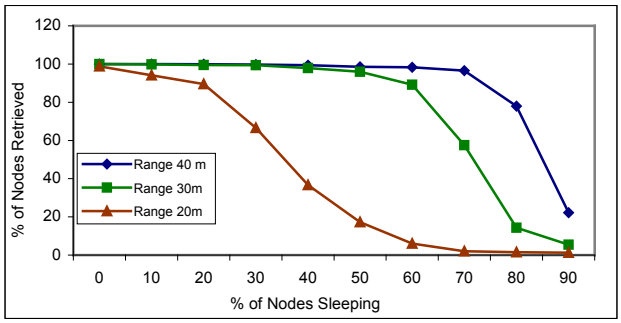

Figure 14 Effect of sleeping nodes on the Retrieved Topology. The network consists of 1000 node in a field of 400x400m^2. Simulations are carried out for three different communication ranges with varying % of nodes sleeping.

d= average degree of each node.

p=probability of node to become black (from equation 4 and using N')

p_s=probability of a sleeping node being reported

b_x=expected black neighbors of each node:

$$b_x = p\sum_{i=1}^{d} {}^{d}C_i i(1-x)^i (x)^{d-i} \quad \ldots 9$$

Each active node would always be reported because the active network is assumed to be connected. The probability of a sleeping node being reported is the probability of that any one of its b_x black neighbors report it. Since black nodes report *f fraction* of their edges, the probability of a sleeping node being reported is:

$$p_x = \begin{cases} 1-(1-f)^{b_x} & b_x \geq 1 \\ b_x f & b_x < 1 \end{cases} \quad \ldots 10$$

We tested the performance of STEM when varying percentage of sleeping nodes. The simulations are set up for three different communication ranges (*R=20, 30, 40 m*) with the parameter values *r=R* and *f=1*. The results in figure 14 show that STEM is able to retrieve information about nearly all the nodes if the number of active neighbors of each node is around *8*. The results also imply that if the active node set is connected, then we get information about almost all nodes.

IV. MAPPING QUERIES TO PARAMETERS.

In this section we describe various type of queries that would be relevant in the context of sensor networks. While this is not meant to be an exhaustive list of queries, it would encompass a broad range of applications. Along with the queries, we show how the parameters for STEM need to be set in order for it to provide an answer in the desired form. STEM algorithm can be invoked using three parameters: virtual range, resolution factor and query type. The first gives us the set of nodes to respond. The second gives the fraction of their edge information that each returns and the third one decides the type of aggregation or filtering to be performed. Thus a query is of the form:

– *STEM (virtual-range, resolution-factor, Query-Type)*

To handle various queries we frequently face the question: given the required topology resolution what values of parameters should be chosen? Recall figure 10, the contours showed the range of parameter values, which returns topology at a particular resolution. In general the choice of parameter values would depend on the following factors:

1. *Overhead:* Choose parameter values, which require the minimum overhead for a given resolution.

2. *Properties of Multi-resolution Topologies:* Given a resolution what property needs to be maintained in the retrieved graph.

In this work, we try to minimize the overhead while getting a desired resolution. For executing the queries at minimum overhead, we choose the largest possible virtual range for a given resolution and then select an appropriate resolution factor. The underlying intuition here is that the average number of hops traveled by packets is nearly the same for all parameter values, for a desired resolution. Thus to reduce the overhead, the number of packets has to be reduced.

Note that, the retrieved topology at a desired resolution might be required to possess certain properties. The trade-off here is between the overhead and maintenance of graph properties. For example, to test whether each node has two edge-disjoint paths to the sink, the retrieved topology should have minimum node degree of two. We believe that that different trade-offs are required for different applications. However, exploring these tradeoffs is orthogonal to the focus of this paper and is part of our future work.

In this section, we use simulations to show the effectiveness of STEM in dealing with different types of queries. The simulation results shown here are for random 1000 node topologies with nodes spread uniformly across a field area of 400m x 400m. The communication range of all nodes is set to 40m. The tables show the average, maximum and minimum resolutions achieved by STEM in 20 runs on the same topology with randomly chosen initiating nodes.

TABLE I. PERFORMANCE OF NODE-CONSTRAINED QUERY ON A RANDOM 1000 NODE TOPOLOGY IN A 400X400M^2, R =40M.

Required Node Resolution	Returned Node Resolution		
	Average	*Maximum*	*Minimum*
0.8	0.813	0.832	0.791
0.6	0.621	0.646	0.605
0.4	0.439	0.456	0.416
0.2	0.238	0.269	0.213

A. *Node Constrained Query:*

The node-constrained query is of the form: *"Return x% of the nodes in the network"*. Such a query helps in determining the approximate density distribution of the network while not requiring all the nodes to be returned. For this query, we construct the minimal backbone topology by setting the parameter *virtual range (r)* equal to communication range *R* and *resolution-factor (f)* equal to the desired fraction *x*. Thus the topology discovery query takes the form:

– *STEM(R, x, Node-Constrained-Query).*

When the parameter *Node-Constrained-Query* is passed, each black node finds its neighbor nodes, which have *not* been reported by its children black nodes. Out of these unreported neighbors, it picks *x%*, of nodes and aggregates with the children's information before sending it upstream.

A node-constrained query can be used to find virtual backbone of a network. The STEM query would be:

– STEM(R, 0, Node-Constrained-Query)

Table 1 gives the performance of node-constrained query using STEM for the different required resolutions in the simulation setup described earlier.

The node-constrained query can be modeled to approximate the exposure in the sensor network [13]. This query would take a form:

– *STEM(r, 0, Exposure-Query)*

In the query-type *Exposure-Query,* each node just returns its location. The number of nodes responding is controlled by virtual range *r*. As has been observed in [13], the minimum exposure path or exposure does not change significantly after a certain node density. STEM can return the minimal number of nodes to calculate the exposure with desired precision.

B. *Edge Constrained Query:*

The form of an edge-constrained query is: *"Return x% of the edges in the network"*. This type of query is useful in determining the connectivity properties of the network.

TABLE II. PERFORMANCE OF EDGE CONSTRAINED QUERY ON A RANDOM 1000 NODE TOPOLOGY IN A 400X400M^2, R =40M.

Required Edge Resolution	Virtual Radius	Returned Edge Resolution		
		Average	*Maximum*	*Minimum*
0.8	6.38	0.710	0.715	0.706
0.6	9.03	0.556	0.561	0.546
0.4	11.93	0.424	0.427	0.421
0.2	20.45	0.179	0.188	0.175

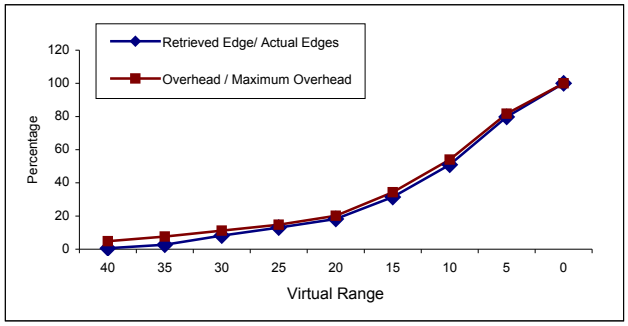

Figure 15. Graph shows the relative overhead (with respect to maximum overhead) incurred in retrieving topologies at different resolutions. The simulations were run on topologies of 1000 nodes, communication range 40m, and field size 400x400 m^2

We evaluate connectivity properties of network such as shortest and all-pairs shortest paths lengths, number of edge, node disjoint paths, etc. for different resolutions of topology retrieved.

Before we can send the topology discovery query, we first compute the values of the parameters to be passed based on the required resolution. I.e. we have to find the appropriate values for *virtual range r*. The *resolution-factor* is set to 1. Note that setting resolution-factor to 1 ensures that at least one edge pertaining to each node is reported back. *Let*

d = average node degree.

$$d = \frac{N\pi R^2}{A} - 1$$

E= expected total number of edges in the topology.

$$E = \frac{Nd}{2},$$

Since every *black* node reports all edges originating from it, the total number of edges reported back only depends on the number of *black* nodes chosen to reply back. To get the required fraction of edges (*x)* the following condition must be satisfied.

$$\frac{Bd}{2} = xE \qquad \Rightarrow B = \frac{2xE}{d}$$

Also from equation 4,

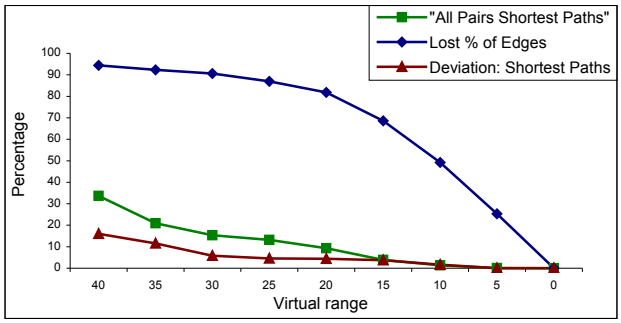

Figure 16: Effect of Edge Resolution on All-pairs and single source shortest path lengths, (N=1000, R=40)

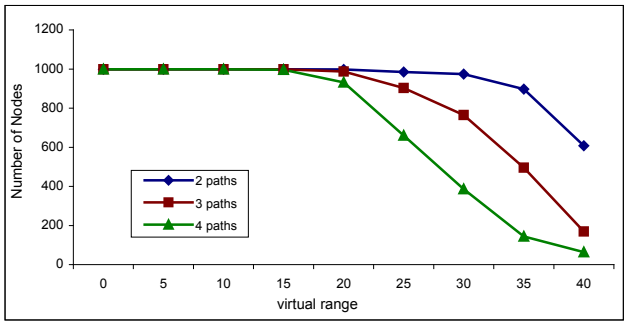

Figure 17: Effect of Edge Resolution on number of Edge Disjoint Paths to Sink. (N=1000, R=40)

$$B = pN \Rightarrow p = \frac{B}{N}$$

This gives the virtual radius as follows:

$$r = \sqrt{\frac{A}{N\pi}\left(1 + \frac{\log(p)}{\log(1-p)}\right)} \qquad \ldots 11$$

Now for equation 11 to be consistent with Equation 4, the expected number of nodes in virtual range should be greater than 1. Thus r is valid only if:

$$\frac{\log(p)}{\log(1-p)} > 1$$

Otherwise r is computed using equation 4, as follows:

$$r = \sqrt{\frac{4A(1-p)}{N\pi}} \qquad \ldots 12$$

The topology discovery query takes the following form:

– *STEM(r, 1, Edge-Constrained-Query).*

Table 2 shows the mapping of required edge-resolution to the value of parameter r for the simulation setup described at the beginning of the section IV. Using the above queries, we see that STEM is able to retrieve edges at resolution close to the desired values, using the above query.

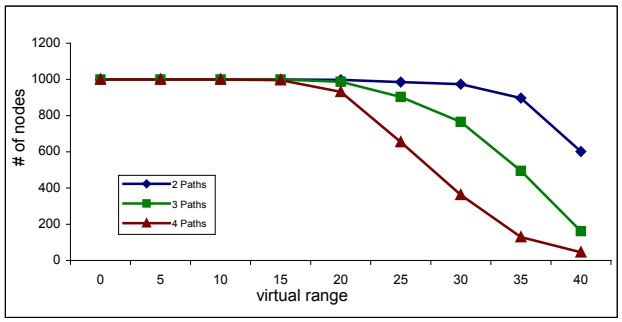

Figure 18: Effect of Edge Resolution on number of Node Disjoint Paths to Sink. N=1000, R=40

Another set of simulations was run to test the overhead incurred by STEM in retrieving a desired resolution and the impact of the lower resolutions on graph properties. The overhead incurred by edge-constrained queries is shown in Figure 15 for different virtual ranges. Observe that the relative overhead curve (as a percentage of the overhead incurred for retrieving the complete topology) closely follows the resolution. Figure 16 shows the effect of edge resolution on the length of single source shortest path lengths (rooted at initiating node) and all-pairs shortest path lengths. Recall that edge-constrained query ensures that each node is reported back. The graph shows the relative degradation in the single source shortest path length and all pair shortest path lengths as the topology resolution increases. At the lowest resolution topology returned (for *virtual range 40m)* STEM retrieves only 5.6% edges. In this case, single source shortest path deviates by only 17% from the optimal whereas for all pairs shortest paths, the deviation is around 35%. We see that at virtual range of 15m (corresponding to 31% edges), deviation for both path lengths are lesser than 5% although the number of edges missing is still very high at 69%. Thus by recovering only 17% of edges to compute single source shortest path and 31% edges to compute all pairs shortest path for the topology considered, we get a negligible deviation from the optimal.

Figure 17 and 18 shows the effect of edge resolution on the number of edge and node disjoint paths present from each node to the monitoring node. For different resolutions returned using different virtual ranges, the plot shows the number of nodes which have 2, 3 and 4 node and edge disjoint paths. The initial behavior for both node and edge disjoint shortest is similar. For all the cases, almost all nodes have paths at *virtual range of 15m* where 69% of the edges are missing.

We note that the savings in topology discovery increases as the density of the graph increases, i.e., lesser percentage of edges are required as density increases for

similar deviation from optimal behavior. In general, since sensor networks are expected to be dense, we can attain significant savings by discovery topology at the required resolution, using STEM.

C. Overhead Constrained Query:

This query has the form: *"Return the best resolution topology using a given Overhead budget."* The total overhead in topology discovery is due to request forwarding (network-wide flooding) and the discovery acknowledgement. Probabilistic forwarding may be required to cut down on the energy spent on the forwarding phase. To support this query, we use the analytical results on expected resolution retrieved and the expected overhead for given values of parameters *r* and *f*.

If O_M = Maximum bytes allowable for topology retrieval,

$O_T(r, f)$ = Total overhead for a given r and f.

$Re(r, f)$ = Retrieved resolution for given r, f.

Then we have to find a pair *r, f*, which satisfies the following:

$$\begin{aligned} & Max(\mathrm{Re}(r,f)) \\ & s.t.\ \ O_T(r,f) \le O_M \\ & \qquad 0 \le r \le R \\ & \qquad 0 \le f \le 1 \end{aligned} \qquad \ldots 13$$

The iso-overhead contours in Figure 12 provide a range of parameter values which return topology at a given overhead. The contour for the given query is computed by intersecting the surface O_T with the hyper-plane defined by equation 13. In general, the analytical computation of parameters by this method is not required if we assume uniform random graphs. Setting *resolution-factor* to 1 and then selecting the minimum possible virtual range for the given overhead would almost always retrieve the maximum resolution. This is because by setting *f=1,* we are maximizing the amount of information in each packet and thus minimizing the share of constant packet overhead in the total overhead. However, other considerations may come into account when selecting the parameter values as discussed at the beginning of the section.

V. DISCUSSION

In this section, we discuss the issues that have an impact on the performance of STEM and the directions for extending STEM to make it general purpose information retrieval framework.

A. Extending STEM as a General Framework for Multi-Resolution Information Retrieval

Initially, we had claimed that STEM can be used as a general purpose multi-resolution information retrieval framework. We now substantiate these claims. The query parameters as described till now, are specific to topology discovery. We now describe how the parameters can support general queries.

Changes in the STEM protocol has to be implemented so that *virtual-range* can take any arbitrary value. For topology discovery the node selection process finds a suitable MVDS. In general, the criterion for node selection could depend on any characteristic of the information sought. For example, if the desired information has spatial correlations and neighboring nodes have correlated data, then such a *neighborhood criterion* could be used for determining the parameter *virtual-range.*

The *resolution-factor* determines the number of neighbors about which a reporting node responds. Each node can collect information about its neighboring nodes by eavesdropping, although this information may be stale. The responding node can choose a fraction *f* of these neighbors (probably giving priority to fresh information) and apply the filter specified by the query-type parameter.

The third parameter, *query-type* specifies the special functions that a responding node should execute before sending the data in order to support the query. The type of information retrieved determines the semantics used to filter information of neighborhood when a node responds to a query. For example, in case of topology discovery, we simply merged the retrieved data. A different type of aggregation could be used for energy information as proposed in [5]. A second perspective for this parameter could also be use of schemes like smart messages as proposed in [25] which would carry specific code along with the query. This would be useful to support some infrequent yet complex queries for which permanently storing code at the sensor is a waste of resources.

B. Handling Channel Errors and Node Failures.

The mechanisms described assume a zero error rate for channels and no node failure. However, we make minor changes to the algorithm to account for them.

We note that the topology discovery initially floods to the whole network. Since the network is dense, with many paths existing between any pair of nodes, even for reasonably bad cannel error rates, all nodes would receive a discovery packet with very high probability. Hence, in the coloring phase, channel error does not create a significant impact. The number of *black* nodes and path lengths to the initiating nodes can increase due to packet losses.

However, topology acknowledgement packets are returned through single prescribed paths and hence reliability is required in packet delivery. We use *passive acknowledgement scheme* from DSR [11] for this purpose.

The protocol to account for node failures is described below:

- When a *black* node forwards a topology discovery request, it also sends its *acknowledgement timer* value. All its *red* neighbors are thus aware when the black node they are attached to, is supposed to send a topology discovery response.
- When the *acknowledgement timer* expires, nodes listen to their communication range to see if the *black* node covering them sends a response.
- When a topology discovery response arrives from children black nodes, all red neighbors of the parent black nodes that can hear it will buffer and aggregate the topology information from the children nodes in anticipation of a black node failure.
- If the black node does not respond at the specified time, each of its *red* neighbor sets up a timer to become *black*. This timer is proportional to the distance from the original *black* node and the number of hops they are away from the monitoring node. Such a timer value is used to minimize the distortion caused by node failures. Also nodes further from the monitoring node respond earlier, to enable aggregation of information upstream.
- When the timer of a *red* node expires, it becomes *black* and sends its neighborhood list up the tree towards the monitoring node. Any such responses also contain *attached_to_id* number, which is the *id* of the original *black* node they were attached to.
- If any *red* node listens to a black node responding, with *attached_to_id* same as the original black node it was attached to, it cancels its own timer to respond.

With a high probability, this protocol ensures that a black node failure would not result in loss of information, if network is still connected. However, some additional overhead and delays might be incurred to compensate for the node failure. Moreover, aggregation at each level cannot be guaranteed.

C. *Asymmetric Links*

The description of STEM assumes the network graph to have symmetric links. If this were not the case then reverse path aggregation mechanism may not work in all cases, i.e., when a node becomes black, it forwards its responses to the parent black node. However, if the reverse link does not exist it has to send the acknowledgement through the default routing path to the monitoring node. Note that, in an asymmetric graph, each pair of nodes would be able to communicate only if the graph is strongly connected.

Using strong connectivity assumption and presence of routing information, a responding node either forwards a packet to its next hop to the parent black node (if a link exists) or sends it through the default routing path to the monitoring node. The asymmetry property has no effect on the coloring phase of STEM and the protocol remains the same.

The worst-case performance of the algorithm in case of asymmetric links would equal the probabilistic direct response mechanism. The probabilistic aggregated response performs poorly because it has a high packet overhead, which is not compensated in this case by aggregations. In general, the performance of STEM as compared to probabilistic direct and aggregated response mechanism is expected to be similar to the trends seen in figure 15, although each method would incur a higher overhead. A comprehensive study of STEM on asymmetric networks is part of our future work.

D. *Push Based Topology Discovery*

Nodes can piggyback network state information along with data packets allowing construction of approximate topology over time. However this may not be appropriate for sensor networks where the following would be common:

- Nodes may sleep for long duration. Hence there would be no information available about them at the monitoring node.
- Monitoring node would not be able to distinguish between sleeping nodes, dead nodes and nodes which do not send anything.

- Intermediate nodes may aggregate data along the path, which could result in loss of topology specific node information.

- The topology of sensor networks is not as static as the Internet. So the converged topology may not reflect the true topology if it takes long time for topology to converge.

If the number of parameters defining network state is large, piggybacking this information along with each packet might incur significant overhead. Hence it is desirable to decouple the network state recovery from data dissemination.

VI. CONCLUSIONS AND FUTURE WORK

In this paper, we have described an algorithm for sensor topology extraction at multiple resolutions (STEM). The algorithm takes parameters, which control the resolution of the returned topology trading expended overhead for accuracy. We analyzed and simulated STEM and showed its effectiveness in retrieving multi-resolution topology at proportional cost. We described various classes of topology queries and how STEM supports them using its parameters. In future, we plan to explore the parameter space of STEM qualitatively and quantitatively. STEM also provides the basic framework required for any information-overhead tradeoff. This opens up a wide design space for multi-resolution information retrieval. This would require knowledge of the properties of queries and the corresponding aggregation/filter functions to compress that property. We intend to explore this design space for different types of information in future.

VII. ACKNOWLEDGEMENTS

We would like to thank Dr. S. Muthu Muthukrishnan, Dr. Thu Nguyen, Dr. Richard Martin, and Chris Perry for their invaluable comments to improve the technical quality and presentation of this work.

REFERENCES

[1] J.M. Kahn, R.H. Katz, K.S.J. Pister, *Next century challenges: Mobile networking for 'smart dust'*, Proc. MOBICOM, 1999, Seattle, 271-278.

[2] G.J. Pottie and W.J. Kaiser, *Wireless Integrated Network Semsors.* Communications of the ACM, vol. 43 no. 5 (2001), 51-58.

[3] A. Downey, "*Using pathchar to estimate Internet link characteristics*, Proc. SIGCOMM 1999, Cambridge, MA, pp. 241-250, Sept. 1999.

[4] J. Broch, D. A. Maltz, D. B. Johnson, Y. C. Hu, and J. Jetcheva. *A Performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols.* In Proc. of the ACM/IEEE MobiCom, October 1998.

[5] Jerry Zhao, Ramesh Govindan and Deborah Estrin, "Residual Energy Scans for Monitoring Wireless Sensor Networks", *IEEE Wireless Communications and Networking Conference, 2002.*

[6] C. Intanagonwiwat, R. Govindan and D. Estrin; *Directed diffusion: a scalable and robust communication paradigm for sensor networks*; in Proceedings of Mobicom '00, 2000.

[7] Cormen, Leiserson, Rivest (1990) *Introduction to Algorithms*, MIT Press, McGraw Hill.

[8] Bruce Lowekamp, David R. O'Hallaron, and Thomas R. Gross, *"Topology Discovery for Large Ethernet Networks,"* In Proceedings of ACM SIGCOMM 2001 (San Diego, California), ACM Press, August 2001.

[9] Y. Breitbart, M. Garofalakis, C. Martin, R. Rastogi, S. Seshadri and A. Silberschatz. *" Topology Discovery in Heterogeneous IP Networks"* In Proceedings of IEEE INFOCOM, 2000.

[10] Nancy Miller and Peter Steenkiste, *"Collecting Network Status Information for Network-Aware Applications"*In Proceedings of IEEE INFOCOM 2000.

[11] David B Johnson and David A Maltz, *"Dynamic Source Routing in Ad Hoc Wireless Networks",* Mobile Computing, Volume 353, Kluwer Academic Publishers, Edited by Imielinski and Korth.

[12] Seapahn Meguerdichian, Farinaz Koushanfar, Gang Qu, Miodrag Potkonjak, *"Exposure In Wireless Ad Hoc Sensor Networks."* Proc.. of 7th Annual International Conference on Mobile Computing and Networking, MOBICOM 2001.

[13] Seapahn Meguerdichian, Farinaz Koushanfar, Miodrag Potkonjak, Mani Srivastava, *"Coverage Problems in Wireless Ad-Hoc Sensor Networks,"*In Proc. of IEEE INFOCOM 2001.

[14] Bhaskar Krishnamachari, Stephen B. Wicker, and Ramon Bejar, *"Phase Transition Phenomenon in Wireless Ad hoc Networks", Symposium on Ad-Hoc Wireless Networks, GlobeCom2001*, San Antonio, Texas, November 2001.

[15] L. Li, Z. Haas and J.Y. Halpern,*"Gossip-Based Ad Hoc Routing", IEEE INFOCOM,* June 2002.

[16] Romit RoyChouldhury, S. Bandyopadhyay and Krishna Paul, *"A Distributed Mechanism for Topology Discovery in Ad hoc Wireless Networks using Mobile Agents",* Proc. of Workshop On Mobile Ad Hoc Networking & Computing (MOBIHOC 2000).

[17] Baruch Awerbuch and Yuval Shavitt, "*Topology Aggregation for Directed Graphs.",* IEEE/ACM Transactions on Networking, Vol.9, N0.1, Feb 2001.

[18] B. Das, V Bhargavan, *"Routing in Ad Hoc Networks Using Minimum connected dominating Sets"*. IEEE International Conference on Communications (ICC '97), Jun, 1997.

[19] A. Ephremides, J.E. Wieselthier, and D.J. Baker, "A design concept for trliable mobile radio networks with frequency hopping signaling." Proceeding of IEEE, 75, 1 (Jan 1987) 56-73.

[20] M. Gerla, J.T. C. Tsai, Multicluster, mobile, multimedia radio networks," ACM J. Wireless Networks , 1, 3 (1995) 255-265.

[21] S. Guha and S. Khuller, " Approximation Algorithms for Connected Dominating Sets," European Symposium on Algorithms. 179-193, 1996.

[22] P. Sinha, R. Sivakumar and V. Bharghavan, "*CEDAR: a Core-Extraction Distributed Ad hoc Routing Algorithm"* In proceedings of IEEE Infocom 1999.

[23] Suman Bannerjee, Samir Khuller, "*A Clustering Scheme for Hierarchical Control in Wireless Networks",* In Proceedings of IEEE INFOCOM 2001.

[24] B. Das, R. Sivakumar and V. Bharghavan. "*Routing in ad hoc networks using a virtual backbone*." In Proc. IEEE (IC3N'97).

[25] Cristian Borcea, Deepa Iyer, Porlin Kang, Akhilesh Saxena, Liviu Iftode, *"Cooperative Computing for Distributed Embedded Systems"*, International Conference of Distributed Computing Systems, 2002.

[26] B. Clark , C. Colbourn and D. Johnson, *"Unit disk graphs",* Discrete Mathematics, vol. 86, pp. 165--177, 1990.

[27] A. Bestavros and J. Byers and K. Harfoush, *"Inference and Labeling of Metric-Induced Network Topologies"* In Proceedings of Infocom'02: The IEEE International Conference on Computer Communication*, June 2002.*

[28] LBNL Network Research Group, *"UCB/LBNL/VINT Network Simulator- ns(version 2)",* http://www.isi.edu/nsnam.

[29] J.E. Wieselthier, G.D. Nguyen, and A. Epremedes, *"On the construction of energy-efficient broadcast and multicast trees in wireless networks"* In IEEE INFOCOM 2000, Tel Aviv, Israel 2000.